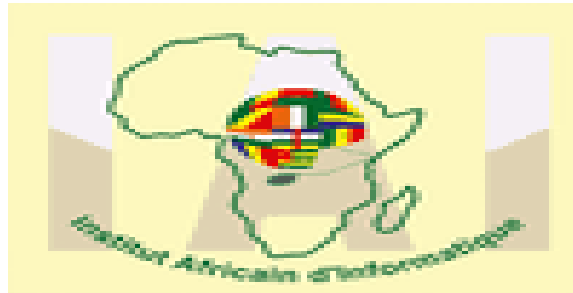


REPUBLIQUE TOGOLAISE

-----  
Travail – Liberté - Patrie  
Ministère de la Planification du  
Développement



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

Tél : (+228) 22 20 47 00

Email : [iaitogo@iai-togo.tg](mailto:iaitogo@iai-togo.tg)/[iaitogo@yahoo.fr](mailto:iaitogo@yahoo.fr)

Site Web: [www.iai-togo.tg](http://www.iai-togo.tg)

07 BP 12456 Lomé, TOGO

**RAPPORT DE STAGE PRATIQUE EN ENTREPRISE**

Option : Génie Logiciel

**THEME :**

**GESTION AUTOMATISEE DES CONCOURS DE BOURSES ET DES  
SEANCES D'ORIENTATION**

**Période** : 27 Juillet 2020 – 26 Septembre 2020

**Rédigé par :**

**WOAGOU Yendouboame Aimé**

Etudiant en deuxième année Tronc Commun

Année Universitaire 2019-2020

**Superviseur**

**M. WOAMEY Mike**

Enseignant à l'IAI-TOGO



## INTRODUCTION

Depuis la découverte de l'informatique, de nombreuses activités de la vie courante ont été simplifiées. Actuellement, les individus peuvent facilement traiter des informations en se servant des logiciels et des réseaux informatiques. Compte tenu de son évolution, ce système caractérise la majorité des grandes entreprises quel que soit le secteur d'activité. Nous sommes dans un monde de plus en plus connecté, de plus en plus reliés aux outils informatiques. La technologie est aujourd'hui un élément fondamental pour toute entreprise. L'évolution dans le domaine de l'informatique offre des opportunités pour améliorer les processus de flux de travaux, favoriser l'innovation et acquérir un avantage concurrentiel. Par conséquent, dans le cadre de la formation des Ingénieurs des Travaux Informatiques, l'Institut Africain d'Informatique représentation du TOGO (IAI-TOGO) a mis en place, pour les étudiants de deuxième (2ème) année, un stage pratique de deux (2) mois en entreprise. Son objectif est la mise en pratique des connaissances en analyse et programmation acquises pendant les deux premières années de formation, et la réalisation d'un projet concret, qui devra fournir une application viable et prête à l'emploi.

A cet effet, nous avons effectué un stage de Juillet à Octobre 2020 à l'institut en raison de la situation de la crise sanitaire qui frappe le pays. Notre travail consistait à développer une application desktop de gestion en interne, d'où le thème de notre stage : **GESTION AUTOMATISEE DES CONCOURS DE BOURSES ET DES SEANCES D'ORIENTATION**

Notre document sera réparti en deux (02) grandes parties : *le rapport de pré programmation et le rapport de réalisation et mise en Œuvre.*

## I. PRESENTATION

### 1. Brève présentation de l'IAI-TOGO

L'institut Africain d'Informatique (IAI) est une école supérieure en informatique. La convention portant création de l'institut et les statuts y afférents ont été signés le 29 janvier 1972 à Fort Lamy (actuel N'Djamena) en république du TCHAD. L'accord de siège entre l'IAI et le GABON (siège de l'Institut) a été signé en janvier 1975. Il est par conséquent un Etablissement Inter-Etats d'Enseignement Supérieur.

En application de la décision du Conseil d'Administration de délocaliser l'IAI, la Représentation du TOGO (IAI-TOGO) a ouvert ses portes le 24 Octobre 2002. IAI-TOGO est situé sur la rue de la Kozah à Nyékonakpoè, logé dans le bâtiment du Centre National d'Etudes et de Traitements Informatique (CENETI) et placé sous la tutelle du Ministère de la Planification du Développement. L'accord d'établissement entre la République Togolaise et l'Institut Africain d'Informatique a été signé le 12 mai 2006 à Lomé.

L'Institut Africain d'informatique, représentation du TOGO (IAI-TOGO) propose actuellement les cycles de formation d'Ingénieurs de Travaux Informatiques (Licence professionnelle en informatique), en Génie Logiciel (GL), Système & Réseaux (SR) et Multimédia Technologies Web et Infographie (M-TWI). La durée de ce cycle est de trois (03) ans. Au terme des trois années de formation, les diplômés peuvent poursuivre leurs études supérieures au siège, au GABON ou dans les universités occidentales ou asiatiques (UTBM en FRANCE, Université-Laval du Québec au CANADA, etc.), bref dans le monde entier.

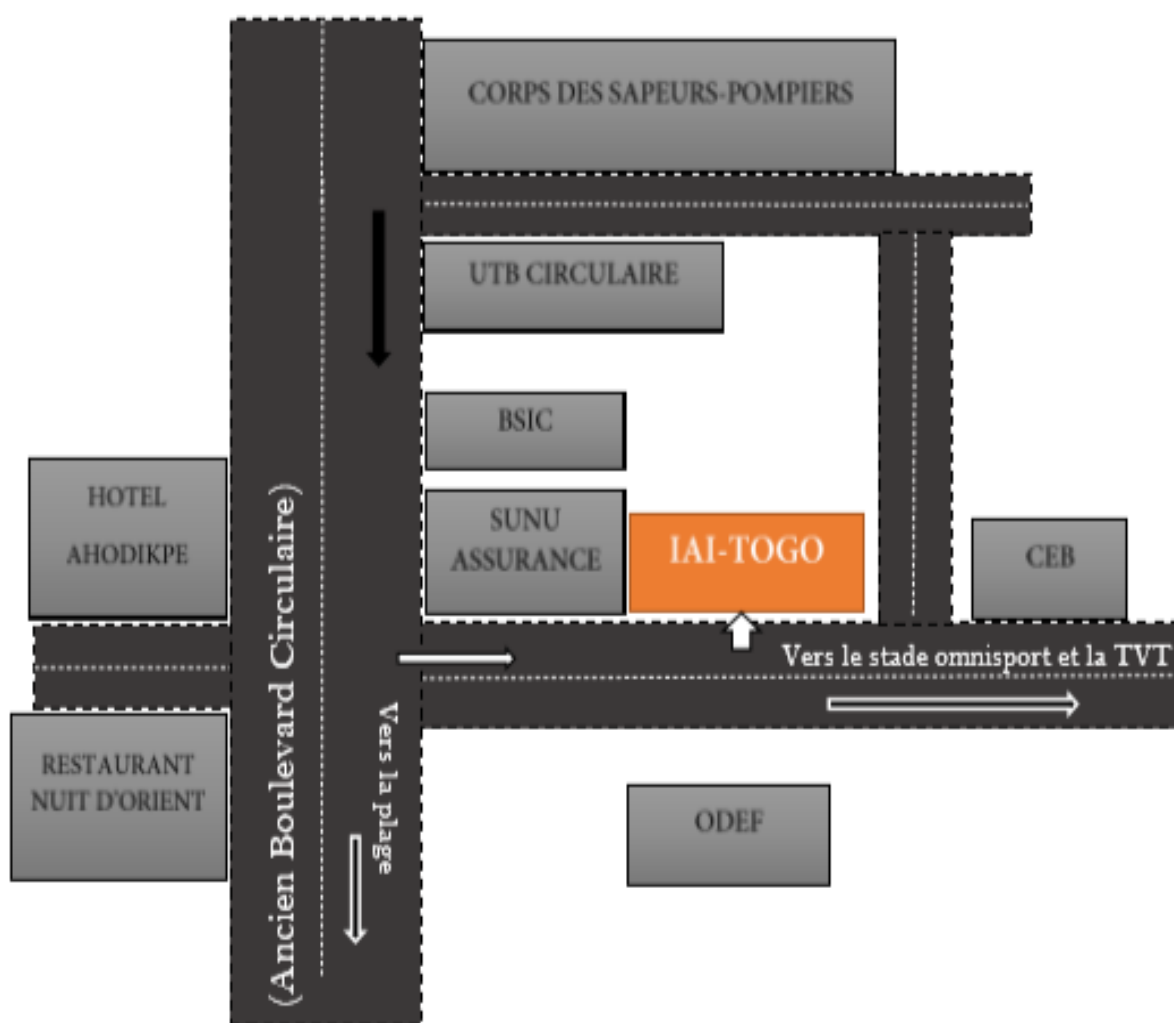
*L'IAI-TOGO est joignable à l'adresse suivante :*

***BP : 07 BP 12456 Lomé 07, TOGO***

***Tél : (+228) 22 21 27 06 / 22 20 47 00***

***E-mail : [iaitogo@iai-togo.com](mailto:iaitogo@iai-togo.com)***

## a. Localisation géographique de l'IAI-TOGO



*Figure 1 : Localisation géographique de L'IAI-TOGO.*



## II. THEME DE STAGE

### 1. Présentation du sujet

Dans le souci d'être plus productif, L'ESGIS doit se doter d'outils pouvant les aider dans la gestion de ses bourses et séance d'orientation. Ces outils doivent pouvoir suivre les flux entrant et sortant tout en aidant les prises de décision sur la base d'information statistiques précise. C'est dans cette optique d'une meilleure gestion des bourses et séance d'orientation que s'inscrit notre thème de stage, Gestion automatisée des concours de bourses et des séances d'orientation.

### 2. Problématique du sujet

ESGIS comme beaucoup d'autre université a mis en place une gestion simple de ses étudiants. Les informations sur les candidats sont enregistrées dans des registres et les données des différentes mobilités sont conservé dans des cahiers. Ces outils ne sont pas adaptés à une bonne gestion des bourses et séance d'orientation et le traitement des informations au fil du temps devient complexe et fastidieux. De cette gestion surgit aussi des problèmes de centralisation et de sécurité des données qui peuvent être facilement perdues.

Comment alors mettre en place une gestion plus optimisée et automatisée des concours de bourses et des séances d'orientation tout en assurant sécurité et la pérennisation de ces informations ?

### 3. Intérêt du sujet

#### a. Objectifs

Notre travail consiste à mettre en place une base de données sécurisée et fiable qui sera interfacée par une application Desktop permettant de gérer les séances d'orientation organisées par ESGIS ainsi que les concours de bourses d'étude.

L'application devra en outre :

- Permettre d'enregistrer les personnes se présentant pour les séances d'orientation et d'éditer leur liste par période
- Permettre d'enregistrer le chargé de la séance d'orientation, connaître la date et l'heure de la séance
- Permettre d'enregistrer les masses médias par lesquelles le concours a été connu et pouvoir éditer les statistiques

- Permettre de connaitre parmi ceux qui se sont présentés à la séance d'orientation ceux qui se sont inscrits (éditer des statistiques)
- Permettre de programmer les concours de bourses par différents cycles et éditer le planning
- Permettre d'enregistrer les candidats aux concours de bourses, inscription au concours et paiement des frais d'inscription
- Permettre d'enregistrer les notes obtenues par chaque candidat lors du concours
- Permettre d'octroyer les bourses aux candidats en tenant compte des notes obtenues et de la catégorie sociale du candidat

b. Résultats attendus

L'application que nous aurons à développer permettrait :

- D'enregistrer un candidat avec ses données
- D'enregistrer une séance d'orientation



---

*PARTIE II : RAPPORT DE PRE-  
PROGRAMMATION*

---

## I. ETUDE DE L'EXISTANT

Pour mener à bien ce projet, des informations nous ont été fournies pendant une phase de recueil d'informations. Grace à ces informations, nous avons pu constater qu'à ESGIS, la gestion des concours de bourses et des séances d'orientation se fait dans un fichier Excel.

### 1. Traitement des concours de bourses

Un candidat désireux de passer un concours pour obtenir une bourse d'études programmée par ESGIS doit se rendre au secrétariat dudit institut. Il se voit alors attribuer une fiche d'inscription (nom, prénoms, date de naissance, lieu de naissance, nationalité, sexe, situation de famille, profession, pays, ville, quartier, e-mail, téléphone, nom du parent, prénoms du parent, profession du parent, adresse du parent, école de provenance, série, année et mention du BAC, filière choisie (BTS Action Commerciale et Force de Vente, BTS Commerce International, Licence Banque Finance, Master Comptabilité Contrôle Audit, etc.)) qu'il doit remplir. Ensuite les informations du candidat sont recueillies dans un fichier Excel. De même les informations concernant les résultats qu'il aura obtenus au concours ainsi que la bourse qui lui aura été octroyée par le Directeur des études seront stockées dans un fichier Excel. Après l'octroi de la bourse le candidat dispose d'un délai généralement (2 semaines) pour venir confirmer sa place c'est-à-dire s'inscrire au risque de se voir retirer sa bourse. Toutes ces informations sont soit stockées dans des fichiers Excel soit sur des papiers A4 prévus à cet effet.

### 2. Traitement des séances d'orientation

Un candidat désireux de s'informer sur les filières disponibles à ESGIS doit se rendre au secrétariat de l'institut. Il se voit alors attribuer une fiche (Nom, prénoms, date de naissance, lieu de naissance, nationalité, sexe, situation de famille, profession, pays, ville, quartier, e-mail, téléphone, nom du parent, prénoms du parent, profession du parent, adresse du parent, école de provenance, série, année et mention du BAC, filière choisie (BTS Action Commerciale et Force de Vente, BTS Commerce International, Licence Banque Finance, Master Comptabilité Contrôle Audit, etc.)) qu'il doit remplir. Ensuite, les informations du candidat sont recueillies dans un fichier Excel. Puis, il se voit attribuer un agent chargé de l'informer sur les filières qu'il a choisies. Il ne peut choisir au plus que trois filières (3). Les observations et autres informations sont portées sur des fiches prévues à cet effet.

## II. CRITIQUE DE L'EXISTANT

Les informations recueillies et notre analyse faite de l'existant ont permis de déceler certaines failles au niveau du traitement des informations. Ainsi, le traitement étant manuel, il peut y avoir :

- Possibilité de perte d'informations

- Microsoft Excel bien qu'étant un logiciel tout entier destiné faire plus rapidement les calculs on note quelques difficultés dans son utilisation à savoir : les difficultés pour un non-initié à ce produit de se retrouver facilement.

- Chaque traitement étant fait au niveau du secrétariat et de la direction des études l'archivage des informations devient de plus en plus difficile, faute de place.

