



# GESTION DE SURVEILLANCE DES EXAMENS

**Réalisé par :**

SABOUR Ilham

NAKACH Fatima-Zahrae

**Encadré par :**

M. Redouan

## **SOMMAIRE :**

### **-Introduction générale**

### **Chapitre I : Problématique et cadre général du projet.**

- 1. Introduction**
- 2. Problématique**
- 3. Analyse et spécification des besoins**
  - 3.1 Spécification des besoins fonctionnels**
  - 3.2 Spécification des besoins non fonctionnels**
- 4. Caractéristiques de l'application**
- 5. Les étapes de la résolution du problème**

### **CHAPITRE II: MISE EN ŒUVRE ET REALISATION**

- 1. Introduction**
- 2. Les outils et techniques de développement**
- 3. Scenario d'exécution**
- 4. La console (Mode utilisateur et administrateur)**

### **-Conclusion générale**

## INTRODUCTION GENERALE :

L'homme a créé la machine afin de lui rendre des services et de lui faciliter des tâches qui lui sont répétitives ou complexes. Il exploite cette dernière dans différents domaines. Ainsi, la machine est intégrée, au fur et à mesure, dans toute notre vie quotidienne.

Le capital humain est le centre de cet exploit, ce qui fait que la gestion de ce capital est l'une des principales pratiques au sein de toute organisation ou regroupement de personnes. De ce fait, nous tenons à réaliser une application de gestion de ces ressources afin d'affiner l'organisation au sein d'un groupe et durant une période précise.

Cette application consiste à organiser sous forme de fichiers les données d'un établissement scolaire tout au long de la période des examens. Elle permet à l'utilisateur d'accéder à son compte où se trouvera un planning de contrôles et où il pourra faire les modifications souhaitées. Cette application doit être à la portée de tout l'organisme éducatif, ainsi elle doit être simple à manipuler, claire et efficace.

# ***CHAPITRE 1:***

# **Problématique et cadre général du projet.**

## **1. Introduction :**

Ce chapitre nous donne une vision globale sur le projet et nous pose la problématique qui nous a poussés à penser à la réalisation d'une telle application.

## **2. Problématique :**

Un établissement scolaire regroupe un réseau très vaste composé de la direction, du comité pédagogique, des enseignants ainsi que des étudiants. Toutes ces ressources doivent être organisées et traitées de sorte que tout élément ait ses propres droits dont il doit être satisfait. Pour cela il faut une gestion bien équilibrée et organisée afin de mettre en ordre le système éducatif surtout lors de la période des examens.

Le schéma ci-dessous illustre en gros la fonction de l'application :

