



Projet C sous titre

Gestion de formations des formateurs



REALISE PAR

GHANOUCI Issam
ZGUINDOU Abdelghafour

ENCADRE PAR

Mm B.BERRADA



A Nos chers parents

Les mots nous manquent pour exprimer toute la reconnaissance, la fierté et le profond amour que nous vous portons pour les sacrifices que vous avez consentis pour notre réussite.

Que Dieu vous préservent bonne santé et longue vie.

A Nos chers frères

Nous espérons avoir atteint le seuil de vos espérances. Nous vous remercions pour le soutien moral et l'encouragement que vous nous avez accordés. Nous vous souhaitons un brillant avenir.

A Toute personne qui vient de chercher son nom ici.

Un mot final pour vous tous, nous espérons que nous avons été à la hauteur de vos espérances et sachez que jamais nous pourrions oublier ce que vous avez fait pour nous.

Remerciement

Nous remercions toutes les personnes qui ont participées de près ou de loin à la réalisation de ce projet. Nous tenons sincèrement à remercier tous les professeurs de l'ENSIAS et particulièrement Mme. B.BERRADA pour son guide et ainsi que pour son encadrement durant toutes les phases de réalisation de ce projet. Nous espérons que le présent écrit présentera notre travail décemment et que notre programme sera à la hauteur des espérances du corps enseignant à l'ENSIAS qui a contribué, grâce à la formation de qualité qu'il dispense à l'aboutissement de ce travail.

Résumé

Ce projet s'inscrit dans le cadre de la mise en œuvre d'un programme de gestion des formations. Les principaux objectifs de ce projet étaient de mettre en pratique les acquis pratiques qu'on n'a pas pu avoir durant notre formation théorique.

Le programme de la gestion des formations consiste à offrir deux types de formations : accréditation destinée aux formateurs, ou normales destinées aux simples candidats.

Table des matières

Remerciement.....	3
Résumé.....	3
Table des figures	5
INTRODUCTION	6
I. Analyse et Conception	7
1. Analyse	7
Présentation du projet	7
Analyse fonctionnel.....	7
2. Conception :	9
3. Structure de programme.....	10
II. Réalisation	18
Conclusion	31

Table des figures

<i>Figure 1 : Arbre des formateurs.....</i>	<i>9</i>
<i>Figure 2 : Arbre des candidats.....</i>	<i>9</i>
<i>Figure 3 : Arbre des enseignants.....</i>	<i>10</i>
<i>Figure 3.1 : Menu principale de l'application.....</i>	<i>16</i>
<i>Figure 3.2 : Authentification de l'administrateur.....</i>	<i>17</i>
<i>Figure 3.3 : Menu principale de l'administrateur.....</i>	<i>17</i>
<i>Figure 3.4 : Menu principale de gestion des classes.....</i>	<i>18</i>
<i>Figure 3.5 : Affichage des classes avec le nombre d'effectif.....</i>	<i>18</i>
<i>Figure 3.6 : Menu principale de gestion des candidats</i>	<i>19</i>
<i>Figure 3.7 : Menu de modification de profil d'un candidat.....</i>	<i>19</i>
<i>Figure 3.8 : Menu d'attribution d'un candidat à une classe.</i>	<i>20</i>
<i>Figure 3.9 : Classe avec son nombre d'effectif maximum</i>	<i>21</i>
<i>Figure 3.10 : dernière candidat chez un enseignant qui devient Formateur</i>	<i>22</i>
<i>Figure 3.11: Menu principale de gestion des formateurs</i>	<i>22</i>
<i>Figure 3.12 : Affichage des formateurs</i>	<i>23</i>
<i>Figure 3.13 : Menu de Mettre à jour un formateur</i>	<i>23</i>
<i>Figure 3.14 : Ajouter une activée (Niveau Régional).....</i>	<i>24</i>
<i>Figure 3.15 Menu offrant quelques statistiques.....</i>	<i>25</i>
<i>Figure 3.16 Ajouter un nouvel enseignant</i>	<i>25</i>
<i>Figure 3.17 Modifier le profil d'un enseignant.....</i>	<i>26</i>
<i>Figure 3.18 : Enseignant complète les 3 parties de la formation suivie.....</i>	<i>26</i>
<i>Figure 3.19 : Affichage des formations.....</i>	<i>27</i>
<i>Figure 3.20 : Affichage d'en enseignant.....</i>	<i>27</i>
<i>Figure 3.21 : S'authentifier en tant qu'un candidat.....</i>	<i>28</i>
<i>Figure 3.22 : S'authentifier en tant qu'un formateur.....</i>	<i>28</i>

INTRODUCTION

Dans le cadre de notre première année d'études, à l'Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS), nous avons l'occasion de réaliser un projet C pendant la deuxième période du premier semestre.

Ce projet consiste à la création d'un programme écrit en langage C qui répond à un besoin tiré de la réalité. Il a été l'occasion idéale pour mettre en pratique les différentes notions acquises durant les nombreux cours : d'« Algorithmique », de « Technique de programmation » et de « Structures de données ». et ainsi développer nos connaissances en langage C concrètement, loin du cadre théorique du cours.

C'est également une bonne opportunité afin de découvrir les procédés en termes d'organisation, de conception et de réalisation d'un projet.

Actuellement, la gestion des formations doit être plus fiable repose sur une automatisation globale. Ainsi la création d'un outil adéquat était donc nécessaire pour faciliter les tâches d'enregistrements. Pour ce faire il nous a été confié la mission d'examiner et de mettre en place un programme d'automatisation des formations de formateurs par M BERRADA.

Dans le premier chapitre, une présentation générale, s'ensuivra la modélisation de l'application puis nous choisirons l'architecture à utiliser. En un second temps, nous verrons comment a été mise en place cette mini-application en parcourant les choix technologiques et les subtilités techniques de sa mise en œuvre.

Ce document établit donc une description détaillée des déroulements des travaux réalisés à savoir :

- Etude détaillée du projet.
- La détermination des différents éléments à analyser et leurs axes.
- La réalisation de programme.

I. Analyse et Conception

1. Analyse

Présentation du projet

Notre programme permettra aux acteurs d'automatiser la gestion des formations de formateurs à l'échelle locale, nationale et régionale. Permettant de plus aux administrateurs de contrôler efficacement les formations et d'être en mesure de répondre aux besoins de candidature, Cependant il nous donne aussi la possibilité de consulter un ensemble de statistique momentanée suivant plusieurs critères.

Analyse fonctionnel

Contraintes des enseignants :

- Un enseignant pour qu'il puisse devenir facilitateur tuteur, doit obligatoirement :
 - 1- Recevoir une formation sous 3 parties.
 - 2- Enseigne 3 classes minimum de 30 élèves par chaque classe.
 - 3- Suivre une formation d'accréditation chez un facilitateur tuteur.

Contraintes Des Formateurs :

- Après avoir vérifié que l'enseignant a pu d'accomplir les conditions. Il sera considéré comme formateur facilitateur tuteur, tout en mémorisant son historique.
- l'administrateur doit obligatoirement indiquer la formation que le formateur facilitateur tuteur souhaite former, qui doit être de type Normal (Destiné aux candidats) ou de type accréditation destinée aux enseignants; ainsi si la formation existe, une nouvelle classe sera créer Pour l'assurer, sinon un formulaire devra être remplie par les informations nécessaires de la formation souhaitée.
- Le formateur Sera toujours capable de modifier son profil en respectant un ensemble de contraintes à savoir :

- 1-- Modifier son nom
- 2-- Ouvrir une nouvelle classe soit de type Normal ou de type accréditation.
- 3-- Inscrire dans une formation d'accréditation assurée par un formateur de niveau supérieur.
- 4-- A condition qu'il soit du niveau régional, le formateur pourra ajouter des nouvelles activités.
- 5-- Après avoir ouvrir 3 classes de type normal, chacune contient 30 élèves, 3 classes de type accréditations, chacune contient 20 formateurs et suivre une formation d'accréditation chez un formateur national .le facilitateur tuteur sera informé automatiquement qu'il a réussi de passer au niveau suivant (**Formateur National**).
- 6-- Après avoir ouvrir plus de 3 classes normal, 3 classes d'accréditations et avoir une accréditation différente à celle antérieure, à travers suivre une formation d'accréditation chez un formateur régional, Le formateur National passe aussi au niveau Régional.

Contraintes des Candidats :

- Le candidat peut suivre sa formation chez n'importe quel formateur ou enseignant.
- En recueillant la première demande d'inscription d'un Candidat, et après avoir remplir quelque informations personnel, Le programme le permettra de choisir la formation souhaitée à suivre, pour que l'opération d'inscription passera en succès ; la formation doit exister dans le système, sinon la demande sera rejetée, normal puisque 'il n y a pas des formateurs de cette formation. Supposant que la formation existe, alors le programme affichera une liste des classes qui s'occupent de la formation, et il ne reste qu'une confirmation avec l'identifiant de la classe souhaitée ; ainsi il y'a 3 possibilités :
 - 1 -- Si le nombre effectif de la classe choisis a atteint son maximum (30 élèves) la demande ne sera pas prise en considération ; le candidat devra choisir une autre classe.
 - 2 -- Si le candidat est l'élève 30 d'une classe d'un enseignant (respectivement formateur), et supposant qu'il a toute autre condition ;

l'enseignant concerné sera se considérer comme un formateur facilitateur tuteur et un message sera affiché automatiquement en pleine écran facilitant l'enseignant (resp. formateur).

3 -- le nombre d'effectif de la classe s'incrémente normalement si l'effectif de la classe ne dépasse pas 30.

- Le candidat peut manipuler son profil, ainsi plusieurs modifications sont présentes. Le nom complet, le code et la date d'inscription.

Contraintes des Formations et des Classes :

- Une formation ne peut pas être créée sans formateur ; pour s'assurer que lors du choix de la formation par le candidat, il aura obligatoirement un formateur ou plusieurs qui assure cette dernière.
- Une formation peut être assurée par un ou plusieurs formateurs, mais nécessairement chaque classe est ouverte par un seul formateur ; de plus il y a deux types de formations : normal ou d'accréditation.

Contraintes des statistiques :

- Une statistique globale momentanée sur le nombre d'effectif de chaque entité, à savoir : candidats, enseignants, formateurs, formation, classes. Ainsi que le nombre des formateurs suivant leurs niveaux.

2. Conception :

A la base des algorithmes classiques des différents méthodes de structure des données, on a essayé de réaliser notre application à l'aide d'une plateforme structuré basé sur le langage de programmation c.

3. Structure de programme

L'application est basée sur les notions de programmation modulaires, cette technique nous a permis de structurer le code de programme sous forme de fichiers header (.h) et des fichiers sources (.c), qui permet de définir les prototypes des fonctions qu'on a déclaré dans les headers. Ainsi on a utilisé l'avantage de programmation C pour gérer dynamiquement la mémoire central et optimiser son utilisation.

