



**Ecole Nationale D'ingénieurs de  
Sfax**

**DEPARTEMENT DE GENIE  
ELECTRIQUE ENIS**



**RAPPORT MINI PROJET C++ : APPLICATION DE  
GESTION DE COMPTE ADMINISTRATEUR**

**REALISE PAR :**

**BEN SALAH NAWEL DORSAF**

**SAOUDI OUMAYMA**

**CLASSE : GE2 S4**

---

**ANNEE UNIVERSITAIRE 2022/2023**

# SOMMAIRE

<b>1. INTRODUCTION :</b>	<b>3</b>
<b>2. LES CLASS DU PROJET :</b>	<b>3</b>
2.1 LISTE DES CLASS :	3
2.2 CLASSE COMPTE :	3
2.2.2 <i>Diagramme de la classe Compte :</i>	3
2.2.3 <i>La description détaillée de la classe Compte</i>	4
2.3 CLASS ADMINAPPL	7
2.3.1 <i>Diagramme de la classe AdminAppl</i>	8
2.3.2 <i>La description détaillée de la classe AdminAppl</i>	8
2.4 CLASSE ADMINGESTION	11
2.4.1 <i>Diagramme de la classe AdminGestion :</i>	11
2.4.2 <i>La description détaillée de la classe AdminGestion</i>	12
<b>3. PARTIE MAIN</b>	<b>15</b>
3.1 DESCRIPTION DE L'APPLICATION	15
3.2 CONNEXION A L'APPLICATION	15
3.3 LE MENU DES OPERATIONS :	15
3.3.1 <i>Liste de compte Admin :</i>	16
3.3.2 <i>Création de nouveaux Comptes</i>	16
3.3.3 <i>Mise à jour de comptes</i>	16
3.3.4 <i>Mise à jour du Password de compte</i>	17
3.3.6 <i>Gestion de mot de passe oublié</i>	18
3.3.7 <i>Dupliquer un compte</i>	18
3.3.8 <i>Supprimer un compte</i>	18
3.3.9 <i>Gestion du code secret</i>	18
3.3.10 <i>Quitter l'application</i>	19

## 1. INTRODUCTION :

---

Le document décrit le mini-projet de C++ réalisé dans le cadre du module ....

L'application implémenté doit gérer les comptes de type administrateurs.

Nous avons créé une application de type console à partir de l'IDE Code blocks.

## 2. LES CLASS DU PROJET :

---

### 2.1 LISTE DES CLASS :

Nous avons trois Class qui ont été implémenté. Compte, AdminAppl et AdminGestion.

La Class Compte représente la Class de base pour les comptes représentantes comptes utilisateurs simple sans privilèges.

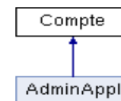
### 2.2 CLASSE COMPTE :

#### Compte Class Reference

classe representant les compte utilisateurs simple sans privileges [More...](#)

```
#include <Compte.h>
```

Inheritance diagram for Compte:



#### 2.2.2 Diagramme de la classe Compte :

## Diagramme de classe

### Compte

#**string** nom  
#**int** id  
# **string** mail  
# **char** passwordappli [10]  
#**string** code secret

+ Compte ()  
+Compte ( const Compte &Admin )  
+ ~Compte()  
+**String** getNom()  
+**Void** setNom(string pnom)  
+**int** getid()  
+**void** setid(int pId)  
+ **string** getMail()  
+**Void** setMail(string pMail)  
+**void** setpasswordappli (string &pPasswordappli)  
+**String** getCodesecret()  
+**void** setCodesecret(string pCodesecret)  
+**Char** \* getPasswordappli(bool pbool)

### 2.2.3 La description détaillée de la classe Compte

#### 2.2.3.1 Les attributs de la classe Compte

## Member Data Documentation

### ◆ codesecret

string Compte::codesecret

Parametre Code secret du compte utilisateur

### ◆ id

int Compte::id

Parametre id du compte utilisateur

### ◆ mail

string Compte::mail

Parametre mail du compte utilisateur

### ◆ nom

string Compte::nom

Parametre Nom du compte utilisateur

### ◆ passwordappli

char Compte::passwordappli[10]

Parametre Password du compte utilisateur

## 2.2.3.2 Les constructeur & destructeur de la classe compte :

## Constructor & Destructor Documentation

### ◆ Compte() [1/2]

Compte::Compte ( )

Constructeur.

Constructeur de la classe **Compte** sans parametre

### ◆ Compte() [2/2]

Compte::Compte ( const **Compte** & Admin )

Constructeur de copie.