- Temps de traitement des informations très lourd
- L'agent, en faisant la saisie manuellement des informations peut commettre des erreurs.
- Difficultés de mise à jour des informations concernant un candidat.
- Notons aussi en ce qui concerne la sécurité des données qu'elles peuvent être altérées par un tiers et aussi l'impossibilité d'éditer de statistiques etc...

## 1. PROPOSITIONS DE SOLUTIONS

Face aux critiques énoncées ci-dessus, deux solutions s'offrent à nous et qui peuvent être envisagées par ESGIS.

### a. Première solution

Cette première solution consiste à créer un site web via un CMS (Content Management System : désigne une famille d'applications qui ont pour but de créer et de mettre à jour facilement un site web dynamique). Comme exemple, nous pouvons citer :

- Joomla : logiciel libre, gratuit et dont la plupart des fonctionnalités sont gratuites ; c'est l'un des CMS les plus utilisés et permet de concevoir tout type de site web.
- WordPress : c'est un outil gratuit et complet qui comme Joomla permet de concevoir tout type de site web.

### ❖ Avantages

- Gain de temps
- Possibilité de changer le thème du site au moment voulu
- Facilité d'utilisation
- Faire des mises à jour régulière

### ❖ Inconvénients

- Les fonctionnalités spécifiques ne sont pas prises en compte
- Problème de ralentissement au niveau du chargement des pages
- Sécurité peu fiable liée à la popularité du logiciel

## b. Deuxième solution

Cette solution consiste à faire développer une application desktop efficace. Cette solution présente des avantages comme des inconvénients :

❖ **Avantages**

- La nouvelle application pourra permettre d'exécuter rapidement les traitements ;
- Faire des recherches rapides et faciles ;
- La maintenance de l'application sera plus aisée
- Notre application prend en compte les nouveautés en matière de web

❖ **Inconvénients**

- Durée de développement de l'application sera plus longue.

### III. EVALUTION FINANCIERE

On se doit d'estimer le coût de chaque solution qu'ils soient hébergement, humains, matériels, ou logiciel et d'en ressortir les couts totaux.

## 1. Première solution

## a. Coût d'hébergement

Tableau 1 : coût d'hébergement de la première solution

Frais par mois (euro)	Coût (FCFA)
7,19(*)	4.714
<b>Total</b>	<b>4.714</b>
Frais par an (euro)	Coût (FCFA)
86,28(*)	56.568
<b>Total</b>	<b>56.568</b>

(\*) Source : <https://www.ovh.com/fr/hebergement-web/>

## b. Coût conception

Tableau 2 : coût de conception de la première solution

Qualité	Nombre	Tâches	Nombre d'heures	Coût horaire		Coût (FCFA)
<b>Analyste programmeur</b>	2	Conception de l'application	<b>210</b>	4000		840.000
Total						840.000

## c. Coût de maintenance

Tableau 4 : coût de maintenance de la première solution

Qualité	Nombre	Tâches	Nombre d'heures	Coût horaire	Coût (FCFA)
<b>Maintenancier</b>	2	Maintenance de l'application	<b>120</b>	3000	360.000
Total					360.000

## d. Coût matériel

Tableau 5 : coût matériel de la première solution

Désignation	Exigence	Prix Unitaire	Quantité	Coût (FCFA)
Ordinateur	DELL-PC Core i3 4 Go RAM 500Go	433,06 € (*)	2	283.654,3
<b>Total</b>				<b>283.654,3</b>

(\*) source : <https://www.cddiscount.com/>

## e. Coût total

Tableau 6 : coût total de la première solution

Coût	Totaux
Coût d'hébergement	56.568
Coût de conception	840.000
Coût de formation	60.000

Coût de maintenance	360.000
Coût matériel	283.650
<b>Total</b>	<b>1.600.218</b>

## 2. Deuxième solution

### a. Coût de conception

Tableau 7 : coût de conception de la deuxième solution

Qualité	Nombre	Tâches	Nombre d'heures	Coût horaire		Coût (FCFA)
<b>Analyste programmeur</b>	2	Conception de l'application	240 (8 heures *30)	4000		960.000
<b>Total</b>						<b>960.000</b>

### b. Coût de maintenance

Tableau 8 : coût de maintenance de la première solution

Qualité	Nombre	Tâches	Nombre d'heures	Coût horaire	Coût (FCFA)
<b>Maintenancier</b>	2	Maintenance de l'application	270 (5 heures*54jours)	3000	810.000
<b>Total</b>					<b>810.000</b>

### c. Coût matériel

Tableau 9 : coût matériel de la première solution

Désignation	Exigence	Prix Unitaire	Quantité	Coût (FCFA)
Ordinateur	DELL-PC Core i3 4 Go RAM 500Go	433,06 € (*)	2	283.654,3
<b>Total</b>				<b>283.654,3</b>

(\*) source : <https://www.cddiscount.com/>

## d. Coût total

Tableau 10 : coût total de la première solution

Coût	Totaux
Coût de conception	960.000
Coût de formation	60.000
Coût de maintenance	810.000
Coût matériel	283.650
<b>Total</b>	<b>2.119.650</b>

## 3. Choix de solution

Pour pallier à aux insuffisances techniques détectées pour la première solution, nous avons choisis de développer l'application desktop de gestion des concours de bourses et des séances d'orientation.

## a. Spécifications fonctionnelles du projet

Tableau 11 : spécification fonctionnelle du projet

PROFILS	DROITS
Utilisateur	<ul style="list-style-type: none"> <li>• S'authentifier</li> <li>• Enregistrer une inscription</li> <li>• Consulter la liste des étudiants déjà inscrits</li> <li>• Octroyer une bourse</li> <li>• Annuler une bourse</li> <li>• Consulter la liste des boursiers</li> <li>• Programmer un concours</li> <li>• Editer les résultats obtenus par les candidats</li> <li>• Enregistrer une séance d'orientation</li> <li>• Consulter la liste des séances effectuées</li> <li>• Editer les statistiques</li> <li>• Envoyer des notifications dans la boîte mail des candidats</li> </ul>
Administrateur	<ul style="list-style-type: none"> <li>• S'authentifier</li> </ul>

	<ul style="list-style-type: none"> <li>• Ajouter un utilisateur</li> <li>• Modifier un utilisateur</li> <li>• Supprimer un utilisateur</li> <li>• Ajouter un profil</li> <li>• Modifier un profil</li> <li>• Supprimer un profil</li> <li>• Attribuer des droits</li> </ul>
--	---

b. Spécification de sécurité

La sécurité est prise en compte dans notre application ; Il faudra donc pour un utilisateur un compte qui lui permettra de s'authentifier avant de faire usage de l'application. Il disposera donc de droits qui lui permettront de réaliser des actions précises.

c. Spécifications techniques de l'application

Tableau 12 : spécifications techniques du projet

Méthode d'analyse	Approche objet (UML + 2 TUP)
Outils de modélisation	Sybase PowerAMC
Système de gestion de base de données	MySQL
Langage de programmation	C#
Environnement de Développement	Visual Studio 2017

d. Planning prévisionnel



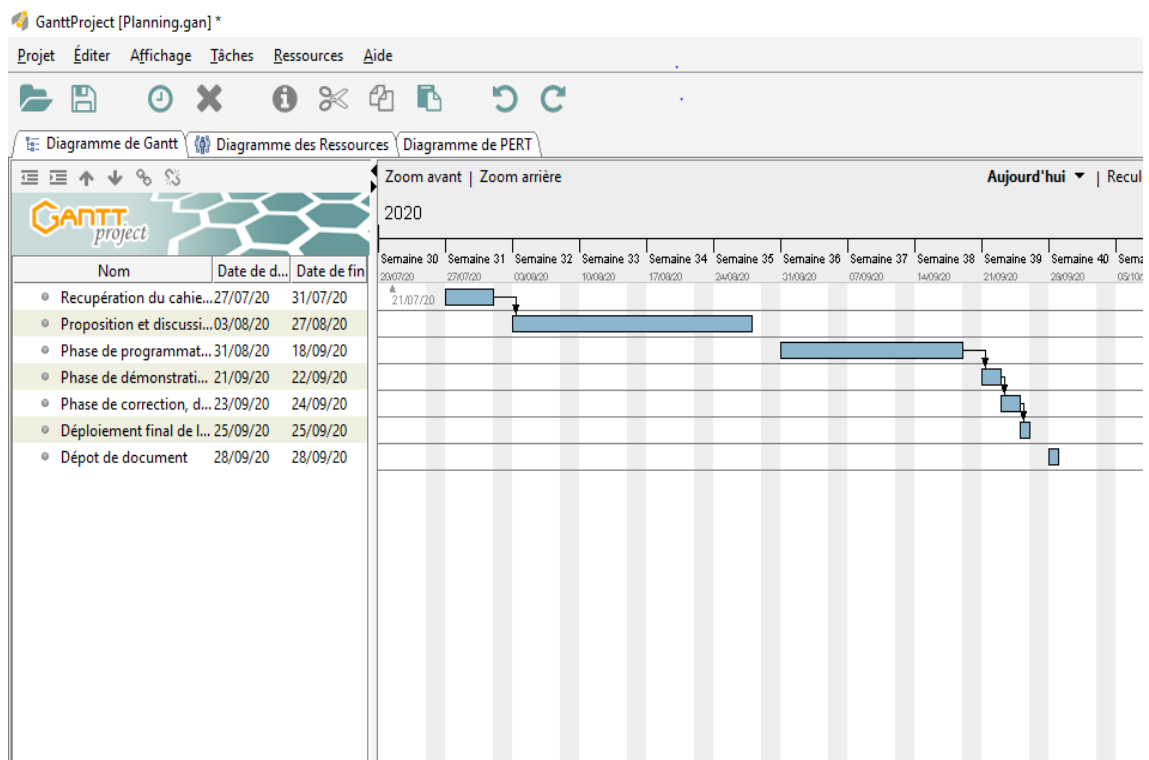


Figure 3 : Planning Prévisionnel

## IV. PRESENTATION DE LA METHODE D'ANALYSE

1. Le langage de modélisation
  - a. Présentation



Figure 4 : Logo d'UML

Le langage de modélisation unifié, en anglais Unified Modeling Language (UML) est un langage de modélisation graphique 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 est le résultat de

la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

b. UML est formalisme

UML est un langage de conception objet qui permet de modéliser les solutions informatiques par des diagrammes (cités ci-dessus). Pourtant, le langage ne propose pas d'ordre qui doit régir ces diagrammes. C'est ainsi que les méthodes viennent appuyer le langage pour rendre totale et parfaite l'approche objet. Les méthodes imposent au langage l'ordre et la démarche. Il existe une multitude de méthodes de conception objet telles que : 2TUP (Two Track Unified Process), XP (Extreme Programming), Scrum, RUP (Rational Unified Process). La méthode retenue pour notre projet est le 2TUP.

c. Processus de modélisation

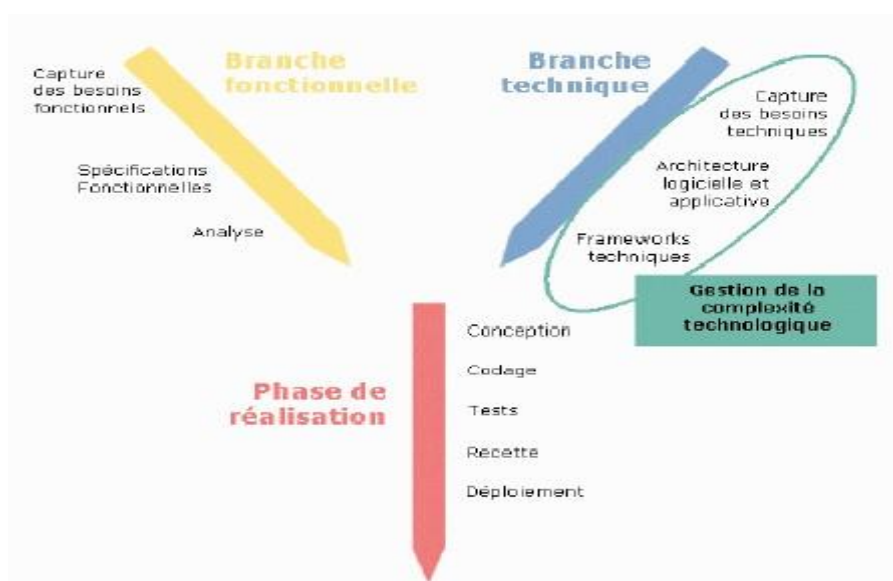


Figure 5 : Illustration du 2TUP

2 Track Unified Process (2TUP), à prononcer "toutiyoupi" est un processus de développement logiciel qui implémente le Processus Unifié. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système.

Ensuite à produire le cahier des charges et à modéliser le contexte (le système est une boîte) noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Le processus s'articule ensuite autour de 3 phases essentielles :

- Une branche technique ;
- Une branche fonctionnelle ;
- Une phase de réalisation.

➤ ***La branche technique (droite)***

- La capture des besoins techniques qui recense toutes les contraintes sur les choix de dimensionnement et la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (prérequis d'architecture technique) ;

- La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype.

➤ ***La branche fonctionnelle (gauche)***

- Elle capture des besoins fonctionnels qui produisent le modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs ;

- L'analyse qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier.

➤ ***La phase de réalisation (milieu)***

Elle correspond à :

- Une conception préliminaire qui représente une étape délicate car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;

- La conception détaillée qui étudie comment réaliser chaque composant ;
- L'étape de codage qui produit ses composants et teste au fur et à mesure les unités

de code réalisées ;

L'étape de recette, qui consiste à valider les fonctionnalités du système développé.

### 1. Présentation de l'outil de modélisation

Dans la modélisation, on pouvait se contenter d'utiliser uniquement UML, mais sur quels supports ? De nos jours, on ne peut pas se contenter des dessins faits sur papier ni au tableau ; c'est là qu'intervient les logiciels de modélisation qui permettent la réutilisation des modèles, leur portabilité (réalisées une fois et utilisables partout), etc.

L'outil retenu pour la modélisation est : Sybase PowerAMC version 15.1.

#### a. QU'est-ce que PowerAMC ?



*Figure 6 : Logo de PowerAMC*

PowerAMC est un environnement graphique de modélisation d'entreprise très simple d'emploi. Il a été créé par la société SDP sous le nom de AMC\*Designor, racheté par PowerSoft qui lui-même a été racheté par Sybase en 1995. PowerAMC prend en compte plusieurs outils de modélisation tels UML et MERISE.