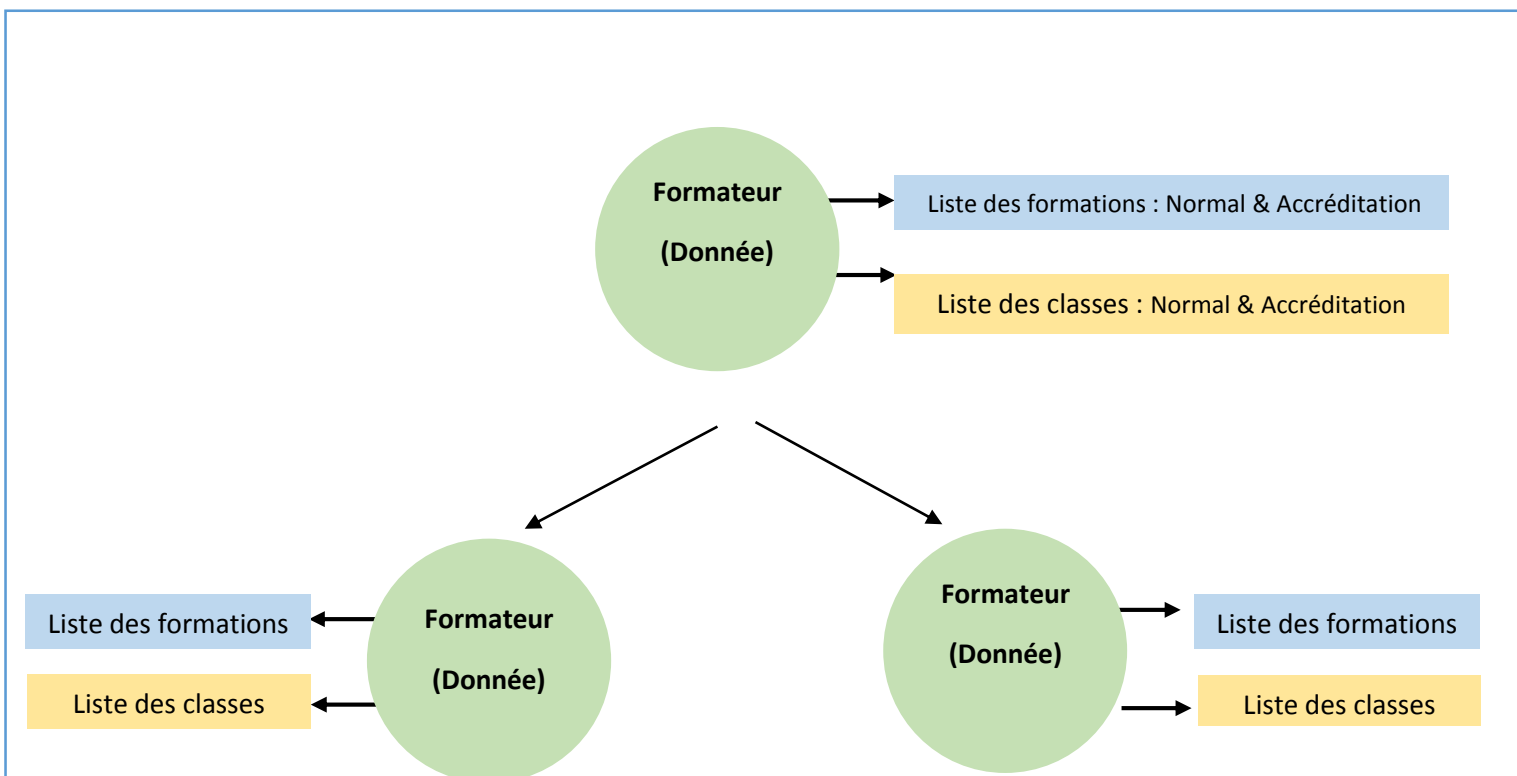
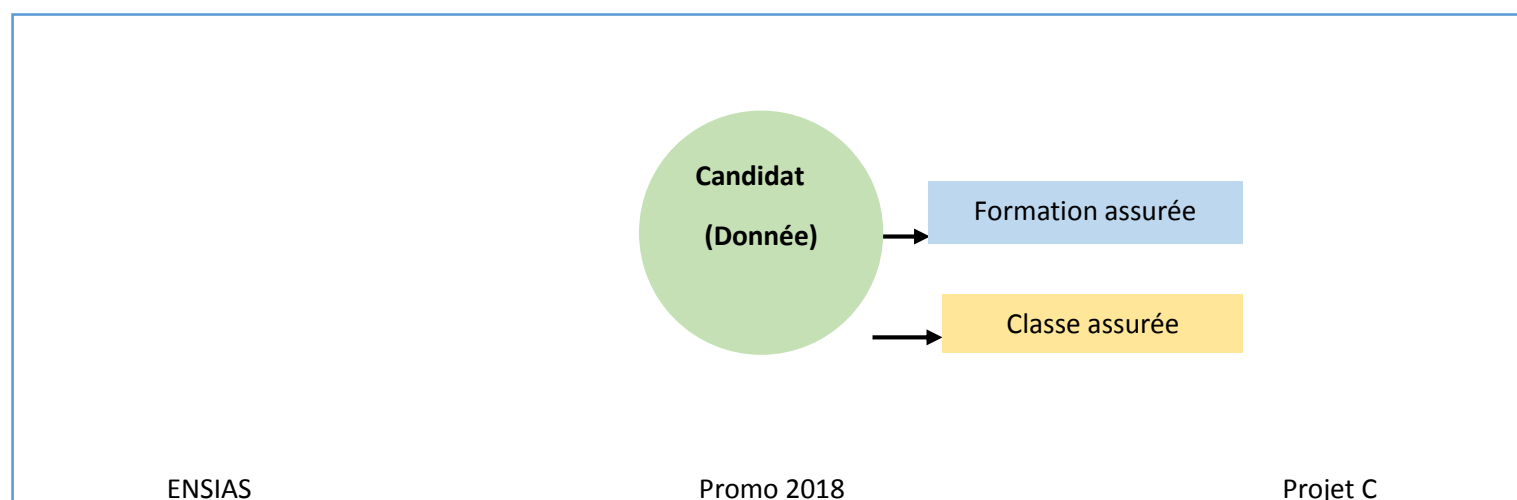


Figure 1 : Arbre des formateurs



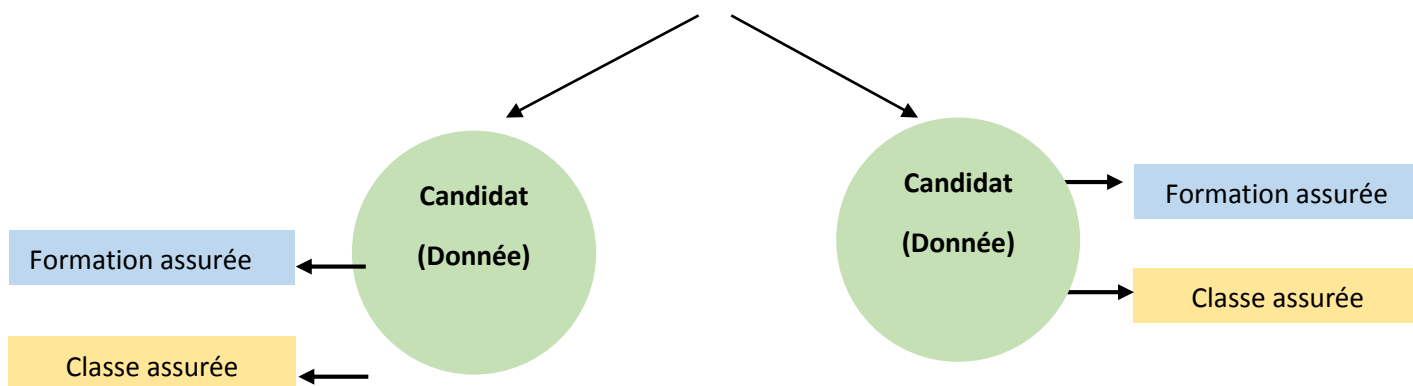


Figure 2 : Arbre des candidats

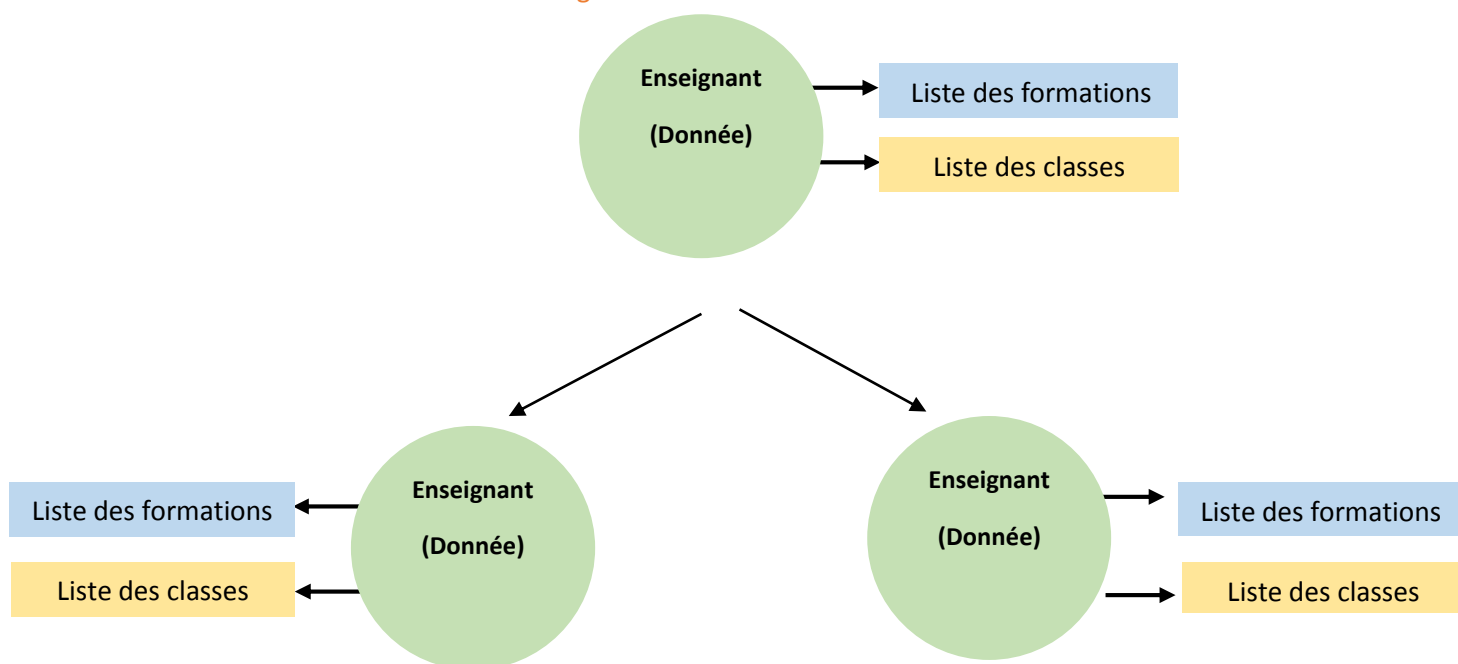


Figure 3 : Arbre des enseignants

Le code de programme est reparti sur six modules : Formation.h, Class.h, Former.h, Candidate.h Enseignant.h, Menu.h et le fichier principale de l'application. Chaque module comporte un ensemble des structures et des fonctions.

