**2SIO**

**Application gestion administrative**

**Présentation du projet**

**SOMMAIRE**

[I. Historique 4](#_Toc161660542)

[II. Présentation du projet 5](#_Toc161660543)

[a) Contexte 5](#_Toc161660544)

[III. Description de l’existant 6](#_Toc161660545)

[a) Base de données 6](#_Toc161660546)

[IV. Application 7](#_Toc161660547)

[V. Diagramme de classe 9](#_Toc161660548)

[VI. DIagramme d’activité 10](#_Toc161660549)

[VII. pRE-REQUIS 13](#_Toc161660550)

[VIII. Organisation 14](#_Toc161660551)

[a) Découpage du projet 14](#_Toc161660552)

[b) Détail des anomalies 14](#_Toc161660553)

[c) Évolutions 14](#_Toc161660554)

[IX. Contraintes 15](#_Toc161660555)

[X. Groupes 16](#_Toc161660556)

[XI. Planning 17](#_Toc161660557)

[XII. rENDU 18](#_Toc161660558)

**TABLE DES FIGURES**

[Figure 1 : Use Case application 5](#_Toc161660600)

[Figure 2 : MCD BDD 6](#_Toc161660601)

[Figure 3 : Description BDD 6](#_Toc161660602)

[Figure 4 : Ecran de connexion 7](#_Toc161660603)

[Figure 5 : Ecran d'accueil 7](#_Toc161660604)

[Figure 6 : Ecran liste des étudiants 8](#_Toc161660605)

[Figure 7 : Ecran ajouter un étudiant 8](#_Toc161660606)

[Figure 8 : Diagramme de classe 9](#_Toc161660607)

[Figure 9 : Diagramme activité "ajout étudiant" 10](#_Toc161660608)

[Figure 10 : Diagramme d'activité "ajout cours" 11](#_Toc161660609)

[Figure 11 : Diagramme d'activité "Consultation d'une section" 12](#_Toc161660610)

[Figure 12 : ajout du package 13](#_Toc161660611)

[Figure 13 : ligne de commande 13](#_Toc161660612)

# Historique

|  |  |  |  |
| --- | --- | --- | --- |
| Identification du Document | | | |
| Client : | Magali GOUDET | **Projet** : | Gestion |
| Rédacteur : | Magali GOUDET | **Professeurs** : | Magali GOUDET  Bastien THOUVEREZ  Christophe REYNARD  Christophe VIRMONT |

|  |  |  |
| --- | --- | --- |
| Historique des Modifications | | |
| 07/03/2023 | V1.0 | Initialisation |
| 15/03/2023 | V1.1 | Mise à jour |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| Documents utiles | |
|  |  |
|  |  |
|  |  |

# Présentation du projet

## Contexte

Le centre de Formation Supérieur en Informatique a fait appel à vous pour faire évoluer et corriger leur application de gestion administratif.

Cette application a pour but de gérer les classes, les étudiants et les cours.

Une première version de l’application existe mais cette dernière contient des bugs et n’est pas fini.

Vous travaillez au sein d’une ESN et votre chef de projet vous a demander de finir l’application et de la rendre opérationnelle.

Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquement

Figure 1 : Use Case application

# Description de l’existant

La première version de l’application est disponible sur un moodle, vous devez donc dans un premier temps vous familiarisez avec l’existant.

## Base de données

La base de données est à réaliser sur PostGreSQL. Ci-dessous la première version de l’application qui va falloir faire évoluer. Les scripts sont disponibles sur le dépôt moodle.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Figure 2 : MCD BDD

|  |  |  |
| --- | --- | --- |
| Nom de la table | Champs | Description |
| ETUDIANT | idEtudiant | Clé primaire de la table étudiant. Ce champ est auto-incrémenté |
| nomEtudiant | Nom de l’étudiant |
| PrenomEtudiant | Prénom de l’étudiant |
| SECTION | idSection | Clé primaire de la table section. Ce champ est auto-incrémenté |
| libelleSection | Libellé de la section. Ce champ permet de savoir de quelle formation on parle |
| COURS | idCours | Clé primaire de la table cours. Ce champ est auto-incrémenté |
| libelleCours | Libellé du cours |
| descriptionCours | Description détaillée du cours. Cela permet de savoir ce qu’il contient, le volume horaire … |
| UTILISATEUR | idUtilisateur | Clé primaire de la table utilisateur. Ce champ est auto-incrémenté |
| loginUtilisateur | Login de connexion à l’application |
| mdpUtilisateur | Mot de passe de connexion à l’application |