Constructeur de la classe **Compte** de copie d objet **Compte**

#### Parameters

**Objet** de classe **Compte**

### ◆ ~Compte()

Compte::~Compte ( )

Destructeur.

Destructeur de la classe **Compte**

### 2.2.3.3 Les méthodes de la classe AdminAppl

#### Member Function Documentation

##### ◆ getCodeSecret()

string Compte::getCodeSecret ( )

retourne le code secret

Methode qui permet de retourner le code secret

##### Returns

de type string (renvoi la valeur du codesecret)

##### ◆ getId()

int Compte::getId ( )

retourner l id

Methode qui permet de retourner le id d utilisateur d un objet de type compte

##### Returns

de type int

##### ◆ getMail()

string Compte::getMail ( )

retourne le mail

Methode qui permet de retourner le mail d utilisateur d un objet de type compte

##### Parameters

aucun parametre

##### Returns

de type String (renvoi la valeur de l attribut mail)

##### ◆ getNom()

string Compte::getNom ( )

Retourne le nom.

Methode qui permet de retourner le nom d utilisateur d un objet de type compte

##### Returns

de type String (renvoi la valeur du nom de l'utilisateurs)

##### ◆ getPasswordAppli()

char \* Compte::getPasswordAppli ( bool pbool )

retourne le password

Methode qui permet d afficher le mot de passe d un compte . Cette methode demande de verifier le code secret de l utilisateur qui doit le fournir avant d afficher le mot de passe.

/param pbool de type boolean (true si la verification du code secret a deja ete effectuee ou false dans le cas contraire) /return de type char \* (renvoi la valeur de l attribut PasswordAppli)

##### ◆ setCodeSecret()

void Compte::setCodeSecret ( string pcodesecret )

modifier le code secret

Methode qui permet de modifier le code secret

##### Parameters

pcodesecret de type string

##### Returns

de type void

#### ◆ setId()

void Compte::setId ( int pid )

Modifier l'id.

Methode qui permet de modifier le id d'utilisateur d'un objet de type compte

##### Parameters

**pid** de type int

##### Returns

de type void

#### ◆ setMail()

void Compte::setMail ( string pmail )

Modifier le mail.

Methode qui permet de modifier le mail d'utilisateur d'un objet de type compte

##### Parameters

**pmail** de type string

##### Returns

de type void

#### ◆ setNom()

void Compte::setNom ( string pnom )

Modifier le nom.

Methode qui permet de modifier le nom d'utilisateur d'un objet de type compte

##### Parameters

**pnom** de type String (la valeur modifie l'attribut nom)

##### Returns

de type void

#### ◆ setPasswordapli()

void Compte::setPasswordapli ( string & pPasswordapli )

Modifier le password.

Methode qui permet de modifier le mot de passe d'un objet de type compte

##### Parameters

**pPasswordapli** de type string (modifie la valeur de l'attribut Passwordapli)

##### Returns

de type void

## 2.3 CLASS ADMINAPPL

### AdminAppl Class Reference

classe representant les compte utilisateurs administrateurs [More...](#)

```
#include <AdminAppl.h>
```

Inheritance diagram for AdminAppl:



⇒ classe représentante les comptes utilisateurs administrateurs : cette classe permet de créer un compte utilisateur de type administrateurs et de gérer toutes les actions de gestion du compte administrateur.

### 2.3.1 Diagramme de la classe AdminAppl

<u>Diagramme de classe</u>
AdminAppl
<b>Static</b> int numeroAdmin=10
 + AdminAppl () +AdminAppl( int pnumeroAdmin ) + AdminAppl(const AdminAppl & Admin) +~AdminAppl() +AdminAppl& operator =(const AdminAppl &source) + <b>bool</b> operator ==(const AdminAppl& source ) const + <b>bool</b> operator=(const AdminAppl & source)const

### 2.3.2 La description détaillée de la classe AdminAppl

#### 2.3.2.1 Les attributs de la classe AdminAppl

La classe AdminAppl hérite de la classe Compte. Elle hérite donc de tous les attributs de la Classe Compte.

#### Static Public Attributes

```
static int numeroAdmin = 10
```

#### Additional Inherited Members

► Protected Attributes inherited from **Compte**



### 2.3.2.2 Les constructeurs & destructeurs de la classe AdminAppl

#### Constructor & Destructor Documentation

##### ◆ AdminAppl() [1/3]

AdminAppl::AdminAppl ( )

Constructeur.