Module 1 : Menu.h

```
//Menu principale de l'administrateur
void Menu_Principal(int *, const int, const int, const int, const int);
//Menu de gestion des classes
void Menu_Cl(int *);
//Menu de gestion des formateurs
void Menu_Fo(int *);
//Menu de gestion des formations
void Menu_Ft(int *);
//Menu de gestion des candidats
void Menu_Cd(int *);
//Menu pour mettre a jour un formateur
void Menu_Upadte_Fo(int *);
//Menu pour mettre a jour un candidat
void Menu_Upadte_Cd(int *);
//Menu du profil d'un candidat
void Menu_Cd_Profile(int *, const char *);
//Menu du profil d'un formateur
void Menu_Fo_Profile(int *, const char *);
//Menu principale de l'application
void _MENU_(int *);
//Menu des statistiques
void Menu_STATISTICS(int *,int ,int ,int );
//Fonction permet l'affichage d'une remarque
void REMARQUE(char*);
//Fonction permet de récupérer les choix d'un menu
int getChoice(const int, const int);
//Fonction permet de retourner un mot de passe
char *newPassowrd();
```

Module 2 : Formation.h

```
//Structure de location de formation
typedef struct _Local
{
    char Area[20];
    char City[20];
    char University[30];
    char Establishment[30];
} Local_t;
//Type de formation
typedef enum Ft_Type {Normal, Accreditation} Ft_Type_t;

//Structure de données des formations sous forme d'une liste chaînée
typedef struct _Formation
{
    int Ft_Id;
    int Ft_Duration;
    char Ft_Name[15];
    //Location de formation
    Local_t Ft_Location;
    //Type de formation
    Ft_Type_t Ft_TYPE;
    struct _Formation *Next;
} Formation_t;

//Fonction permet de sauvegarder la liste des formations
void Ft_Save(Formation_t *);
//Fonction permet d'afficher la liste des formations
void Ft_Print(Formation_t *);
//Fonction permet d'afficher les informations sur une formation
void Ft_ProfilPrint(Formation_t *);
//Fonction permet de charger la liste des formations à partir d'un fichier binaire
Formation_t *Ft_load();
//Fonction permet de chercher une formation par son id
Formation_t *Ft_SearchById(Formation_t *, const int);
//Fonction permet de copier les informations d'une formation
Formation_t *Ft_CopyData(Formation_t *);
//Fonction permet de récupérer les informations d'une formation
Formation_t *Ft_GetData(char c);
//Fonction permet d'ajouter une formation dans la liste des formations
Formation_t *Ft_Add(Formation_t *, Formation_t *);
//Fonction permet de supprimer une formation
Formation_t *Ft_Delete(Formation_t *, const int);
//Fonction permet de mettre à jour les informations d'une formation
Formation_t *Ft_Update(Formation_t *, const int);
//Fonction permet de chercher une formation
Formation_t *Ft_Search(Formation_t *, const int);
//Fonction permet de vider la liste des formations
Formation_t *Ft_Destroy(Formation_t *);
```

Module 3 : Class.h

```
//Type de classe soit normale ou accreditation
typedef enum _Cl_Type {Normale, Accreditation} Cl_Type_t;
//Structure de données des classes sous forme d'une liste chaînée
typedef struct _Class
{
    int Cl_Id;
    int Ft_key;
    char Cl_Name[10];

    Cl_Type_t Cl_TYPE; //Type de classe

    struct _Class *Cl_Next;
} Class_t;
//Fonction permet d'afficher la liste des classes
void Cl_Print(Class_t *);
//Fonction permet l'affichage des informations d'une classe
void Cl_ProfilPrint(Class_t *);
//Fonction permet de sauvegarder la liste des classes dans un fichier binaire
void Cl_Save(Class_t *);

//Fonction permet de chercher une classe par son id
Class_t *Cl_SearchById(Class_t *,const int);
//Fonction permet de lire la liste des classes a partir d'un fichier binaire
Class_t *Cl_load();
//Fonction permet de copier les informations d'un classe
Class_t *Cl_CopyData(Class_t *);
//Fonction permet de lire les données d'un classe a partir du clavier
Class_t *Cl_GetData(char c);
//Fonction permet de stocker une classe dans la liste des classes
Class_t *Cl_Add(Class_t *,Class_t *);
//Fonction permet de supprimer une classe

//Fonction permet de modifier les informations d'une classe
Class_t *Cl_Update(Class_t *, const int);
//Fonction permet de chercher une classe par son id
Class_t *Cl_Search(Class_t *, const int);
//Fonction permet de chercher la classe dont une formation est assurée
Class_t *Cl_Search_ByFtID(Class_t *, const int);
//Fonction permet de vider la liste des classes
Class_t *Cl_Destroy(Class_t *);
```

Module 4 : Former.h

```
//Niveau de formateur
typedef enum _level {Tutu, Nation, Region} Level_t;
//Structure de données des formateurs sous forme d'un arbre binaire de recherche
typedef struct _Former
{
    int Fo_Act_Nbr; //Nombre des activites
    int Fo_FtN_Nbr; //Nombre des formations normale
    int Fo_FtA_Nbr; //Nombre des formations d'Accreditation
    int Fo_CLN_Nbr; //Nombre des classes normale
    int Fo_CLA_Nbr; //Nombre des classes d'accréditation

    char Fo_Code[8];
    char Fo_Name[20];
    char *Fo_Activity[15]; //Tableau des activités

    Level_t Fo_Level; //Niveau de formateur

    Formation_t *Fo_Ft_liste; //Liste des formations
    Class_t *Fo_Cl_liste; //Liste des classes

    Formation_t *Ft_Have_Accr; //Formation d'accréditation suivie
    Class_t *Cl_Have_Accr; //Classe d'accréditation suivie

    struct _Former *Fo_left;
    struct _Former *Fo_right;
} Former_t;

//Fonction permet d'afficher l'arbre des formateurs
void Fo_Print(Former_t *);
//Fonction permet d'afficher le profil d'un formateur
void Fo_prifilPrint(Former_t *);
//Fonction permet de sauvegarder l'arbre des formateurs dans un fichier binaire
void Fo_Save(Former_t *);
//Fonction permet de filtrer les formations normal et d'accréditation d'un formateur
//a partir de sa liste des formations
void Ft_Filtrin(Formation_t *, Formation_t **, Formation_t **);
//Fonction permet de filtrer les classes normal et d'accréditation d'un formateur
//a partir de sa liste des classes
void Cl_Filtrin(Class_t *, Class_t **, Class_t **);
//Fonction permet de copier les informations d'un formateur
void Fo_CopyData(Former_t **, Former_t *);
//Fonction permet d'afficher tous les formateurs suivant un niveau donné
void Fo_Print_Niveau(Former_t *Fo_root, Level_t niveau);
//Fonction permet de sauvegarder les données des formateurs dans un fichier binaire
Former_t *Fo_Save_1(Former_t *, FILE *);
//Fonction permet de récupérer les informations d'un formateur
Former_t *Fo_getData(Formation_t **, Class_t **);
//Fonction permet d'ajouter un formateur dans l'arbre des formateurs
Former_t *Fo_Add(Former_t *, Former_t *);
//Fonction permet de supprimer un formateur
Former_t *Fo_Delete(Former_t *, Class_t **, const char *);
//Fonction permet de mettre a jour les informations d'un formateur
Former_t *Fo_Update(Former_t *, const char *, Formation_t **, Class_t **);
//Fonction permet de chercher un formateur
Former_t *Fo_Search(Former_t *, const char *);
//Fonction permet de charger l'arbre des formateurs à partir d'un fichier binaire
Former_t *Fo_load(Formation_t *, Class_t *);
//Fonction pour la gestion d'arbre des formateurs
Former_t *Fo_FindMin(Former_t *);
//Fonction permet de faire une statistique sur le nombre effectif de chaque niveau de formateur
Former_t *Fo_Niveau(Former_t *, int*);
```


Module 5 : Condidate.h

```
//structure date
typedef struct _Date
{
    int DD;
    int MM;
    int YY;
} Date_t;
//Structure des données des candidats sous forme d'un arbre binaire de recherche
typedef struct _Condidat
{
    //Les informations d'un candidat
    char Cd_Code[8];
    char Cd_FirstName[20];
    char Cd_LastName[20];
    int Cd_Id;
    //Date d'inscription
    Date_t Cd_RegDate;
    //Formation suivie
    Formation_t *Cd_Formation;
    //Classe
    Class_t *Cd_Class;
    struct _Condidat *Cd_left;
    struct _Condidat *Cd_right;
} Condidat_t;

//Fonction permet d'afficher l'arbre des candidats
void Cd_Print(Condidat_t *);
//Fonction permet de sauvegarder l'arbre des candidats dans un fichier binaire
void Cd_Save(Condidat_t *);
//Fonction permet d'afficher le profil d'un candidat
void Cd_profilPrint(Condidat_t *);

//Fonction permet de sauvegarder les données des candidats dans fichier binaire
Condidat_t *Cd_Save_1(Condidat_t *, FILE *);
//Fonction permet de charger l'arbre des candidats a partir d'un fichier binaire
Condidat_t *Cd_load(Formation_t *, Class_t *);
//Fonction permet de copier les données d'un candidat
Condidat_t *Cd_CopyData(Condidat_t *);
//Fonction pour traiter l'arbre des candidats
Condidat_t *Cd_FindMin(Condidat_t *);
//Fonction permet de récupérer les informations d'un candidat
Condidat_t *Cd_getData(Formation_t *, Class_t *);
//Fonction permet d'ajouter un candidat dans l'arbre des candidats
Condidat_t *Cd_Add(Condidat_t *, Condidat_t *);
//Fonction permet de supprimer un candidat
Condidat_t *Cd_Delete(Condidat_t *, const char *);
//Fonction permet de modifier les informations d'un candidat
Condidat_t *Cd_Update(Condidat_t *, const char *);
//Fonction permet de chercher un candidat par son code
Condidat_t *Cd_Search(Condidat_t *, const char *);
```