PowerAMC permet d'effectuer les tâches suivantes :

- Modélisation intégrée via l'utilisation de méthodologies et de notation standards :
  - Données (E/R, Merise) ;
  - Métiers (BPMN, BPEL, ebXML) ;
  - Application (UML).
- Génération automatique de code via des Template personnalisables :
  - SQL (avec plus de 50 SGBD) ;
  - Java ;
  - .NET

- Fonctionnalités de reverse engineering pour documenter et mettre à jour des systèmes existants ;
- Une solution de référentiel d'entreprise avec des fonctionnalités de sécurité et de gestion des versions très complètes pour permettre un développement multiutilisateur
- Fonctionnalités de génération et de gestion de rapports automatisés et personnalisables
- Un environnement extensible, qui vous permet d'ajouter des règles, des commandes, des concepts et des attributs à vos méthodologies de modélisation et de codage.

#### b. Modélisation avec PowerAMC

PowerAMC fournit un jeu unique d'outils de modélisation professionnels qui associent les techniques et notations standard de la modélisation de processus métiers, de la modélisation des données et de la modélisation des diagrammes UML et d'autres fonctionnalités puissantes afin d'aider à analyser, concevoir, construire et maintenir des applications, en utilisant les techniques les plus élaborées d'ingénierie logicielle.

La solution de modélisation PowerAMC permet d'intégrer étroitement la conception et la maintenance des couches de données centrales de l'application et exigences de projet, processus métiers, code orienté objet, vocabulaires XML et informations de réplication de base de données.

## V. ETUDE DETAILLEE DE LA SOLUTION

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.

### 1. Vue de modélisation

Diagramme de contexte statique Le diagramme de contexte statique délimite le domaine d'étude en précisant ce qui est à la charge du système et en identifiant l'environnement extérieur au système étudié avec lequel ce dernier communique. Ces composant sont :

- ✓ Les acteurs externes : entités externes au système étudié interagissant avec lui
- ✓ Un processus unique symbolisant le Système Information étudié

✓ Echange entre le système étudié et son environnement

Ce diagramme n'est pas officiellement désigné comme diagramme UML. Il ne fait donc pas partie des diagrammes « officiels », mais il est utile pour la description du domaine d'étude avant d'aborder les aspects plus fins.

## 2. Vue de cas d'utilisation

### a. Le 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.

#### ❖ Élément de modélisation des 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 un employé quelconque de l'ESGIS)
- Administrateur (employé qui a la charge du contrôle des accès à l'application)
- Le cas d'utilisation

Un cas d'utilisation exprime le comportement du système en termes d'actions et réactions face à un besoin d'un utilisateur, c'est une fonctionnalité offerte par le système.

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 13 : Les différents cas d'utilisation du système*

	<b>Cas d'utilisation</b>	<b>Acteurs</b>
Gérer les inscriptions	Enregistrer une inscription	Utilisateur
	Encaissement des règlements	
	Consulter la liste des étudiants déjà inscrits	
Gérer les bourses	Octroyer une bourse	Utilisateur
	Annuler une bourse	
	Consulter la liste des boursiers	
Gérer les concours	Programmer un concours	Utilisateur
	Editer les résultats obtenus par les candidats	
Gérer les séances d'orientations	Enregistrer la séance	Utilisateur
	Consulter la liste des séances effectuées	
	Editer les statistiques	
Envoi des notifications	Envoyer des notifications dans la boîte mail des candidats	Utilisateur
Gérer les utilisateurs	Ajouter, Modifier, Supprimer un utilisateur	Administrateur
Gérer les profils	Ajouter, modifier, supprimer les profils	Administrateur
	Attribuer des droits	
	S'authentifier	Administrateur, utilisateur

#### b. Diagramme de cas d'utilisation par acteur

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

#### ❖ Diagramme de cas d'utilisation du module concours

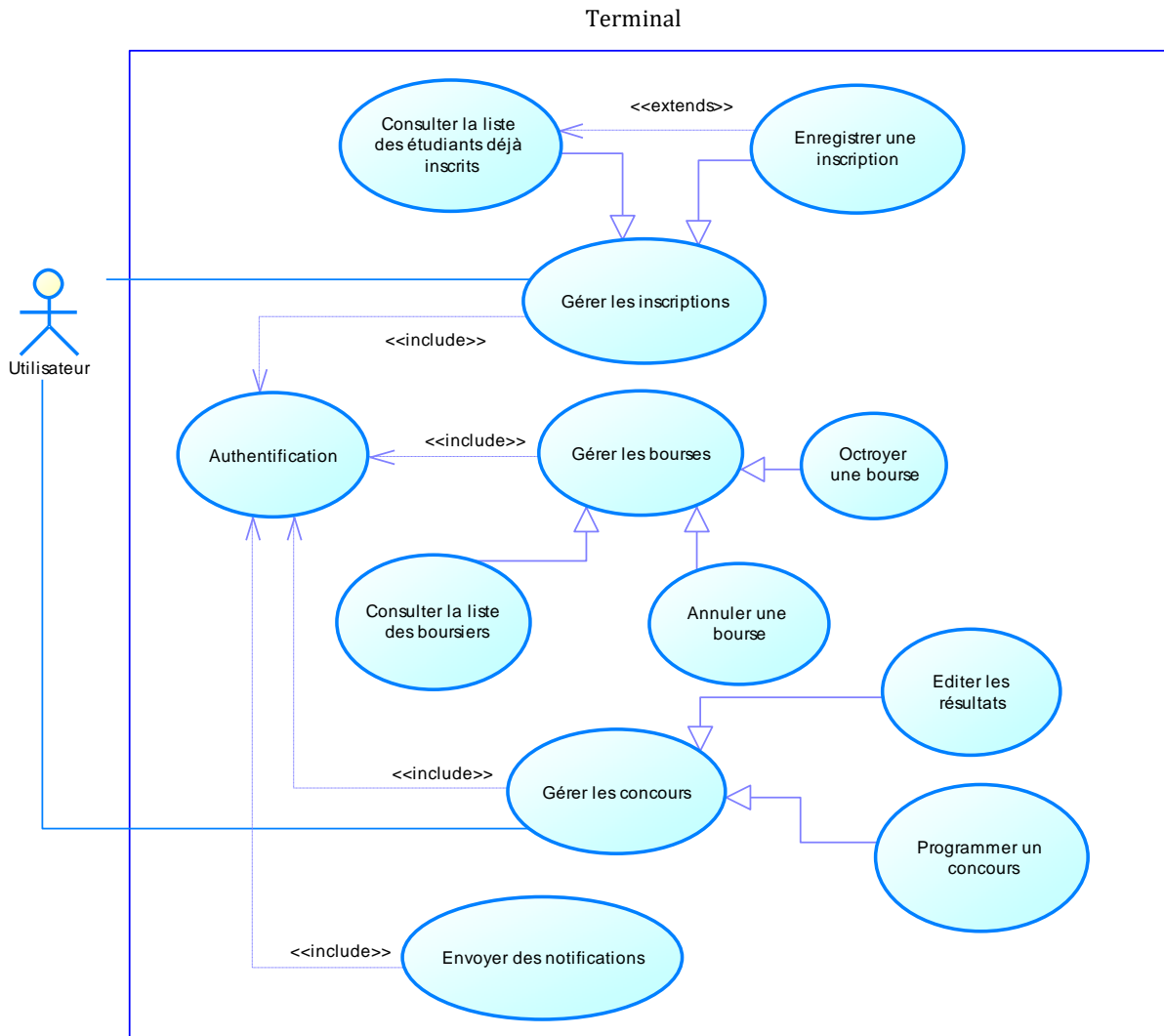


Figure 7 : représentation du diagramme de cas d'utilisation du module concours

❖ Diagramme de cas d'utilisateur du module Séances d'orientation



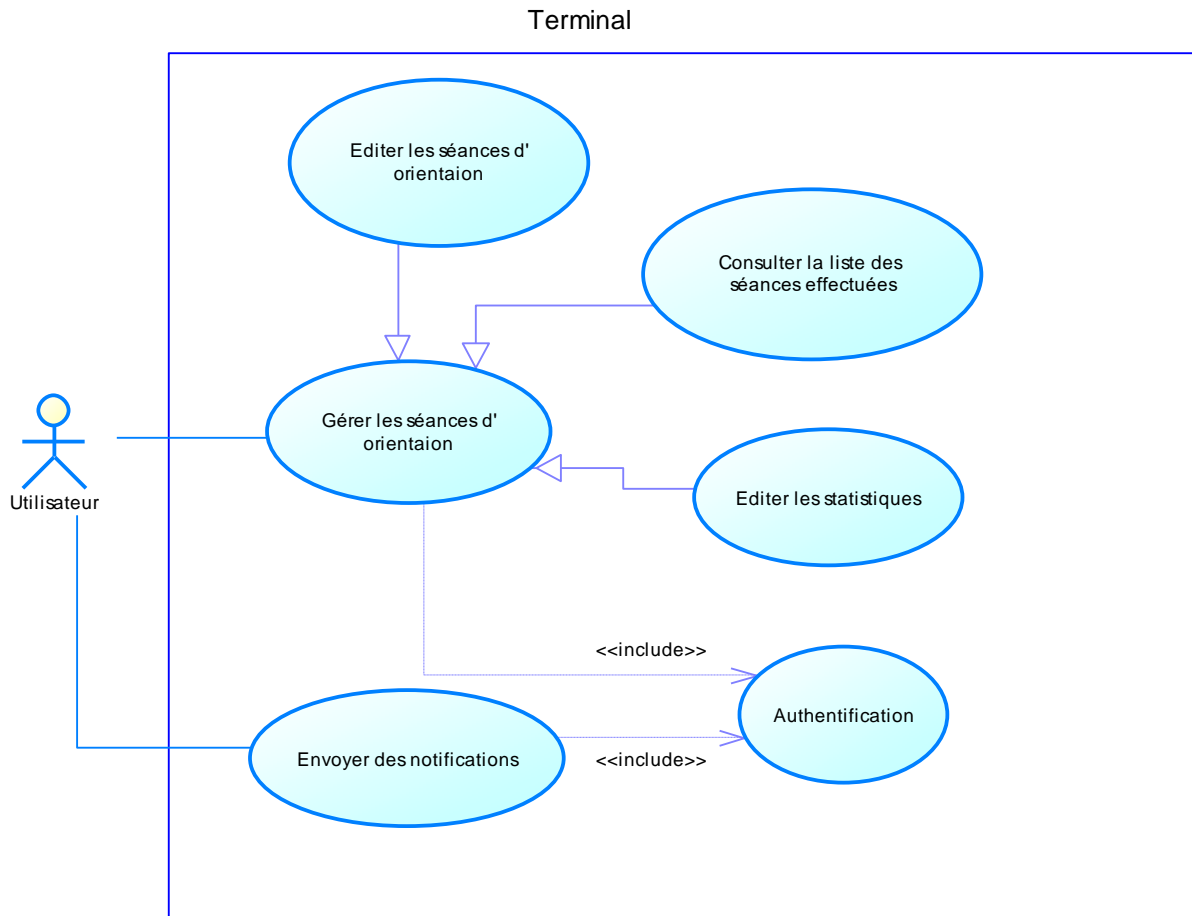


Figure 8 : représentation du diagramme de cas d'utilisation du module séance d'orientation

❖ Diagramme de cas d'utilisation du module Utilisateur

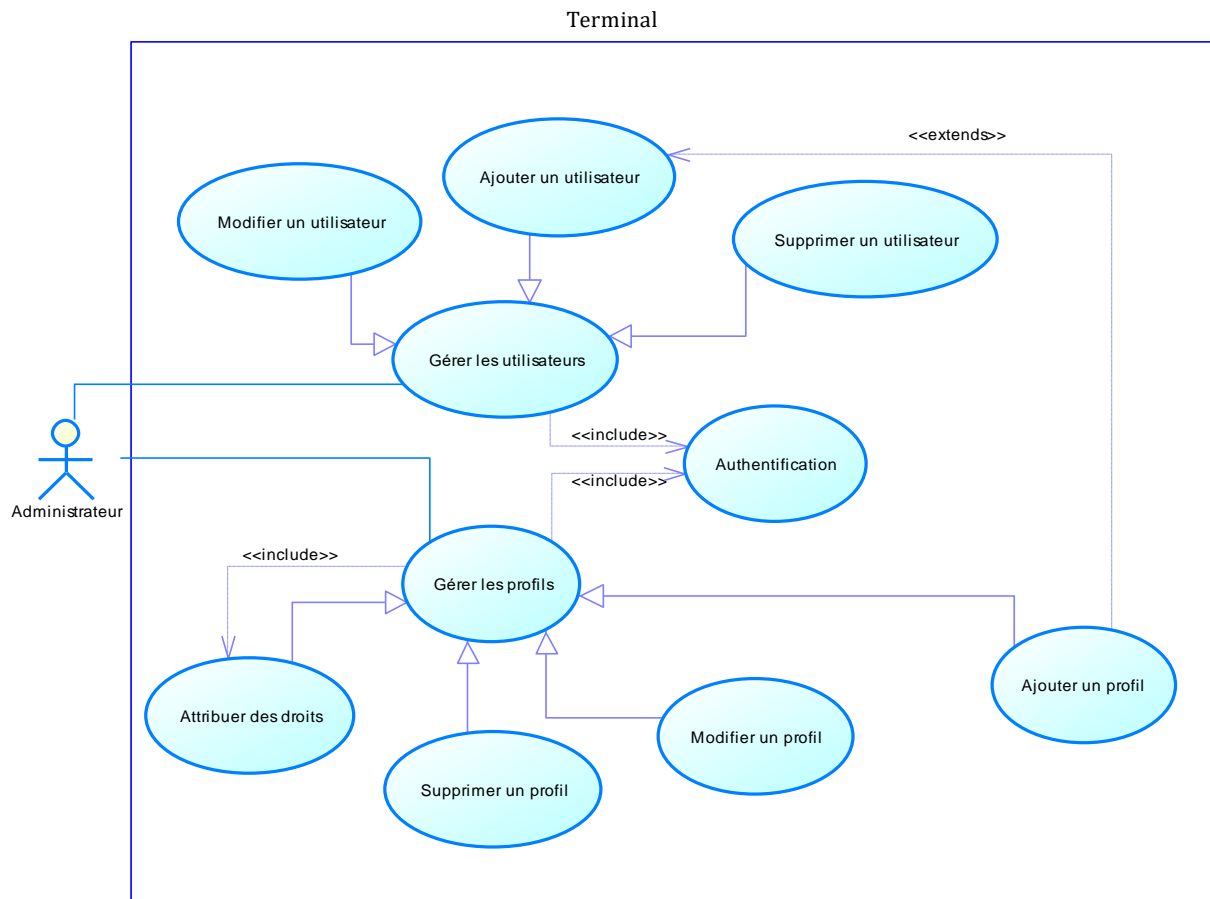


Figure 9 : représentation du diagramme de cas d'utilisation du module Administrateur

