

Projet programmation orientée **objet**

SCHOOL1.0

Professeur M.Balbali

**Étudiants :
Wade Pape Daouda
Monteil Jean-Marc
Perron Thomas**

14/04/2018

Table des matières

Introduction :	3
Objectif du projet :	3
Diagramme d'utilisation :	4
Les différents acteurs :	5
Diagramme de Class :	6
Class individu :	6
Class groupe :	7
Class étudiant :	7
Class administrateur :	7
Class directeur :	7
Class enseignant :	7
Class secrétaire :	7
Class classe :	7
Class établissement :	7
Class adresse :	8
Class examen :	8
Class matériel :	8
Class matière :	8
Class salle :	8
Class séance :	8
Diagramme de Séquence :	9
Interfaces Homme-machine :	14
Page d'accueil :	14
Page étudiant :	15
Page directeur :	16
Page enseignant :	17
Page secrétaire :	18
Page administrateur :	19
Conclusion :	22

Introduction :

Ce projet informatique, SCHOOL1.0, porte sur la création d'un logiciel de gestion d'établissements scolaires, il vise à l'organisation d'une base de donnée regroupant les étudiants ainsi que les professionnels de différents établissements, de la gestion des salles, des emplois du temps et des inscriptions de nouveaux étudiants.

De plus, notre logiciel sert également à la gestion des programmes et réunions pédagogiques, aux actions marketing et tout cela de manière instantanée et chaque donnée peut être mise à jour à tout moment.

Pour cela nous allons étudier plusieurs diagrammes :

- Un diagramme d'utilisation pour avoir un rendu global des comportements et fonctionnements de notre logiciel.

- Un diagramme de class pour étudier plus en profondeur les différentes class et fonctions de notre logiciel, permet une plus simple compréhension des class à coder en java.

- Un diagramme de séquence pour pouvoir étudier les liens entre les différents acteurs et le système.

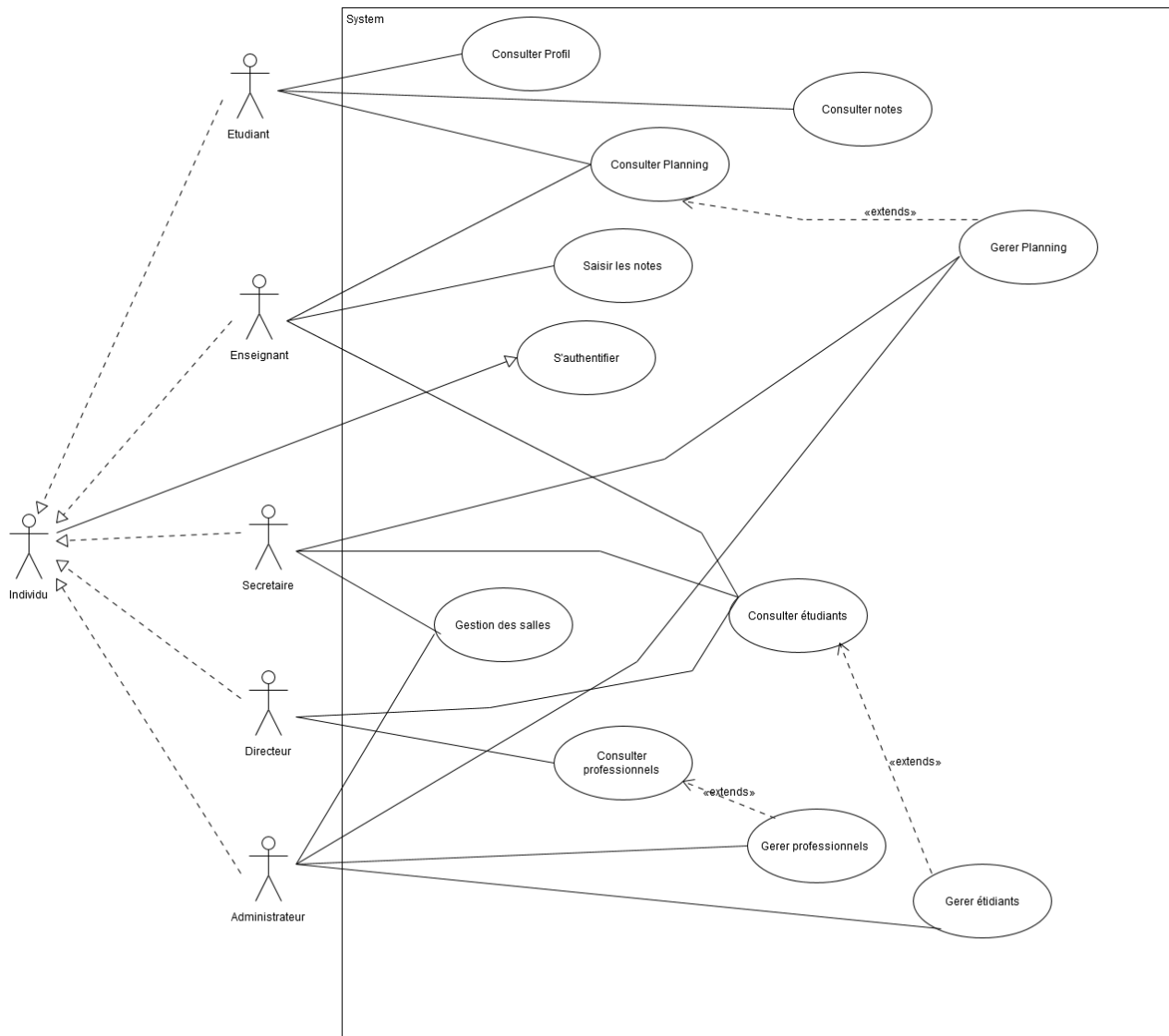
Objectif du projet :

Ce projet à pour objectif d'agrandir nos connaissances en Java, et tout particulièrement le Java orienté objet, en effet ce projet porte essentiellement sur la création d'individus sous forme d'objet ainsi que leur interaction avec le système.

De plus nous allons gérer une base de données et l'interface homme-machine de ce logiciel.

Diagramme d'utilisation :

Nous allons décrire l'interaction entre les acteurs et le système.



Les différents acteurs :

-Étudiant :

Les étudiants n'ont aucun droit de gestion dans notre logiciel, il ne peuvent que s'identifier sur la page de connexion, et ensuite accéder à leur emploi du temps pour y consulter leurs matières, salles, et horaires de travail et connaître leurs notes.

-Enseignant :

Les enseignants une fois identifiés, peuvent consulter leur emploi du temps pour savoir dans quelle salle et à quelle heure ils doivent enseigner leurs cours et aussi savoir à quels étudiants ils enseignent.

-Secrétaire :

Les secrétaires doivent s'identifier pour pouvoir gérer les emplois du temps étudiant et enseignant, c'est à dire ajouter, supprimer ou modifier un créneau horaire. De plus les secrétaires peuvent approvisionner les établissements en matériel.

-Directeur :