Module 5 : Enseignant.h

```
typedef struct _Enseignant
{
    int Ens_Ft_Nbr;
    int Ens_Cl_Nbr;

    char Ens_Code[8];
    char Ens_Name[20];

    Formation_t *Ens_Ft_liste; //Liste des formations
    Class_t *Ens_Cl_liste; //Liste des classes

    Formation_t *Ft_Have_Accr; //Formation d'accréditation suivie
    Class_t *Cl_Have_Accr; //Classe d'accréditation suivie

    int Part_Of_Formation[3]; // Les parties qu'il suivie durant la formation

    struct _Enseignant *left;
    struct _Enseignant *right;
} Enseignant_t;

//Return VRAI si l'enseignant a réalisé toute les conditions pour être Facilitateur
BOOLEAN Ens_Graduate(Enseignant_t *, Class_t *);
// fonction cherche l'enseignant qui s'occupe d'une classe , verifie et fait appel à TRANSFERT
void Ens_Gradute_My_Teachers(Former_t **, Enseignant_t *, Class_t *, int);
// fonction ajoute un enseignant avec les formateurs et reinitialise les valeurs
void Transfert(Former_t **, Enseignant_t *);
// Passage par adresse pour retourner nombre d'enseignants
void Return_Ens(int *);
// afficher le profil d'un enseignant
void Ens_Print_Profil(Enseignant_t *, Class_t *);
// afficher tous les enseignants
void Ens_Print(Enseignant_t *, Class_t *);
// Enregistrer les enseignants dans un fichier binaire
void Ens_Save(Enseignant_t *);
// return 1 si l'enseignant a pu de suivre les 3 parties de sa formation
int Part_Fini(Enseignant_t *);
// recuperer les donnees d'un enseignant , en ajoutant la formation et la classe aux principaux
Enseignant_t *Ens_getData(Formation_t **, Class_t **);
// Ajouter un enseignant dans l'arbre des enseignant .
Enseignant_t *Ens_add(Enseignant_t *, Enseignant_t *);
// Rechercher un enseignant avec son code d'enregistrement .
Enseignant_t *Ens_Search(Enseignant_t *, char *);
// Modifier le profil d'un enseignant ( Ajouter Formation , Compléter une partie .... )
void Ens_Update(Enseignant_t **, Former_t **, const char *, Formation_t **, Class_t ** );
// Enregistrer la liste des enseignants
Enseignant_t * Ens_Save_l(Enseignant_t *, FILE *);
// charger la liste des enseignants à partir d'un fichier binaire
Enseignant_t *Ens_load(Formation_t *, Class_t *);

#endif // ENSEIGNANT_H_INCLUDED
```

II. Réalisation

L'application réalisée peut être exécutée dans un poste de travail équipé par un système d'exploitation Windows ou Linux, un processeur Pentium ou plus, mémoire centrale de 1 Go au moins et un disque dur de 100 Go. L'application utilise l'interface homme-machine non graphique (CMD), après l'exécution de l'application le menu principal apparaît sur l'écran, dans ce cas l'utilisateur choisit l'un des numéros pour exécuter la fonction souhaitée parmi ceux qui figurent sur ce menu (figure 3.1). Les différentes fonctions permettent de récupérer le profil de l'administrateur ou d'un formateur ou d'un candidat et ainsi l'utilisateur peut quitter l'application par le dernier choix.

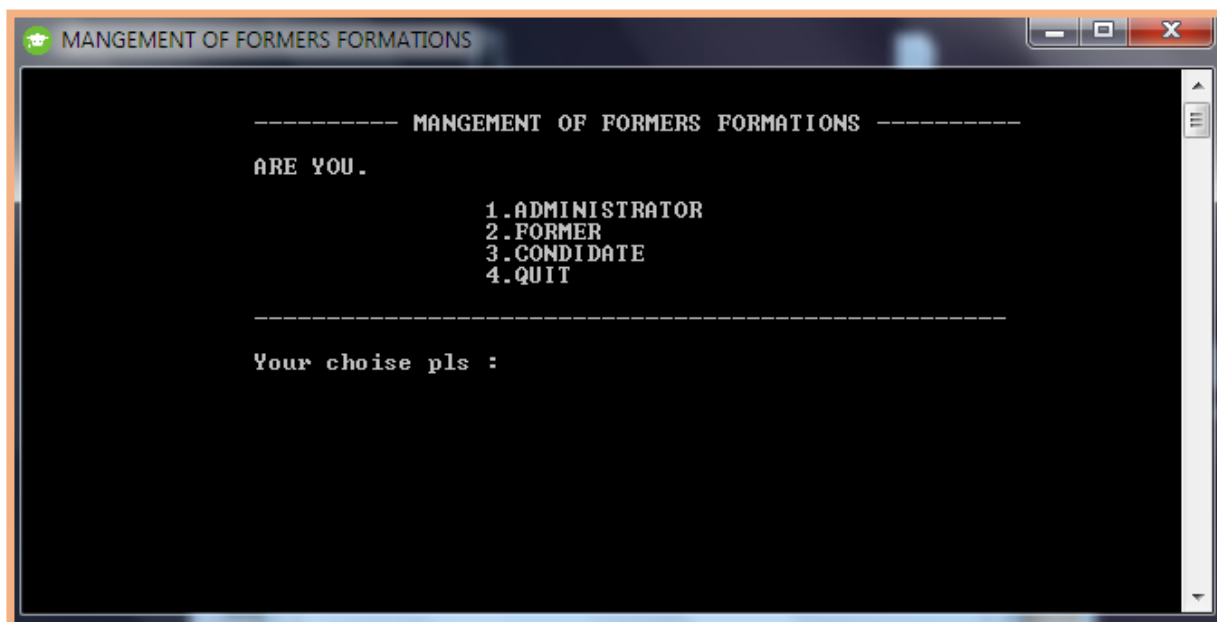


Figure 3.1 : Menu principale de l'application

→ Une authentification est obligée pour chaque session.

3 – 1. Session administrateur :



Figure 3.2 : Authentification de l'administrateur

→ L'administrateur doit saisir son login et son mot de passe afin d'ouvrir sa propre session.

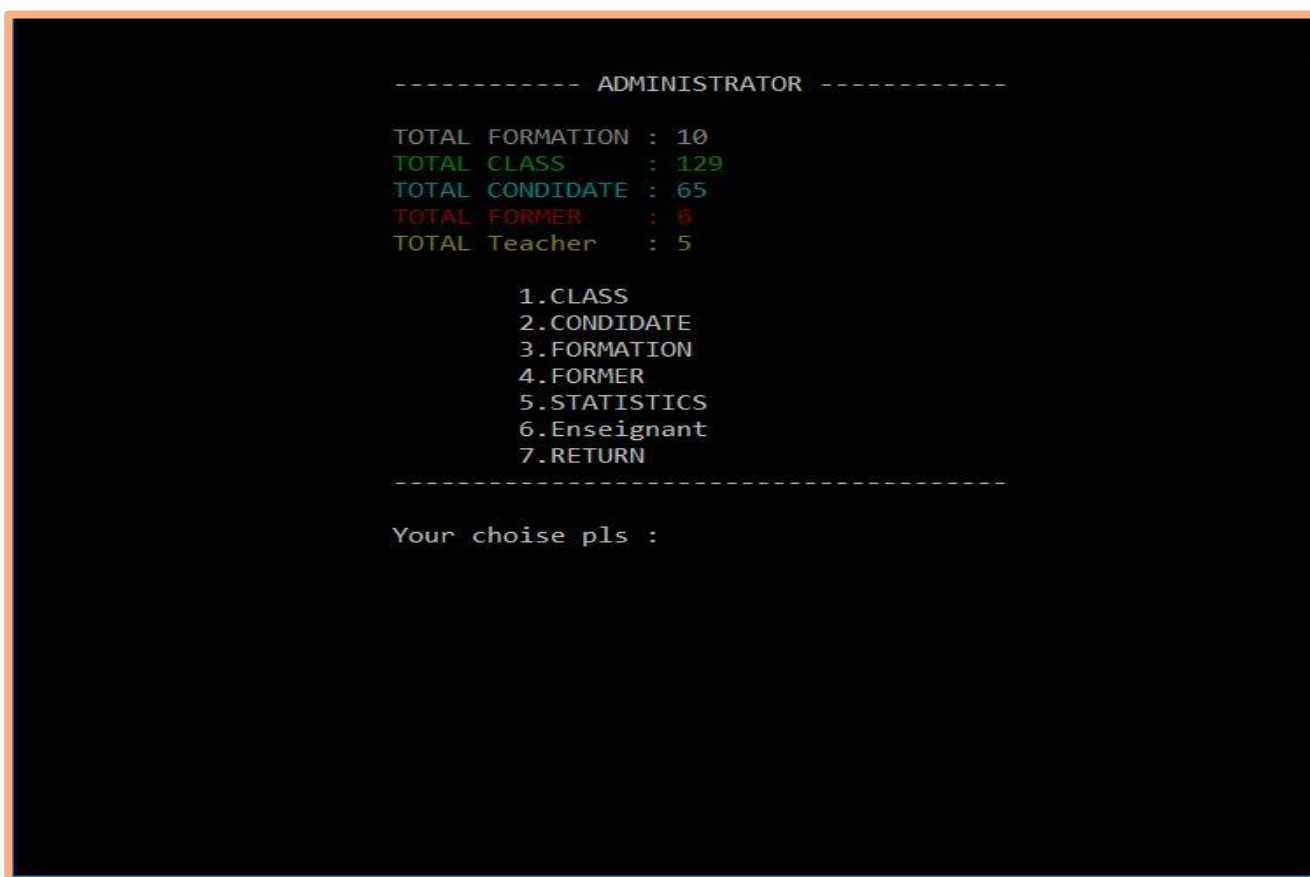


Figure 3.3 : Menu principale de l'administrateur

→ A travers cette interface, L'administrateur a le droit de gérer un ensemble d'entité, de même, Il a le nombre d'effectif de chacune.