Figure 3 : Description BDD

# Application

Voici une description des différents écrans existant :

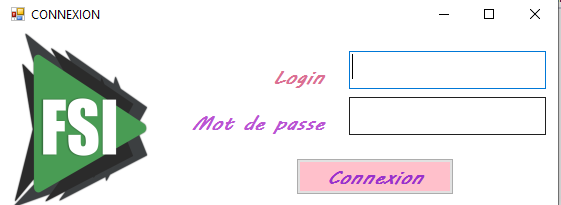


Figure 4 : Ecran de connexion

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Utilisateur connecté

Bouton Quitter

Menu Classe

Menu étudiant

Retour à l’écran d’accueil

Figure 5 : Ecran d'accueil

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

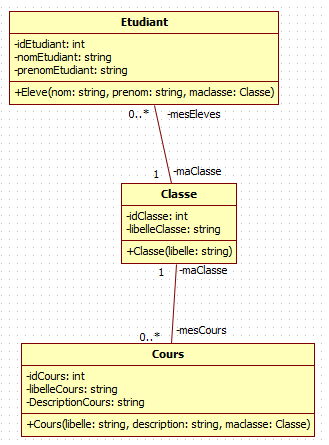
Figure 6 : Ecran liste des étudiants



Figure 7 : Ecran ajouter un étudiant

# Diagramme de classe

Le diagramme de classe (et par conséquence les classes dans l’application) est pour le moment incomplet et comprend deux classes.



Une image contenant texte

Description générée automatiquement

Figure 8 : Diagramme de classe

# DIagramme d’activité

Voici le diagramme d’activité concernant l’ajout d’un étudiant, l’ajout d’un cours et la consultation d’un cours.

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

Figure 9 : Diagramme activité "ajout étudiant"

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

Figure 10 : Diagramme d'activité "ajout cours"

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Figure 11 : Diagramme d'activité "Consultation d'une section"

# pRE-REQUIS

Pour pouvoir se connecter à une BDD PostGreSQL, il faut ajouter la dll Npgsql, voici la marche à suivre :