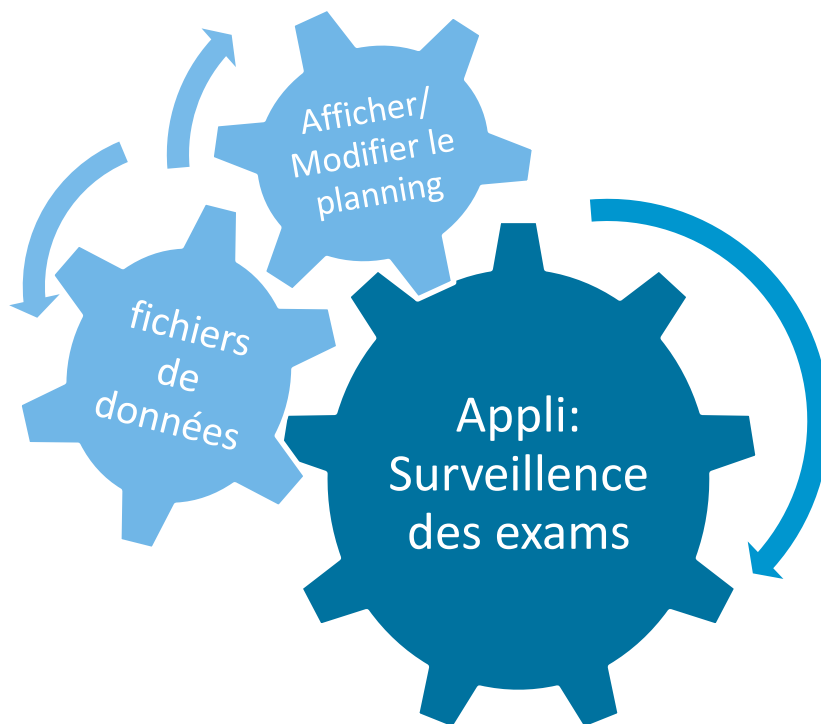


Diagramme1 : Fonctionnalités de l'application

### **3. Analyse et spécification des besoins:**

#### **3.1 Spécification des besoins fonctionnels:**

L'application doit pouvoir gérer deux types d'utilisateurs : l'administrateur et l'enseignant.

Par rapport à l'administrateur, notre application doit répondre aux exigences suivantes :

- L'administrateur doit avoir un identifiant et un mot de passe à saisir pour se connecter.
- Il faut pouvoir visualiser l'ensemble des surveillances dans une même liste avec la possibilité d'appliquer des filtres sur celle-ci.

- Il faut pouvoir manipuler les surveillances : ajouter, modifier ou supprimer une surveillance.

- Il faut pouvoir manipuler les comptes des enseignants (changer le mot de passe, supprimer ou créer un compte )

Par rapport aux enseignants, notre application doit répondre aux exigences suivantes :

- Les enseignants doivent avoir un compte pour se connecter.
- Une fois connecté L'enseignant pourra voir son planning durant la période des examens et pourra accéder à ses informations ( Nom, prénom, login, mot de passe, email et filière)
- L'enseignant ne peut pas modifier le planning ni voir la totalité des surveillances.
- l'enseignant peut modifier son mot de passe et son email mais ne peut pas supprimer son compte.

#### **3.2 Spécification des besoins non fonctionnels :**

Afin d'assurer un bon fonctionnement de l'application et de garantir la satisfaction de l'utilisateur, des contraintes doivent être prises en compte tout au long du développement du projet :

- Ergonomie et convivialité : L'application doit fournir une interface simple et élégante pour l'utilisateur afin de faciliter l'exploitation des services de l'application ;

- Portabilité, maintenance et évolution : L'application doit être multiplateforme c'est à-dire fonctionnelle sur n'importe quel système d'exploitation. Le code doit être bien lisible, compréhensible et modulaire pour garantir la souplesse, l'évolution et la maintenance de la solution et ainsi répondre aux changements du marché.

- Rapidité et robustesse : L'application doit assurer la rapidité de réponse et la fiabilité du résultat procuré.

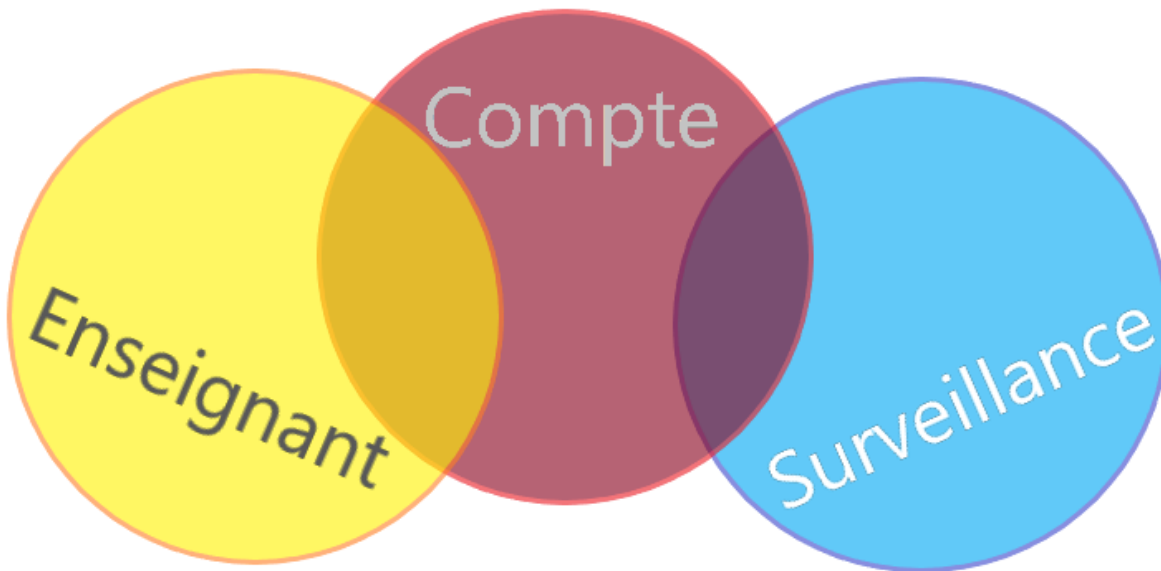
- Sécurité : L'application doit tenir compte de la confidentialité des données des utilisateurs.