❖ Diagramme de cas d'utilisation du système

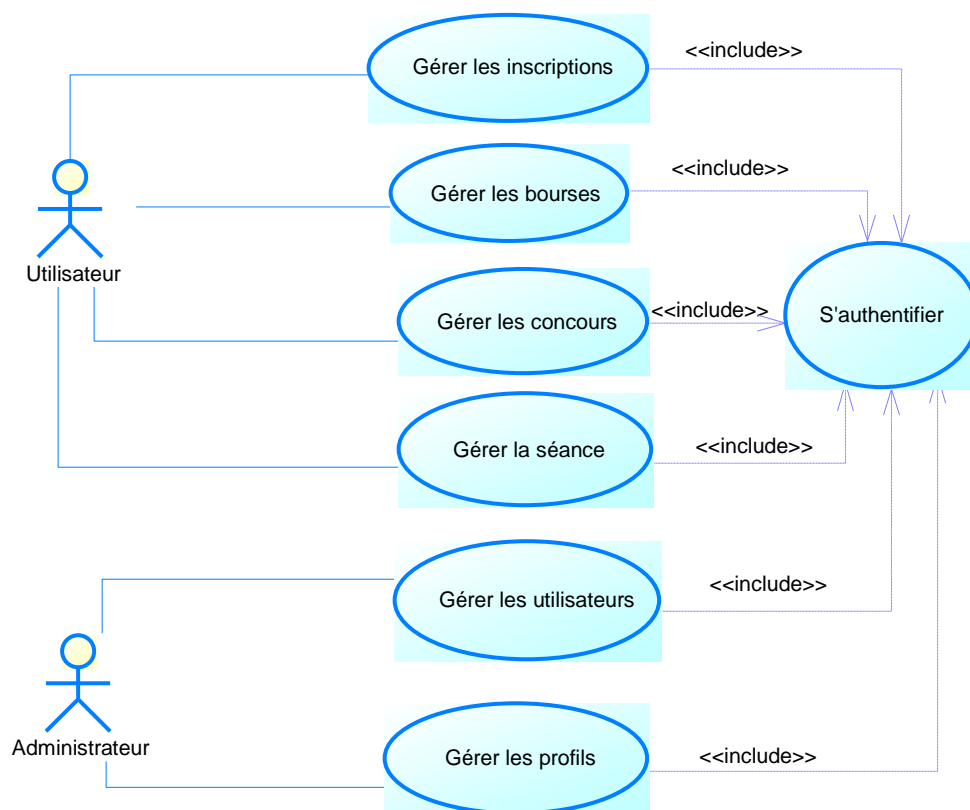


Figure 10 : Diagramme de cas d'utilisation du système

### 3. Vues des processus

#### a. Description textuelle de quelque cas d'utilisation

La description 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 : Utilisateur, Administrateur

Résumé : Permet à un utilisateur du système d'être reconnu par ce dernier en saisissant les paramètres de connexion afin d'y effectuer les actions qui lui sont attribuées.

Auteur : WOAGOU Yendouboame Aimé

Version : 1.0

Date de création : 15-08-2020

Date de modification : 15/09/20

Description des scénarii :

- **Pré conditions**

- Le système est fonctionnel
- La page d'authentification est accessible
- Disposer d'un compte

- **Scénario nominal**

1. L'utilisateur saisit les paramètres de connexion
2. L'utilisateur valide les paramètres saisis
3. Le système vérifie les paramètres saisis par l'utilisateur (A1) (E1)
4. Le système nous donne accès au menu principal

- **Scénario alternatif**

A1 : les paramètres de connexion sont incorrects pour la première ou la deuxième fois : le scénario alternatif est déclenché au point 2 du scénario nominal

- ⇒ Le système signale que le mot de passe est erroné, le système enregistre l'échec : le scénario reprend au point 1 du scénario nominal

- **Scénario d'exception**

E1 : Les paramètres de connexion sont incorrects pour la troisième fois : le scénario est déclenché au point 2 du scénario nominal

- ⇒ Le système indique que le mot de passe est erroné pour la troisième fois et met fin au cas

- **Post conditions**

- L'utilisateur est authentifié
- Le système est toujours fonctionnel

❖ **Enregistrer une inscription**

Titre : Enregistrer une inscription

Acteur : Utilisateur

Résumé : Permet à un utilisateur d'ajouter un nouveau candidat au système

Responsable : WOAGOU Yendouboame Aimé

Version : 1.0

Date de création : 15/08/2020

Date de modification : 15/09/20

Description des scénarii :

- **Pré conditions :**

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

- **Scénario nominal :**

1. L'utilisateur saisit les informations sur le compte du candidat
2. L'utilisateur valide les informations
3. Le système contrôle les informations saisies par l'utilisateur (A1)
4. Le système enregistre les informations saisies par l'utilisateur
5. L'utilisateur reçoit un message de réussite de l'enregistrement

- **Scénario alternatif :**

*A1 : Les informations saisies sont incorrectes* : le scénario alternatif est déclenché au point 3 du scénario nominal

- L'utilisateur est invité à saisir des informations correctes : le scénario reprend au point 1 du scénario nominal

- **Post conditions**

- Nouveau compte candidat créé
- Le système est toujours fonctionnel

❖ **Octroyer une bourse**

Titre : Octroyer une bourse

Acteur : Utilisateur, Candidat

Résumé : Permet à un utilisateur d'octroyer une bourse aux candidats

Responsable : WOAGOU Yendouboame Aimé

Version : 1.0

Date de création : 15/08/2020

Date de modification : 15/09/20

Description des scénarii :

- **Pré conditions :**

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

- **Scénario nominal :**

1. L'utilisateur vérifie les notes du candidat (E1)
2. L'utilisateur vérifie la catégorie sociale du candidat (E2)
3. L'utilisateur valide l'octroi de la bourse
4. Le système informe l'utilisateur de l'octroi avec succès de la bourse

- **Scénario d'exception**

E1, E2 : Les notes sont mauvaises et la catégorie sociale n'est pas la bonne

⇒ Echec d'octroi de bourse

- **Post conditions**

- Le système reste toujours fonctionnel

❖ **Programmer un concours**

Titre : Programmer un concours

Acteur : Utilisateur

Résumé : Permet à l'utilisateur de programmer un concours

Responsable : WOAGOU Yendouboame Aimé

Version : 1.0

Date de création : 15/08/2020

Date de modification : 15/09/20

Description des scénarii :

- **Pré conditions :**
  - Le système est fonctionnel
  - L'utilisateur s'est authentifié
- **Scénario nominal :**
  1. Le système établit la liste des candidats inscrits
  2. L'utilisateur valide les informations
  3. Le système enregistre les informations concernant la liste des inscrits
  4. L'utilisateur programme un concours
- **Post conditions**
  - Le concours est programmé
  - Le système est toujours fonctionnel

### ❖ Enregistrer une séance d'orientation

Titre : Enregistrer une séance d'orientation

Acteur : Utilisateur

Résumé : Permet à un utilisateur d'enregistrer une séance d'orientation dans le système

Responsable : WOAGOU Yendouboame Aimé

Version : 1.0

Date de création : 15/08/2020

Date de modification : 15/09/2020

Description des scénarii :

- **Pré conditions :**
  - Le système est fonctionnel
  - L'utilisateur s'est authentifié
- **Scénario nominal :**
  1. L'utilisateur saisit les informations concernant la séance
  2. L'utilisateur valide les informations
  3. Le système contrôle les informations saisies par l'utilisateur (A1)
  4. Le système enregistre les informations saisies par l'utilisateur
  5. L'utilisateur reçoit un message de réussite de l'enregistrement
- **Scénario alternatif :**

*A1 : Les informations saisies sont incorrectes* : le scénario est déclenché au point 3 du scénario nominal

- L'utilisateur est invité à saisir des informations correctes : le scénario reprend au point 1 du scénario nominal

- **Post conditions**

- Nouvelle séance d'orientation créée
- Le système est toujours fonctionnel

#### 4. Diagramme de séquence

##### a. Généralités

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Le diagramme de séquences système lui, est présenté en montrant l'acteur principal à gauche, puis un objet unique représentant le système en boîte noire, et enfin les éventuels acteurs secondaires sollicités durant le scénario à droite du système. Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction. Les diagrammes de séquence représentent la séquence de messages transmis entre des objets. Ils peuvent également représenter les structures de contrôle entre des objets.

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie :

- Ils sont représentés par des flèches
- Ils sont présentés du haut vers le bas le long des lignes de vie, dans un ordre chronologique

Un message définit une communication particulière entre des lignes de vie (objets ou acteurs).

Plusieurs types de messages existent, dont les plus courants :

- L'envoi d'un signal ;
- L'invocation d'une opération (appel de méthode) ;
- La création ou la destruction d'un objet.

Un message synchrone bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.



- Si un objet A envoie un message synchrone à un objet B, A reste bloqué tant que B n'a pas terminé.
- On peut associer aux messages d'appel de méthode un message de retour (en pointillés) marquant la reprise du contrôle par l'objet émetteur du message synchrone.

Un message asynchrone n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.

#### b. Quelques diagrammes de séquences

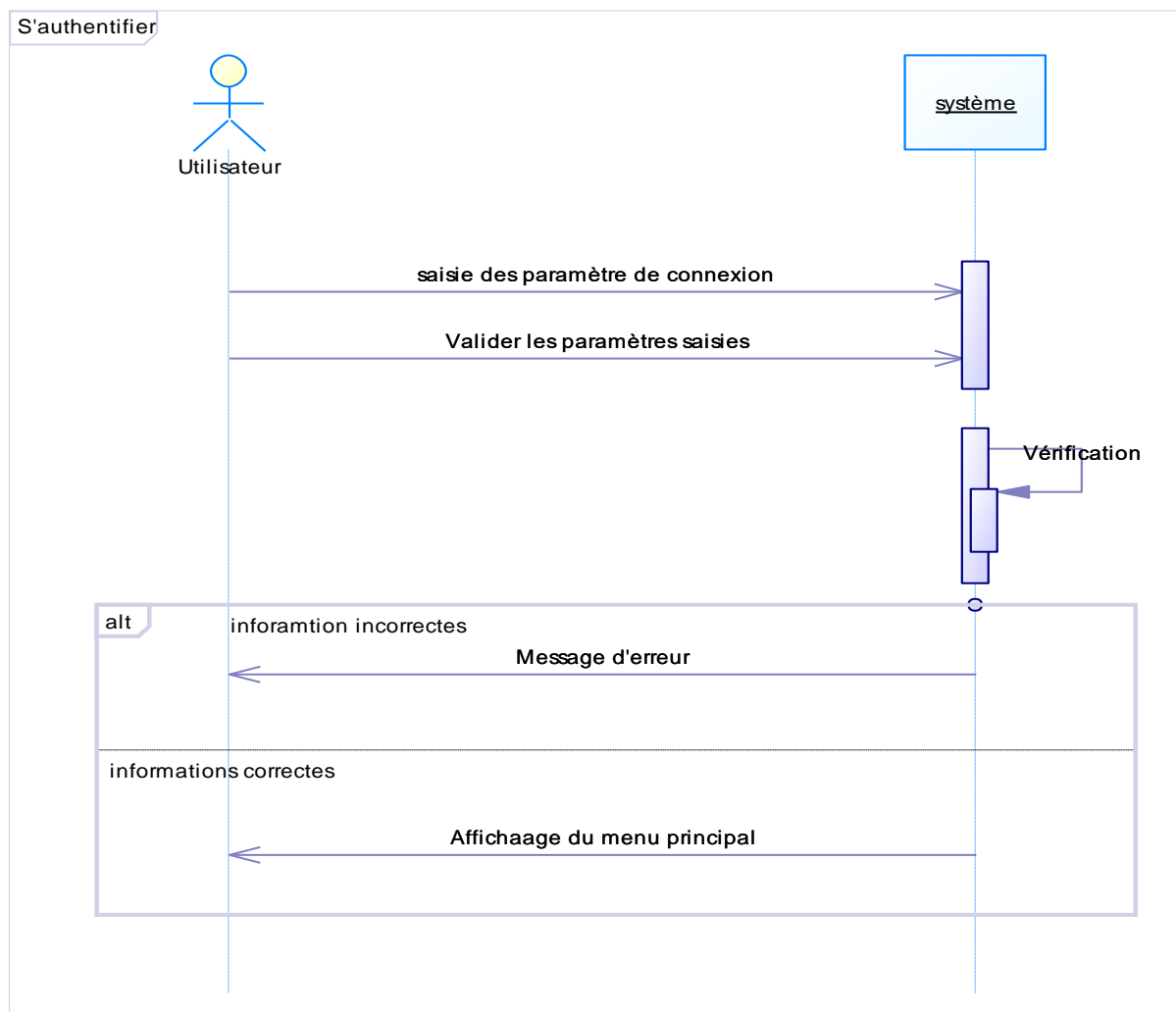


Figure 11 : Diagramme de séquence du cas d'utilisation « S'authentifier »

»

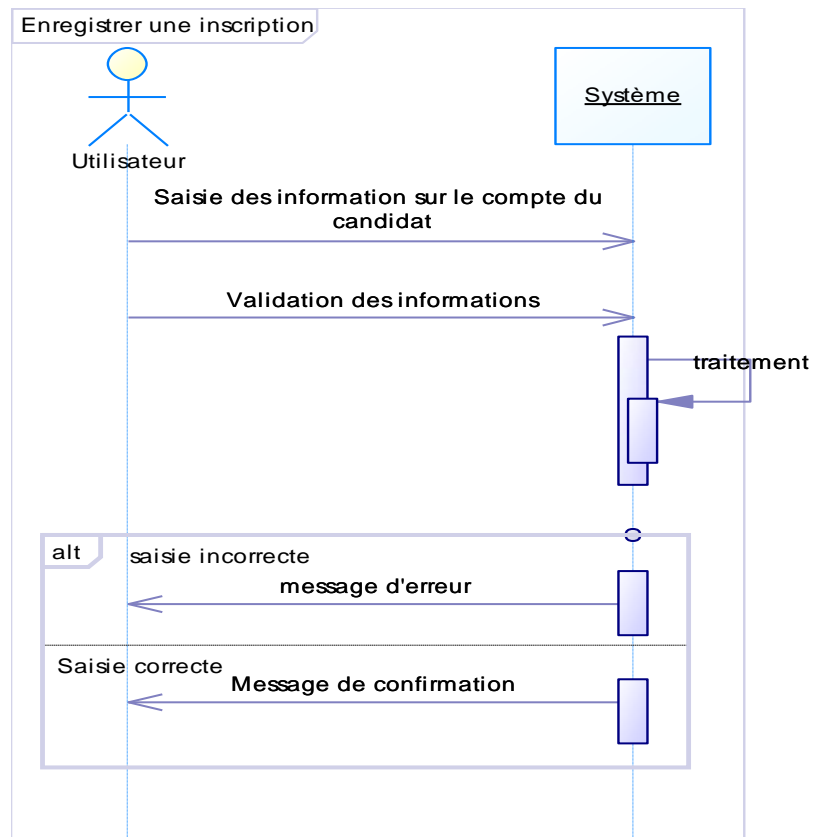


Figure 12 : Diagramme de séquence du cas d'utilisation « enregistrer une inscription »