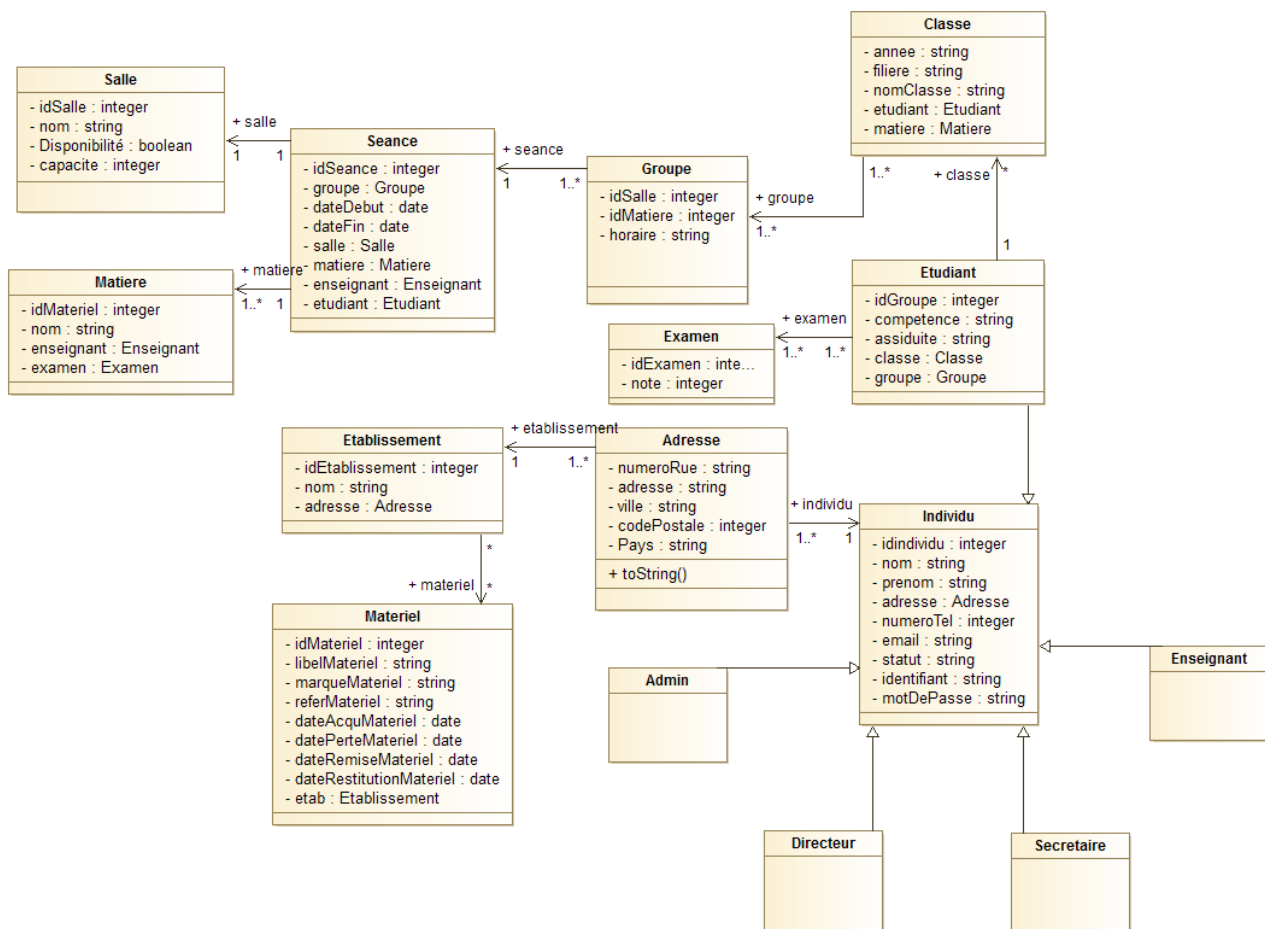
Les directeurs doivent s'identifier pour pouvoir consulter la liste des professionnels de son établissement, c'est à dire la liste des enseignants et secrétaires fréquentant l'établissement.

-Administrateur :

L'administrateur du logiciel une fois identifié, à la possibilité de gérer la base de données des individus et gérer tous les objets, et consulter les fonctionnalités de tous les agents.

Diagramme de Class :

Nous allons étudier les différentes class de notre logiciel.



Toutes nos variables dans notre code sont déclarées en "private" pour que l'utilisateur de l'application ne puisse pas modifier les valeurs.

Class individu :

Cette class est utilisée pour la création des objets "individu", en effet grâce à un constructeur nous créons chaque objet avec des variables associées, tel que son Nom, prénom, adresse, son numéro de téléphone, son email, son statut c'est à dire qu'elle rôle cet individu possède (Administrateur, étudiant, directeur ...), et enfin son identifiant et son mot de passe.

La class individu hérite des class des différents "Statut".

Class groupe :

Cette class est utilisée pour la création des objets "groupe", en effet les groupes sont utiles pour regrouper les étudiants selon une option de matière, attribuer une matière à ce groupe d'étudiants ainsi que l'horaire de la matière et la salle associée.

Class étudiant :

Cette class est une class extends de la class individu. De plus on lui attribue un entier comme Idgroupe ainsi que des String pour ces compétences et son assiduité, enfin il est lié aux objets classe, groupe et matériel.

Class administrateur :

Cette class est une class extends de la class individu.

Class directeur :

Cette class est une class extends de la class individu.

Class enseignant :

Cette class est une class extends de la class individu.

Class secrétaire :

Cette class est une class extends de la class individu.

Class classe :

Cette class sert à gérer une classe d'étudiants , pour cela on utilise un String pour y attribuer son année, sa filière et son nom de classe, de plus on lie à cette class les objets étudiant et matière.

Class établissement :

Cette class sert à gérer l'objet établissement, on lui attribue un entier comme Id d'établissement et un String comme nom et enfin un objet adresse.

Class adresse :

Cette class est utilisée pour l'attribution d'une adresse à un individu, on attribut un numéro de rue, une adresse, une ville, un code postal et un pays, pour cela on utilise une méthode toString() pour gérer l'ensemble de l'adresse de l'individu comme un simple String.

Class examen :

Cette class est utilisée pour la création d'un objet composé de son Idexamen et des notes sous forme d'entier.

Class matériel :

La class matériel sert à la création d'un objet matériel, il est composé d'un entier pour son Idmatériel, d'un String pour son libellé, sa marque et sa référence, de plus il est relié aux objets Date : dateacqumateriel, datepertemateriel, dateremisemateriel et daterestitutionmateriel qui sont respectivement la date d'acquisition du matériel, potentiellement la date de perte du matériel, la date limite de remise et la date de restitution du matériel à l'établissement par l'étudiant et enfin le matériel est lié à son établissement (qui est un objet).

Class matière :

La Class matière est composée de son Idmatière sous forme d'entier, de son nom en String et est associée aux objets enseignant et examen.

Class salle :

Cette class sert à la création de l'objet salle, il est défini grâce à son Idsalle sous forme d'entier, de la capacité de places disponibles (entier), de son nom (String), et de sa disponibilité (Boolean).

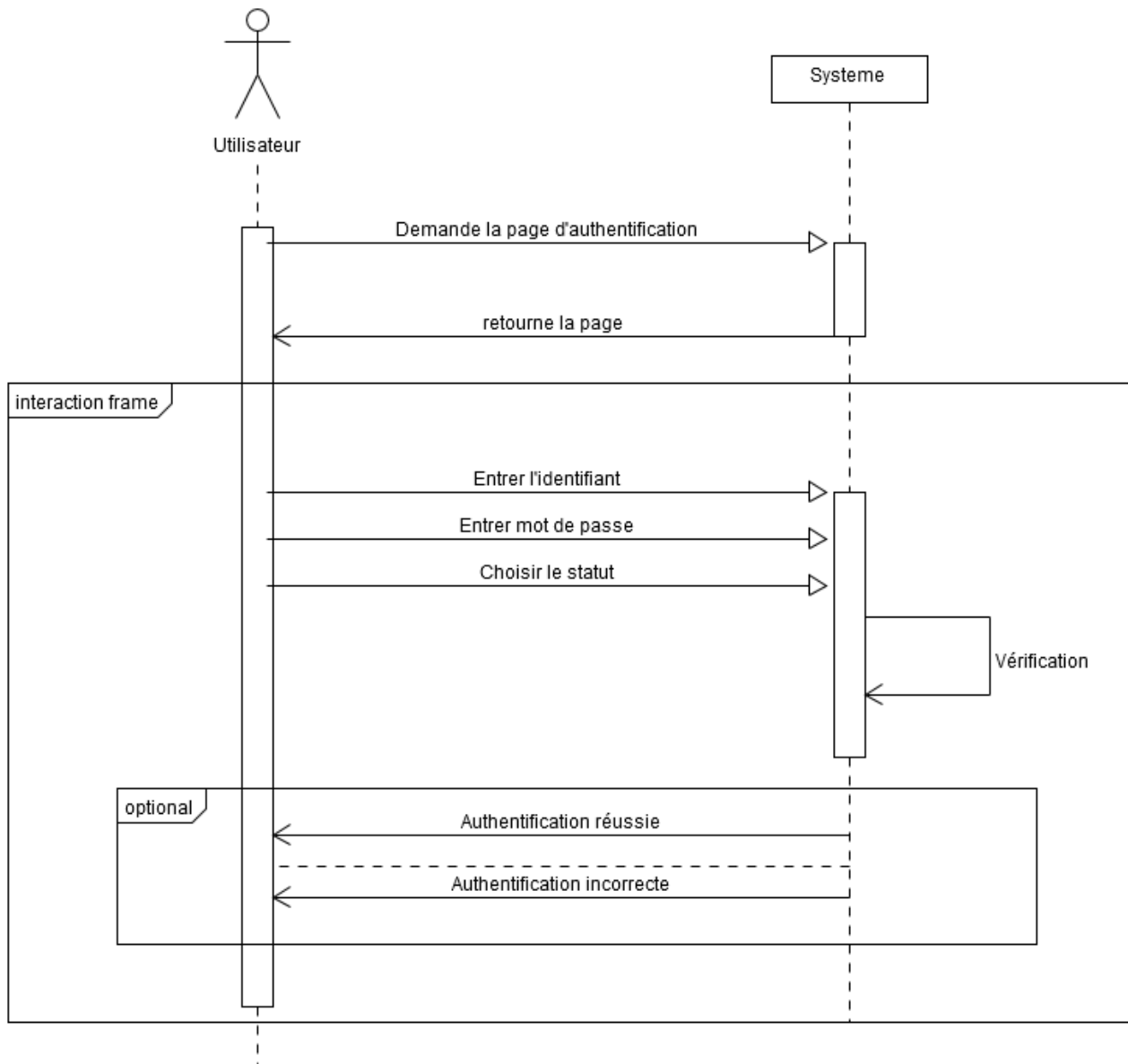
Class séance :

La class séance est composée d'un entier Idseance, et est associée aux objets groupe, datedebut, datefin, salle, matière et enseignant.