#### **4. Caractéristiques de l'application :**

- Rapide et sécurisée ;
- Affichage simplifié sur la console ;
- Langue utilisée: français.

## 5. Les étapes de la résolution du problème

La gestion du programme s'articule essentiellement sur la gestion de 3 éléments: « Surveillance », « Compte » et « Enseignant ». Cette division s'applique à la fois sur les structures, les fonctions et les fichiers.



### 5.1 LES STRUCTURES :

Le programme est basé sur 3 structures:

-La structure « Compte » :

```
typedef struct Compte{  
    bool admin;  
    char login[90];  
    char mdp[20];  
    char enseignant[90];  
}_Compte;
```

Chaque compte est caractérisé par les informations suivantes: un booléen qui prend la valeur 1 si le compte correspond à celui d'un administrateur et 0 sinon, le login, le mot de passe et le nom du propriétaire du compte.



**-La structure « Enseignant » :**

```
typedef struct Enseignant{  
    char nom[95];  
    char prenom [95];  
    char filiere[90];  
    char email[90];  
}_Enseignant;
```

Chaque élément de type « enseignant » contient les informations suivantes: nom, prénom, département et email de l'enseignant.

**-La structure « Surveillance » :**

```
typedef struct Surveillance{  
    char salle[4];  
    char date[11];  
    char duree[12];  
    char enseignant[100];  
    char module[70];  
    char filiere[100];  
}_Surveillance;
```

Chaque surveillance est caractérisée par le numéro de la salle où se déroule l'épreuve, la date de l'examen sous la forme (jour/mois/année), la durée de l'épreuve sous la forme (heure début – heure de fin), le nom du surveillant qui est un enseignant, le module de l'épreuve et la filière concernée.

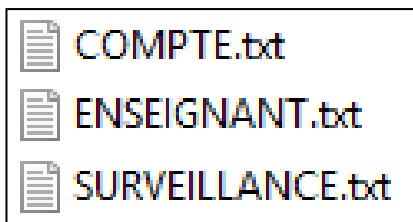
**5.2 LES FICHIERS :**

Le programme est composé de 3 types de fichiers :

- Des fichiers textes qui contiennent les informations des bases de données.
- Des fichiers « .c » qui contiennent le main et les fonctions utilisés dans le programme.
- Un fichier « .h » qui contient le header du programme.

Et par la suite on détaillera les caractéristiques de chaque fichier.

-On dispose de 3 fichiers textes :



Dans tous ces fichiers une ligne représente un élément et contient tout les informations qui le correspondent. Deux informations d'une même ligne sont séparées par une tabulation et chaque ligne est suivie d'un retour à la ligne (même la dernière).

Si l'une de ces conditions n'est pas respectée le programme affiche un message d'erreur et s'arrête.

## -Le fichier COMPTE.txt :

Chaque ligne représente les informations d'un compte :

0 ou 1 (Si le compte est celui d'un administrateur ou non)

Le login, qui est le nom du surveillant en minuscule suivit de « 2019 »

Le mot de passe (aléatoire), l'utilisateur ou l'administrateur peut le changer par la suite.

Le nom de l'utilisateur.