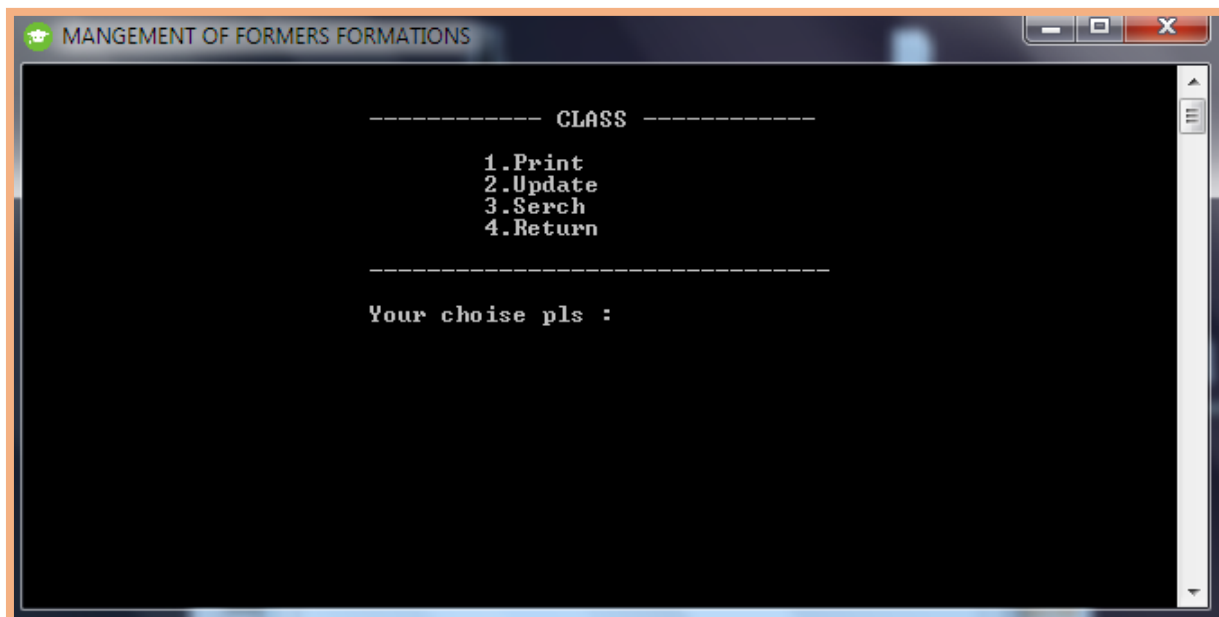


Figure 3.4 : Menu principale de gestion des classes

➔ A travers cette interface, l'admirateur peut faire une gestion globale sur les classes.

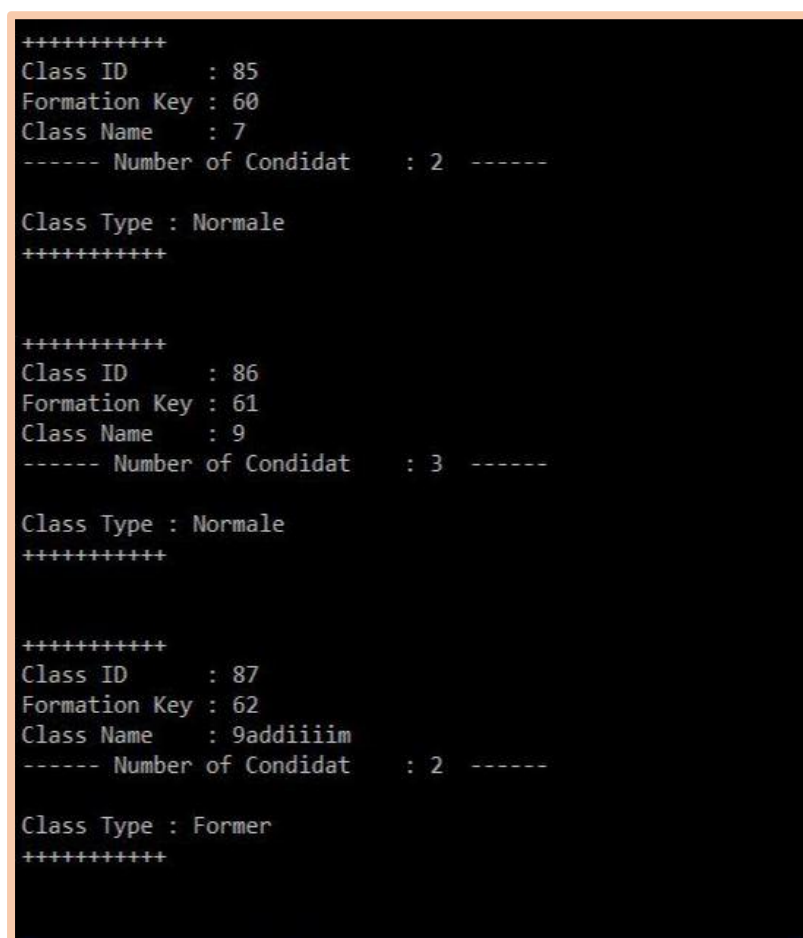


Figure 3.5 : Affichage des classes avec le nombre d'effectif.

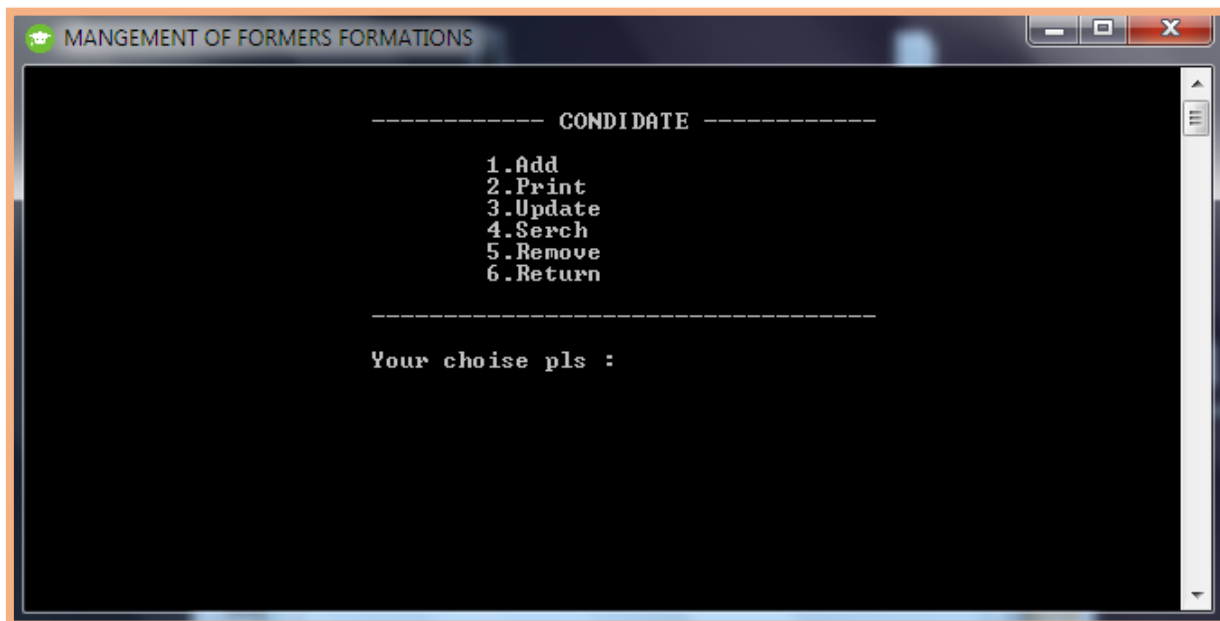


Figure 3.6 : Menu principale de gestion des candidats

➔ A travers cette interface, l'admirateur peut faire une gestion globale sur le candidat.

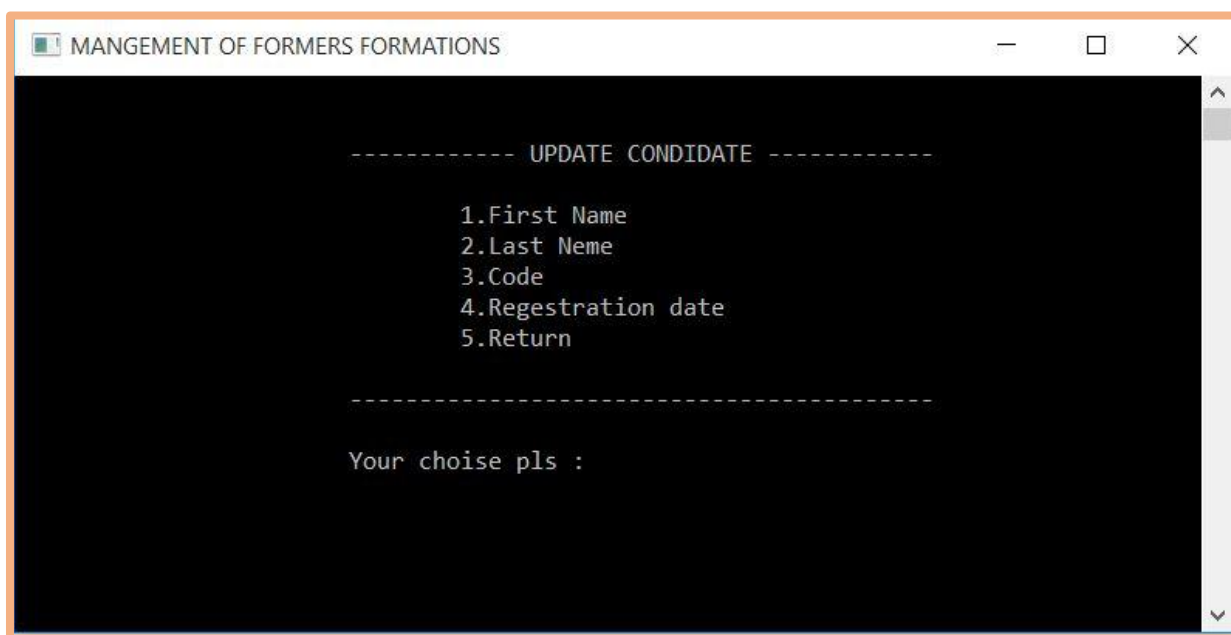


Figure 3.7 : Menu de modification de profil d'un candidat.

➔ A travers cette interface, l'admirateur peut faire une modification sur le profil d'un candidat ; ainsi il peut corriger sa date de registration, son nom ou son prénom.

```

Canditate First Name : ISSAM
Canditate Last Name : GHANOUGH
Canditate Code : QA156204

Resestration Data
-----
Day   : 31
Month : 01
Year  : 2016

Choice your formation & your Class

Formation ID : (0 Quit) 5

-----
Id of Formation : 5
The Name of Formation : Oracle
Duration : 50 jours
Type Is : Accreditation
        Area           : Casa
        City           : Casa
        University     : Hasan1
        Establishment  : FS

0. Confirm & 1. No : 0

Secces Operation with Cdrmatation

---- This is the list of all classes that has Oracle like Formation

-----
Class ID      : 6
Formation Key : 5
Class Name    : 500
----- Number of Canditat : 1 -----

Class Type : Normale
+++++++

+++++++
Class ID      : 102
Formation Key : 5
Class Name    : 1
----- Number of Canditat : 3 -----

YOUR CHOICE PLS (ID OF CLASS) (-1 : EXIST ) : 102
This is a not a normal class , Sorry !
YOUR CHOICE PLS (ID OF CLASS) (-1 : EXIST ) : 6

Secces Operation with class

```

Figure 3.8 : Menu d'attribution d'un candidat à une classe.