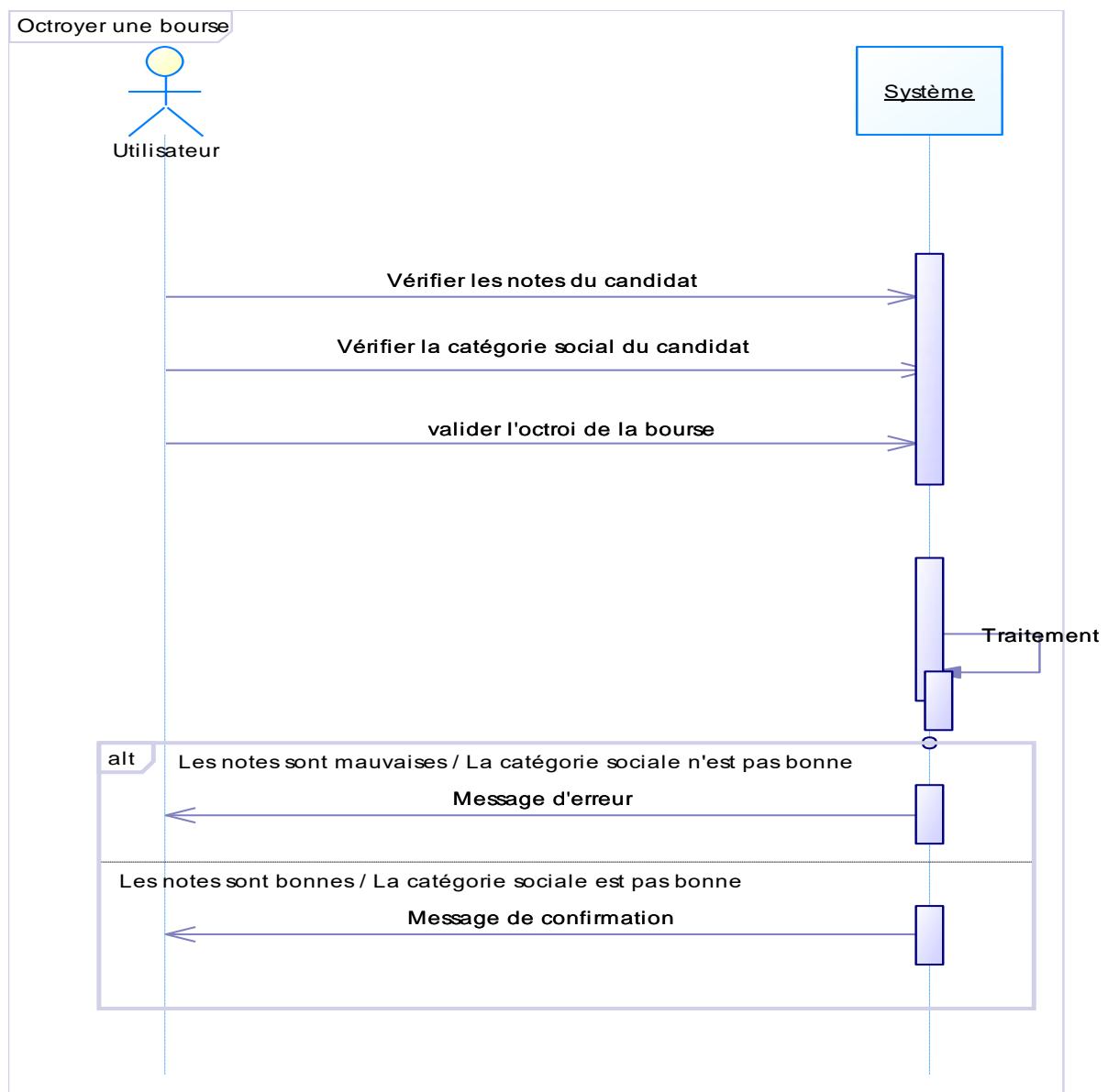


Figure 13 : Diagramme de séquence du cas d'utilisation « Octroyer une bourse »

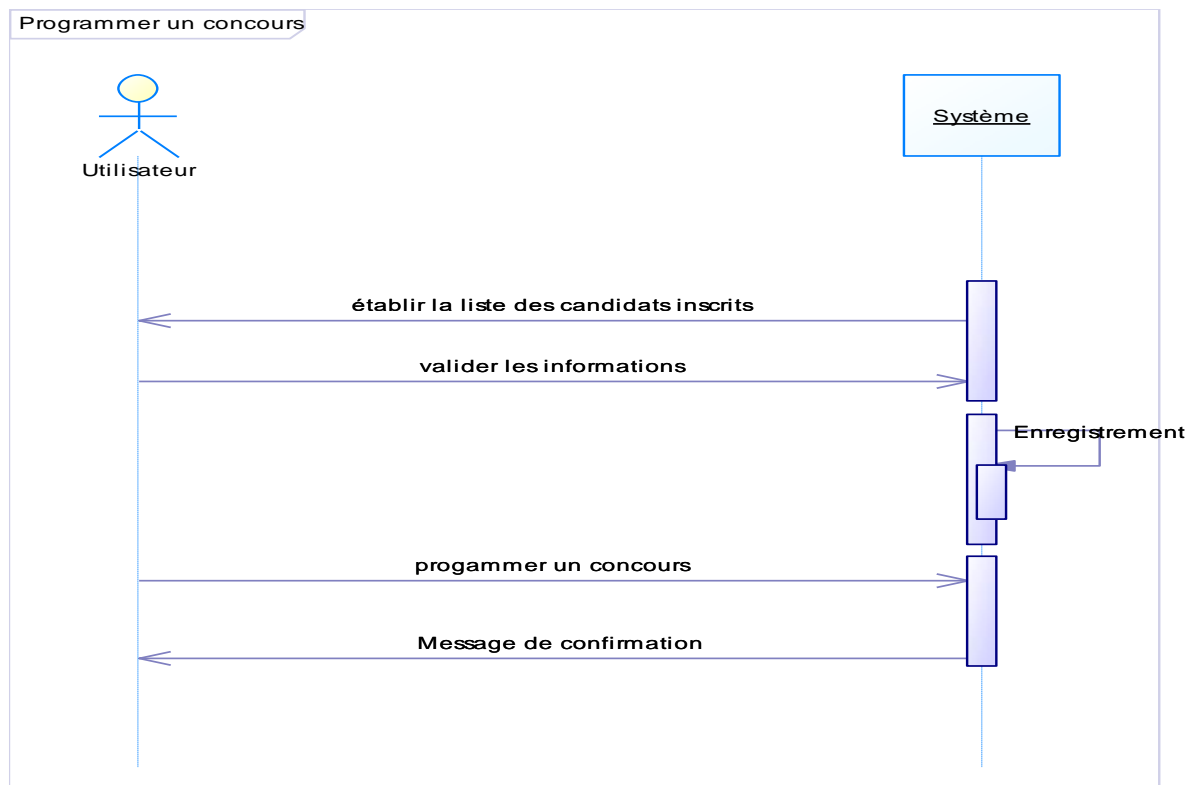


Figure 14 : Diagramme de séquence du cas d'utilisation « Programmer un concours »

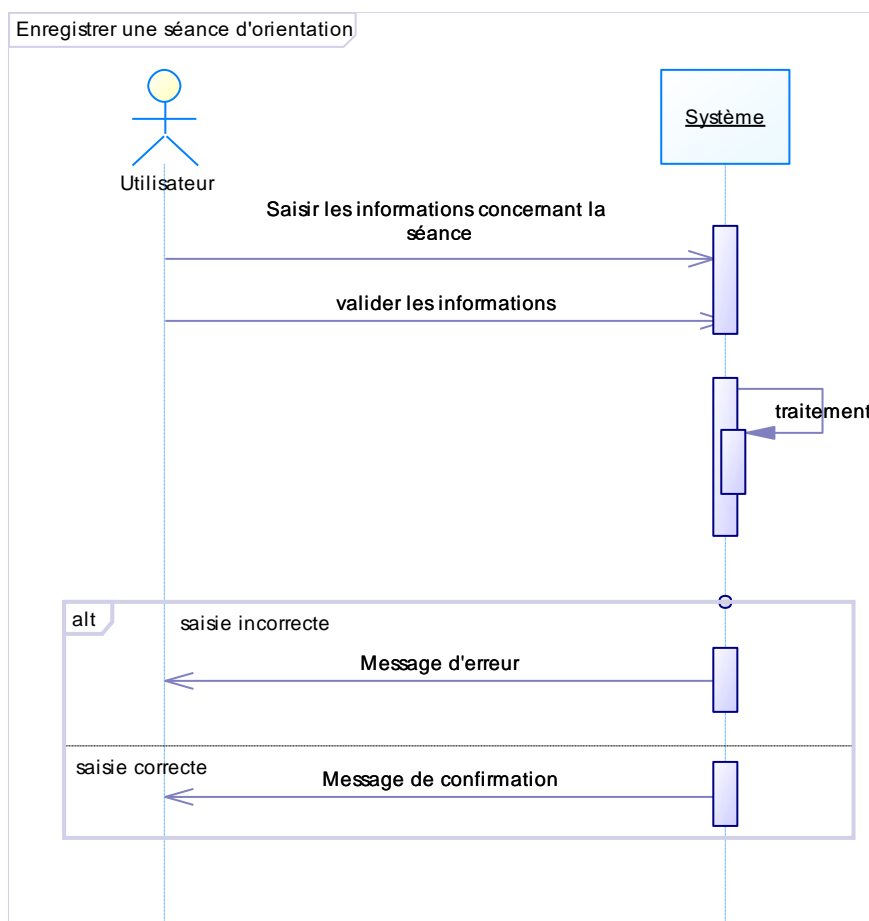


Figure 15 : Diagramme de séquence du cas d'utilisation « enregistrer une séance d'orientation »

## 5. Diagramme d'activité

### a. Généralités

Le diagramme d'activités se présente comme un organisme qui décrit les différents enchainements ou étapes du déroulement d'un cas d'utilisation ou groupe de cas d'utilisation.

Le diagramme d'activité est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Certains des composants les plus courants d'un diagramme d'activités sont :

- Actions : étape dans l'activité où les utilisateurs ou le logiciel exécutent une tâche donnée.
- Nœud de décision : embranchement conditionnel dans le flux qui est représenté par un losange. Il comporte une seule entrée et deux sorties ou plus.

- Flux de contrôle : autre nom des connecteurs qui illustrent le flux entre les étapes du diagramme.
- Nœud de départ : symbolise le début de l'activité. Cet élément est représenté par un cercle noir.
- Nœud de fin : représente l'étape finale de l'activité. Il est représenté par un cercle noir avec un contour.

b. Quelques diagrammes d'activité

Le diagramme d'activités présente uniquement les actions du système. Voici quelques diagrammes d'activité de notre système :

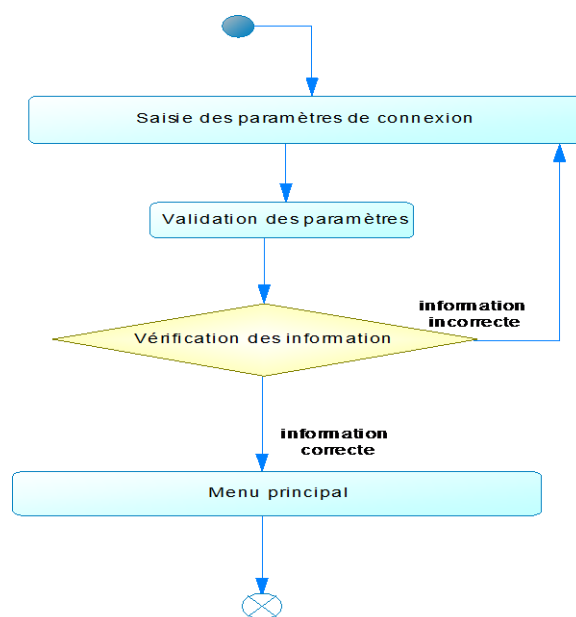


Figure 16 : Diagramme d'activité du cas d'utilisation « s'authentifier »

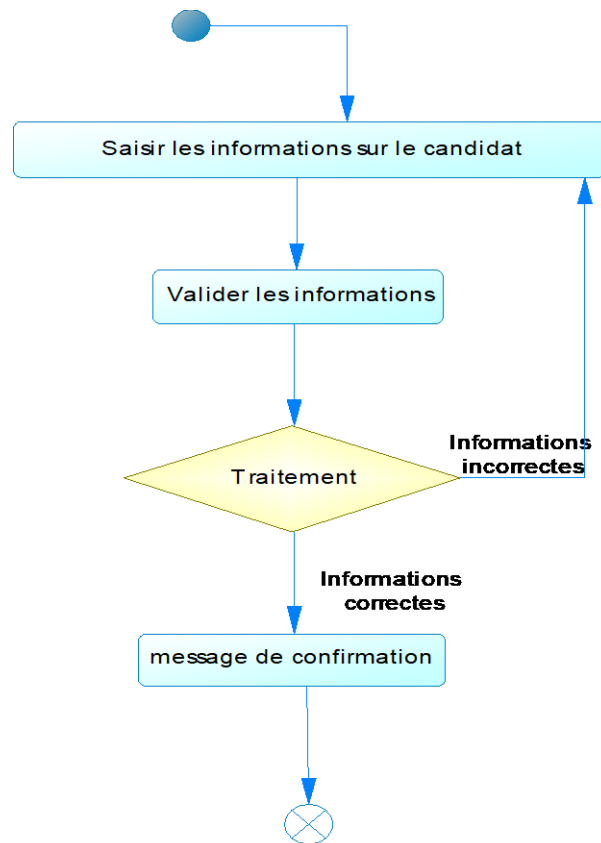


Figure 17 : Diagramme d'activité du cas d'utilisation « enregistrer une inscription »