0	aantri2019	abc	AANTRI
0	amrani2019	123	AMRANI
0	janati2019	ABC	JANATI
0	el_faker2019	a1b2c3	EL FAKER
0	rachid2019	000	RACHID
1	admin	admin	ADMINISTRATEUR

## -Le fichier ENSEIGNANT.txt :

Chaque ligne représente les informations d'un enseignant :

Nom, prénom, département et email.

Si l'enseignant n'a pas encore donner son email « un espace vide » le remplace par défaut.  
*L'utilisateur ou l'administrateur peut par la suite modifier l'email.*


## -Le fichier SURVEILLANCE.txt :


Chaque ligne représente les informations d'une surveillance :


Salle, date, durée, le nom de surveillant, module et filière.


LC1	02/01/2019	14:30-16:00	EL FAKER	M.1.2	IeL
A7	03/01/2019	09:00-11:00	JANATI	M.1.5	e-MBI
LC2	04/01/2019	14:30-16:00	HAMYANI	M.1.3	ISEM
LC3	05/01/2019	09:00-11:00	AMRANI	M.1.6	SSE
A9	09/01/2019	14:30-16:00	SENHADJI	M.1.4	GL
A7	03/01/2019	09:00-11:00	RACHID	M.1.5	e-MBI
LC2	04/01/2019	14:30-16:00	RACHID	M.1.3	ISEM
LC3	05/01/2019	09:00-11:00	RACHID	M.1.6	SSE
A9	09/01/2019	14:30-16:00	RACHID	M.1.4	GL


## -Les fichiers « .c » :


 Compte.c

 Gestioncmp.c

 menu.c

 donnees.c

 main.c

 Surveillance.c

**-Le fichier « main.c » contient le main du programme, et les fonctions qui vérifient le login, le mot de passe et les droits de l'utilisateur.**

```
char *Nom_ens(_Compte *Surv, int n, char login[90],char mdp[90]);
```

```
//Retourne le nom du professeur à partir de son login
```

```
int Verifier_admin(_Compte *Surv, int n, char login[90]);
```

```
//Vérifie si c'est un compte d'administrateur ou non
```

**-Le fichier « menu.c » contient le menu de l'utilisateur et celui de l'administrateur.**

```
void ft_admin(_Compte *Compte, _Surveillance *SURV, _Enseignant *Ens,char lg[90],char nom[90],int *N, int *C,int E);
```

//Cette fonction permet d'afficher le menu de l'administrateur, demande à l'utilisateur d'entrer l'option qu'il veut et exécute la fonction qui correspond à chaque option.

```
int ft_surveillant(_Surveillance *SURV, _Compte *Compte, _Enseignant *Ens ,char lg[90],char nom[90],int *N,int *C,int E);
```

//Cette fonction permet d'afficher le menu de l'enseignant, demande à l'utilisateur d'entrer l'option qu'il veut et exécute la fonction qui correspond à chaque option.

**-Le fichier « Compte.c » contient les fonctions qui se trouvent dans le menu de l'utilisateur.**

```
void cp_modifier_mdp(_Compte *Surv, int n,char login[90]);
```

```
//Modifie le mot de passe
```

```
void cp_modifier_login(_Compte *Surv, int n,char login[90]);
```

```
//Modifie le login
```

```
void cp_modifier_email(_Compte *Surv,_Enseignant *ens, int n,int m,char login[90]);
```

```
//Modifie l'email
```

```
int Afficher_sur(_Surveillance *Surv, int n, char num[100]);
```

```
//Affiche les surveillances de l'enseignant
```

```
int Afficher_cp(_Compte *Surv,_Enseignant *ens,int n,int m,char login[90]) ;
```

```
//Affiche les informations du compte
```

```
int Enregister_ens(_Enseignant *Ens,int n);
```

```
// Enregistre les modifications des informations dans le fichier.
```

