



2SIO

Application gestion administrative

Présentation du projet

SOMMAIRE

I.	HISTORIQUE.....	4
II.	PRESENTATION DU PROJET.....	5
	A) CONTEXTE	5
III.	DESCRIPTION DE L’EXISTANT	6
	A) BASE DE DONNEES.....	6
IV.	APPLICATION	7
V.	DIAGRAMME DE CLASSE	9
VI.	DIAGRAMME D’ACTIVITE.....	10
VII.	ORGANISATION	13
	A) DECOUPAGE DU PROJET	13
	B) DETAIL DES ANOMALIES.....	13
	C) ÉVOLUTIONS	13
	➤ <i>Partie Etudiant.....</i>	<i>13</i>
	➤ <i>Partie Section.....</i>	<i>13</i>
	➤ <i>Partie cours.....</i>	<i>13</i>
	➤ <i>Partie Utilisateur.....</i>	<i>14</i>
VIII.	CONTRAINTES.....	15
IX.	GROUPES	16
X.	PLANNING.....	17
XI.	RENDU.....	18



TABLE DES FIGURES

<i>Figure 1 : Use Case application</i>	<i>5</i>
<i>Figure 2 : MCD BDD.....</i>	<i>6</i>
<i>Figure 3 : Description BDD</i>	<i>6</i>
<i>Figure 4 : Ecran de connexion</i>	<i>7</i>
<i>Figure 5 : Ecran d'accueil.....</i>	<i>7</i>
<i>Figure 6 : Ecran liste des étudiants.....</i>	<i>8</i>
<i>Figure 7 : Ecran ajouter un étudiant</i>	<i>8</i>
<i>Figure 8 : Diagramme de classe</i>	<i>9</i>
<i>Figure 9 : Diagramme activité "ajout étudiant"</i>	<i>10</i>
<i>Figure 10 : Diagramme d'activité "ajout cours"</i>	<i>11</i>
<i>Figure 11 : Diagramme d'activité "Consultation d'une section"</i>	<i>12</i>



I. HISTORIQUE

Identification du Document			
Client :	Magali GOUDET	Projet :	Gestion
Rédacteur :	Magali GOUDET	Professeurs :	Magali GOUDET Bastien THOUVEREZ Oualid RAHAIMI Christophe VIRMONT

Historique des Modifications		
07/03/2023	V1.0	Initialisation
15/03/2023	V1.1	Mise à jour
14/03/2025	V2.0	Evolution du projet, bascule en java

Documents utiles	

II. PRESENTATION DU PROJET

a) 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 finie.