Constructeur de la classe **AdminAppl** sans parametre

##### ◆ AdminAppl() [2/3]

AdminAppl::AdminAppl ( int pnumeroAdmin )

Constructeur avec parametre.

Constructeur de la classe **AdminAppl** avec le parametre numero d'utilisateur d'un objet **AdminAppl**. Le nom d'utilisateur et le mot de passe sont saisis par l'utilisateur et le mot de passe est masque par des \*.

##### Parameters

**pnumeroAdmin** de type int

##### ◆ AdminAppl() [3/3]

AdminAppl::AdminAppl ( const AdminAppl & Admin )

Constructeur.

Constructeur de la classe **AdminAppl** de copie d'objet **AdminAppl**

##### Parameters

**Admin** de type **AdminAppl**

##### ◆ ~AdminAppl()

AdminAppl::~AdminAppl ( )

Destructeur.

Destructeur de la classe **AdminAppl**

### 2.3.2.3 Les méthodes de la classe AdminAppl

#### Member Function Documentation

##### ◆ operator!=()

```
bool AdminAppl::operator!= ( const AdminAppl & source ) const
```

surcharge de l'opérateur de différence (!=)

surcharge de l'opérateur de différence pour comparer deux objet de type AdminAppl

##### Parameters

**source** de type AdminAppl

##### Returns

de type boolean

##### ◆ operator=()

```
AdminAppl & AdminAppl::operator= ( const AdminAppl & source )
```

surcharge de l'opérateur d'affectation

surcharge de l'opérateur d'affectation pour comparer deux objet de type AdminAppl

##### Parameters

**source** de type AdminAppl

##### Returns

objet de type AdminAppl

##### ◆ operator==()

```
bool AdminAppl::operator== ( const AdminAppl & source ) const
```

surcharge de l'opérateur d'affectation

surcharge de l'opérateur d'affectation pour comparer deux objet de type AdminAppl

##### Parameters

**source** de type AdminAppl

##### Returns

de type boolean

#### Friends And Related Function Documentation

##### ◆ operator<<

```
std::ostream& operator<< ( std::ostream & out,  
                          AdminAppl & source  
                          )
```

surcharge de l'opérateur de flux <<

Surcharger l'opérateur << qui permet d'afficher le nom d'utilisateur et le mot de passe d'un objet AdminAppl non masqué si le code secret est correct

##### Parameters

**out** de type ostream

**source** de type AdminAppl

##### Returns

de type ostream

### 2.3.2.4 Les amis de la classe AdminAppl

#### Friends

class	<b>AdminGestion</b>
std::ostream &	<b>operator&lt;&lt;</b> (std::ostream &out, <b>AdminAppl</b> &source) surcharge de l'opérateur de flux << <a href="#">More...</a>

## 2.4 CLASSE ADMINGESTION

### AdminGestion Class Reference

classe de gestion des compte utilisateurs de type administrateurs [More...](#)

```
#include <AdminGestion.h>
```

Classe de gestion des comptes utilisateurs de type administrateurs.  
Cette classe permet de gérer tous les actions de gestion des comptes administrateurs.

### 2.4.1 Diagramme de la classe AdminGestion :

<u>Diagramme de classe</u>
AdminGestion
String SuperAdmin = 'admin' ; String SuperPassword = 'admin' ; String UserCorrection ; String PasswordCorrection;
+ AdminGestion (string puserld, string puserPassword) +AdminGestion( int a,string puserled, string puserPassword ) +~AdminGestion() +int getNumeroAdmin() +void setNumeroAdmin( int pnumeroAdmin_G) +void listercompteAdmin(int id) +void create userAdmin (int id) +void updateUserAdmin(int id) +void updateUserPasswordAdmin(int id) +void readUser Admin(int id) +void manageForgetUserPasswordAdmin(int id) +void duplicateUserAdmin(int id , int idcopie) +void deleteUserAdmin(int id) +void managecodeSecret(int id) +void manageExitfunct() +bool AdminConnecte()

## 2.4.2 La description détaillée de la classe AdminGestion

### 2.4.2.1 Les attributs de la classe AdminGestion

```
std::map<int, AdminAppl*> mapAdmin; /*< Parametre de type map mapAdmin pour référencer les comptes administrateurs*/
```

```
string SuperAdmin = "admin"; /*< Parametre SuperAdmin du compte de gestion des comptes administrateur*/  
string SuperPassword = "admin"; /*< Parametre SuperPassword du compte de gestion des comptes administrateur*/  
string UserConnection; /*< Parametre UserConnection du compte de connexion utilise*/  
string PasswordConnection; /*< Parametre PasswordConnection du compte de connexion utilise*/
```

### Static Public Attributes

```
static int numeroAdmin_G = 1
```

### 2.4.2.2 Les constructeurs & destructeurs de la classe AdminGestion :

#### Constructor & Destructor Documentation

##### ◆ AdminGestion() [1/2]

```
AdminGestion::AdminGestion ( string puserId,  
                             string puserPassword  
                             )
```

Constructeur.