**-Le fichier « Surveillance.c » contient les fonctions de gestion des surveillances.**

```
int Gestion_surv(_Surveillance *Surv,int *N);
```

```
// Affiche le menu de gestion des surveillances
```

```
int Enregister(_Surveillance *Surv,int n);
```

```
// Enregistre les modifications dans les surveillances dans le fichier
```

```
void Afficher(_Surveillance *Surv,int n);
```

```
//Affiche toutes les surveillances
```

```
void Supprimer(_Surveillance *Surv, int *n) ;
```

```
// Supprime une surveillance en entrant toutes ces informations
```

```
void Modifier(_Surveillance *Surv, int *n);
```

```
//Modifie une surveillance
```

```
void Inserer(_Surveillance *Surv, int *n);
```

```
//Insère une surveillance
```

```
int existe(_Surveillance *Surv, int n, char salle[3],char date[11],char duree[11],char enseignant[100]) ;
```

```
// Retourne 1 si le surveillant existe et 0 sinon.
```

**-Le fichier « Gestioncmp.c » contient les fonctions de gestion des comptes des surveillants.**

```
int Enregister_cmp(_Compte *Cmp,int n);
```

```
//Enregistre les modifications des informations des comptes dans le fichier
```

```
void Gestion_cmp(_Compte *Cmp,_Enseignant *Ens,int *N,int E,char login[99]);
```

```
//Affiche le menu de la gestion des comptes des enseignants
```

```
void Supprimer_cmp(_Compte *Surv, int *n, char login[90]);
```

```
//Supprimer le compte qui correspond au login que l'admin a entré
```

```
int Modifier_cmp(_Compte *Surv, int n,char login[90]);
```

```
//Modifie le compte qui correspond au login que l'admin a entré
```

```
void Afficher_cmp(_Compte *Surv,_Enseignant *ens,int n,int m);
```

```
// Affiche le compte qui correspond au login que l'admin a entré
```

```
void Afficher_ens(_Enseignant *Surv,int n) ;
```

```
//Affiche la liste des enseignants avec leurs infos
```

```
void Creer_cmp(_Compte *Cmp,_Enseignant *Ens, int *n,int E);
```

```
//Crée un nouveau compte
```

```
int cmp_existe(_Compte *Cmp, int n,char login[95],char enseignant[100]);
```

```
//Retourne 0 si le compte existe déjà 1 sinon
```

```
int lg_existe(_Compte *Cmp, int n,char login[95]);
```

```
// retourne 0 si un compte avec ce login existe et 1 sinon
```

```
int ens_existe(_Enseignant *Ens,char enseignant[100],int n);
```

// retourne 0 si l'enseignant possède déjà un compte et 1 sinon

**-Le fichier « donnees.c » contient les fonctions qui manipulent les fichiers textes, et enregistrent les données dans les tableaux qui convient.**

int checkfile(FILE \*file, int lines, int tab);

// Retourne 1 si le fichier est valide 0 sinon

int countLines(FILE \*file) ;

//Retourne le nombre de lignes dans un fichier text

\_Surveillance \*surv(int \*N);

//Retourne un tableau de pointeurs sur des éléments de type « surveillance » à partir d'un fichier « SURVEILLANCE.txt »