➔ A travers cette interface, quand le candidat souhaite de s'inscrire, il doit saisir l'identifiant de la formation souhaitée ; ensuite le programme affiche les descriptions de la formation souhaitée, après une confirmation de la formation ; une liste des classes qui s'occupent de former cette formation seront affichées, ensuite le candidat choisit l'identifiant de la classe : ainsi il se peut d'avoir 3 états :

- 1 - Le candidat s'inscrit dans une classe de type normal et non une classe d'accréditation : le nombre d'effectif de la classe s'incrémente normalement.
- 2 - Si le nombre effectif de la classe choisie a atteint son maximum (30 élèves) la demande ne sera pas prise en considération ; le candidat devra choisir une autre classe (Figure 3.9).
- 3- Si le candidat est l'élève 30 d'une classe d'un enseignant (respectivement Formateur), et supposant qu'il a toute autre condition ; l'enseignant concerné sera se considérer comme un formateur facilitateur tuteur et un message sera affiché automatiquement en pleine écran facilitant l'enseignant (resp. formateur). (Figure 3.10).

```

+++++++
Class ID      : 105
Formation Key : 1
Class Name    : 1
----- Number of Candidat      : 1 -----

Class Type : Normale
+++++++

+++++++
Class ID      : 117
Formation Key : 1
Class Name    : 1
----- Number of Candidat      : 1 -----

Class Type : Normale
+++++++

YOUR CHOICE PLS (ID OF CLASS) (-1 : EXIST ): 77
Sorry ! This Class Have Already 30 candidat => You must choice an other Class
:
YOUR CHOICE PLS (ID OF CLASS) (-1 : EXIST ): 104
Secces Operation with class

```

Figure 3.9 : Classe avec son nombre d'effectif maximum


```

+++++++
Class ID      : 110
Formation Key : 11
Class Name    : 1
----- Number of Candidat : 1 -----

Class Type : Normale
+++++++

+++++++
Class ID      : 126
Formation Key : 11
Class Name    : 1
----- Number of Candidat : 0 -----

Class Type : Normale
+++++++

YOUR CHOICE PLS (ID OF CLASS) (-1 : EXIST ) : 126

```

1 : Graduate as Facilitateur TuTeur In our System

PRESS ANY KEY FOR CONTINUE

Figure 3.10 : dernière candidat chez un enseignant qui devient Formateur

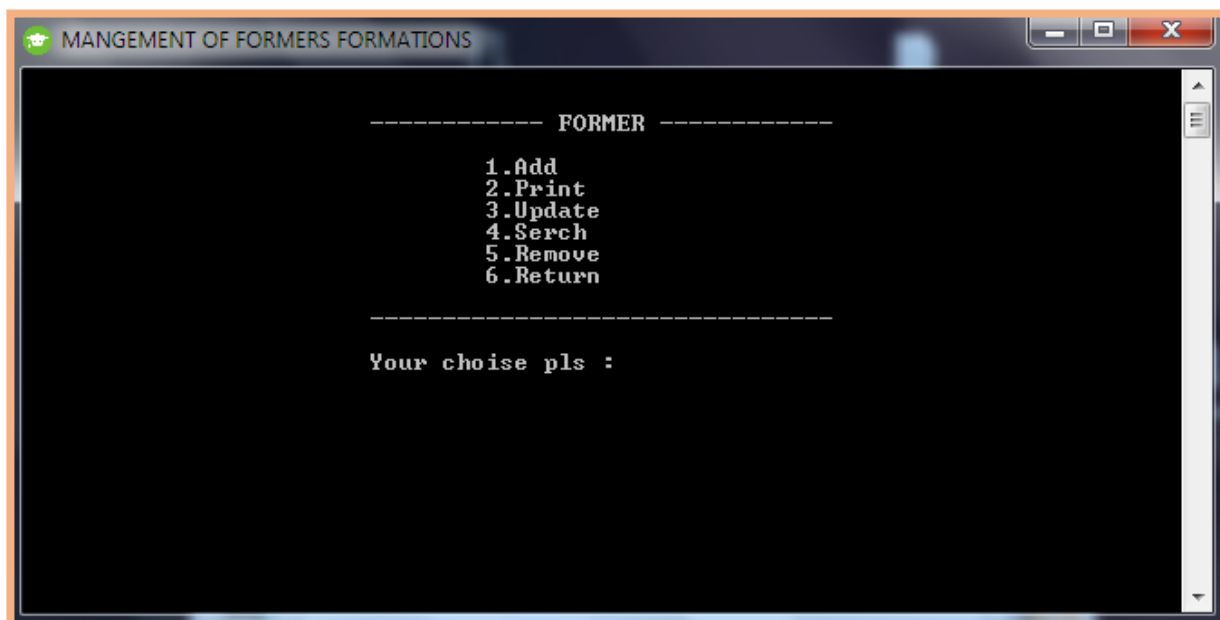
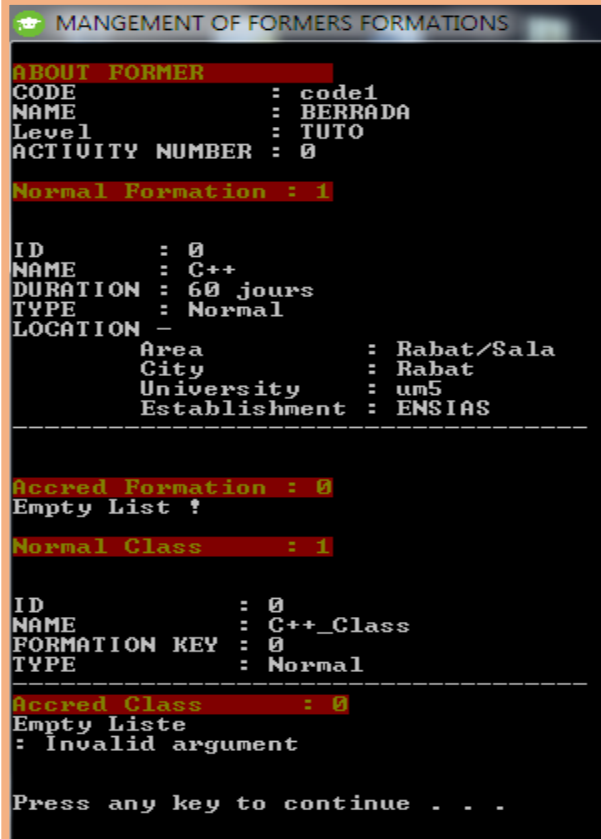


Figure 3.11 : Menu principale de gestion des formateurs

➔ A travers cette interface, l'admirateur peut faire une gestion globale sur les formateurs.



```

MANGEMENT OF FORMERS FORMATIONS
ABOUT FORMER
CODE          : code1
NAME          : BERRADA
Level         : TUTO
ACTIVITY NUMBER : 0

Normal Formation : 1

ID           : 0
NAME         : C++
DURATION     : 60 jours
TYPE        : Normal
LOCATION      :
Area         : Rabat/Sala
City         : Rabat
University   : um5
Establishment : ENSIAS

-----
Accred Formation : 0
Empty List ?

Normal Class   : 1

ID           : 0
NAME         : C++_Class
FORMATION KEY : 0
TYPE        : Normal

-----
Accred Class   : 0
Empty Liste
: Invalid argument

Press any key to continue . . .

```

Figure 3.12 : Affichage des formateurs

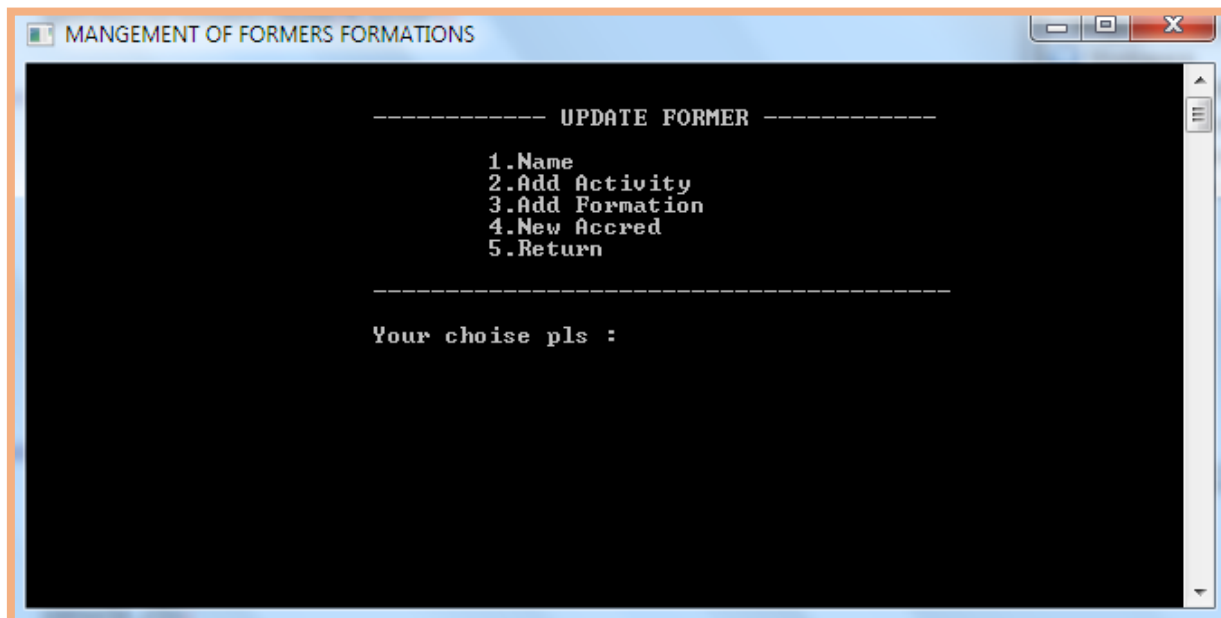


Figure 3.13 : Menu de Mettre à jour un formateur

- Le formateur Sera toujours capable de modifier son profil en respectant un ensemble de contraintes à savoir :

- 1-- Modifier son nom
- 2-- Ouvrir une nouvelle classe soit de type Normal ou de type accréditation.
- 3-- Inscrire dans une formation d'accréditation assurée par un formateur de niveau supérieur.
- 4-- A condition qu'il soit du niveau régional, le formateur pourra ajouter des nouvelles activités.
- 5-- Après avoir ouvrir 3 classes de type normal, chacune contient 30 élèves, 3 classes de type accréditations, chacune contient 20 formateurs et suivre une formation d'accréditation chez un formateur national .le facilitateur tuteur sera informé automatiquement qu'il a réussi de passer au niveau suivant (**Formateur National**).
- 6-- Après avoir ouvrir plus de 3 classes normal, 3 classes d'accréditations et avoir une accréditation différente à celle antérieure, à travers suivre une formation d'accréditation chez un formateur régional, Le formateur National passe aussi au niveau Régional.