Constructeur de la classe **AdminGestion** avec parametre

##### Parameters

**puserId** de type string (user de connection)  
**puserPassword** de type string (password de connection)

##### ◆ AdminGestion() [2/2]

```
AdminGestion::AdminGestion ( int a,  
                             string puserId,  
                             string puserPassword  
                             )
```

Constructeur.

Constructeur de la classe **AdminGestion** avec parametre

##### Parameters

**a** de int pour la gestion des entrees dans la map referencant les utilisateurs de type administrateurs  
**puserId** de type string (user de connection)  
**puserPassword** de type string (password de connection)

##### ◆ ~AdminGestion()

```
AdminGestion::~AdminGestion ( )
```

Destructeur.

Destructeur de la classe **AdminGestion**

### 2.4.2.3 Les méthodes de la classe AdminGestion :

## Member Function Documentation

### ◆ AdminConnecte()

bool AdminGestion::AdminConnecte ( )

gestion de connexion

Methode qui permet la gestion de connexion

#### Returns

de type void

### ◆ createUserAdmin()

void AdminGestion::createUserAdmin ( int id )

créer un compte administrateurs

Methode qui permet de créer un compte administrateur

#### Returns

de type void

### ◆ deleteUserAdmin()

void AdminGestion::deleteUserAdmin ( int id )

supprimer un compte administrateurs

Methode qui permet de supprimer un compte administrateurs

#### Parameters

de type int pour definir l id du compte administrateur a traiter

#### Returns

de type void

### ◆ duplicateUserAdmin()

void AdminGestion::duplicateUserAdmin ( int id,  
int idcopie  
)

dupliquer un compte administrateurs

Methode qui permet de dupliquer un compte administrateurs

#### Parameters

de type int pour definir l id du compte administrateur a traiter

#### Returns

de type void

### ◆ getNumeroAdmin()

int AdminGestion::getNumeroAdmin ( )

Retourne le NumeroAdmin\_G.

Methode qui permet de retourner le NumeroAdmin\_G

#### Returns

de type int (renvoi la valeur du numeroAdmin\_G)

### ◆ ListerCompteAdmin()

void AdminGestion::ListerCompteAdmin ( )

lister les compte administrateurs

Methode qui permet de lister les compte administrateurs

#### Returns

de type void

#### ◆ manageCodeSecret()

void AdminGestion::manageCodeSecret ( int id )

gestion du code secret d un compte administrateurs

Methode qui permet gestion du code secret d un compte administrateurs

##### Parameters

de type int pour definir l Id du compte administrateur a traiter

##### Returns

de type void

#### ◆ manageExitfunct()

void AdminGestion::manageExitfunct ( )

gestion de sortie de fonction

Methode qui permet la gestion de sortie de fonction

##### Returns

de type void

#### ◆ manageForgetUserPasswordAdmin()

void AdminGestion::manageForgetUserPasswordAdmin ( int id )

gerer le mot de passe oublie d un compte administrateurs

Methode qui permet de gerer le mot de passe oublie d un compte administrateurs

##### Parameters

de type int pour definir l Id du compte administrateur a traiter

##### Returns

de type void

#### ◆ readUserAdmin()

void AdminGestion::readUserAdmin ( int id )

affiche les donnees d un compte administrateurs

Methode qui permet de affiche les donnees d un compte administrateur

##### Parameters

de type int pour definir l Id du compte administrateur a traiter

##### Returns

de type void

#### ◆ setNumeroAdmin()

void AdminGestion::setNumeroAdmin ( int pnumeroAdmin\_G )

Modifier NumeroAdmin\_G.

Methode qui permet de modifier le NumeroAdmin\_G (celui-ci permet de gerer les entrées dans la map)

##### Parameters

pnom de type int (la valeur modifie l attribut numeroAdmin\_G)

##### Returns

de type void

#### ◆ updateUserAdmin()

void AdminGestion::updateUserAdmin ( int id )

modifi un compte administrateurs

Methode qui permet modifi un compte administrateur

##### Parameters

de type int pour definir l Id du compte administrateur a traiter

##### Returns

de type void

#### ◆ updateUserPasswordAdmin()

void AdminGestion::updateUserPasswordAdmin ( int id )

modifi le password d un compte administrateurs

Methode qui permet de modifi le password d un compte administrateur

##### Parameters

de type int pour definir l Id du compte administrateur a traiter

##### Returns

de type void

### 3. PARTIE MAIN

#### 3.1 DESCRIPTION DE L'APPLICATION

L'application a été réalisée à partir d'un projet console.