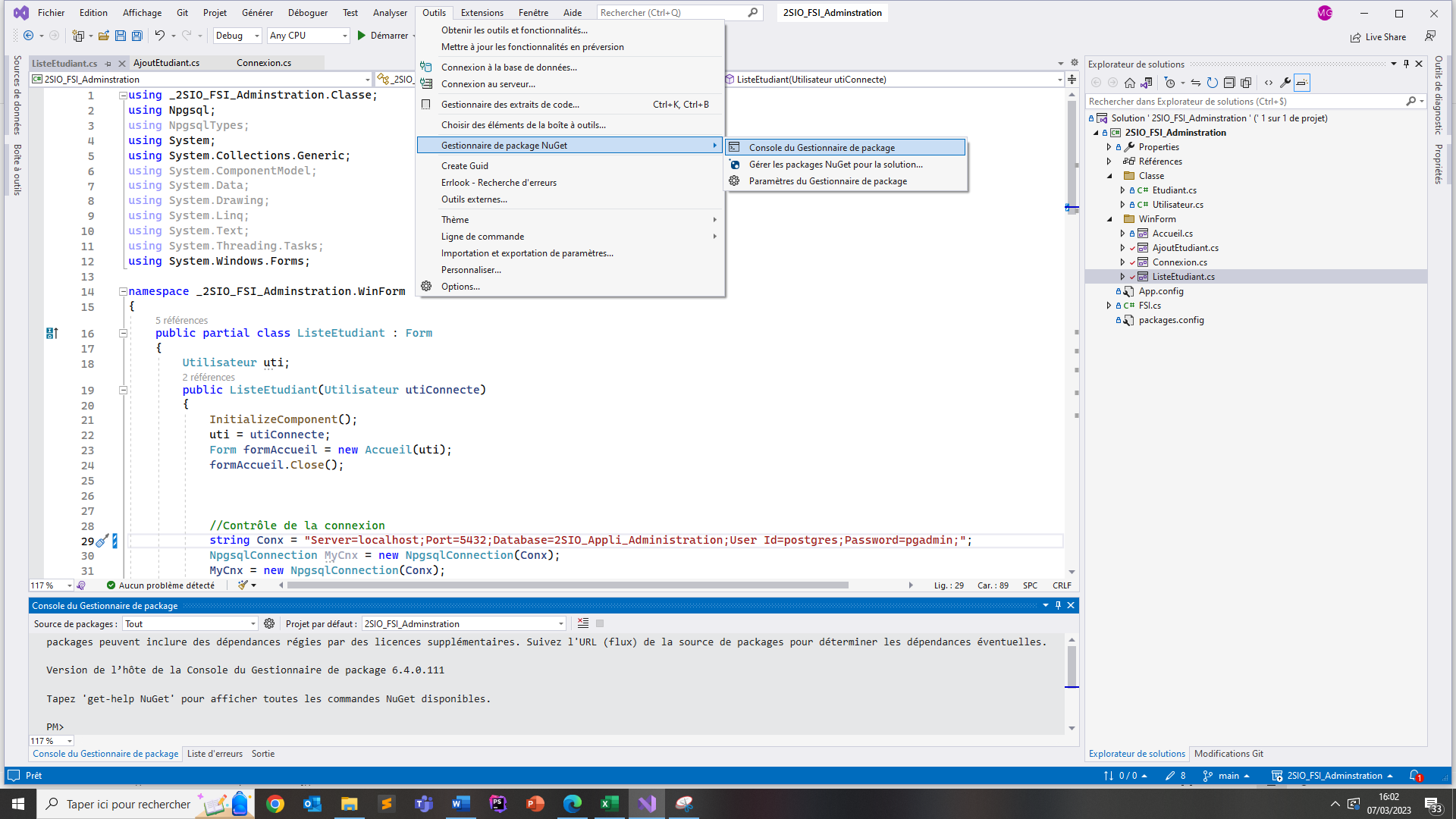


Figure 12 : ajout du package

Install-Package Npgsql -Version 6.0.3

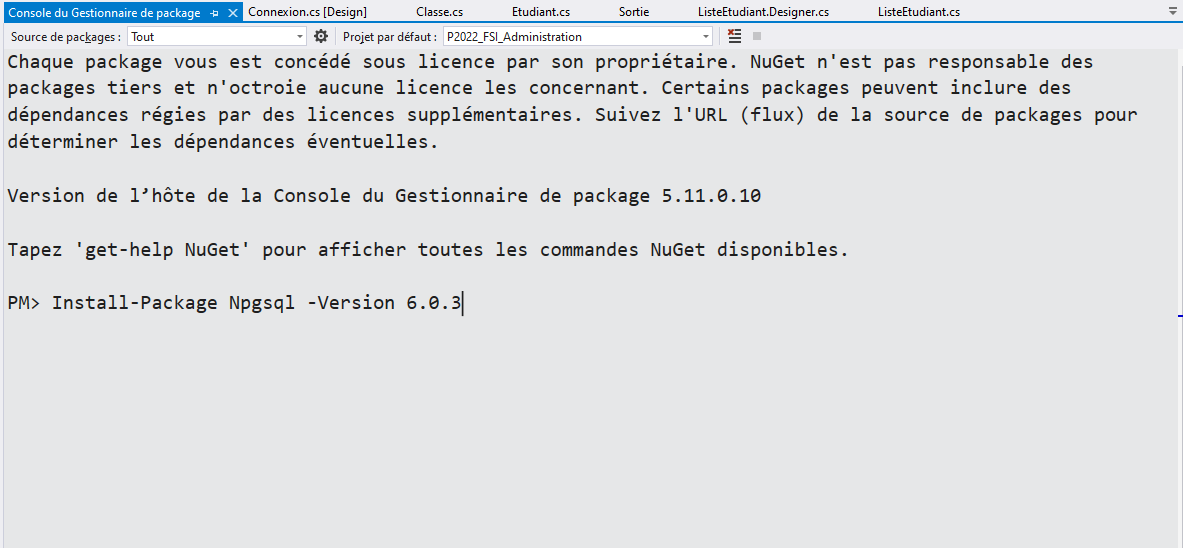


Figure 13 : ligne de commande

De plus pour dans le projet il est utilisé un composant « DataGridView » qui permet de consulter les données d’une table, regardez son fonctionnement.