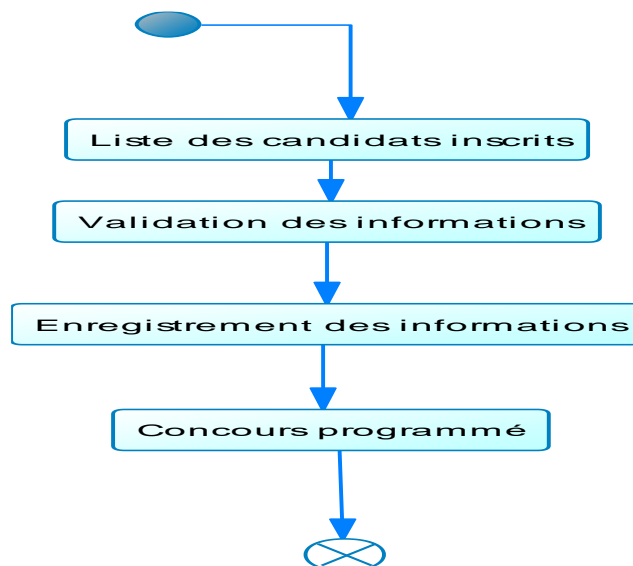


Figure 18 : Diagramme d'activité du cas d'utilisation « Programmer un concours »

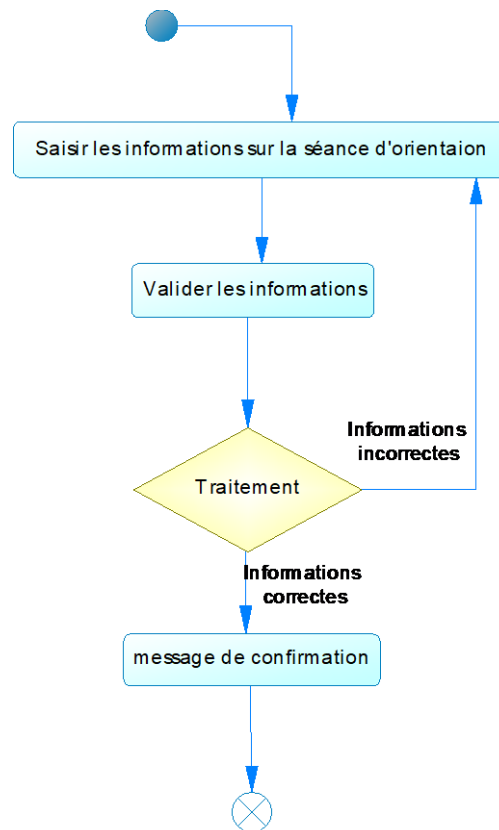


Figure 19 : Diagramme d'activité du cas d'utilisation « enregistrer une séance d'orientation »

»



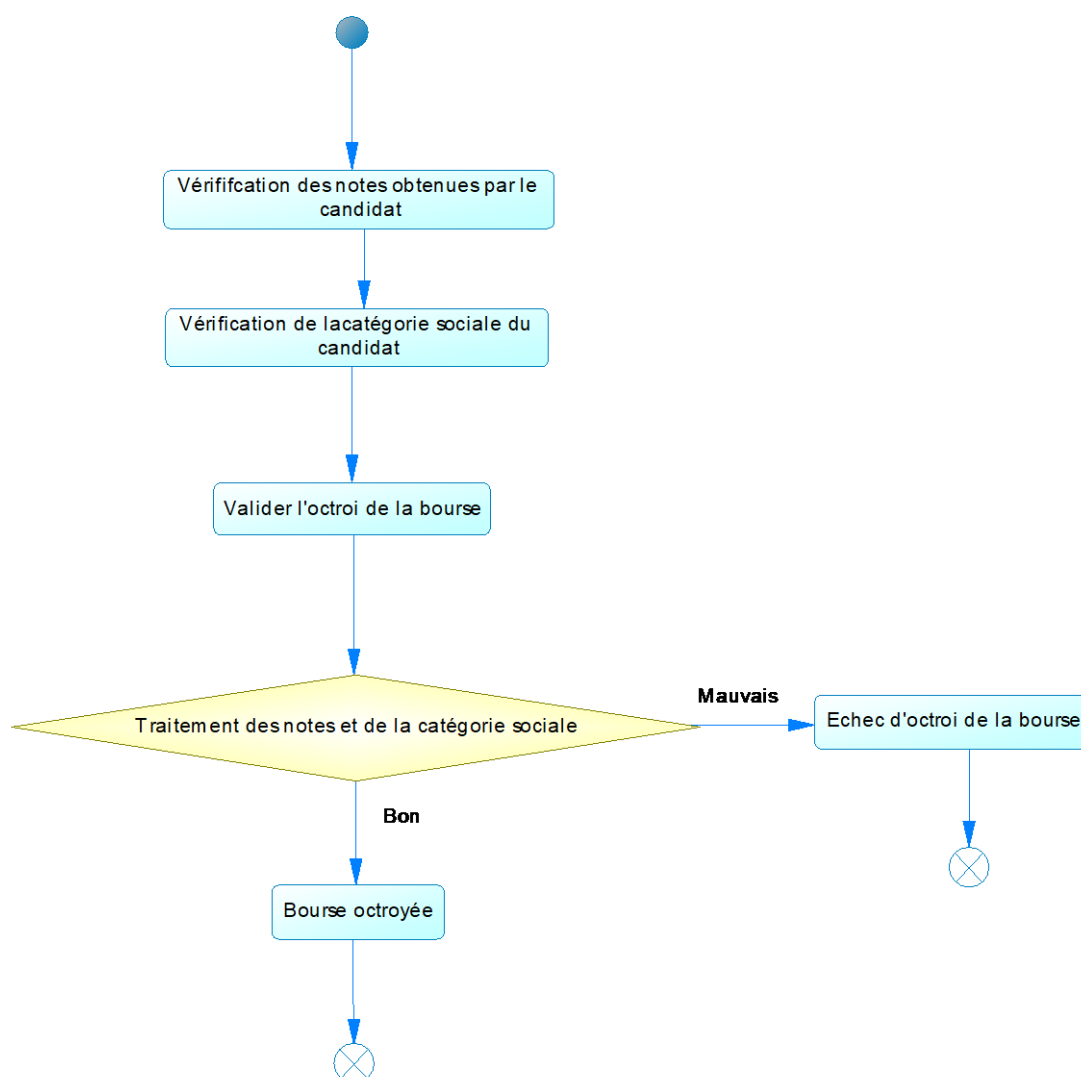


Figure 20 : Diagramme d'activité du cas d'utilisation « octroyer une bourse »

## 6. Vue logique

### a. Généralités sur un diagramme de classes

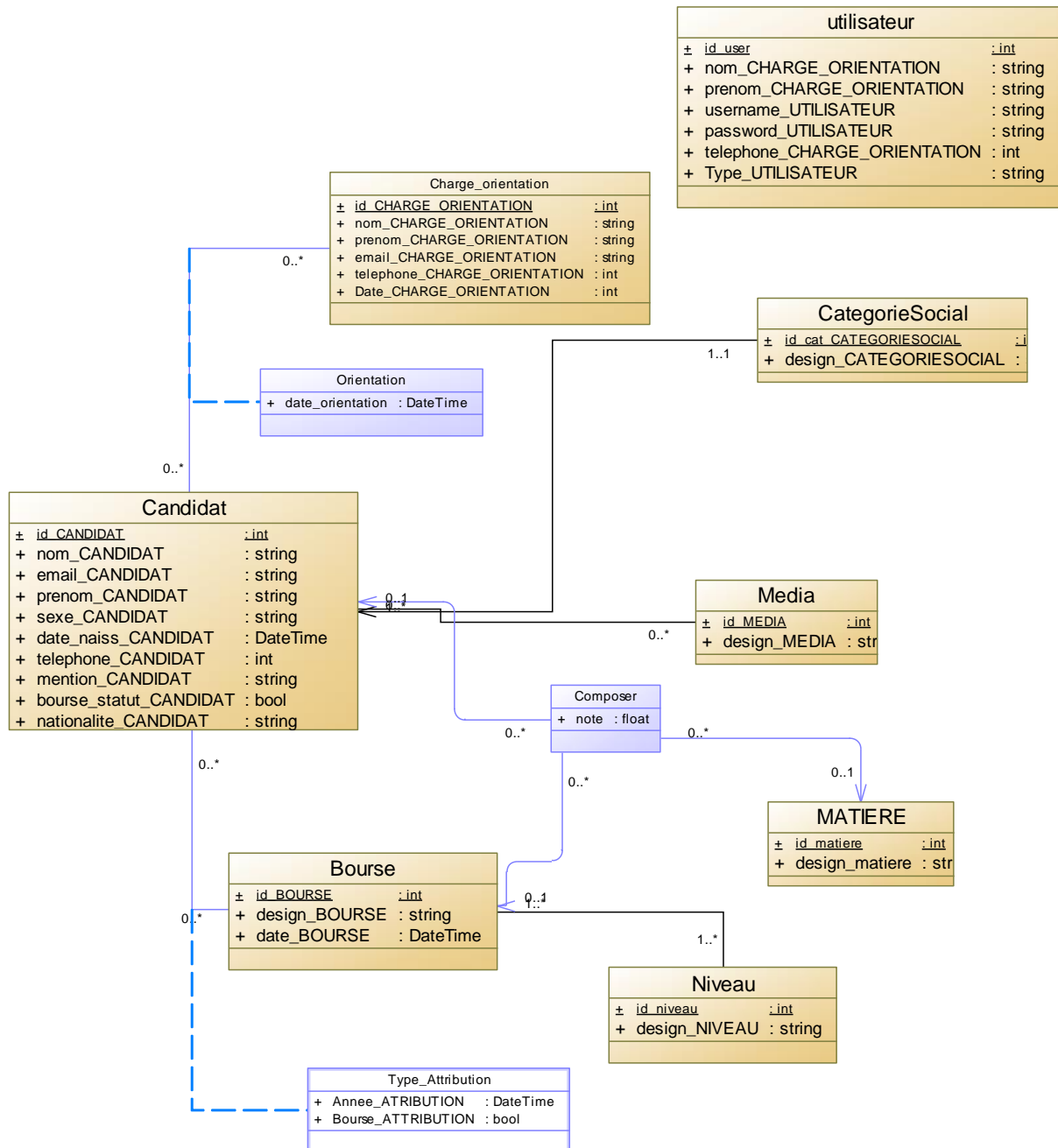
Le diagramme de classe exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classe est de modéliser les entités du système d'informations. Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont structurées, c'est-à-dire qu'elles sont regroupées dans des classes. Le diagramme des classes met en évidence d'éventuelles relations entre ces classes. Le diagramme des classes est le diagramme le plus répandu dans les spécifications d'UML.

Le diagramme de classes comporte 6 concepts que sont :

#### ➤ Classe

- Attribut
- Identification
- Relation
- Opération
- Généralisation/Spécialisation

## b. Représentation du diagramme de classe



---

### *PARTIE III : REALISATION ET MISE EN OEUVRE*

---

## INTRODUCTION

La phase d'analyse abordée dans le rapport de pré programmation nous a permis de cerner les contours du système à automatiser, nous conduisant à la phase de réalisation et de mise en œuvre. Cette partie présentera les diverses technologies employées pour le développement, et le déploiement de l'application.

## I. Mise en Œuvre

### 1. Choix matériel

Notre matériel de développement durant la période de stage est un ordinateur bureau.

Caractéristiques

- Fabricant : HP
- Processeur : Intel® Celeron® CPU N3060 @ 1.60GHz 1.60GHz
- Mémoire RAM : 4Go
- Disque dur : 500Go
- Type de Système : Système d'exploitation 64 bits (Windows 10 Professionnel), processeur x64

### 2. Choix logiciel

Plusieurs outils ont fait partie de l'ouvrage de notre projet. Nous les classons en 2 : les outils d'implémentation de la base de données et les outils de programmation et de développement.

#### a. Les outils d'implémentation de la base de données

Pour la création de la base de données nous avons opter pour : Microsoft SQL Server 2016



Microsoft SQL Server est un système de gestion de base de données relationnelle développé par Microsoft. Ce produit est conçu pour la fonction de base de stockage des données, de données de récupération tel que requis par d'autres applications. Il peut être exécuté sur le même ordinateur ou sur un autre à travers un réseau.

### **Utilisation de SQL SERVER**

- ✓ Créer des bases de données
- ✓ Maintenir les bases de données
- ✓ Analyser les données via SQL Server Analysis Services (SSAS)
- ✓ Générer des rapports par le biais de SQL Server Reporting Services (SSRS)
- ✓ Effectuer des opérations ETL par SQL Server Integration Services (SSIS)

### **Composants de SQL SERVER**

SQL Server fonctionne en architecture client/serveur, il prend donc en charge deux types de composants : le workstation et le server.

- ✓ Workstation components : Ils sont installés sur les machines de chaque xappareil/opérateur SQL Server. Ce ne sont que des interfaces pour interagir avec les composants du serveur. Nous avons notamment SSMS, SSCM, Profiler, BIDS, SQLEM.
- ✓ Server components : Ils sont installés sur le serveur centralisé. Ce sont des services. Nous avons entre autres : SQL Server, SQL Server Agent, SSIS, SSAS, SSRS, navigateur SQL, recherche de texte SQL Server complète.

b. Les outils de programmation et de développement

Pour le développement de l'application nous avons opter pour : Visual Studio 2017



Les environnements de développement intégré (EDI) offrent des outils permettant l'édition du code, le débogage, la génération, la mise en œuvre d'applications. Parmi les EDI les plus populaires, on retrouve en bonne position Visual Studio, EDI propriétaire, développé par Microsoft. Visual Studio est un ensemble d'outils de développement permettant de développer différents types d'applications (Windows, Web, Android, iOS, Cloud, etc.), en utilisant de nombreux langages de programmation dont C#, VB.NET, C++, JavaScript et bien plus. Au fil des versions, Visual Studio est devenu un EDI robuste et peut, selon Microsoft, être « utilisé par n'importe quel développeur, pour mettre en place n'importe quelle application, pour n'importe quelle plateforme ».

En effet Visual studio dispose de nombreuses documentations et aides en ligne pour son apprentissage et d'un mode de débogage sans pareille. De plus la beauté de l'EDI de chez Microsoft et ses librairies intégrées font partis des grandes raisons qui nous ont poussé à l'EDI de la géante firme américaine.

