



**UNIVERSITÉ
DE LORRAINE**



nancy Charlemagne
Département Informatique

Projet Tutoré

S23 : Outil graphique et interactif de
placement d'étudiants en salle d'examen

Réalisé par :

**Plaid Justin
Guezennec Lucas
Biancalana Théo
Muller Geoffroy**

Tuteur :

Nadia Bellalem

2018-2019

Table des matières

1. Introduction

2. Analyse

- a. Diagramme cas d'utilisation

- b. Diagramme d'activités

- c. Diagramme de séquence

- d. Modèle relationnelle

3. Réalisation

- a. Architecture logicielle

- b. Tests de validation

- c. Difficultés rencontrées

4. Conclusion

1. Introduction

Ce document a pour objectif de décrire toutes **les étapes** et **les cheminements** ayant permis la **réalisation** du projet tuteuré: outil graphique et interactif de placement d'étudiant dans une salle d'examen. Tout le long de ce document, il sera appelé **EtuPlacement**.

Il s'agit donc d'un logiciel programmé en JAVA, ayant pour objectif de **placer des étudiants dans une salle d'examen**. Pour ce faire, le logiciel utilise des groupes d'étudiant qui sont soit importé depuis **un fichier Excel** ayant une mise en forme particulière, ou **soit créés directement depuis le logiciel**.

EtuPlacement aura également besoin de **salle** pour pouvoir placer les étudiants, pour répondre à ce besoin, le logiciel intègre un module permettant **une gestion de salle**, ainsi, il est possible de créer facilement des salles et de les modifier par la suite, un groupe de salle est déjà intégré dans le logiciel pour permettre une utilisation rapide de celui-ci.

Enfin, le système doit **placer les étudiants** en respectant certaines contraintes comme **l'écart entre les étudiants, les places indisponibles, les spécificités des étudiants...** Certaines de ces **contraintes**, seront directement sélectionnable lors de la création de l'examen et d'autres sont directement prises en compte, lorsque le système à terminer la tâche de placement, le résultat est proposée sous la forme **d'un fichier Excel**.

Afin de mieux exposer notre projet ainsi que son fonctionnement nous avons fait ce document qui le présente sous différentes formes.

Pour ce faire, nous avons effectué des diagrammes et parmi ceux-ci, on pourra retrouver des **diagrammes cas d'utilisation** qui permettent de décrire l'ensemble des fonctionnalités offertes par le système, des **diagrammes d'activité** pour

représenter les états du système, des **diagrammes séquence** qui vont permettre de rentrer dans les détails d'une fonctionnalité spécifique (Par exemple : la création d'examen) afin de montrer les **interactions entre l'utilisateur et le système**, enfin un **diagramme de classe** qui permet de décrire la structure interne du système.

L'équipe qui a effectué le projet est composée de MULLER Geoffroy, PLAID Justin, BIANCALANA Théo et GUEZENNEC Lucas. Nous nous sommes réparties les tâches de la façon suivante, le projet se découpant en trois modules (Examen, Étudiant et Salle), nous nous sommes chacun attribué un module, pour la personne restante, il s'agissait de la conception des modèles Java et de l'intégration d'une base de données.

Lucas GUEZENNEC	Module Salle
Geoffroy MULLER	Module Examen
Théo BIANCALANA	Module Étudiant
Justin PLAID	Gestion des modèles et base de données

Pour concevoir ce projet, nous avons procédé par itération, 3 au total, et avons utilisé l'outil Trello afin de procéder à la séparation des tâches et à l'avancement de celle-ci.

À l'itération 1 nous avons choisi de nous concentrer sur les modules importants tel que Examen et Étudiant. Nous avons aussi fait le choix de faire l'interface totale du logiciel.

Ensuite lors de l'itération 2 nous nous sommes concentrés sur le module Salle tout en continuant la progression sur les autres modules.

Enfin, à l'itération 3 nous nous sommes occupés des détails du logiciel et implémentés des nouvelles fonctionnalités qui nous avaient été conseillées.