# Organisation

## Découpage du projet

Le projet se décompose en deux lots :

* Lot 1 : correction des anomalies. Pour le moment l’application permet juste de gérer les étudiants. Dans un premier temps on vous demande de rendre cette partie fonctionnelle et d’éventuellement modifier les écrans pour que ces derniers soient fonctionnels et conforme à la charte graphique. Pour cela dressez une liste sous Excel des anomalies par écran / fonctionnalités. Une fois la liste finie, merci de présenter votre liste au chef de projet (Mme GOUDET) pour validation avant de faire les corrections.
* Lot 2 : finalisation de l’application avec ajout des écrans, fonctionnalités et classes manquantes pour rendre l’application fonctionnelle.

## Détail des anomalies

L’ESN qui a fourni la 1ère version de l’application a laissé plein d’anomalie (en plus des manques). Bien sûr aucun listing des bugs a été fourni. Vous devez donc tester le comportement de l’application et bien noter l’ensemble des anomalies que vous identifiez.

## Évolutions

* Partie Etudiant :
  + Modifier/ Supprimer un étudiant
  + Consulter les données d’un étudiant
  + Ajouter l’adresse de l’étudiant, cette information doit apparaitre sur l’écran consulter les données et non sur le listing
* Partie Section :
  + Modifier/ Supprimer une section
  + Consulter les données d’une section (avec la liste des étudiants inscrit)
* Partie Cours :
  + Ajouter le menu cours
  + Liste des cours
  + Ajouter un cours à une section
  + Modifier / Supprimer un cours
  + Consulter les données d’un cours et voir la Section concernée
* Partie Utilisateur :
  + Ajouter un nom à l’utilisateur en Base De Données pour que ça soit ce champ qui apparaissent sur tous les écrans.
* Commenter l’application et rendre le code lisible
* Faire les optimisations nécessaires
* Un système de log va devoir être mis en place. Ce dernier aura pour but de stocker en BDD les évènements (ajout / modification / suppression) fait sur la table étudiante et cours.

# Contraintes

Les modifications et les évolutions doivent faire évoluer les classes et/ou les tables. Il est **INDISPENSABLE** de bien faire ses modifications.

La programmation orientée objet est OBLIGATOIRE.

L’ensemble des écrans doivent respecter les contraintes de couleur suivantes :

* Vert : 499c54
* Gris : 3c3f41
* Noir : 000000

Des nuances de verts peuvent être ajouté en gardant une cohérence avec le logo

La police de caractère, peut changer mais attention à ce que cela reste professionnelle.

# Groupes

Vous êtes par équipe de 2. Une fois le listing des anomalies fait, le CP affecta les anomalies à chaque membre de l’équipe.

Il en est de même pour les évolutions, lister les tâches du lot, ensuite le CP fera l’affectation des tickets qui devra être **ABSOLUMENT** respecter.

|  |  |  |
| --- | --- | --- |
| Numéro équipe | Membre 1 | Membre 2 |
| N°1 | Mamoune NIDAM | Luca PICHON |
| N°2 | **Anthony MEUNIER** | **Rachel YAZBEK** |
| N°3 | Benjamin BERNS | Romain BAUDOY |
| N°4 | Eddy KEBRITI | Branis KACI |
| N°5 | Hieu TRAN | Loris BACH |
| N°6 | Ilias LARBI | Anthony FROMAGE |
| N°7 | Angel BOISSIE | Jessim MRHIZER |
| N°8 |  |  |

# Planning

Voir fichier 2SIO\_FSI\_administration\_planning\_initial.mpp

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

# rENDU

A la fin du projet vous devez :

* Déposer votre application sur un GIT
* Fournir le cahier de recette
* Fournir les scripts de création (et d’alimentation avec votre un jeu de test). Les scripts peuvent être dans le dépôt GIT
* Un document d’utilisation de l’application
* Le planning réel fait sous MSPROJECT ou GanttProject