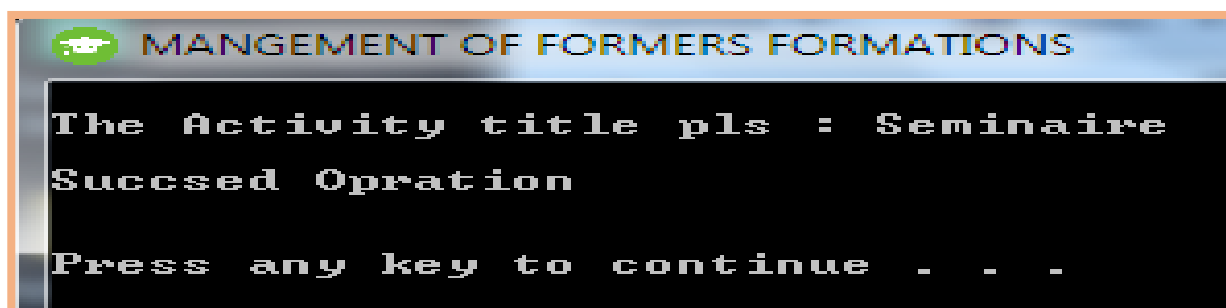


Figure 3.14 : Ajouter une activée (Niveau Régional)

- ➔ Chaque formateur régional, pourra organiser n'importe quel nombre d'activités.

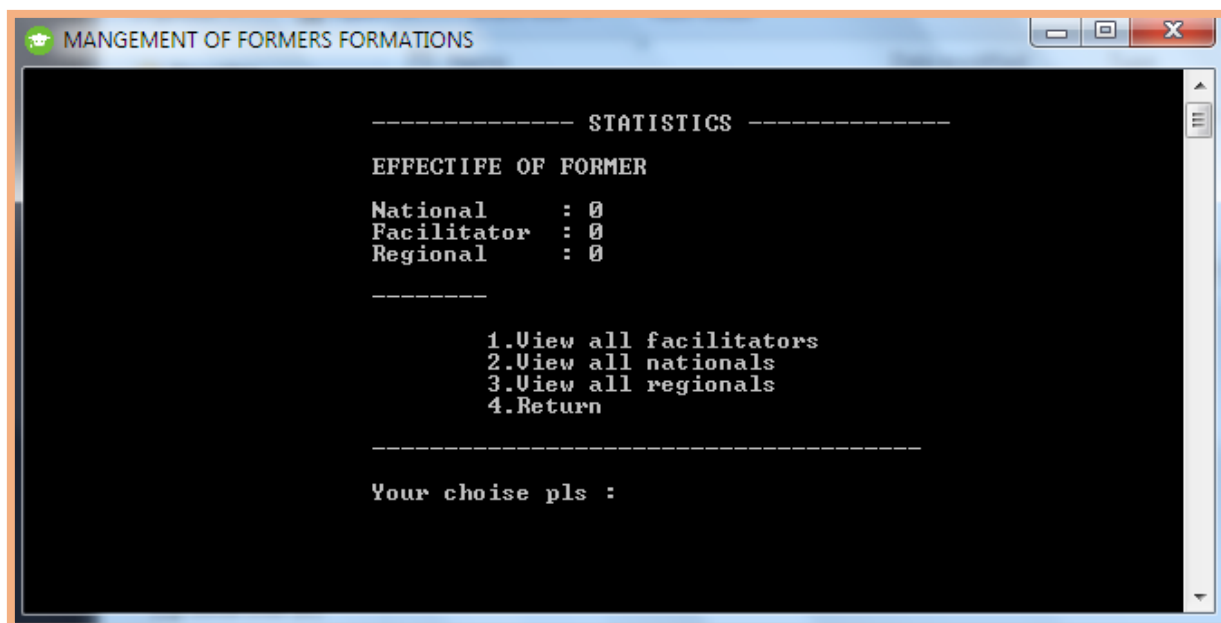


Figure 3.15 : Menu offrant quelques statistiques

- ➔ Cette interface nous présente le nombre d'effectif de chaque niveau, ainsi, il y aura une possibilité d'un filtrage sur les formateurs par les niveaux d'avancement.

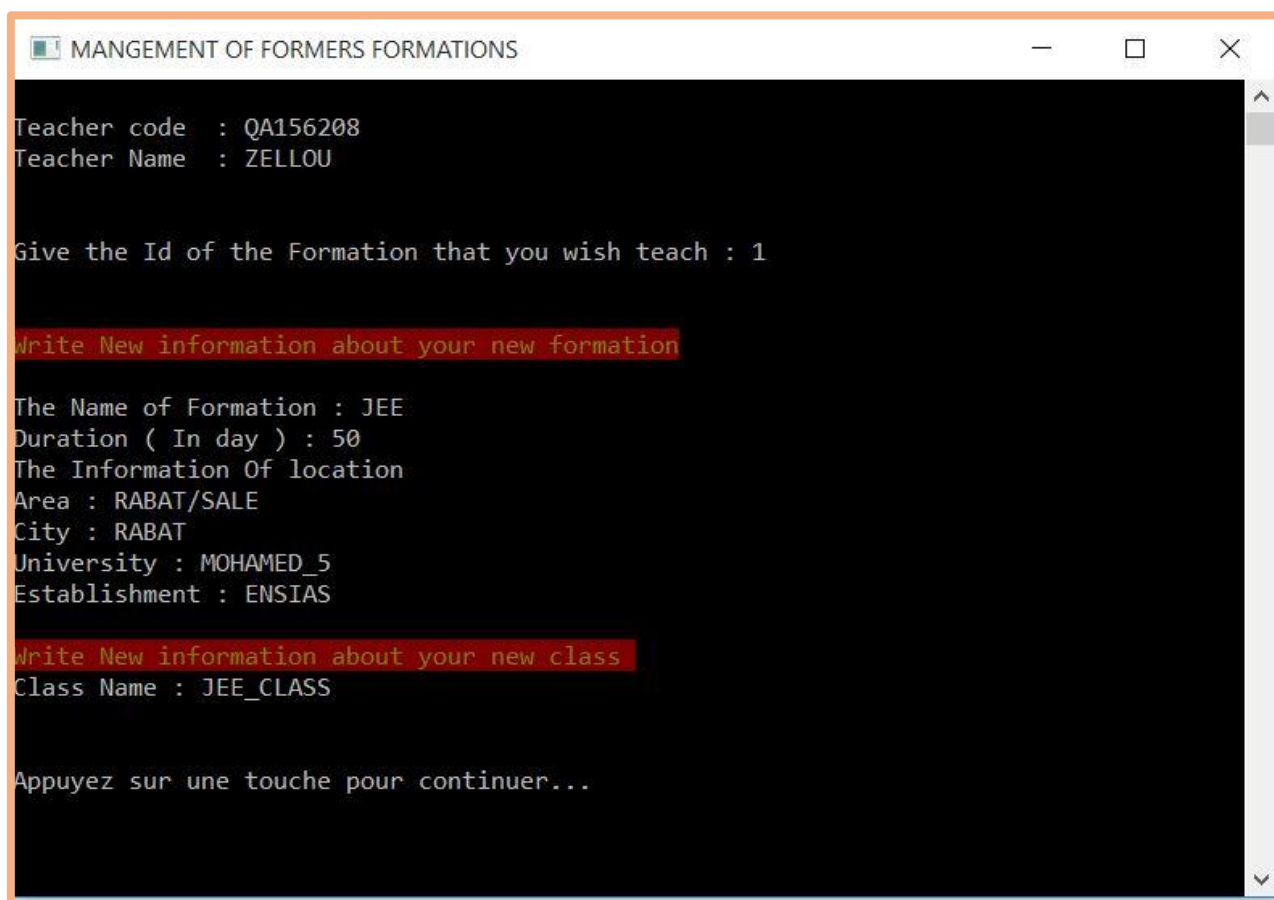


Figure 3.16 : Ajouter un nouvel enseignant

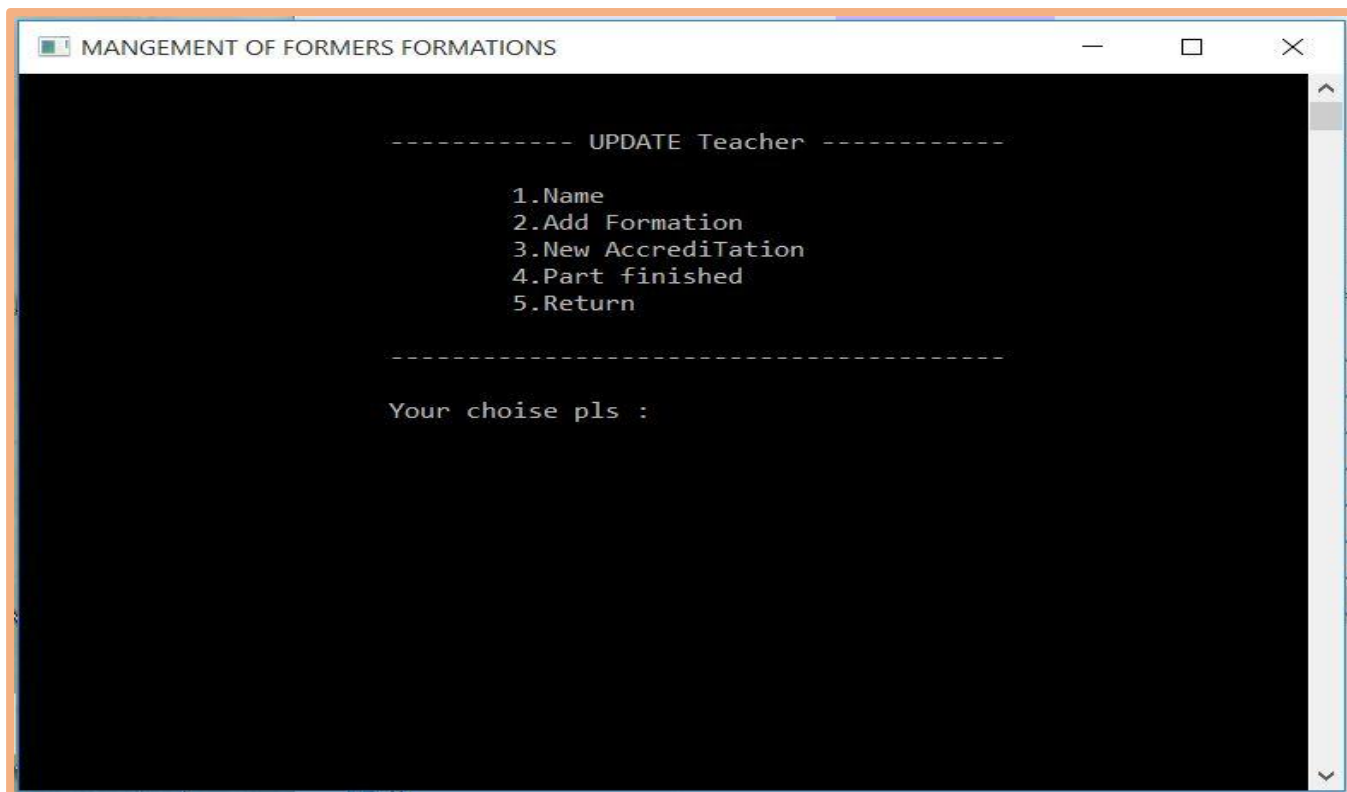


Figure 3.17 : Modifier le profil d'un enseignant.