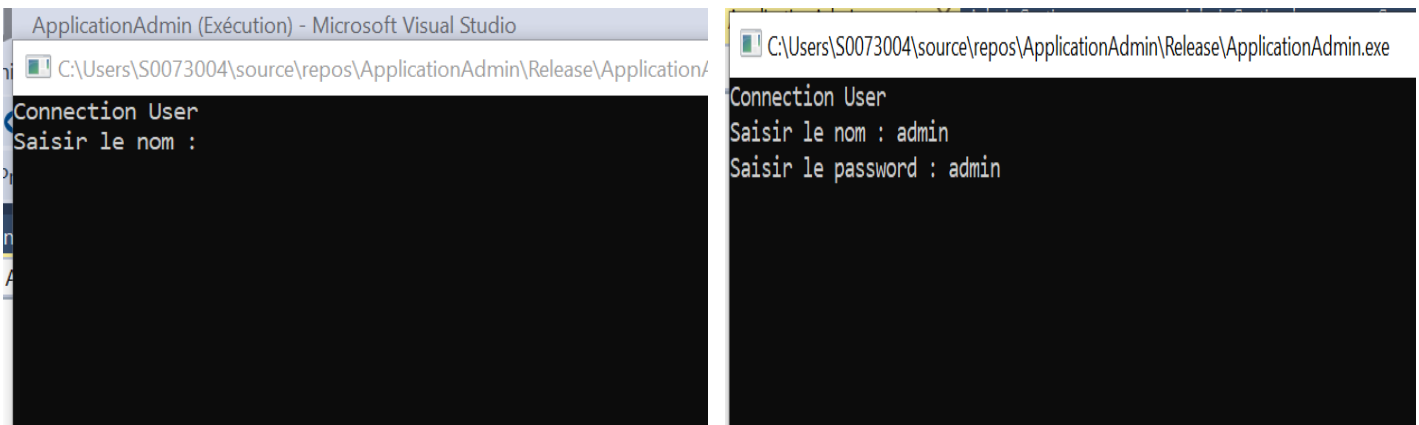
Un fichier main a été créé pour utiliser les différentes classes définies et réaliser les différentes opérations.

L'algorithme est le suivant :

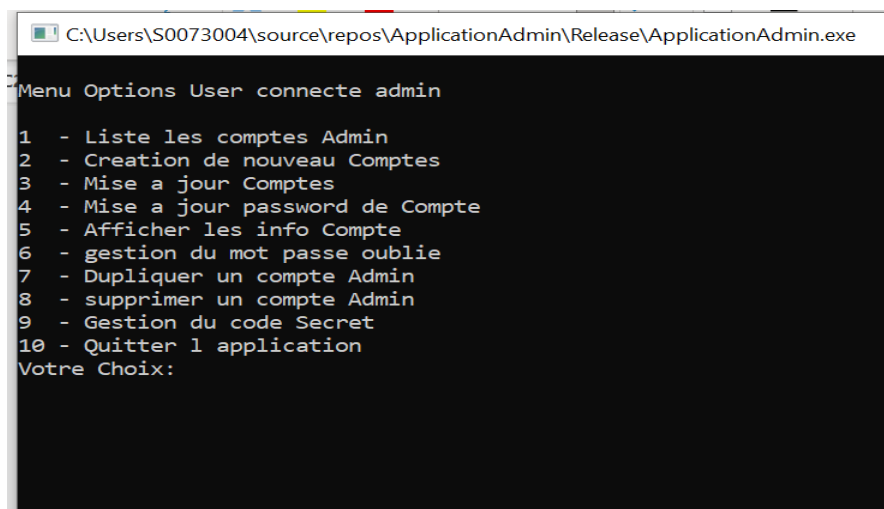
- Connexion au super administrateur pour avoir les droits de gestion totale sur les comptes de types administrateurs
- Affichage du menu des différentes opérations qui permettent de gérer les comptes administrateurs
- Faire un choix pour réaliser une opération

#### 3.2 CONNEXION A L'APPLICATION

Connexion à l'application de gestion des comptes Administrateurs. Il faudra saisir l'utilisateur admin/admin pour avoir l'ensemble des privilèges.