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

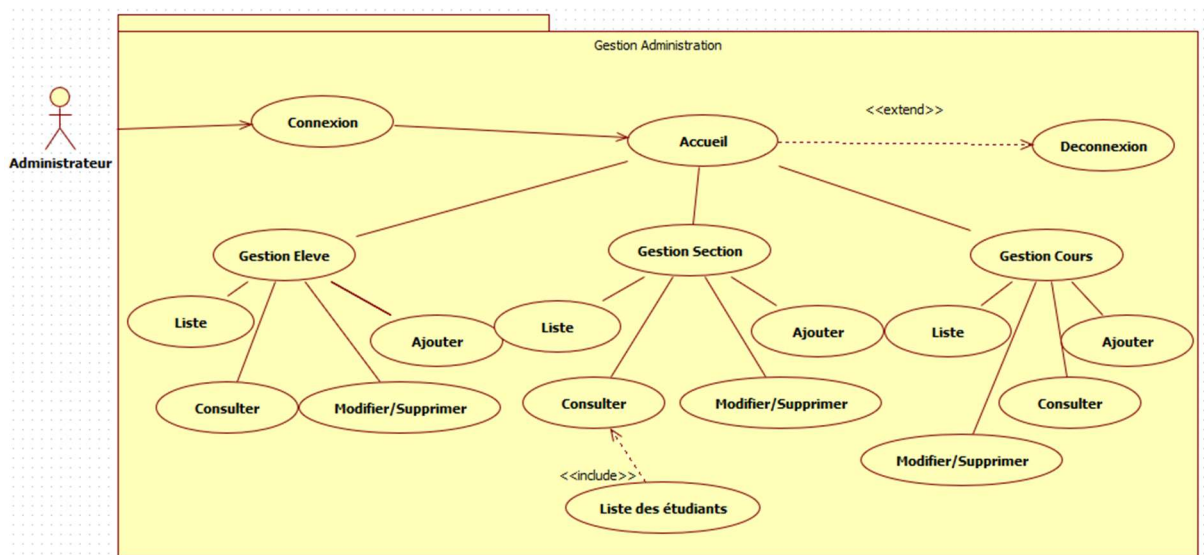


Figure 1 : Use Case application

III. DESCRIPTION DE L'EXISTANT

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

a) Base de données

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

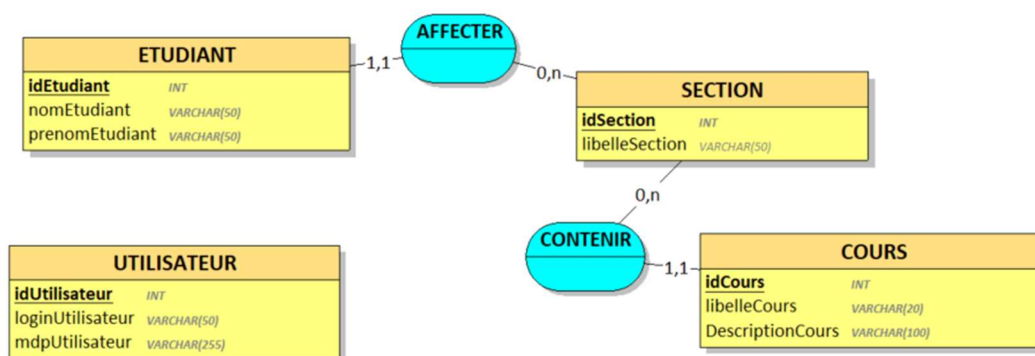


Figure 2 : MCD BDD

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



IV. APPLICATION

Voici une description des différents écrans existant :

CONNEXION

LOGIN :

MOT DE PASSE :