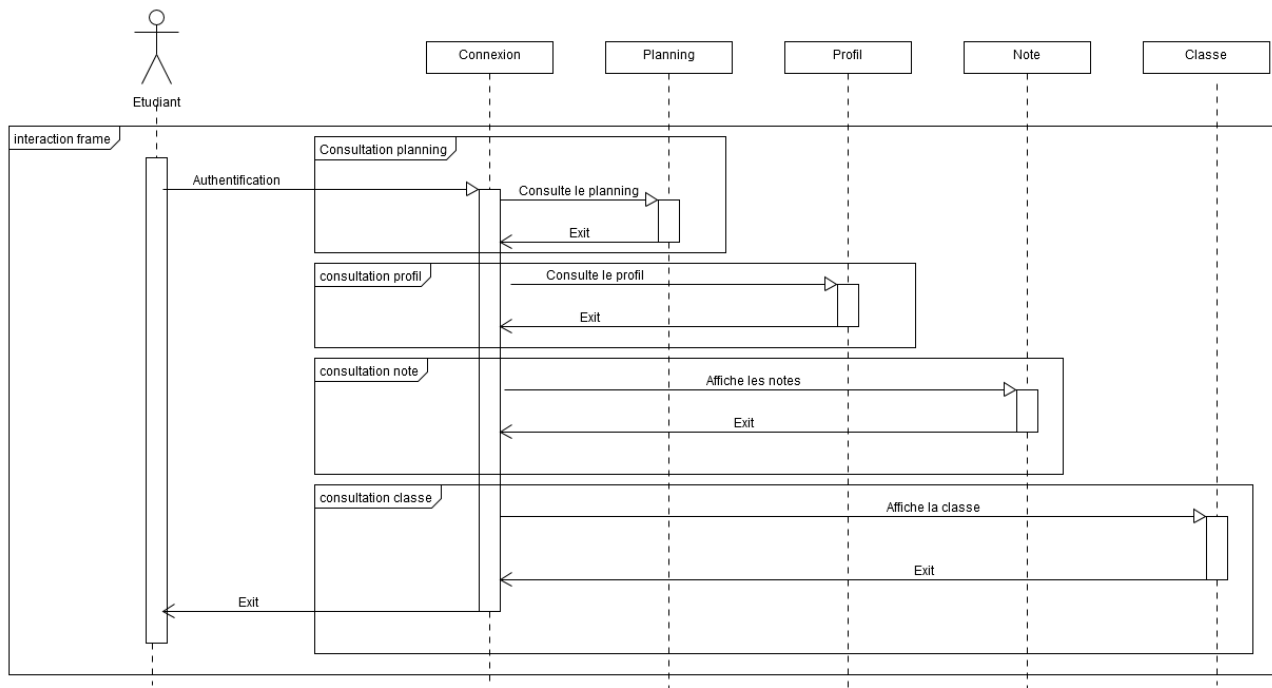
Diagramme de Séquence :

Ce diagramme va nous servir pour voir graphiquement les interactions entre les acteurs et notre logiciel.

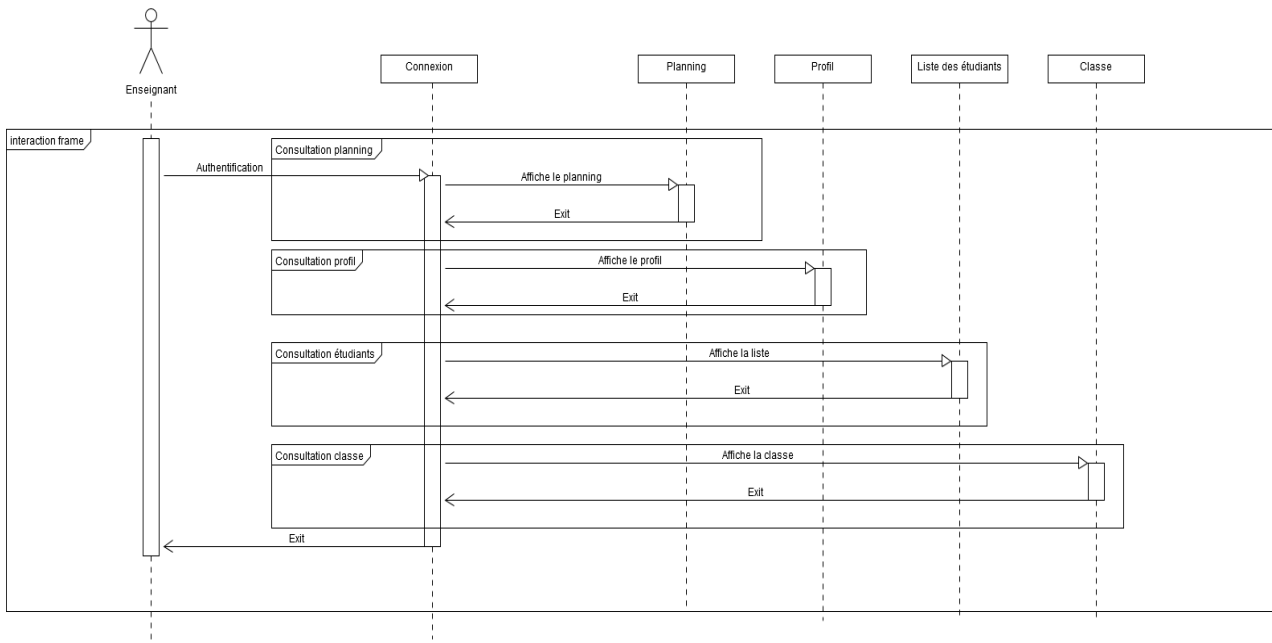
Tout d'abord l'utilisateur doit s'identifier via la page d'accueil, pour cela il doit entrer son identifiant, son mot de passe et choisir statut.



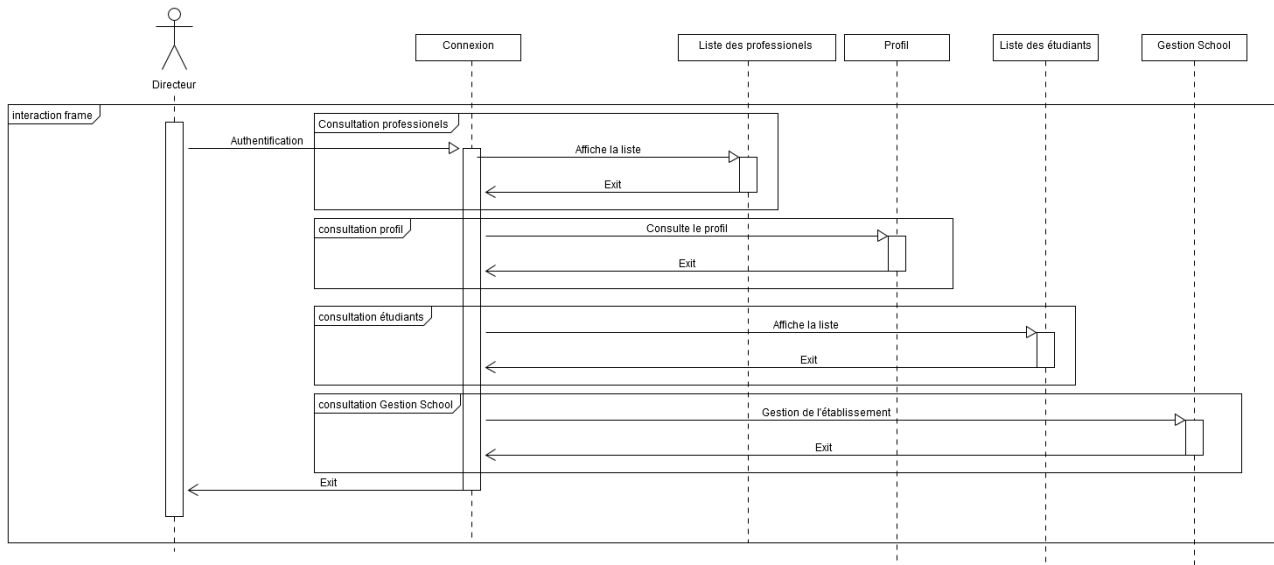
Une fois connecté, l'utilisateur arrive sur la page qui lui est dédiée :
 Pour le statut étudiant, l'utilisateur a quatre choix, il peut consulter son profil, sa classe, ses notes et ses cours.