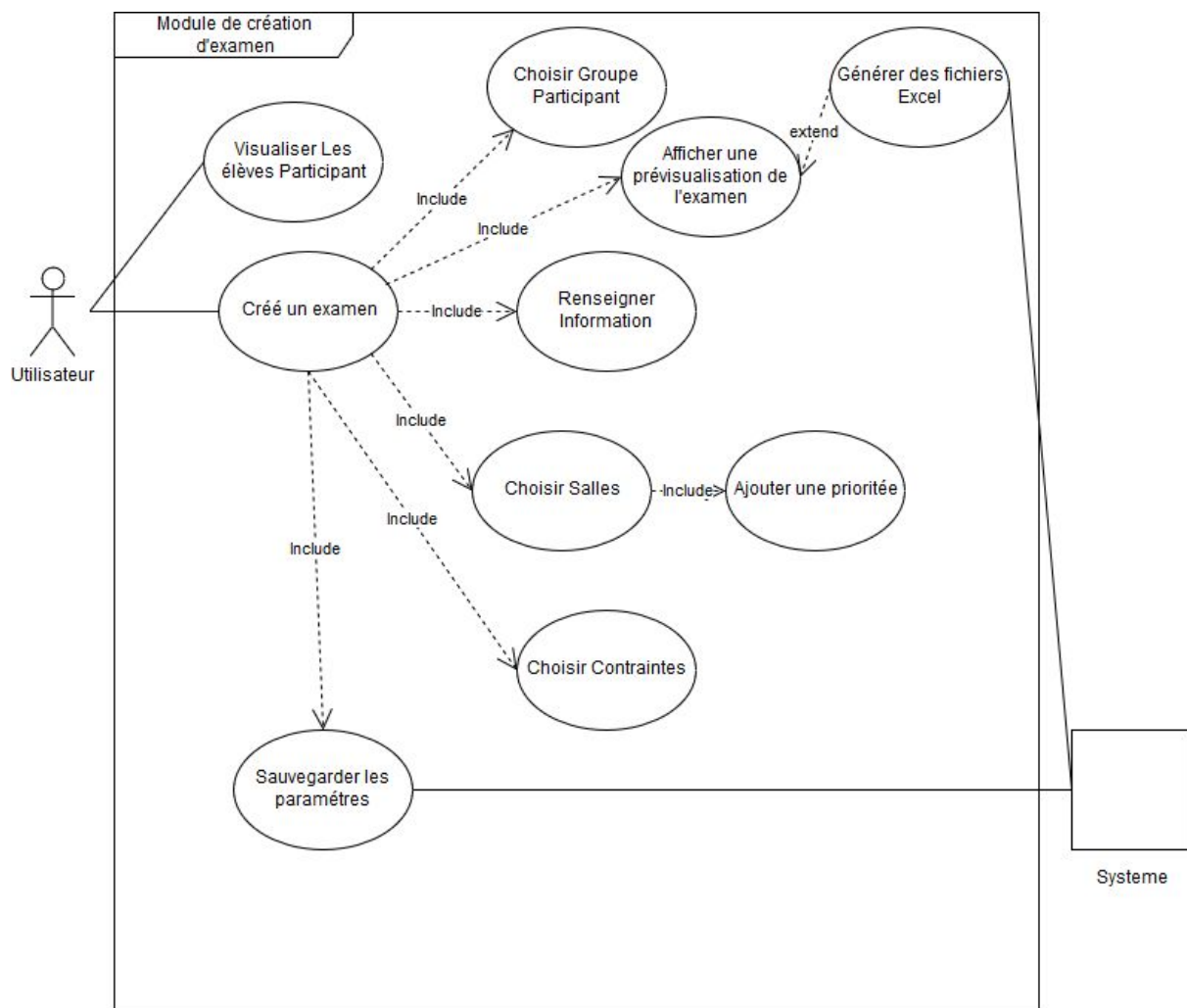
2. Analyse

Avant de commencer la réalisation du projet, nous avons effectué une analyse de celui-ci. Il s'agit donc d'analyser l'environnement dans lequel le logiciel va être utilisé, d'identifier les acteurs et de bien comprendre le besoin exprimé. Mais aussi de réfléchir aux possibilités qu'allait offrir Etuplacement ainsi qu'à ces limites et surtout comment mettre en place toutes ces fonctionnalités. Cette étude se traduit donc par l'élaboration de différents diagrammes : cas d'utilisation, cas d'activité, de séquence, classe et la conception de la base de données.