Connexion

Figure 4 : Ecran de connexion

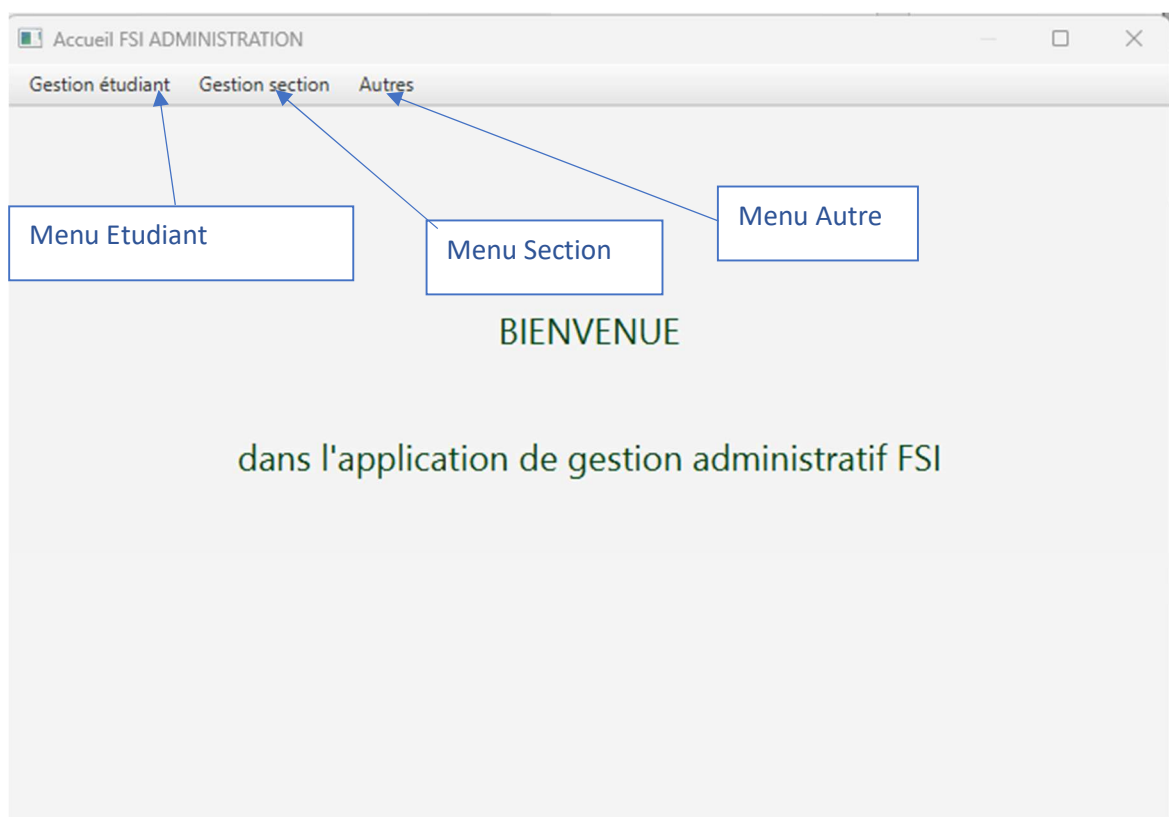


Figure 5 : Ecran d'accueil

Liste étudiant		
Gestion étudiant Gestion section Autres		
nom étudiant	prénom étudiant	
Singier	Romain	
Dupont	Remy	
Castre	Alexandra	
Hazaki	Tokyo	
Fournil	Dimitri	
GOUDET	Hanaé	
Goudet	Joachim	

Figure 6 : Ecran liste des étudiants

ajouter un étudiant	
Gestion étudiant Gestion section Autres	
AJOUT D'UN NOUVEL ETUDIANT	
Nom étudiant	<input type="text"/>
Prénom étudiant	<input type="text"/>
Section étudiant	<div> <div>section 1</div> <div>section 2</div> </div>
<div> <div>ENREGISTRER</div> <div>EFFACER</div> <div>RETOUR</div> </div>	

Figure 7 : Ecran ajouter un étudiant

V. DIAGRAMME DE CLASSE

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

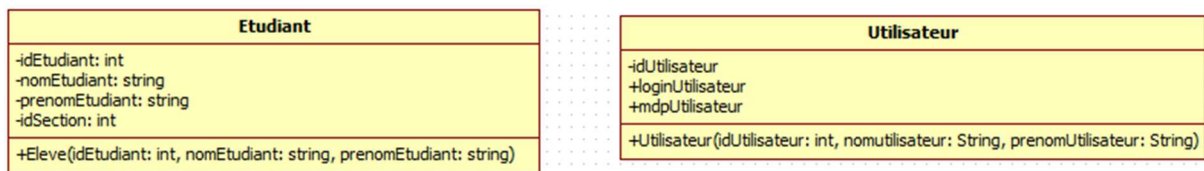


Figure 8 : Diagramme de classe

VI. DIAGRAMME D'ACTIVITE

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

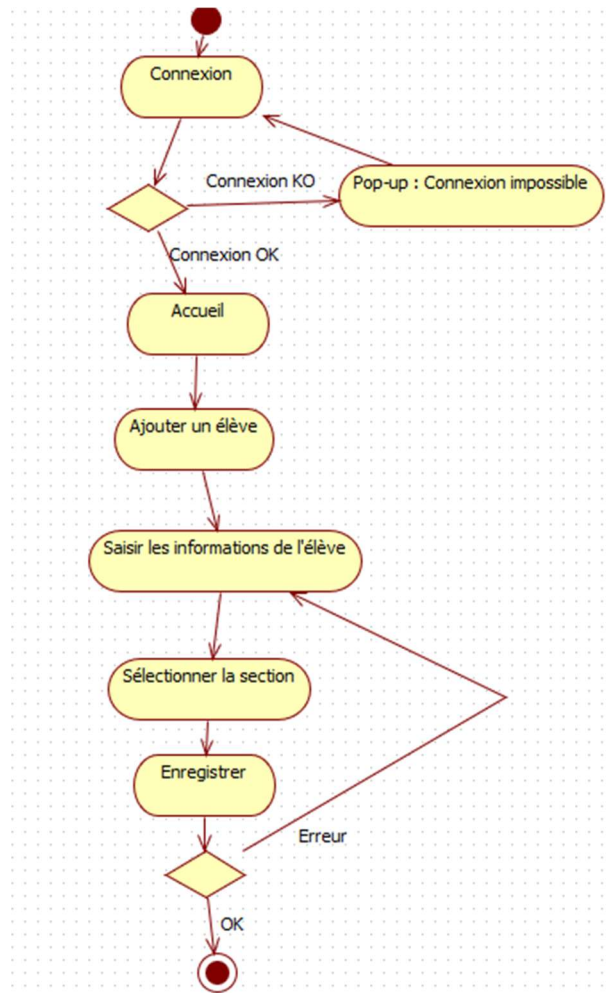


Figure 9 : Diagramme activité "ajout étudiant"