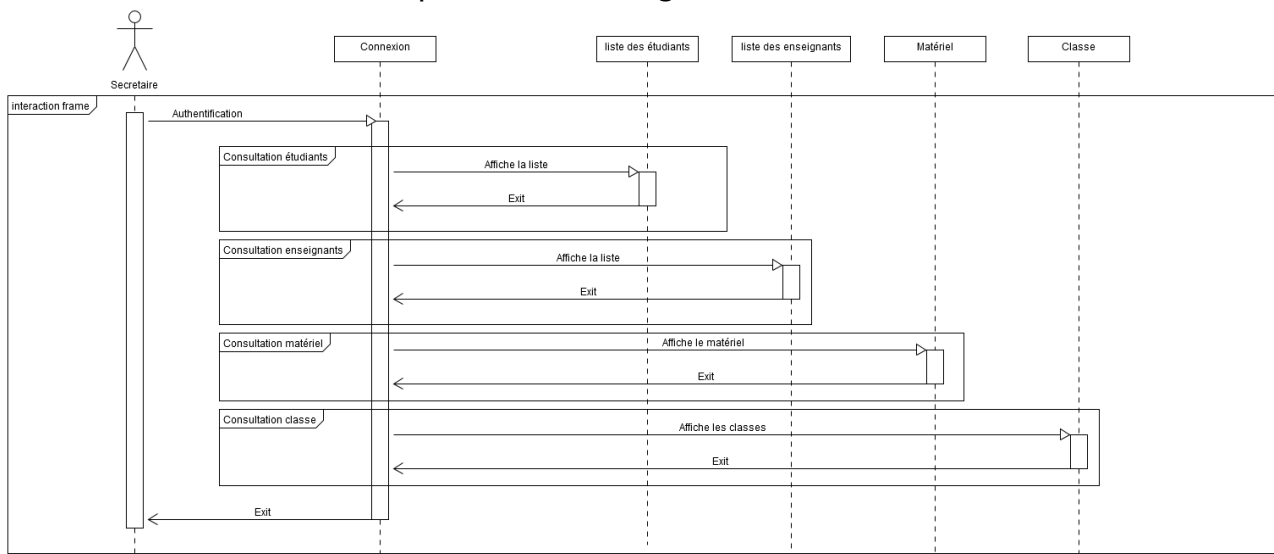
Pour le statut enseignant, l'utilisateur a quatre choix, il peut consulter son profil, ses classes, ces étudiants et ses plannings.



Pour le statut directeur, l'utilisateur a quatre choix, il peut consulter son profil, les étudiants, les professionnels et gérer son établissement.



Pour le statut secrétaire, l'utilisateur a quatre choix, il peut consulter les classes, les étudiants, les professeurs et gérer le matériel.



Pour le statut administrateur, l'utilisateur a cinq choix, il peut gérer les classes, les étudiants, les professeurs, gérer le matériel et gérer les établissements.

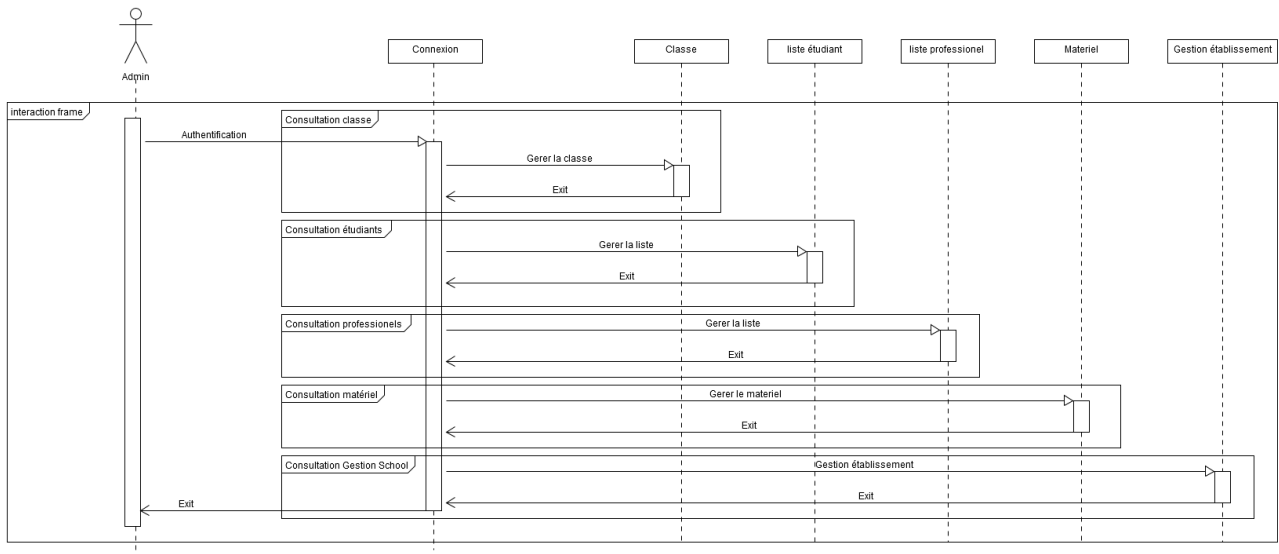


Diagramme de séquence pour inscrire un individu :