Il nous a semblé dès le départ pertinent de séparer le projet en 3 différents modules. Un module examen ayant pour objectif de gérer tout ce qui est relatif à la création d'un examen, celui-ci est totalement dépendant des deux autres. Un module étudiant qui s'occupe de gérer l'ensemble des étudiants présent dans le logiciel, tout ce qui concerne l'ajout, la suppression, la modification d'étudiant ou de groupe concerne ce module. Enfin, le module salle qui gère l'ensemble des salles, on peut donc les visualiser, les modifier, en ajouter et en supprimer. L'ensemble de ces trois modules permet de gérer tout ce qui est nécessaire à la création d'examen.

A. Diagramme cas d'utilisation

Module Examen



Ce diagramme présente les possibilités du module examen, ainsi, on pourra **créer un examen et visualiser les étudiants qui y participent**. On observe également les actions nécessaires à la **création d'examen** comme **l'ajout de contraintes, l'ajout de salle par ordre de priorité de remplissage**. Concernant la bulle "**Renseigner informations**", il s'agit de renseigner : **le nom de l'examen, la matière concernée et la date de celui-ci**. De plus, lors de la génération de

l'examen, les paramètres de celui-ci sont sauvegardés et ainsi l'examen sera pré-rempli pour une création plus rapide d'un prochain examen.

Concernant l'évolution de cette étude, il y eu l'ajout de la prévisualisation avant la génération des fichiers Excel.

Description textuelle du cas d'utilisation créer un examen

Précondition :

Un ou plusieurs groupes sont présents dans le logiciel.

Un ou plusieurs étudiants sont présent dans le logiciel.

Une ou plusieurs salles sont présentes dans le logiciel.

L'utilisateur se trouve dans l'onglet de création d'examen.

Postcondition :

Si l'examen a été approuvé par l'utilisateur alors générer un fichier avec les étudiants participants associé à leur place, le système générera aussi une feuille par salle contenant la liste des étudiants présent dans celle-ci avec un emplacement pour la signature.