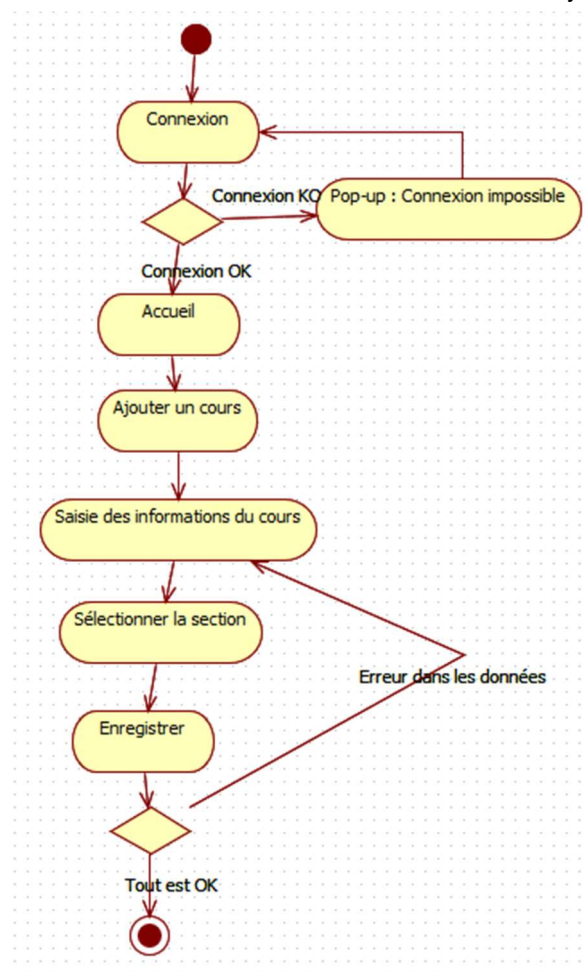


Figure 10 : Diagramme d'activité "ajout cours"

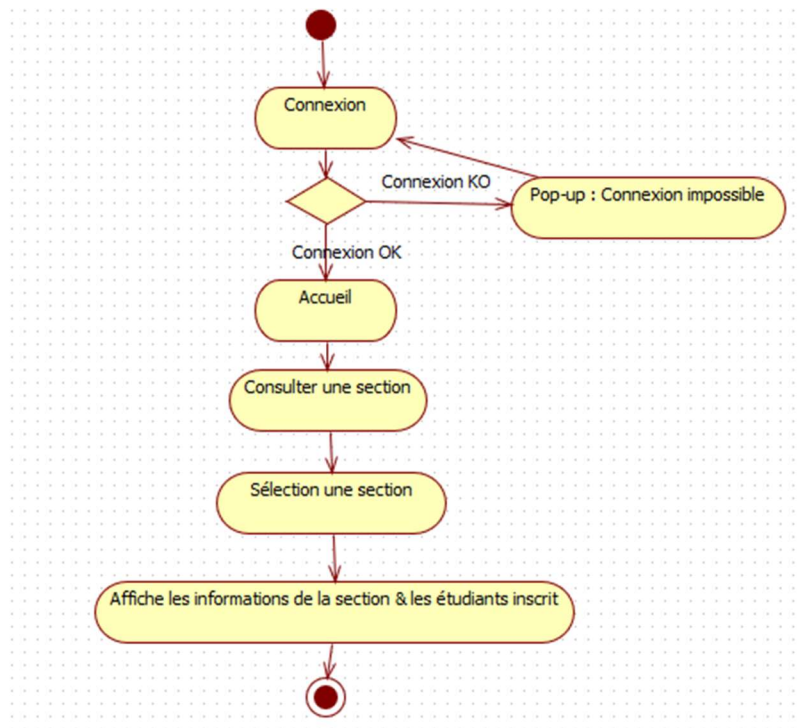


Figure 11 : Diagramme d'activité "Consultation d'une section"

VII. ORGANISATION

a) 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 GLPI 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. Il faudra, bien sûr, créer des tickets sous GLPI.

b) Détail des anomalies

L'ESN qui a fourni la 1^{ère} 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.

c) Évolutions

➤ Partie Etudiant

Sur les étudiants voici les fonctionnalités complètes attendues :

- La liste des étudiants doit contenir toutes les données de l'étudiant (nom, prénom, libelle de sa section ainsi qu'en plus sa date de naissance)
- Quand on ajoute un étudiant, on veut pouvoir renseigner toutes les infos, dont la date de naissance et on veut que sa soit le libellé de la section qui apparait dans le listing.
- On veut pouvoir modifier et supprimer un étudiant

➤ Partie Section

Une section concerne une classe, il a été choisi de l'appeler ainsi pour éviter des problèmes de compréhension entre la classe en notion Objet et la classe (2SIO par exemple).