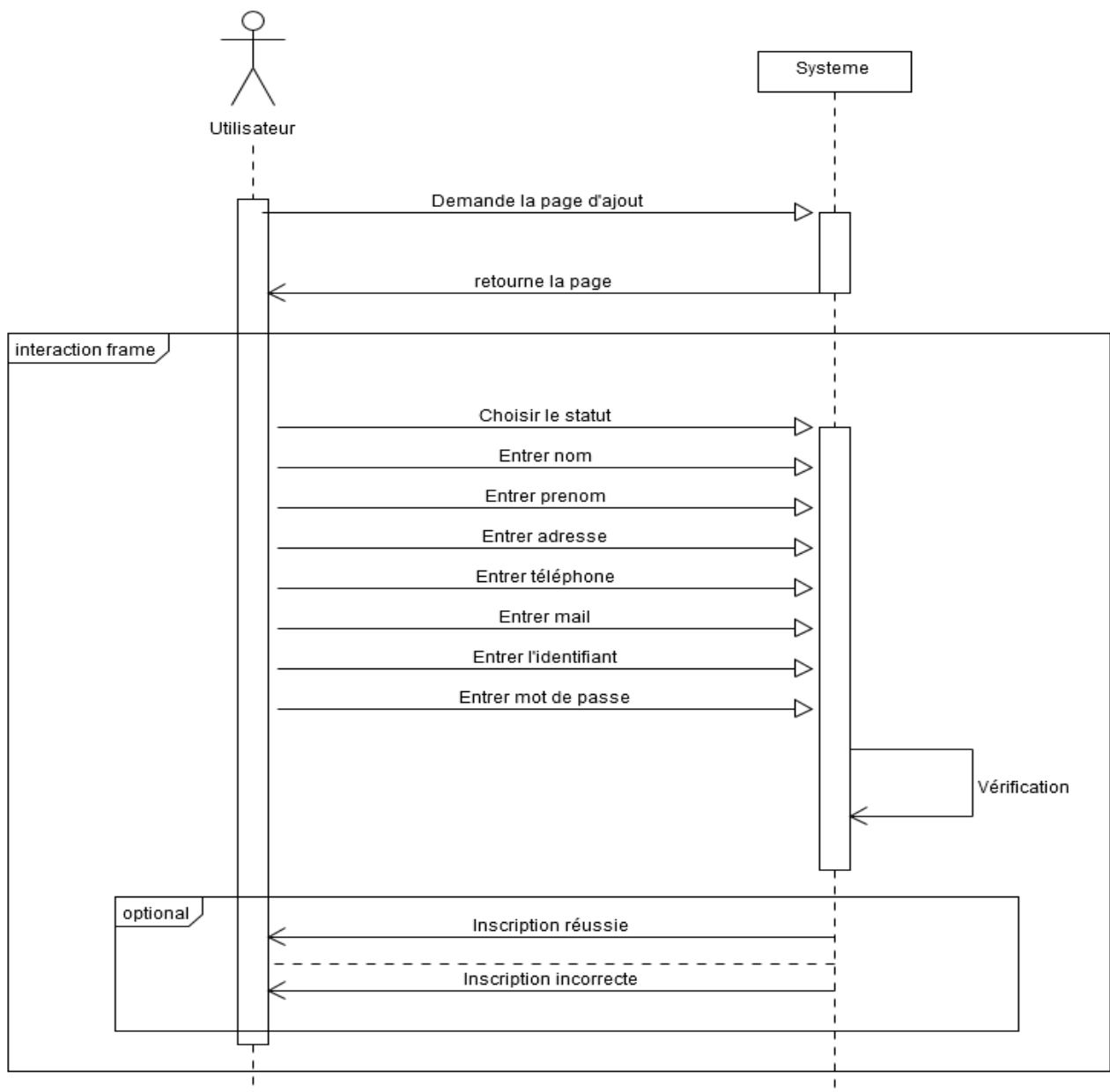
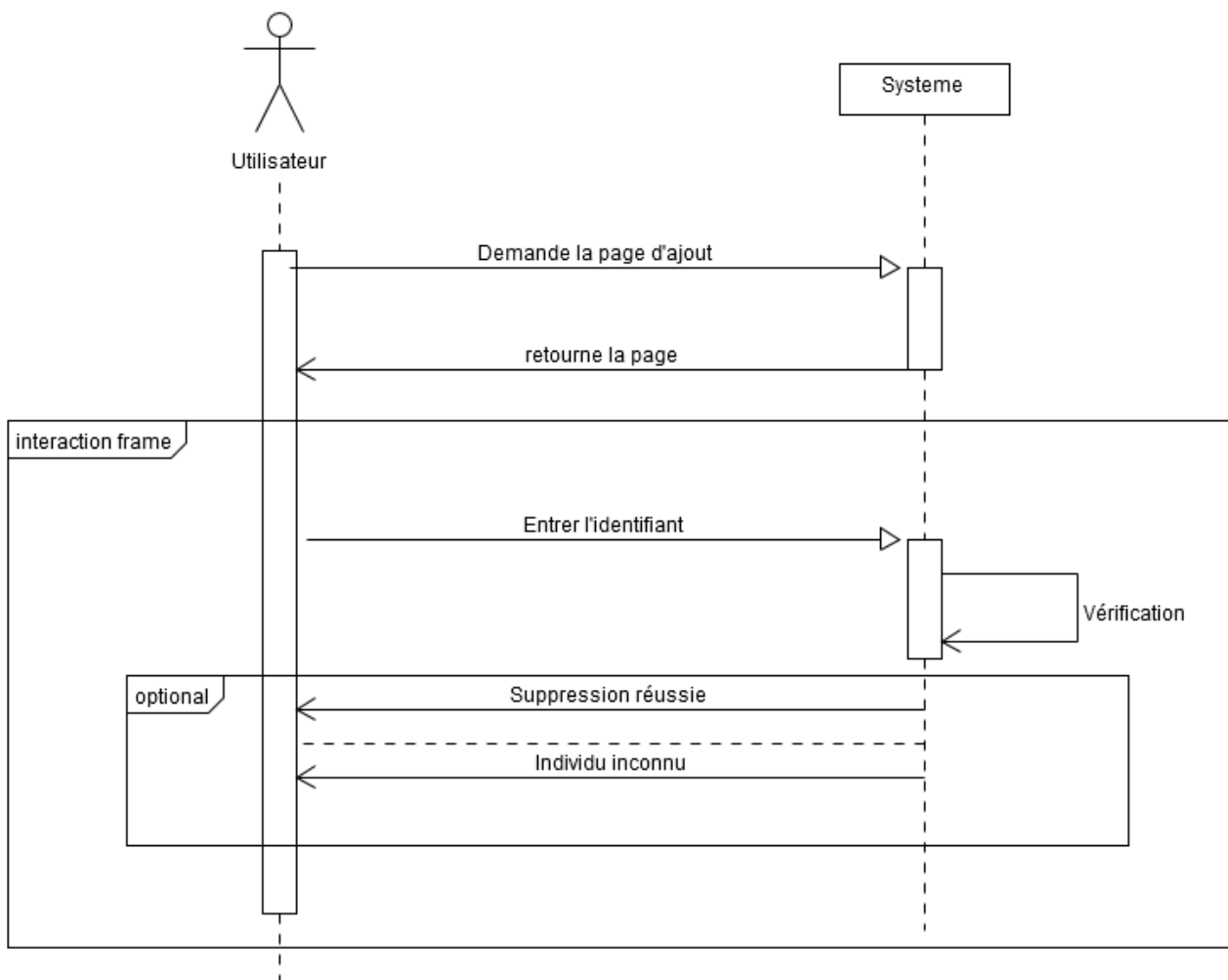
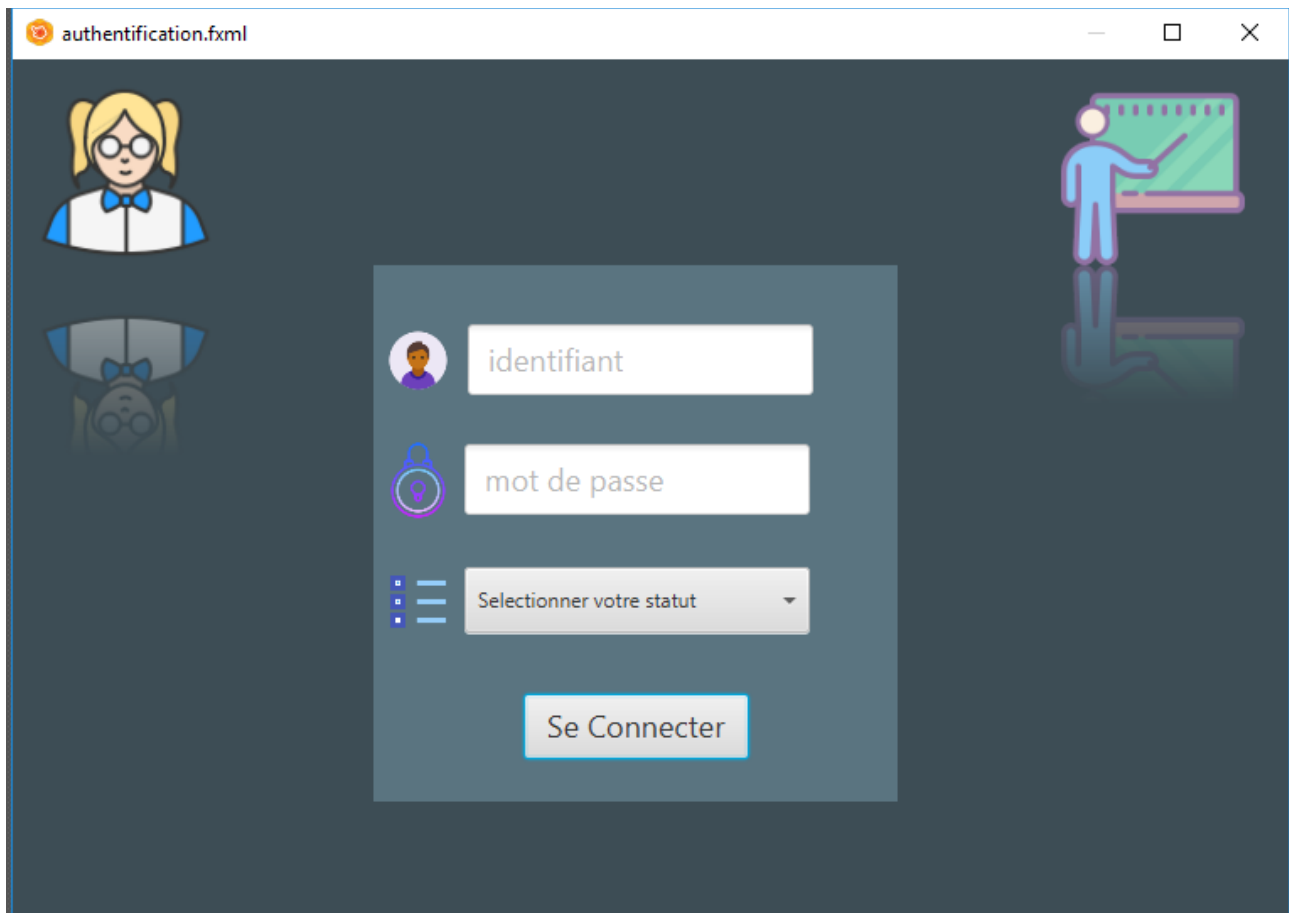


Diagramme de séquence pour supprimer un individu :



Interfaces Homme-machine :