### 3. Sécurité de l'application

La mise en place de la sécurité est une étape importante dans la conception d'une application. Notre application Desktop présente un modèle de sécurité basé sur le code. Ce modèle vient donc s'ajouter aux éventuels schémas de sécurité déjà en place dans le système informatique. Dans cet ordre d'idées, nous avons eu à prendre des mesures pour garantir au maximum la confidentialité, l'intégrité et la disponibilité de l'application et des données traitées :

- ✓ L'accès par authentification à l'application ;
- ✓ Les utilisateurs simples n'ont pas accès au menu Utilisateur ;
- ✓ Les données validées sont contrôlées avant leur insertion dans la base de données ;
- ✓ Les opérations réalisées sont archivées dans le système pour permettre un meilleur suivi ;

Nous notons aussi que pour chaque action à exécuter, se présente un message de confirmation, à valider avant l'effectivité de l'action pour éviter des erreurs de manipulation.

## II. Présentation de l'application

### 1. Présentation

L'application que nous avons développée a pour nom **GESCONDORI**. Cette dernière est écrite en C# avec Visual Studio 2017 et permet la gestion des Concours de bourses et des séances d'orientation de l'ESGIS. Elle offre des interfaces simples et facilement manipulables.

#### a. Script de création de la base de données

```
if exists (select 1
          from sysindexes
          where id = object_id('Affilier')
          and name = 'AFFILIER_FK2'
          and indid > 0
          and indid < 255)
drop index Affilier.AFFILIER_FK2
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('Affilier')
          and name = 'AFFILIER_FK'
          and indid > 0
          and indid < 255)
drop index Affilier.AFFILIER_FK
go
```

```
if exists (select 1
          from sysobjects
          where id = object_id('Affilier')
          and type = 'U')
drop table Affilier
go
```

```
if exists (select 1
          from sysobjects
```

```

where id = object_id('BOURSE')
and type = 'U')
drop table BOURSE
go

if exists (select 1
from sysindexes
where id = object_id('CANDIDAT')
and name = 'APPARTENIR_FK'
and indid > 0
and indid < 255)
drop index CANDIDAT.APPARTENIR_FK
go

```

```

if exists (select 1
from sysobjects
where id = object_id('CANDIDAT')
and type = 'U')
drop table CANDIDAT
go

```

```

if exists (select 1
from sysobjects
where id = object_id('CATEGORIESOCIAL')
and type = 'U')
drop table CATEGORIESOCIAL
go

```

```

if exists (select 1
from sysobjects
where id = object_id('CHARGE_ORIENTATION')
and type = 'U')
drop table CHARGE_ORIENTATION
go

```

```

if exists (select 1

```



```

from sysindexes
where id = object_id('Composer')
and name = 'COMPOSER_FK3'
and indid > 0
and indid < 255)
drop index Composer.COMPOSER_FK3
go

```

```

if exists (select 1
from sysindexes
where id = object_id('Composer')
and name = 'COMPOSER_FK2'
and indid > 0
and indid < 255)
drop index Composer.COMPOSER_FK2
go

```

```

if exists (select 1
from sysindexes
where id = object_id('Composer')
and name = 'COMPOSER_FK'
and indid > 0
and indid < 255)
drop index Composer.COMPOSER_FK
go

```

```

if exists (select 1
from sysobjects
where id = object_id('Composer')
and type = 'U')
drop table Composer
go

```

```

if exists (select 1
from sysindexes
where id = object_id('Consulter')

```

```

    and name = 'CONSULTER_FK2'
    and indid > 0
    and indid < 255)
drop index Consulter.CONSULTER_FK2
go

```

```

if exists (select 1
    from sysindexes
    where id = object_id('Consulter')
    and name = 'CONSULTER_FK'
    and indid > 0
    and indid < 255)
drop index Consulter.CONSULTER_FK
go

```

```

if exists (select 1
    from sysobjects
    where id = object_id('Consulter')
    and type = 'U')
drop table Consulter
go

```

```

if exists (select 1
    from sysobjects
    where id = object_id('MATIERE')
    and type = 'U')
drop table MATIERE
go

```

```

if exists (select 1
    from sysobjects
    where id = object_id('MEDIA')
    and type = 'U')
drop table MEDIA
go

```

```

if exists (select 1
           from sysobjects
           where id = object_id('NIVEAU')
           and type = 'U')
drop table NIVEAU
go

if exists (select 1
           from sysindexes
           where id = object_id('Orientation')
           and name = 'ORIENTATION_FK2'
           and indid > 0
           and indid < 255)
drop index Orientation.ORIENTATION_FK2
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('Orientation')
           and name = 'ORIENTATION_FK'
           and indid > 0
           and indid < 255)
drop index Orientation.ORIENTATION_FK
go

```

```

if exists (select 1
           from sysobjects
           where id = object_id('Orientation')
           and type = 'U')
drop table Orientation
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('TypeAttribution')
           and name = 'TYPEATTRIBUTION_FK2'

```

```

    and indid > 0
    and indid < 255)
drop index TypeAttribution.TYPEATTRIBUTION_FK2
go

if exists (select 1
    from sysindexes
    where id = object_id('TypeAttribution')
    and name = 'TYPEATTRIBUTION_FK'
    and indid > 0
    and indid < 255)
drop index TypeAttribution.TYPEATTRIBUTION_FK
go

if exists (select 1
    from sysobjects
    where id = object_id('TypeAttribution')
    and type = 'U')
drop table TypeAttribution
go

if exists (select 1
    from sysobjects
    where id = object_id('UTILISATEUR')
    and type = 'U')
drop table UTILISATEUR
go

/*=====*/
/* Table: Affilier */
/*=====*/

create table Affilier (
    ID_NIVEAU      int          not null,
    ID_BOURSE      int          not null,
    constraint PK_AFFILIER primary key (ID_NIVEAU, ID_BOURSE)
)

```

go

/\*=====\*/

/\* Index: AFFILIER\_FK \*/

/\*=====\*/

create index AFFILIER\_FK on Affilier (

ID\_NIVEAU ASC

)

go

/\*=====\*/

/\* Index: AFFILIER\_FK2 \*/

/\*=====\*/

create index AFFILIER\_FK2 on Affilier (

ID\_BOURSE ASC

)

go

/\*=====\*/

/\* Table: BOURSE \*/

/\*=====\*/

create table BOURSE (

ID\_BOURSE int IDENTITY not null,

DESIGN\_BOURSE varchar(254) not null,

DATE\_BOURSE datetime not null,

constraint PK\_BOURSE primary key nonclustered (ID\_BOURSE)

)

go

/\*=====\*/

/\* Table: CATEGORIESOCIAL \*/

/\*=====\*/

create table CATEGORIESOCIAL (

ID\_CAT\_CATEGORIESOCIAL int IDENTITY not null,

DESIGN\_CATEGORIESOCIAL varchar(254) null,

constraint PK\_CATEGORIESOCIAL primary key nonclustered (ID\_CAT\_CATEGORIESOCIAL)

```

)
go

/*=====*/
/* Table: CANDIDAT */
/*=====*/

create table CANDIDAT (
    ID_CANDIDAT      int  IDENTITY      not null,
    ID_CAT_CATEGORIESOCIAL int      ,
    NOM_CHARGE_ CANDIDAT varchar(254)    null,
    EMAIL_CHARGE_ CANDIDAT varchar(254)    null,
    PRENOM_CHARGE_ CANDIDAT varchar(254)    null,
    SEXE_CANDIDAT    varchar(254)    null,
    DATE_NAISS_CANDIDAT datetime    null,
    TELEPHONE_CHARGE_ CANDIDAT int      null,
    MENTION          varchar(254)    null,
    BOURSE_STATUT_ CANDIDAT bit        null,
    NATIONALITE_ CANDIDAT varchar(254)    null,
    TYPECANDIDAT_ CANDIDAT varchar(254)    null,
    constraint PK_CANDIDAT primary key nonclustered (ID_CANDIDAT)
)
go

/*=====*/
/* Index: APPARTENIR_FK */
/*=====*/

create index APPARTENIR_FK on CANDIDAT (
    ID_CAT_CATEGORIESOCIAL ASC
)
go

/*=====*/
/* Table: CHARGE_ORIENTATION */
/*=====*/

create table CHARGE_ORIENTATION (
    ID_CHARGE_ORIENTATION int  IDENTITY      not null,

```

```

    NOM_CHARGE_ORIENTATION varchar(254)      null,
    PRENOM_CHARGE_ORIENTATION varchar(254)    null,
    EMAIL_CHARGE_ORIENTATION varchar(254)     null,
    TELEPHONE_CHARGE_ORIENTATION int          null,
    DATE_CHARGE_ORIENTATION int               null,
    constraint PK_CHARGE_ORIENTATION primary key nonclustered (ID_CHARGE_ORIENTATION)
)
go

/*=====*/
/* Table: Composer */
/*=====*/

create table Composer (
    ID_BOURSE      int      null,
    ID_MATIERE     int      null,
    ID_CANDIDAT    int      null,
    NOTE          float     null
)
go

/*=====*/
/* Index: COMPOSER_FK */
/*=====*/

create index COMPOSER_FK on Composer (
    ID_BOURSE ASC
)
go

/*=====*/
/* Index: COMPOSER_FK2 */
/*=====*/

create index COMPOSER_FK2 on Composer (
    ID_MATIERE ASC
)
go

```

```

/*=====*/
/* Index: COMPOSER_FK3 */
/*=====*/
create index COMPOSER_FK3 on Composer (
ID_CANDIDAT ASC
)
go

```

```

/*=====*/
/* Table: Consulter */
/*=====*/
create table Consulter (
ID_CANDIDAT int not null,
ID_MEDIA int not null,
constraint PK_CONSULTER primary key (ID_CANDIDAT, ID_MEDIA)
)
go

```

```

/*=====*/
/* Index: CONSULTER_FK */
/*=====*/
create index CONSULTER_FK on Consulter (
ID_CANDIDAT ASC
)
go

```

```

/*=====*/
/* Index: CONSULTER_FK2 */
/*=====*/
create index CONSULTER_FK2 on Consulter (
ID_MEDIA ASC
)
go

```

```

/*=====*/
/* Table: MATIERE */

```



```

/*=====*/

create table MATIERE (
    ID_MATIERE      int  IDENTITY      not null,
    DESIGN_MATIERE  varchar(254)      null,
    constraint PK_MATIERE primary key nonclustered (ID_MATIERE)
)
go

ALTER TABLE CANDIDAT ADD SERIE NVARCHAR(250);

/*=====*/
/* Table: MEDIA */
/*=====*/

create table MEDIA (
    ID_MEDIA      int  IDENTITY      not null,
    DESIGN_MEDIA   varchar(254)      null,
    constraint PK_MEDIA primary key nonclustered (ID_MEDIA)
)
go

/*=====*/
/* Table: NIVEAU */
/*=====*/

create table NIVEAU (
    ID_NIVEAU      int  IDENTITY      not null,
    DESIGN_NIVEAU   varchar(254)      null,
    constraint PK_NIVEAU primary key nonclustered (ID_NIVEAU)
)
go

/*=====*/
/* Table: Orientation */
/*=====*/

create table Orientation (
    ID_CANDIDAT      int      not null,
    ID_CHARGE_ORIENTATION int      not null,

```

```

DATE_ORIENTATION    datetime        null,
constraint PK_ORIENTATION primary key (ID_CANDIDAT, ID_CHARGE_ORIENTATION)
)
go

/*=====*/
/* Index: ORIENTATION_FK */
/*=====*/
create index ORIENTATION_FK on Orientation (
ID_CANDIDAT ASC
)
go

/*=====*/
/* Index: ORIENTATION_FK2 */
/*=====*/
create index ORIENTATION_FK2 on Orientation (
ID_CHARGE_ORIENTATION ASC
)
go

/*=====*/
/* Table: TypeAttribution */
/*=====*/
create table TypeAttribution (
    ID_BOURSE          int            not null,
    ID_CANDIDAT         int            not null,
    ANNEE_ATTRIBUTION   datetime       null,
    BORSE_ATTRIBUTION   bit            null,
    constraint PK_TYPEATTRIBUTION primary key (ID_BOURSE, ID_CANDIDAT)
)
go

/*=====*/
/* Index: TYPEATTRIBUTION_FK */
/*=====*/

```

```

create index TYPEATTRIBUTION_FK on TypeAttribution (
ID_BOURSE ASC
)
go

/*=====*/
/* Index: TYPEATTRIBUTION_FK2 */
/*=====*/

create index TYPEATTRIBUTION_FK2 on TypeAttribution (
ID_CANDIDAT ASC
)
go

/*=====*/
/* Table: UTILISATEUR */
/*=====*/

create table UTILISATEUR (
    ID_USER          int    IDENTITY          not null,
    NOM_CHARGE_ORIENTATION varchar(254)      not null,
    PRENOM_CHARGE_ORIENTATION varchar(254)    not null,
    USERNAME_UTILISATEUR varchar(254)         not null,
    PASSWORD_UTILISATEUR varchar(254)         not null,
    TELEPHONE_CHARGE_ORIENTATION int          not null,
    TYPE_UTILISATEUR  varchar(254)            not null,
    constraint PK_UTILISATEUR primary key nonclustered (ID_USER)
)
go

ALTER TABLE CANDIDAT ADD CONSTRAINT FK_CATEGORIESOCIAL
FOREIGN KEY (ID_CAT_CATEGORIESOCIAL) REFERENCES CATEGORIESOCIAL
(ID_CAT_CATEGORIESOCIAL);

```

## b. Quelques masques de saisies et codes source de l'application

### ✓ Fenêtre de connexion

Code source

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Gestion_des_Concours_BourseOrientation.PL
{
    public partial class FRM_Connexion : Form
    {
        private DbGesconcontextt db;
        BL.CLS_Connexion C = new BL.CLS_Connexion();
        private Form frmmenu;
        //classe connexion

        public FRM_Connexion(Form Menu)
        {
            InitializeComponent();
            db = new DbGesconcontextt();
            this.frmmenu = Menu;
            //initialiser base de donnée
        }

        string testobligatoire()
        {
            if(txtNom.Text==" " ||txtNom.Text== "Nom d'utilisateur")
            {
                return "Entrer votre Nom";
            }
            if(txtMotdepasse.Text==" " || txtMotdepasse.Text == "Mot de Passe")

```

```

    {
        return "Entrer votre Mot de Passe";
    }
    return null;
}

private void FRM_Connexion_Load(object sender, EventArgs e)
{
}

private void btnQuitter_Click(object sender, EventArgs e)
{
    this.Close();
}

private void txtNom_Enter(object sender, EventArgs e)
{
    if(txtNom.Text== "Nom d'utilisateur")
    {
        txtNom.Text = "";
        txtNom.ForeColor = Color.WhiteSmoke;
    }
}

private void txtMotdepasse_Enter(object sender, EventArgs e)
{
    if (txtMotdepasse.Text == "Mot de Passe")
    {
        txtMotdepasse.Text = "";
        txtMotdepasse.UseSystemPasswordChar = false;
        txtMotdepasse.PasswordChar = '*';
        txtMotdepasse.ForeColor = Color.WhiteSmoke;
    }
}

private void txtNom_Leave(object sender, EventArgs e)
{
    if (txtNom.Text == "")
    {
        txtNom.Text = "Nom d'utilisateur";
        txtNom.ForeColor = Color.Silver;
    }
}

private void txtMotdepasse_Leave(object sender, EventArgs e)
{
    if (txtMotdepasse.Text == "")
    {
        txtMotdepasse.Text = "Mot de Passe";
        txtMotdepasse.UseSystemPasswordChar = true;
        txtMotdepasse.ForeColor = Color.Silver;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (testobligatoire() == null)
    {
        if (C.ConnexionValide(db, txtNom.Text, txtMotdepasse.Text)==true)
        {
            (frmmenu as FRM_Menu).activerForm();
        }
    }
}

```