Sur les sections, on veut pouvoir :

- Voir la liste des sections avec le nombre d'étudiant inscrits
- Modifier / Supprimer une section
- Consulter la liste des étudiants présent dans une section
- Consulter la liste des cours de cette section

➤ Partie cours

Une section est composée de plusieurs cours. Par exemple dans la section « Licence développeur ... » il existe un cours de Front-end, d'UML, ...

Nous voulons pouvoir gérer au niveau des cours :

- La liste des cours avec toutes les informations et le libellé de la section dont il dépend
- L'ajout d'un cours
- La modification et la suppression d'une cours

➤ Partie Utilisateur

On souhaite que le nom de l'utilisateur apparaisse sur l'écran d'accueil

➤ Autres demandes

Pour faciliter la TMA à la suite de votre livraison, nous voudrions que :

- L'ensemble de votre code soit commenté, optimiser et exploitable
- Mettre en place un système de LOG qui a pour but de suivre l'activité de l'application. Les ajouts, modifications, suppression au niveau des étudiants et des cours doivent être tous stockés. On veut pouvoir savoir quel utilisateur a fait telle action.
- Faire apparaitre le logo de l'entreprise sur tous les écrans
- Améliorer le visuel de l'application



VIII. 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 titres doivent être avec la couleur verte #024010.

Vous pouvez changer le design, mais ce dernier doit rester simple, clair et professionnel.

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

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



IX. 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.

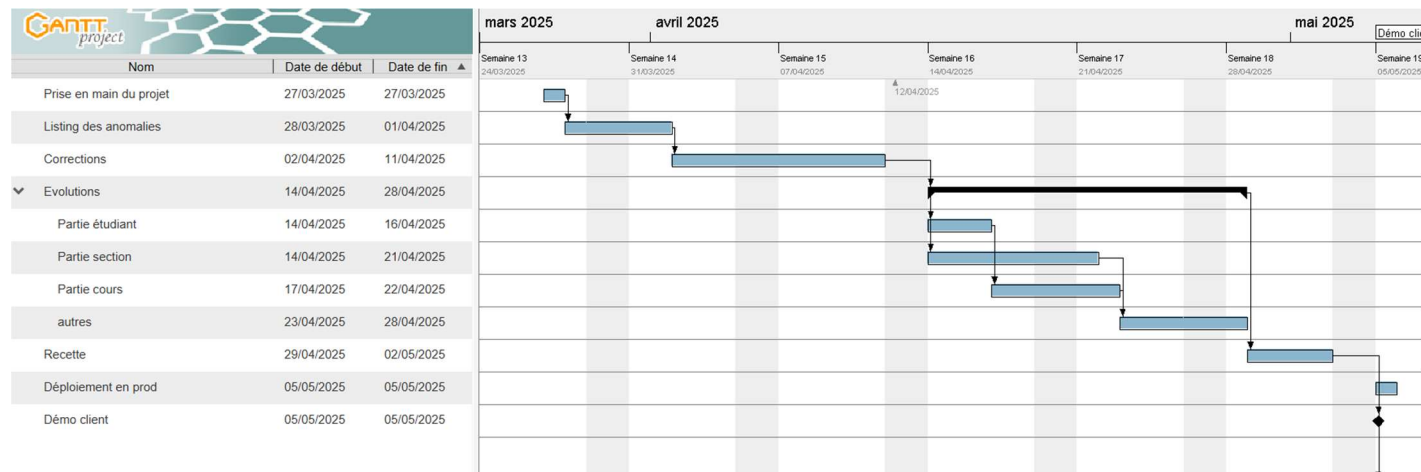
Il en est de même pour les évolutions, lister les tâches du lot, ensuite le CP fera l'affectation des tickets qui devra être **ABSOLUMENT** respecter.

Numéro équipe	Membre 1	Membre 2
N°1	GREFFET Jules	NEUVILLE Yann
N°2	ROUX Maxime	CABOT Robin
N°3	BOUQUET Yanis	LAFFI Wissem
N°4	GHARBI Aymen	ZENASNI Hakim
N°5	POLY Eloi	GUILLLOT Gaspard
N°6	IGLESIAS Alexandre	SAADIA Zakarriya
N°7	TRAN Quang Hieu	BERIOT Alexis
N°8	TROPEL Rémi-Jean	



X. PLANNING

Voir fichier 2SIO_FSI_administration_planning_initial.gan





XI. 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 GanttProject