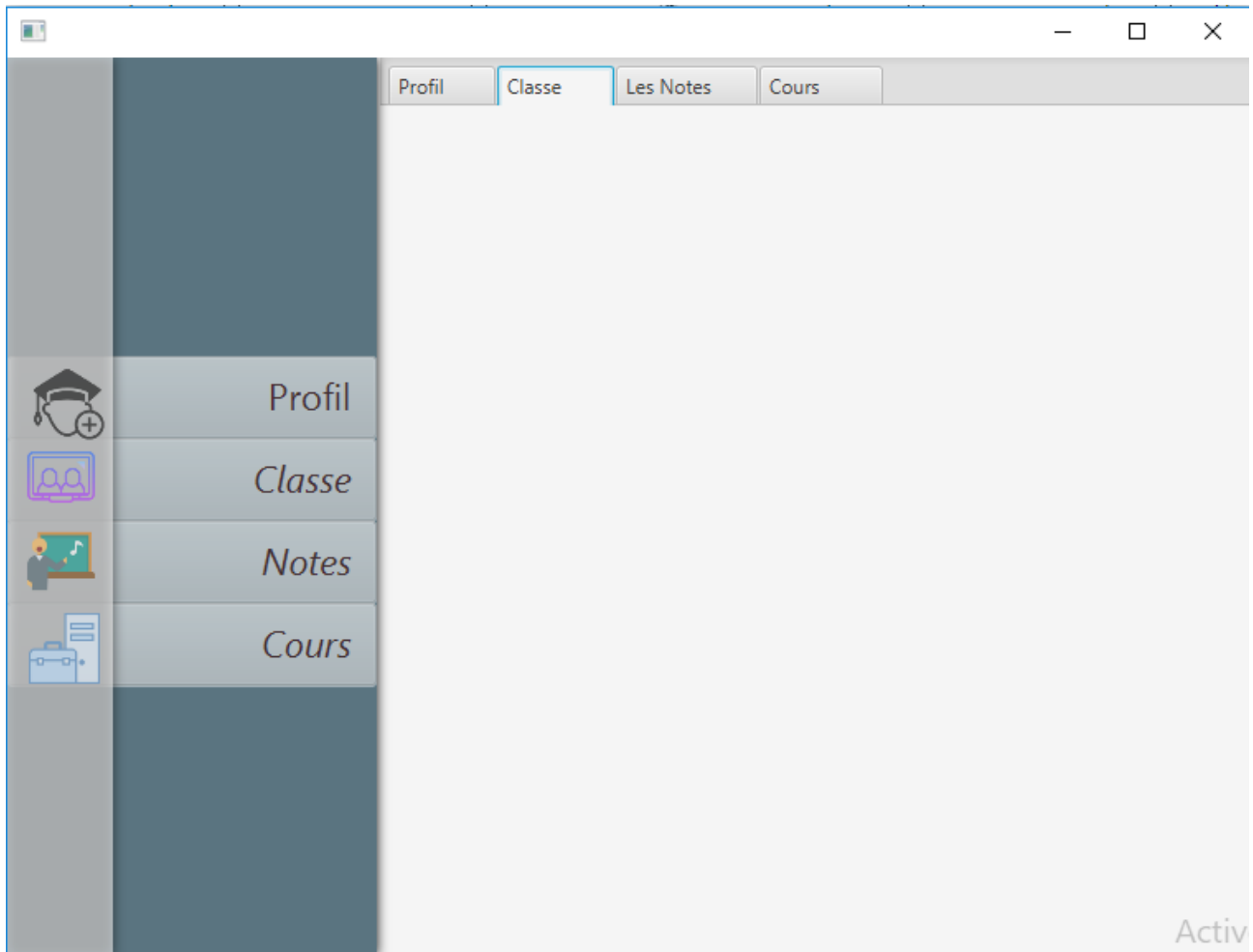
Page d'accueil :



Cette interface a pour but de servir de page d'accueil de notre logiciel, elle sert à l'identification d'un individu, pour cela nous utilisons un identifiant entré via un TextField, un mot de passe sécurisé via un PasswordField, et enfin nous pouvons choisir le statut de l'utilisateur via une ComboBox.

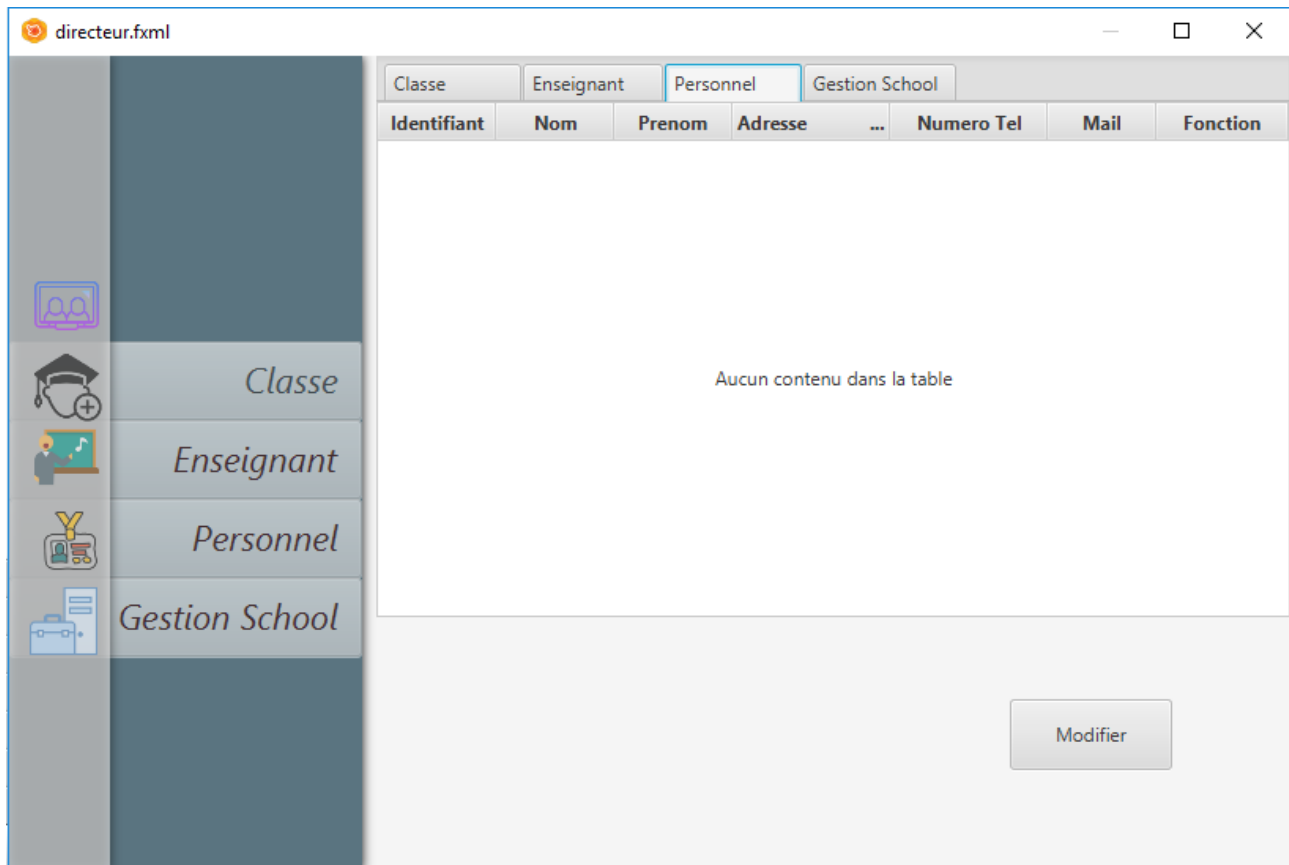
Une fois ces informations remplis, l'utilisateur peut se connecter à son compte et avoir l'interface qui lui est attribué selon son statut.

Page étudiant :



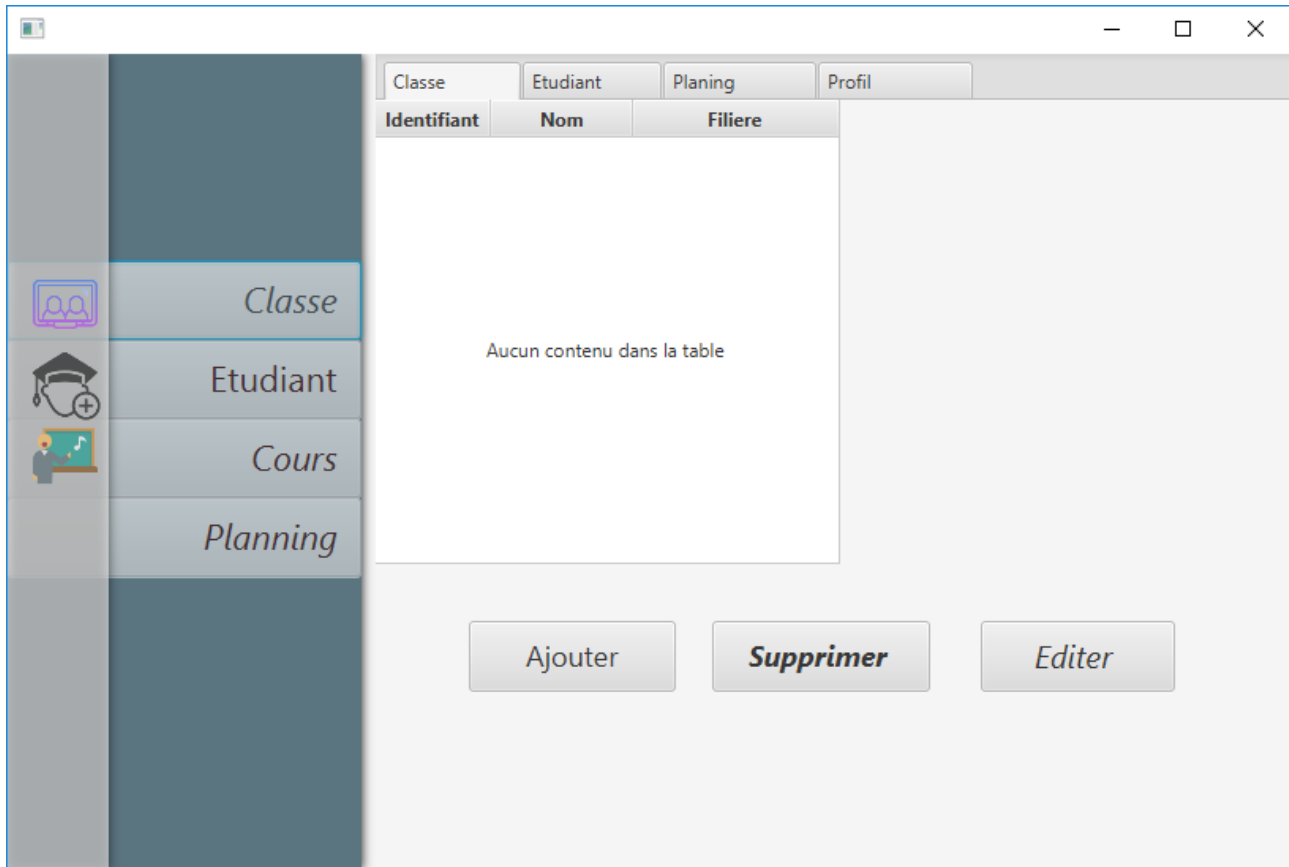
Si l'utilisateur est un étudiant, une fois connecté il arrivera sur cette interface, il arrivera sur son profil étudiant, en changeant de Pane dans un `TablettePane`, il pourra également consulter ces notes, les différents étudiants de son groupe (sa classe d'étudiant), et également consulter son emploi du temps.