```

        this.Close();
    }
    else
    {
        MessageBox.Show( "Connexion a échoué", "Connexion",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        MessageBox.Show(testobligatoire(), "Obligatoire",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void txtNom_TextChanged(object sender, EventArgs e)
{
}
}
}

```

- ✓ Fenêtre d'enregistrement d'un Candidat

Code source

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Net.Mail;

```

```

using Gestion_des_Concours_BourseOrientation.BL;

namespace Gestion_des_Concours_BourseOrientation.PL
{
    public partial class FRM_Ajouter_Modifier_Candidat : Form
    {
        private UserControl usclient;
        public FRM_Ajouter_Modifier_Candidat(UserControl UserC)
        {
            InitializeComponent();
            this.usclient = UserC;
        }
        CLS_Etudiant c = new CLS_Etudiant();
        string testObligatoire()
        {
            if (txtNom2.Text == "" || txtNom2.Text == "Nom")
            {
                return "Entrer le nom du candidat";
            }
            if (txtPrenom.Text == "" || txtPrenom.Text == "Prénom")
            {
                return "Entrer le Prénom du candidat";
            }
            if (cbSexe.Text == "")
            {
                return "Entrer le Sexe du candidat";
            }
            if (txtTel.Text == "" || txtTel.Text == "Telephone")
            {
                return "Entrer le Telephone du candidat";
            }
            if (txtMail.Text == "" || txtMail.Text == "Email")
            {
                return "Entrer le Mail du candidat";
            }
            if (txtNationalite.Text == "" || txtNom2.Text == "Nationalité")
            {
                return "Entrer la Nationalité du candidat";
            }
            if (cbSerie.Text == "")
            {
                return "Entrer la Série de provenance du candidat";
            }
            if (cbMention.Text == "")
            {
                return "Entrer la Mention eu par le candidat au BAC";
            }
            if (cbCat.Text == "")
            {
                return "Entrer la Catégorie sociale du candidat";
            }
            if (cbType.Text == "")
            {
                return "Entrer le type de candidat";
            }
            if (cbNiveaux.Text == "")
            {
                return "Entrer le Niveau du candidat";
            }
        }
    }
}

```

```

        if(cbMedia.Text == "")
        {
            return "Entrer le Média";
        }
        if (txtMail.Text != "" || txtMail.Text != "Email")
        {
            try
            {
                new MailAddress(txtMail.Text); //pour verifier si le mail est
valide
            }
            catch (Exception e)
            {
                return "Email invalide";
            }
        }

        return null;
    }

    private void FRM_Ajouter_Modifier_Candidat_Load(object sender, EventArgs e)
    {

    }

    private void panel10_Paint(object sender, PaintEventArgs e)
    {

    }

    private void textBox5_TextChanged(object sender, EventArgs e)
    {

    }

    private void txtNom2_Enter(object sender, EventArgs e)
    {
        if (txtNom2.Text == "Nom")
        {
            txtNom2.Text = "";
            txtNom2.ForeColor = Color.WhiteSmoke;
        }
    }

    private void txtNom2_Leave(object sender, EventArgs e)
    {

        if (txtNom2.Text == "")
        {
            txtNom2.Text = "Nom";
            txtNom2.ForeColor = Color.Silver;
        }
    }

    private void txtPrenom_Enter(object sender, EventArgs e)
    {
        if (txtPrenom.Text == "Prénom")
        {
            txtPrenom.Text = "";
            txtPrenom.ForeColor = Color.WhiteSmoke;
        }
    }
}

```



```

private void txtPrenom_Leave(object sender, EventArgs e)
{
    if (txtPrenom.Text == "")
    {
        txtPrenom.Text = "Prenom";
        txtPrenom.ForeColor = Color.Silver;
    }
}

private void pictureBox1_Click(object sender, EventArgs e)
{
    Close();
}

private void txtPrenom_TextChanged(object sender, EventArgs e)
{
}

private void txtTel_Enter(object sender, EventArgs e)
{
    if (txtTel.Text == "Telephone")
    {
        txtTel.Text = "";
        txtTel.ForeColor = Color.WhiteSmoke;
    }
}

private void txtTel_Leave(object sender, EventArgs e)
{
    if (txtTel.Text == "")
    {
        txtTel.Text = "Telephone";
        txtTel.ForeColor = Color.Silver;
    }
}

private void txtMail_Enter(object sender, EventArgs e)
{
    if (txtMail.Text == "Email")
    {
        txtMail.Text = "";
        txtMail.ForeColor = Color.WhiteSmoke;
    }
}

private void txtMail_Leave(object sender, EventArgs e)
{
    if (txtMail.Text == "")
    {
        txtMail.Text = "Email";
        txtMail.ForeColor = Color.Silver;
    }
}

private void txtNationalite_TextChanged(object sender, EventArgs e)
{
}

private void txtNationalite_Enter(object sender, EventArgs e)

```

```

{
    if (txtNationalite.Text == "Nationalité")
    {
        txtNationalite.Text = "";
        txtNationalite.ForeColor = Color.WhiteSmoke;
    }
}

private void txtNationalite_Leave(object sender, EventArgs e)
{
    if (txtNationalite.Text == "")
    {
        txtNationalite.Text = "Nationalité";
        txtNationalite.ForeColor = Color.Silver;
    }
}

private void txtTel_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar < 48 || e.KeyChar > 57)//code ascii des numeros
    {
        e.Handled = true;
    }
    if (e.KeyChar == 8)
    {
        e.Handled = false;
    }
}

private void btnAjoutercandidat_Click(object sender, EventArgs e)
{
    if (testObligatoire() != null)
    {
        MessageBox.Show(testObligatoire(), "Obligatoire",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }

    else
    {
        BL.CLS_Etudiant cLS_Etudiant = new BL.CLS_Etudiant();
        if (cLS_Etudiant.Ajouter_CANDIDAT(txtNom2.Text, txtPrenom.Text,
        cbSexe.Text, cbSerie.Text, int.Parse(txtTel.Text), txtMail.Text, txtNationalite.Text,
        cbMention.Text, cbCat.Text, cbType.Text, cbNiveaux.Text, cbMedia.Text) == true)
        {
            MessageBox.Show("Candidat Ajouter avec
succès", "Ajouter", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
            (usclient as USER_Liste_Client).Actualiserdatagrid();
        }
        else
        {
            MessageBox.Show("Le candidat existe deja", "Echec",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void btActualisercandidat_Click(object sender, EventArgs e)
{
    txtNom2.Text = "Nom"; txtNom2.ForeColor = Color.Silver;
}

```

```

txtPrenom.ForeColor = Color.Silver; txtPrenom.Text = "Prénom";
cbSexe.Items.Clear();
cbNiveaux.Items.Clear();
cbMedia.Items.Clear();
txtTel.ForeColor = Color.Silver; txtTel.Text = "Telephone";
txtMail.Text = ""; txtMail.ForeColor = Color.Silver;
txtNationalite.ForeColor = Color.Silver; txtNom2.Text = "Nationalité";
cbSerie.Items.Clear();
cbMention.Items.Clear();
cbCat.Items.Clear();
cbType.Items.Clear();
    }
}
}

```

- ✓ Fenêtre d'enregistrement d'une Séance d'orientation

Code source

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Net.Mail;

namespace Gestion_des_Concours_BourseOrientation.PL
{
    public partial class FRM_Ajouter_Modifier_Orientation : Form
    {
        public FRM_Ajouter_Modifier_Orientation()
        {

```

```

InitializeComponent();
}
string testObligatoire()
{
    if (txtNomcharge.Text == "" || txtNomcharge.Text == "Nom")
    {
        return "Entrer le nom du Charge de l'orientation";
    }
    if (txtPrenomcharge.Text == "" || txtPrenomcharge.Text == "Prénom")
    {
        return "Entrer le Prénom du Charge de l'orientation";
    }

    if (txtTelcharge.Text == "" || txtTelcharge.Text == "Telephone")
    {
        return "Entrer le Telephone du Charge de l'orientation";
    }
    if (txtMailcharge.Text == "" || txtMailcharge.Text == "Email")
    {
        return "Entrer le Mail du Charge de l'orientation";
    }

    if (txtMailcharge.Text != "" || txtMailcharge.Text != "Email")
    {
        try
        {
            new MailAddress(txtMailcharge.Text); //pour verifier si le mail est valide
        }
        catch (Exception e)
        {
            return "Email invalide";
        }
    }

    return null;
}
private void pictureBox1_Click(object sender, EventArgs e)
{
    this.Close();
}

private void pictureBox4_Click(object sender, EventArgs e)
{
    this.Close();
}

private void txtTelcharge_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar < 48 || e.KeyChar > 57) //code ascii des numeros
    {
        e.Handled = true;
    }
    if (e.KeyChar == 8)
    {
        e.Handled = false;
    }
}

private void btnAjoutercandidat_Click(object sender, EventArgs e)

```

```

{
    if (testObligatoire() != null)
    {
        MessageBox.Show(testObligatoire(), "Obligatoire", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
    }
    else
    {

        {
            BL.Orientation orientation = new BL.Orientation();
            if (orientation.Ajouter_Orientation(txtNomcharge.Text, txtPrenomcharge.Text, txtOrientation.Text,
            int.Parse(txtTelcharge.Text), DateTime.Parse(dateTimePicker1.Text)) == true)
            {
                MessageBox.Show("Orientation Ajouter avec succès", "Ajouter", MessageBoxButtons.OK,
                MessageBoxIcon.Asterisk);
            }
            else
            {
                MessageBox.Show("Orientation existe deja existe deja", "Echec", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            }
        }
    }
}

private void btActualiserOrientation_Click(object sender, EventArgs e)
{
    txtNomcharge.Text = "Nom"; txtNomcharge.ForeColor = Color.Silver;
    txtPrenomcharge.ForeColor = Color.Silver; txtPrenomcharge.Text = "Prénom";

    txtTelcharge.ForeColor = Color.Silver; txtTelcharge.Text = "Telephone";
    txtMailcharge.Text = ""; txtMailcharge.ForeColor = Color.Silver;
}

private void txtNomcharge_Leave(object sender, EventArgs e)
{
    if (txtNomcharge.Text == "")
    {
        txtNomcharge.Text = "Nom";
        txtNomcharge.ForeColor = Color.Silver;
    }
}

private void txtNomcharge_Enter(object sender, EventArgs e)
{
    if (txtNomcharge.Text == "Nom")
    {
        txtNomcharge.Text = "";
        txtNomcharge.ForeColor = Color.WhiteSmoke;
    }
}

private void txtPrenomcharge_Enter(object sender, EventArgs e)

```

```

{
    if (txtPrenomcharge.Text == "Prénom")
    {
        txtPrenomcharge.Text = "";
        txtPrenomcharge.ForeColor = Color.WhiteSmoke;
    }
}

private void txtPrenomcharge_Leave(object sender, EventArgs e)
{
    if (txtPrenomcharge.Text == "")
    {
        txtPrenomcharge.Text = "Prénom";
        txtPrenomcharge.ForeColor = Color.Silver;
    }
}

private void txtMailcharge_Enter(object sender, EventArgs e)
{
    if (txtMailcharge.Text == "Email")
    {
        txtMailcharge.Text = "";
        txtMailcharge.ForeColor = Color.WhiteSmoke;
    }
}

private void txtMailcharge_Leave(object sender, EventArgs e)
{
    if (txtMailcharge.Text == "")
    {
        txtMailcharge.Text = "Email";
        txtMailcharge.ForeColor = Color.WhiteSmoke;
    }
}

private void txtTelcharge_Leave(object sender, EventArgs e)
{
    if (txtTelcharge.Text == "")
    {
        txtTelcharge.Text = "Telephone";
        txtTelcharge.ForeColor = Color.WhiteSmoke;
    }
}

private void txtTelcharge_Enter(object sender, EventArgs e)
{
    if (txtTelcharge.Text == "Telephone")
    {
        txtTelcharge.Text = "";
        txtTelcharge.ForeColor = Color.WhiteSmoke;
    }
}

private void txtOrientation_Enter(object sender, EventArgs e)
{
    if (txtOrientation.Text == "Nom Orientation")
    {
        txtOrientation.Text = "";
        txtOrientation.ForeColor = Color.WhiteSmoke;
    }
}

```

```

    }

}

private void txtOrientation_Leave(object sender, EventArgs e)
{
    if (txtOrientation.Text == "")
    {
        txtOrientation.Text = "Nom Orientation";
        txtOrientation.ForeColor = Color.WhiteSmoke;
    }
}

private void txtMailcharge_TextChanged(object sender, EventArgs e)
{
}

private void FRM_Ajouter_Modifier_Orientation_Load(object sender, EventArgs e)
{
}

}
}

```

### Conclusion

La phase de réalisation et de mise en œuvre nous a permis de mettre place la solution retenue dont nous avons eu à exposer les quelques interfaces. Cette mise en place a été facilitée par les différents matériels et logiciels utilisés.

**Conclusion générale**

Durant ces deux (2) mois de stage, nous avons eu à travailler sur un projet informatique complet composé des phases d'étude, d'analyse et de développement. Notre stage qui avait pour objectif la mise en place d'un système de gestion automatisée des mobilités nous a permis de mettre en applications nos connaissances acquises durant les deux années passées à l'IAI-TOGO.



## Bibliographie

### ➤ Ouvrage

- UML 2 en action De l'analyse des besoins à la conception 4ème Edition de Pascal Roques et Franck Vallée

### ➤ Notes de cours

- UML, M. KETEGLO (2019-2020)
- Conception de bases de données, M. TCHANTHO (2017-2018)
- C#, M. GBODUI (2019-2020)

### ➤ Sites Web

- [https :// www.developpez.net](https://www.developpez.net)
- [https :// www.pcsoft.fr](https://www.pcsoft.fr)
- [https :// www.youtube.com](https://www.youtube.com)
- [fr.wikipedia.org](https://fr.wikipedia.org) Les documents annexes