#### 3.3 LE MENU DES OPERATIONS :



### 3.3.1 Liste de compte Admin :

Faire le choix 1 et Entrée

```
C:\Users\S0073004\source\repos\ApplicationAdmin> E:\MINIPROJET\MiniProjet_20221204\bin\Debug\MiniProjet_20221204.exe

Menu Options User connecte admin
1 - Liste les comptes Admin
2 - Creation de nouveau Comptes
3 - Mise a jour Comptes
4 - Mise a jour password de Compte
5 - Afficher les info Compte
6 - gestion du mot passe oublie
7 - Dupliquer un compte Admin
8 - supprimer un compte Admin
9 - Gestion du code Secret
10 - Quitter l application
Votre Choix: 1

liste des Comptes Admin
-----
ID      |      Nom
-----
1       |      omayma
2       |      Cherine
3       |      nawel
-----
fin de la liste
-----

Enter to return Menu :
```

### 3.3.2 Création de nouveaux Comptes

Faire le choix 2 et Enter.

Ensuite suivre les différentes demandes. Et saisir Enter après chaque réponse.

```
C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
createUserAdmin : createUserAdmin
Saisir le nom :
Syrine

C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
createUserAdmin : createUserAdmin
Saisir le nom :
Syrine
Saisir le password :
*****
Saisir le codesecret :
Tunisie
Saisir votre mail :
syrine.bensalah@yahoo.fr

Enter to return Menu :
```

### 3.3.3 Mise à jour de comptes

Faire le choix 3 et Entrée

```
C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe

Menu Options User connecte admin
1 - Liste les comptes Admin
2 - Creation de nouveau Comptes
3 - Mise a jour Comptes
4 - Mise a jour password de Compte
5 - Afficher les info Compte
6 - gestion du mot passe oublie
7 - Dupliquer un compte Admin
8 - supprimer un compte Admin
9 - Gestion du code Secret
10 - Quitter l application
Votre Choix: 3
=====
MISE A JOUR
=====
Choisir un Id du compte que vous voulez modifier:
2
```



```
C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
updateUserAdmin
Nom actuel : Cherine
updateUserAdmin: mettre á jour Nom
Lylia

Enter to return Menu :
```

### 3.3.4 Mise à jour du Password de compte

Faire le choix 4 et Entrée, et suivre les instructions

```
C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
Menu Options User connecte admin
1 - Liste les comptes Admin
2 - Creation de nouveau Comptes
3 - Mise a jour Comptes
4 - Mise a jour password de Compte
5 - Afficher les info Compte
6 - gestion du mot passe oublie
7 - Dupliquer un compte Admin
8 - supprimer un compte Admin
9 - Gestion du code Secret
10 - Quitter l application
Votre Choix: 4
=====
Mise á jour
=====
Choisir un Id du compte pour lequel que vous voulez modifier le password:
5
```

```
C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
updateUserPasswordAdmin
Modification du password
updateUserPasswordAdmin: saisir le code secret
avion
vous pouvez modifier votre password
*****saisie effectuÚ

Enter to return Menu :
```