Page directeur :



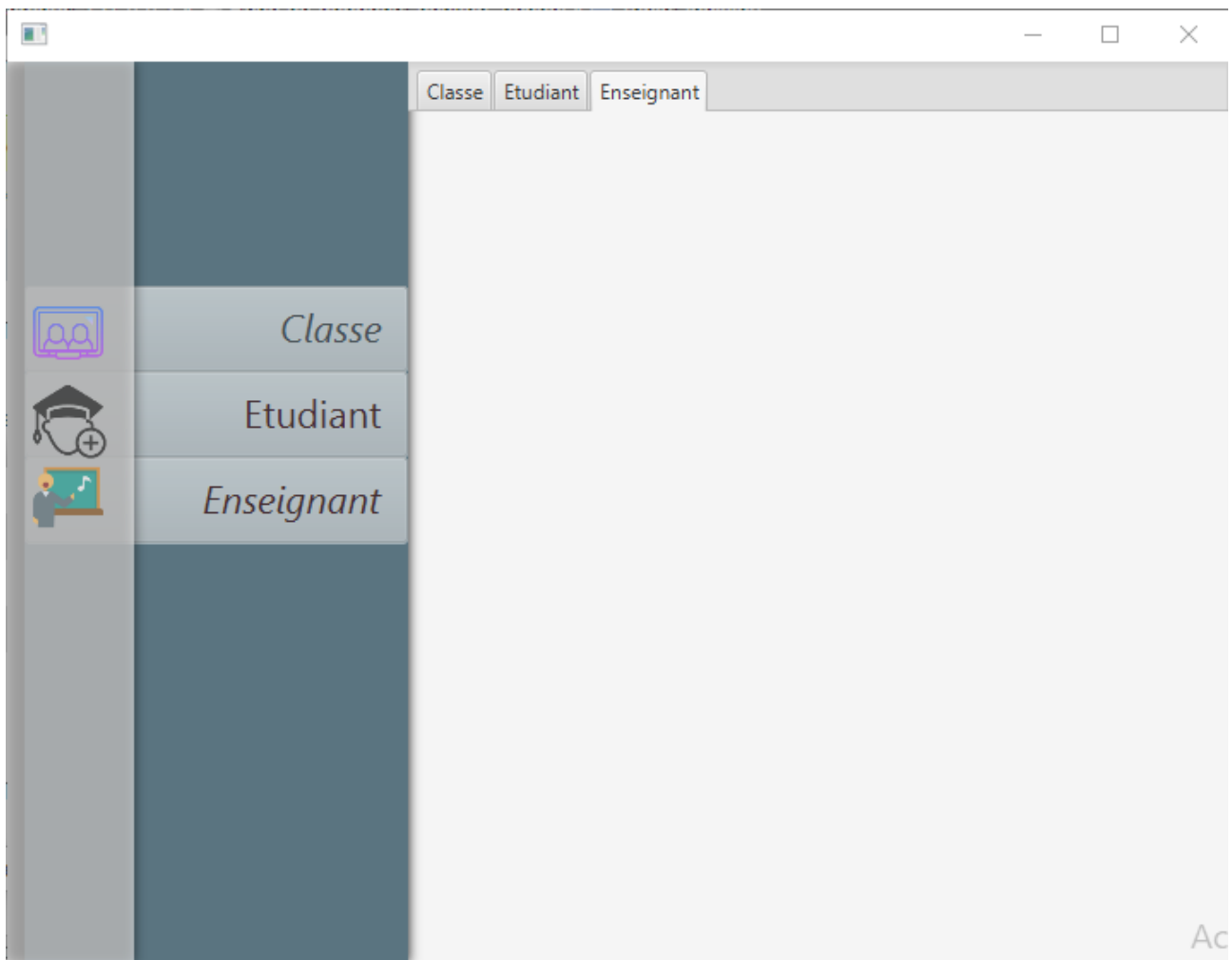
Si l'utilisateur est directeur, il pourra alors obtenir cette interface, celle-ci lui permet de consulter les listes d'individus liées à son établissement, pour cela il pourra utiliser une TableView, accéder aux salles de son établissement et au matériel.

Page enseignant :



Si l'utilisateur est enseignant, il peut via un `TablettePane`, consulter son profil, la liste des étudiants qu'il a en cours, la liste des filières à laquelle il enseigne et son planning de cours.