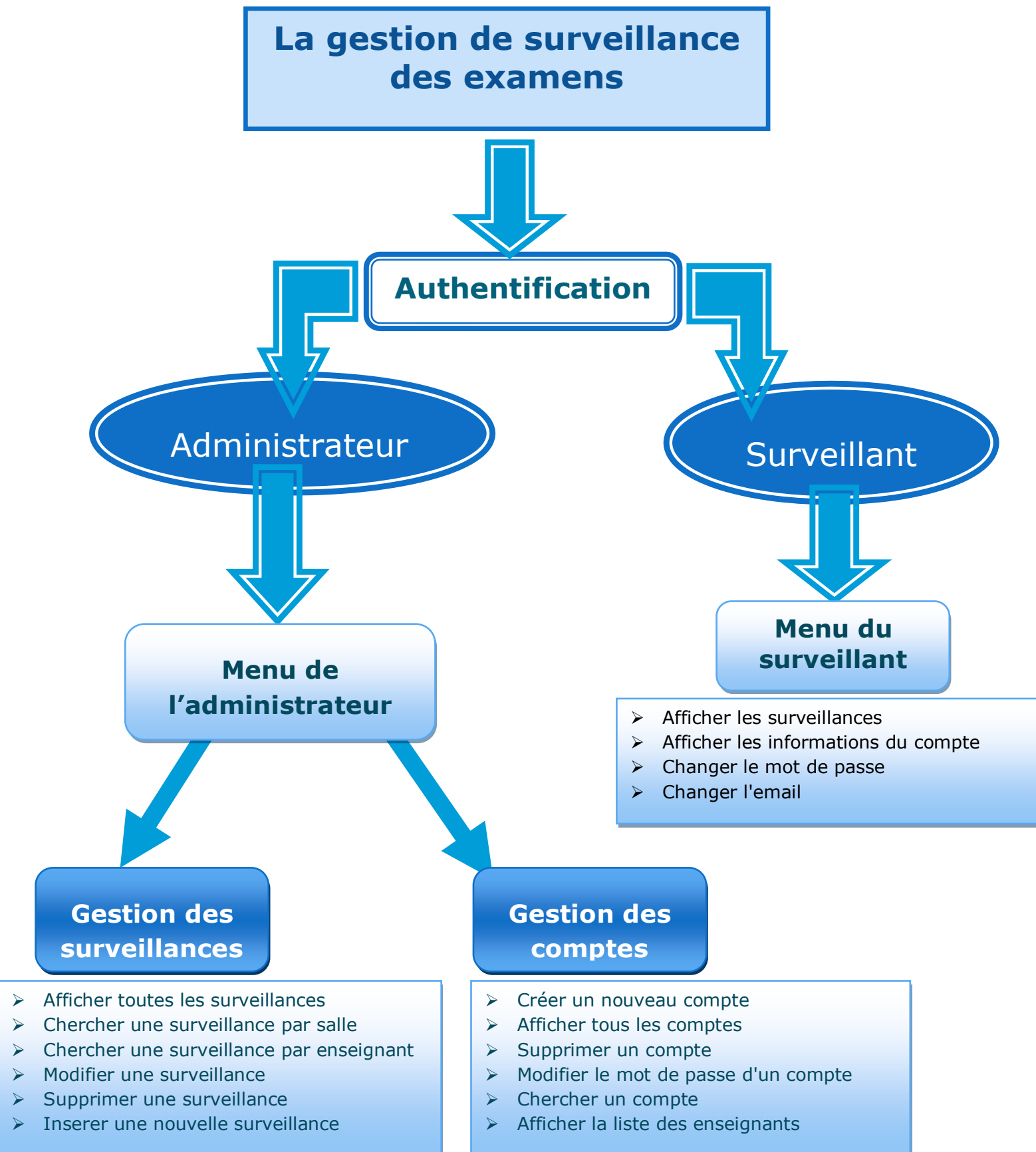
\_Enseignant \*ens(int \*N);

//Retourne un tableau de pointeurs sur des éléments de type « enseignant » à partir du fichier « ENSEIGNANT.tx »

\_Compte \*cmp(int \*N);

//Retourne un tableau de pointeurs sur des éléments de type « compte » à partir du fichier « COMPTE.txt »

Alors, qu'on peut schématiser le programme par le schéma suivant :



# ***CHAPITRE II:***

# **MISE EN**

# **ŒUVRE ET**

# **REALISATION**



## **1. Introduction**

Dans ce chapitre, nous présentons la partie réalisation et mise en œuvre de notre travail. Pour cela, nous présentons, en premier lieu, l'environnement de travail et les outils de développement utilisés. En second lieu, nous élaborons une présentation du programme avec les deux modes utilisateur et administrateur.

## **2. Les outils et techniques de développement :**

### **Code::Blocks :**

Code::Blocks est un environnement de développement intégré libre et multiplateforme.

Il est écrit en C++ et utilise la bibliothèque wxWidgets. Code::Blocks est orienté C et C++, mais il supporte d'autres langages.



Pour la réalisation du travail, nous avons utilisé le langage C dans l'IDE Code::Blocks,

## **3. Scenario d'exécution**

Nous présentons notre application à travers les divers imprimés écrans réalisés en deux côtés, utilisateur (surveillant) et administrateur.