### 3.3.5 Afficher les infos de compte :

Faire le choix 5 et Enter

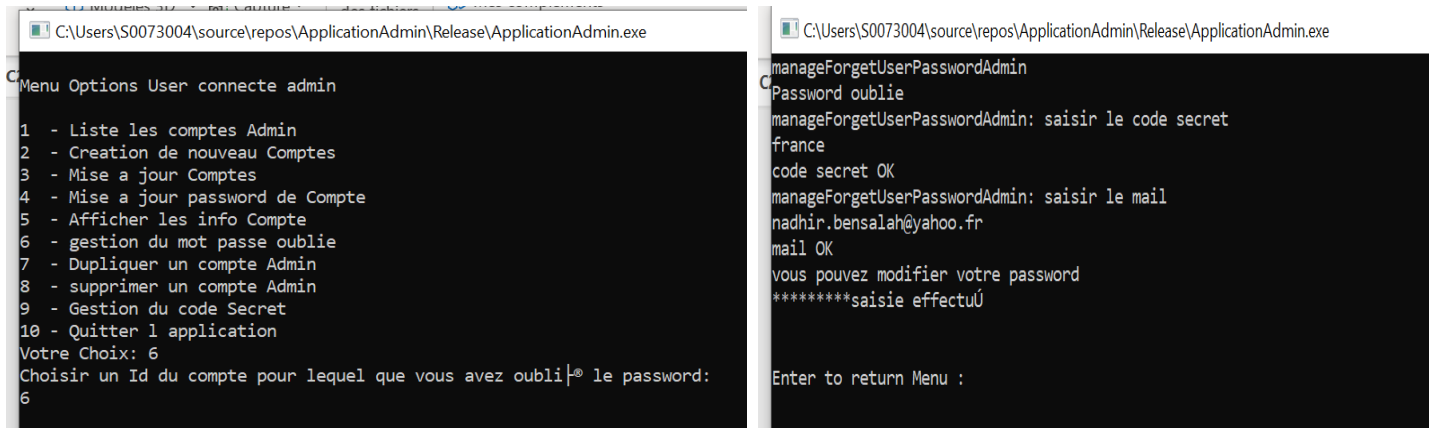
Et choisir un ID de la liste des ID des comptes Administrateurs existants

```
C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
Menu Options User connecte admin
1 - Liste les comptes Admin
2 - Creation de nouveau Comptes
3 - Mise a jour Comptes
4 - Mise a jour password de Compte
5 - Afficher les info Compte
6 - gestion du mot passe oublie
7 - Dupliquer un compte Admin
8 - supprimer un compte Admin
9 - Gestion du code Secret
10 - Quitter l application
Votre Choix: 5
Choisir un Id du compte pour lequel que vous voulez les infos:
5
```

```
C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
readUserAdmin: readUserAdmin
Nom : Yasin
mail: yasin.bensalah@yahoo.fr
numAdmin 15

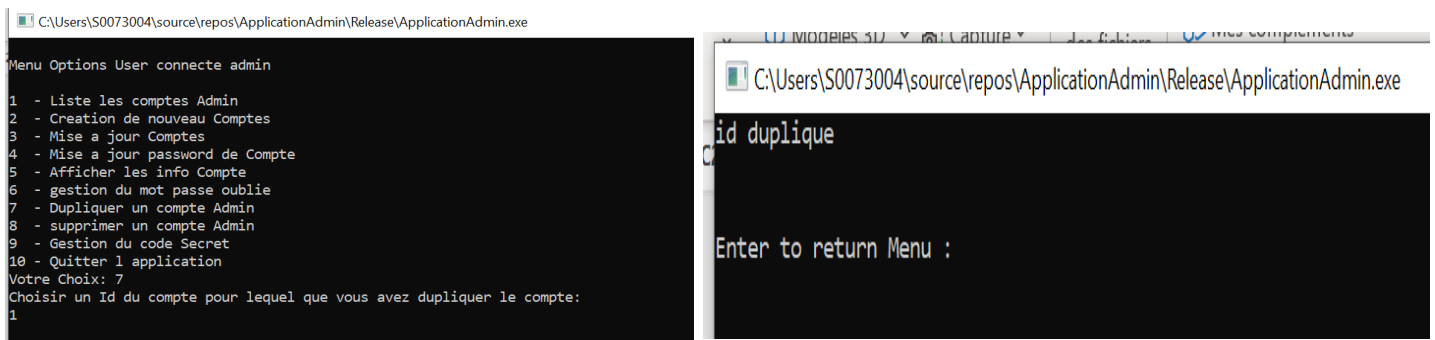
Enter to return Menu :
```