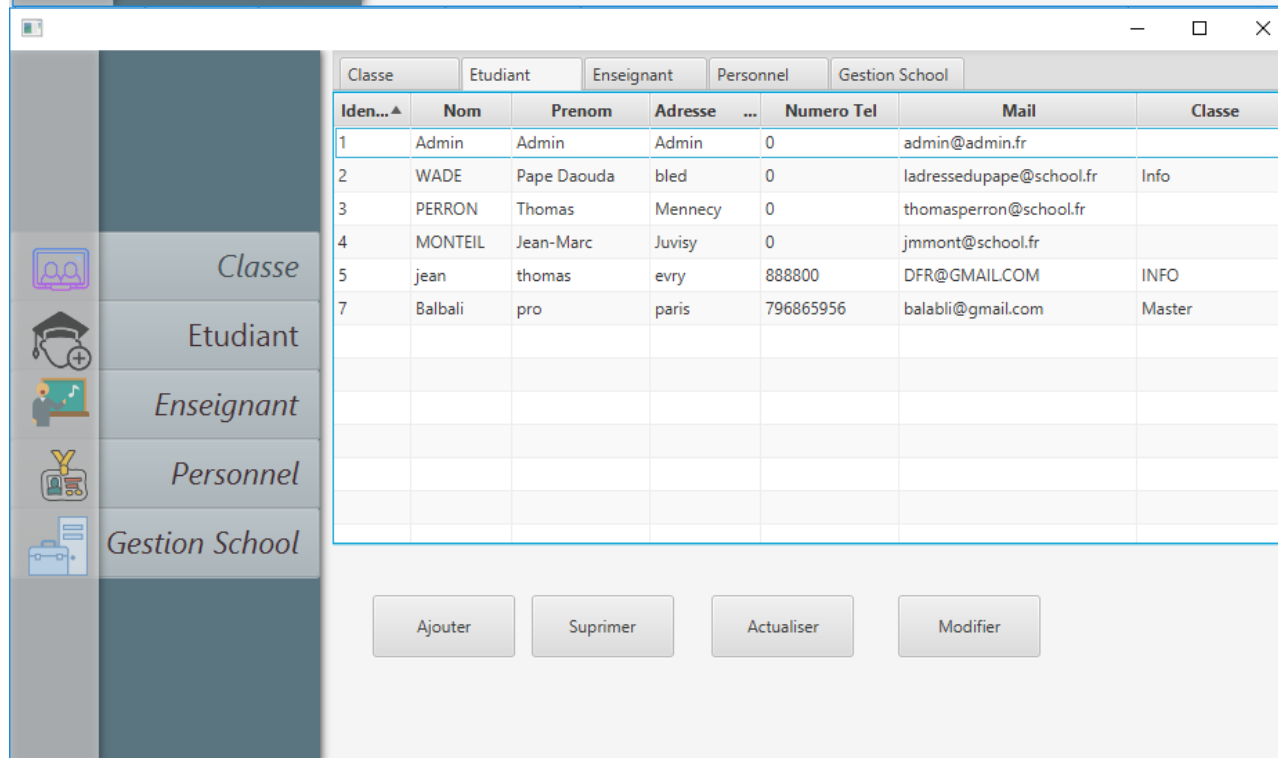
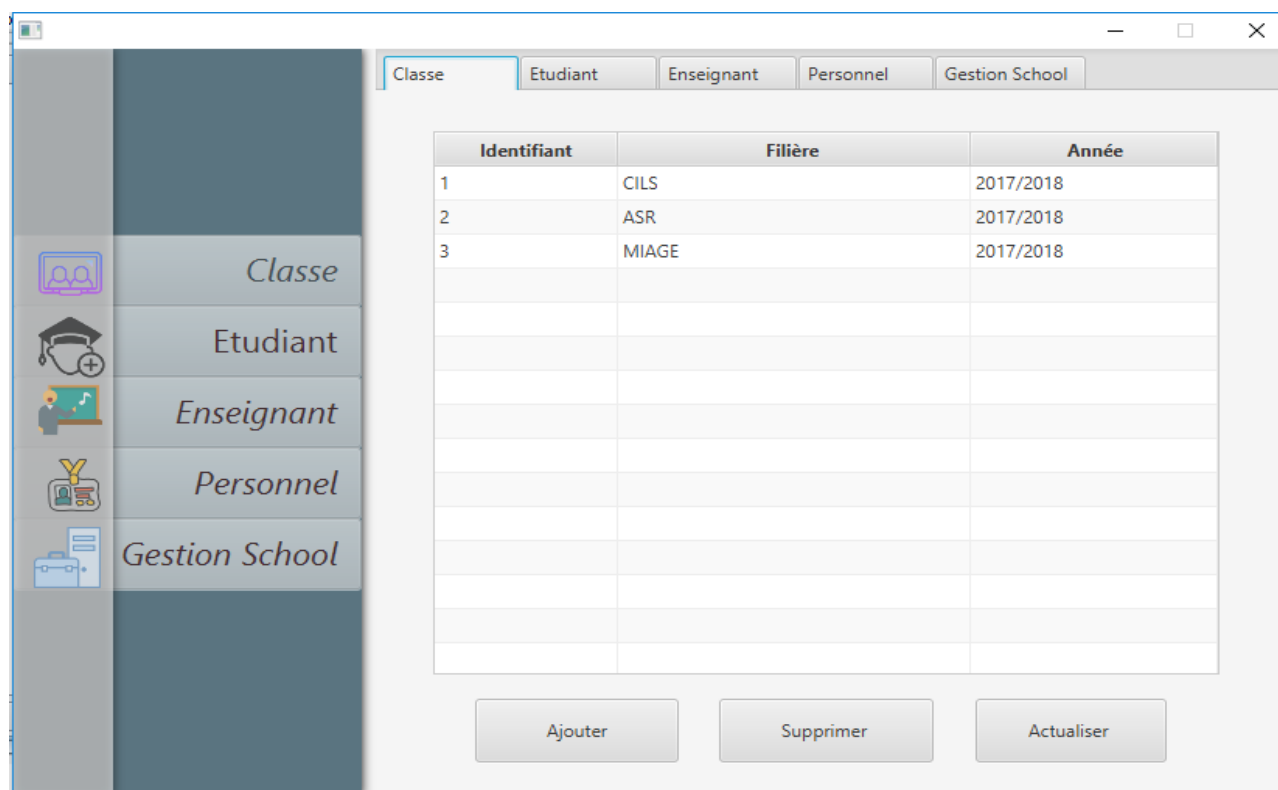
Page secrétaire :



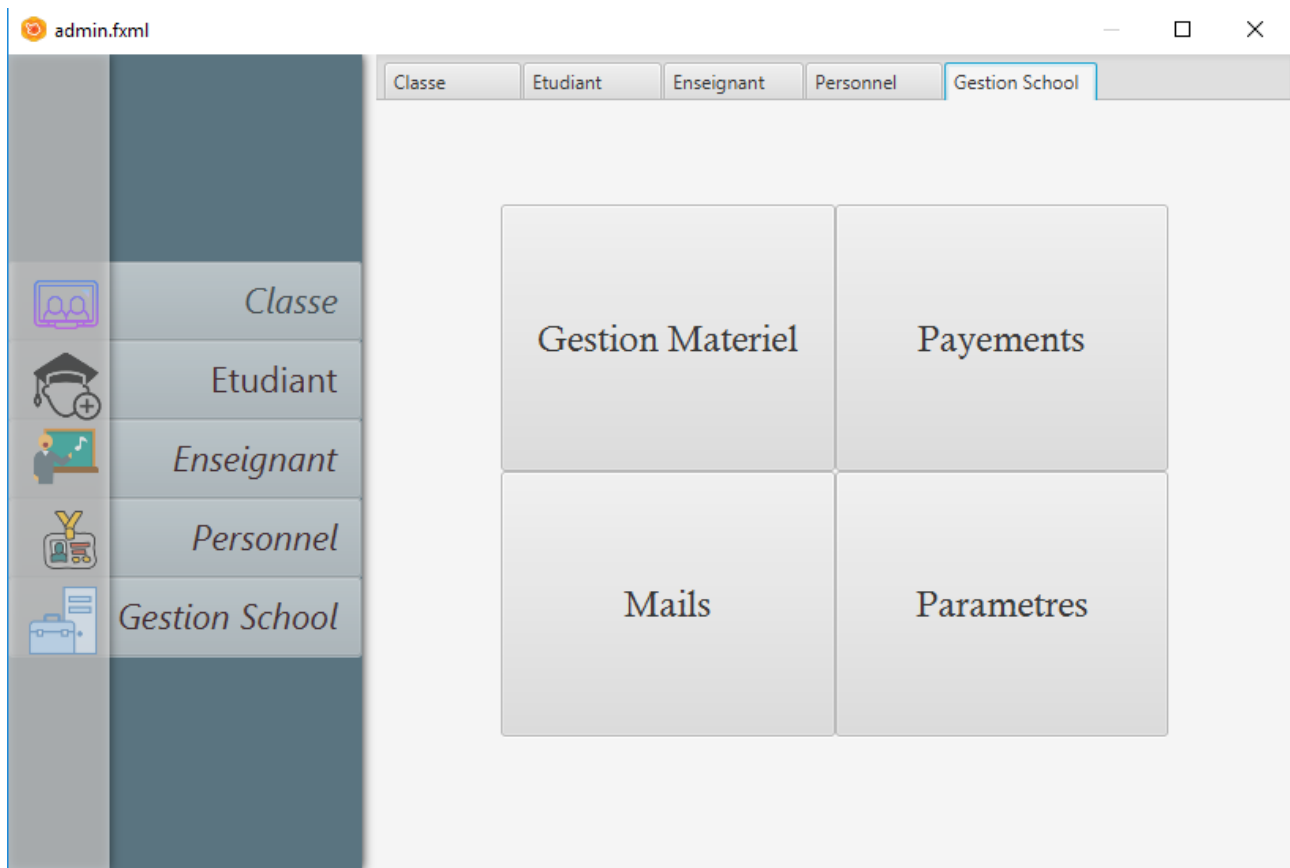
Si l'utilisateur est secrétaire, il a accès à la liste des classes de son établissement, la liste des étudiants associés et enfin la listes des enseignants travaillant dans l'établissement, aussi un accès au matériel disponible au prêt. Le tout est accessible dans plusieurs TableView organisé dans un TablettePane.

Page administrateur :

Si l'utilisateur est administrateur, il a accès via un `TablettePane` à la gestion des classes d'étudiants, des étudiants, des enseignants, du personnel et la gestion des établissements.



Wade Pape Daouda
Monteil Jean-Marc
Perron Thomas



Conclusion :

Au terme de ce projet nous avons appris à nous servir de bases de données grâce au logiciel Wampp, nous servir des interfaces homme-machine avec l'aide du programme JavaFx et approfondir nos connaissances dans le langage Java.

En effet, nous avons réussi à faire interagir notre base de données avec plusieurs interfaces graphiques via le langage Java. Nous avons également pu nous servir de nos connaissances en UML pour élaborer nos différents diagrammes et donc optimiser notre logiciel.

Nous avons également pu améliorer notre cohésion de groupe et avoir une meilleure idée du travail sur de plus gros projet qu'à notre habitude, ainsi que sur l'organisation à avoir sur la gestion du temps sur un projet comportant une date limite proche et enfin comment s'organiser avec des collègues aux horaires et lieux de travail différents.

Projet :

Lien Github : https://github.com/papewade/Projet_School_01

Pour plus de facilité, nous avons utilisé un seul compte Github.