Pour tester le programme, on a pris un exemple « RACHID AHMED » comme surveillant (enseignant).

Les autres données sont ceux des enseignants de l'école avec leurs vraies informations.

Pour les surveillances, on a pris comme exemple le planning des surveillances du premier semestre de la première année.

Et « ADMINISTRATEUR » comme nom de l'administrateur du programme.

### 3.1 Mode Utilisateur (surveillant):

Pour que le surveillant puisse accéder au menu , il doit d'abord passer par l'étape de l'authentification :

```
login :  professeur1
Mot de passe :  *****
```

Si les informations sont correctes, un message de bienvenue s'affiche à l'écran:

```
Bienvenue professeur  AHMED
```

Suivi du menu du surveillant :

```
===== MENU DU SURVEILLANT =====

  1. Afficher les surveillances
  2. Afficher les informations du compte
  3. Changer le mot de passe
  4. Changer l'email
  5. Quitter

Choisir une option (1->6)
```

L'utilisateur pourra donc :

Afficher les surveillances

```
Choisir une option (1->6)      1
A7 03/01/2019 09:00-11:00 M.1.5 e-MBI
LC2 04/01/2019 14:30-16:00 M.1.3 ISEM
LC3 05/01/2019 09:00-11:00 M.1.6 SSE
A9 09/01/2019 14:30-16:00 M.1.4 GL
Tapez entrer pour continuer ...
```

Afficher les informations du compte

```
Choisir une option (1->6)      2
login : rachid2019
Mot de passe : 123
Nom: RACHID
Prenom: AHMED
Filiere : Genie Logiciel
Email: ahmed_rachid@hotmail.fr
Tapez entrer pour continuer ...
```

Changer le mot de passe

```
Choisir une option (1->6)      3
Entrez le nouveau mot de passe "nouveau mot de passe "
```

Changer l'email

```
Choisir une option (1->6)      4
Entrez la nouvelle adresse mail "nouvelle adresse"
```

Ou quitter le programme

### **3.2 Mode administrateur:**

Tout comme l'utilisateur, l'administrateur doit d'abord passer par l'étape de l'authentification pour accéder au programme

```
login : admin
Mot de passe : admin
```

Un message de bienvenue s'affiche suivi du menu de l'administrateur :

```
Bienvenue

===== MENU ADMINISTRATEUR =====

    1. Gerer la base de donnees des comptes des enseignants
    2. Gerer la base de donnees des surveillances des examens
    3. Quitter

Choisir une option (1->3)      _
```

L'administrateur a donc le choix de gérer la base de données des comptes des enseignants (surveillants) ou celle des surveillances.

Selon son choix, il obtient l'un des deux menu suivants :

```
===== MENU DE LA GESTION DES COMPTES =====  
  
1. Creer un nouveau compte  
2. Afficher tous les comptes  
3. Supprimer un compte  
4. Modifier le mot de passe d'un compte  
5. Chercher un compte  
6. Afficher la liste des enseignants  
7. Revenir au menu principal
```

Choisir une option (1->7)

```
===== MENU DE GESTION DES SURVEILLANCES DES EXAMENS=====  
  
1. Afficher toutes les surveillances  
2. Chercher une surveillance par salle  
3. Chercher une surveillance par enseignant  
4. Modifier une surveillance  
5. Supprimer une surveillance  
6. Insérer une nouvelle surveillance  
7. Revenir au menu principal  
8. Quitter le programme
```

Choisir une option (1->8)

## Conclusion générale :

Notre projet a consisté à la réalisation d'une application de gestion de la surveillance des examens au sein d'un établissement scolaire. Elle est faite de manière simple et explicite de sorte qu'elle soit à la portée de tout le monde.

Ce projet nous a aidé à améliorer nos connaissances en langage C et aussi en ce qui concerne la gestion des fichiers.

La réalisation de ce projet nous a rendu fières de pouvoir aider les organismes à gérer leurs données qui est une étape du développement d'une société qui tourne autour de l'informatisation des données.