### 3.3.5 Gestion de mot de passe oublié



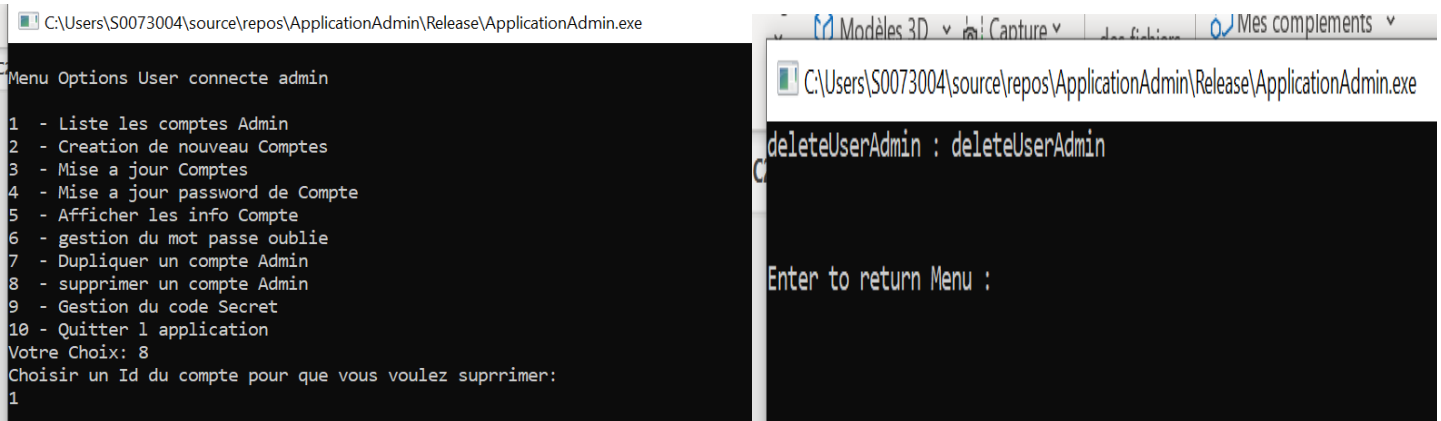
### 3.3.6 Dupliquer un compte

Faire le choix 7 et Enter  
Et suivre les instructions



### 3.3.7 Supprimer un compte

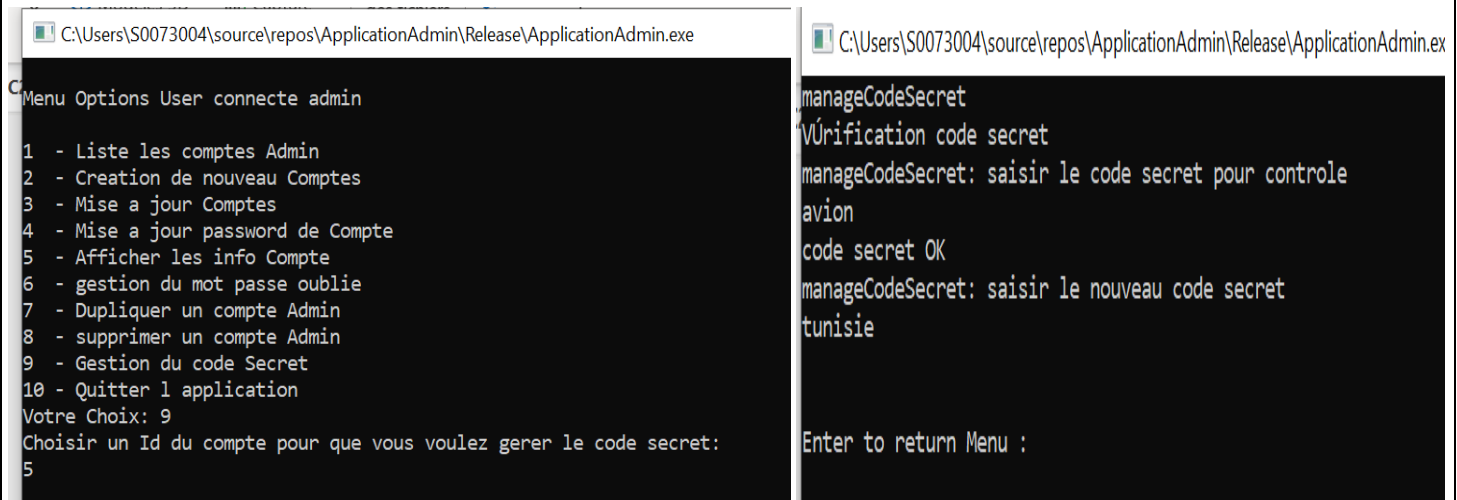
Faire le choix 8 et Enter



### 3.3.8 Gestion du code secret

Faire le choix 9 et Enter

## Et suivre les instructions



```
C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
Menu Options User connecte admin
1 - Liste les comptes Admin
2 - Creation de nouveau Comptes
3 - Mise a jour Comptes
4 - Mise a jour password de Compte
5 - Afficher les info Compte
6 - gestion du mot passe oublie
7 - Dupliquer un compte Admin
8 - supprimer un compte Admin
9 - Gestion du code Secret
10 - Quitter l application
Votre Choix: 9
Choisir un Id du compte pour que vous voulez gerer le code secret:
5

C:\Users\S0073004\source\repos\ApplicationAdmin\Release\ApplicationAdmin.exe
manageCodeSecret
Vérification code secret
manageCodeSecret: saisir le code secret pour controle
avion
code secret OK
manageCodeSecret: saisir le nouveau code secret
tunisie

Enter to return Menu :
```

### 3.3.9 Quitter l'application

Faire le choix 10 Et Enter. L'application se fermera.