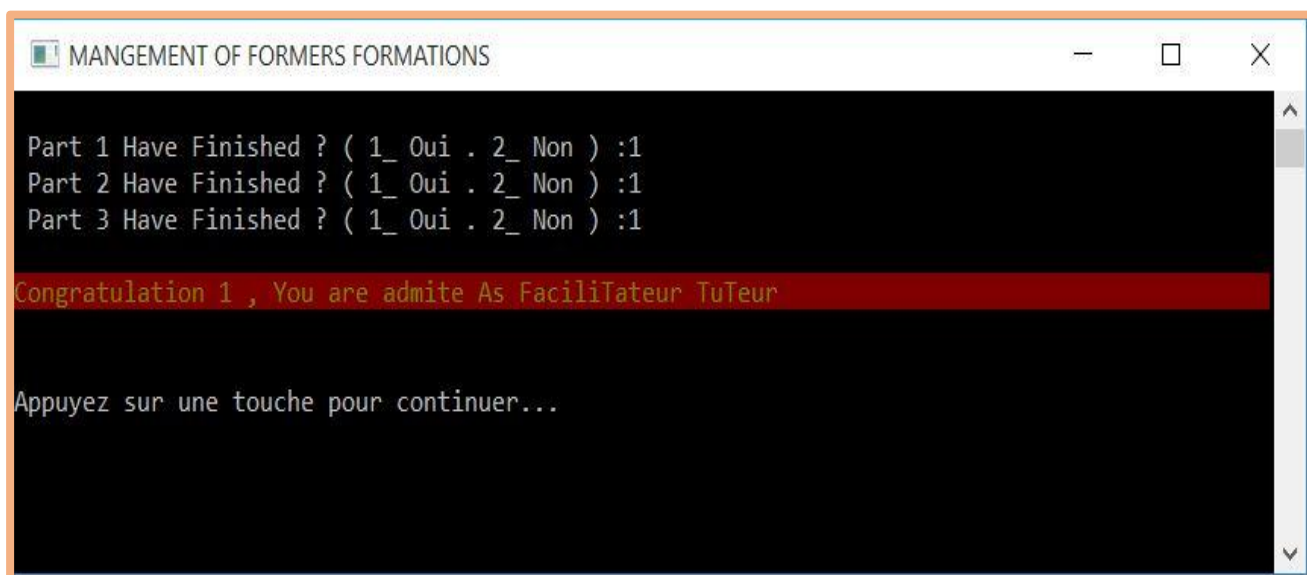


Figure 3.18 : Enseignant complète les 3 parties de la formation suivie

- ➔ Quand l'enseignant a pu de former 3 classes avec 30 élèves chacun, et après avoir obtenir une accréditation ; il reste que compléter les 3 parties de la formation suivie pour qu'il puisse devenir formateur facilitateur tuteur.

```

-----
Id of Formation : 0
The Name of Formation : java
Duration : 30 jours
Type Is : Normal
      Area      : Rabata/Sala
      City       : Rabat
      University : um5
      Establishment : ENSIAS
-----

-----
Id of Formation : 1
The Name of Formation : C++
Duration : 50 jours
Type Is : Normal
      Area      : Rabat/Sala
      City       : Rabat
      University : um5
      Establishment : ENSIAS
-----

-----
Id of Formation : 2
The Name of Formation : Data_Structur
Duration : 50 jours
Type Is : Normal
      Area      : Rabat/Sala
      City       : Rabat
      University : um5
      Establishment : ENSIAS
-----

```

Figure 3.19 : L'affichage des formations

```

NAME      : MNAQUARE

Normal Formation : 1

-----
Id of Formation : 1
The Name of Formation : C++
Duration : 50 jours
Type Is : Normal
      Area      : Rabat/Sala
      City       : Rabat
      University : um5
      Establishment : ENSIAS
-----

Normal Class : 1

+++++++
Class ID      : 132
Formation Key : 1
Class Name    : 1
----- Number of Candidat : 0 -----

Class Type : Normale
+++++++

Accreditate Formation HAVE

-----
Id of Formation : 13
The Name of Formation : 1
Duration : 1 jours
Type Is : Accreditation
      Area      : 1
      City       : 1
      University : 1
      Establishment : 1
-----

Accreditation Class HAVE

+++++++
Class ID      : 129
Formation Key : 13
Class Name    : 11
----- Number of Candidat : 1 -----

Part :      _1._      _2._      _3._
Realise :      .Oui.      .Oui.      .Non.

```

Figure 3.20 : L'affichage d'un enseignant

3 – 2. Session Candidat :

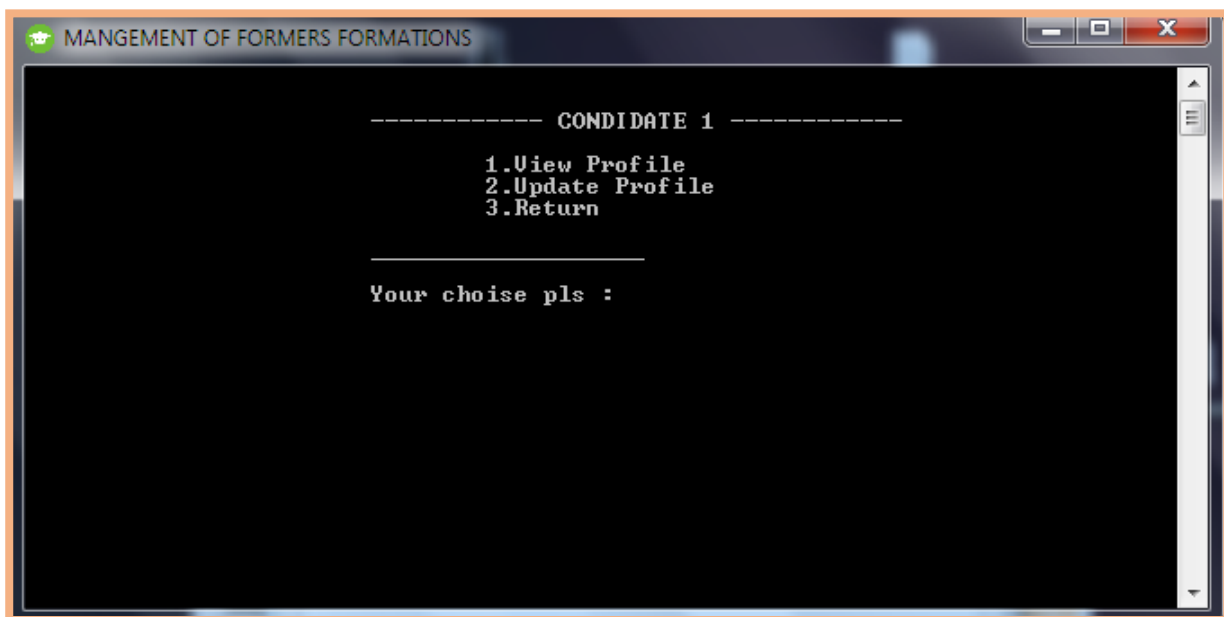


Figure 3.21 : S'authentifier en tant qu'un candidat

- ➔ Une permission d'accès est associée à chaque candidat, tel qu'il ne puisse modifier que son propre profil ou le consulter.

3 – 3. Session Formateur :

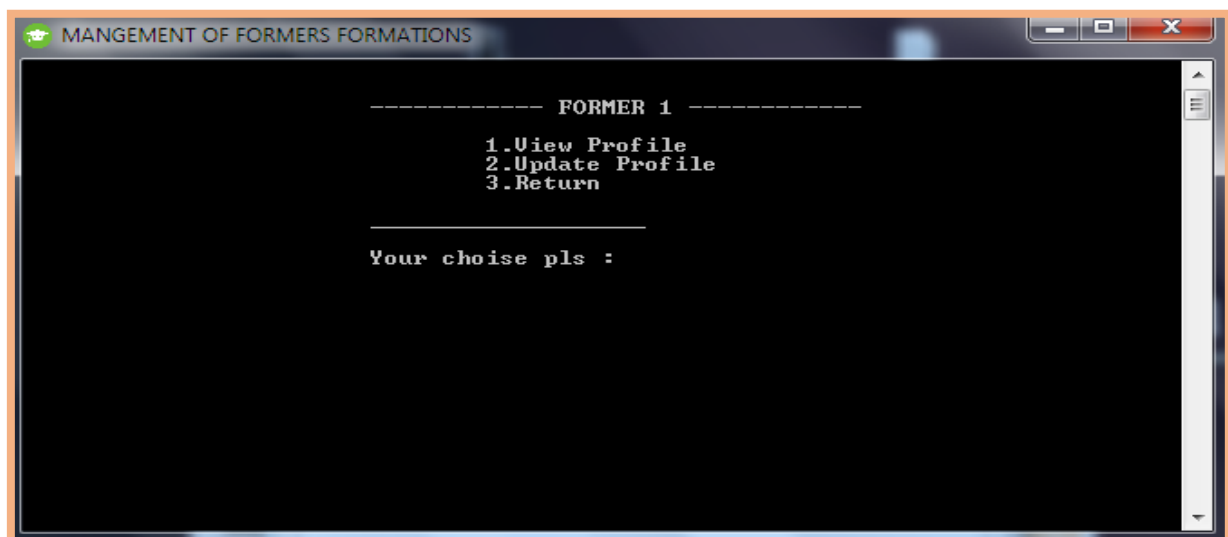


Figure 3.22 : S'authentifier en tant qu'un formateur

- ➔ Un nombre de contrainte est associé à chaque formateur, tel qu'il ne puisse modifier que son propre profil ou le consulter.

Conclusion

Ce rapport explicite une première version du programme demandé, à savoir la gestion des formations. La majorité des fonctionnalités de base de la gestion des candidats, des formateurs, des enseignants, des formations, et des classes ont été implémentées et fonctionnent correctement. Donc Un tel programme est apte à interpréter et enregistrer toutes les informations nécessaires pour le bon fonctionnement d'une gestion de formations.

Le présent projet ouvre des perspectives à plusieurs niveaux. En effet, parmi les perspectives d'améliorations possibles, l'implémentation d'une interface graphique, qui permettra de faciliter les tâches d'enregistrement à travers une interface ergonomique, aussi il y a une possibilité d'offrir des certificats de formations d'accréditation aux différents formateurs ou normaux pour les candidats.