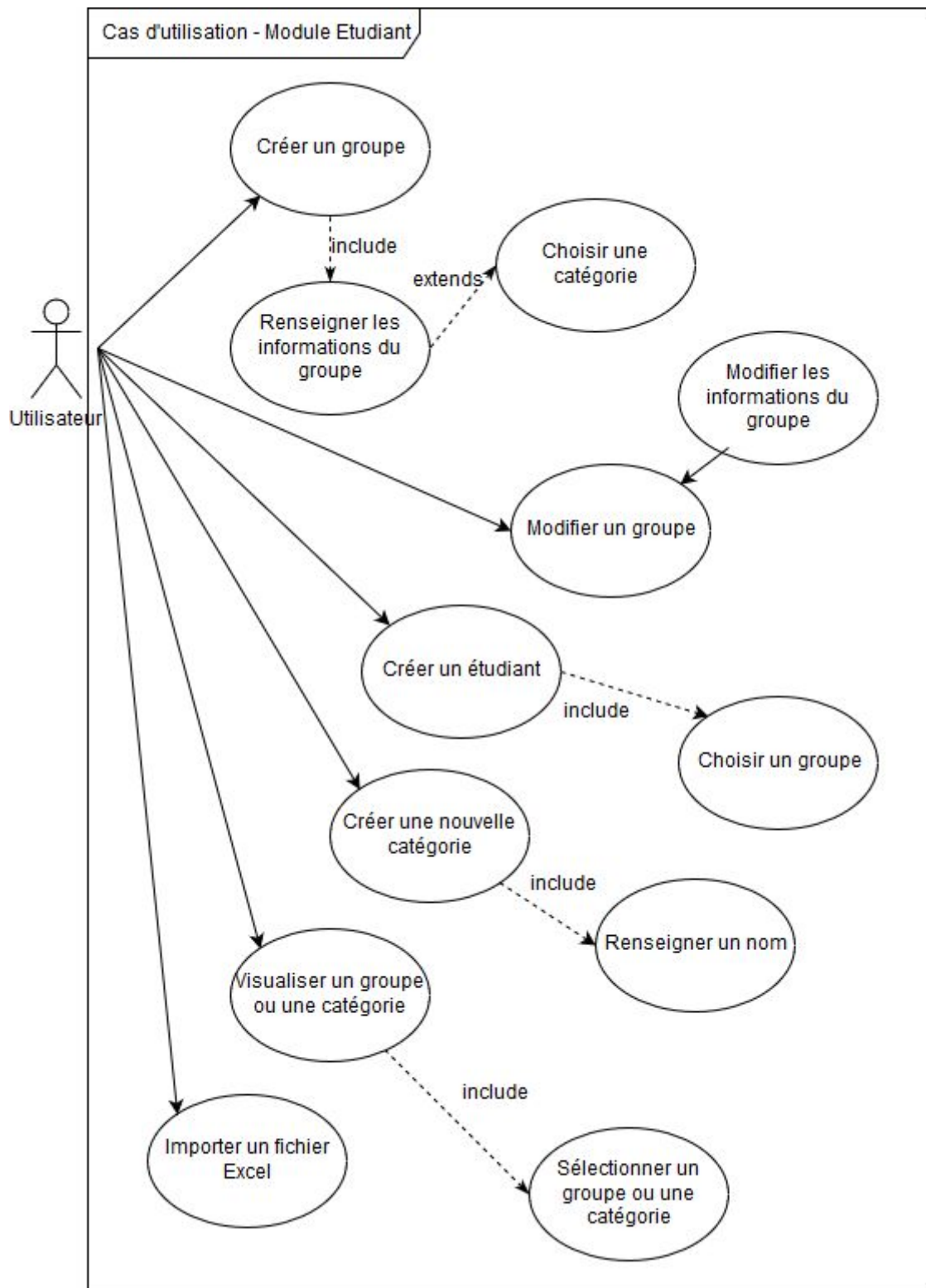
Déroulement normal :

- (1) L'utilisateur se rend dans l'onglet "Examen" du logiciel.
- (2) L'utilisateur renseigne : le nom, la matière et la date de l'examen.
- (3) L'utilisateur choisi les groupes participants et les visualise.
- (4) L'utilisateur choisi les salles par ordre de priorité.
- (5) L'utilisateur choisi des contraintes.
- (6) L'utilisateur confirme l'examen.
- (7) Le système génère un placement.
- (8) L'utilisateur confirme le placement.
- (9) Le système génère un fichier.

Variantes :

- (A) La date choisi lors de l'étape (2) peut être antérieur à la date du jour, dans ce cas le système notifie l'utilisateur que la date doit être modifiée et met par défaut la date du jour.
- (B) L'utilisateur peut ne pas avoir mis de groupe et/ou de salle, dans ce cas le système notifie l'utilisateur lorsque ce dernier souhaite générer l'examen.
- (C) Lors de l'étape (4), il se peut que l'utilisateur choisi un nombre de salle insuffisant pour le nombre d'élèves choisi. Dans ce cas, le système notifie l'utilisateur que l'examen ne peut être créé par manque d'emplacement.
- (D) Un fichier du même nom peut se trouver dans le répertoire où l'on souhaite créer le fichier à l'étape (9), dans ce cas on demande à l'utilisateur si il souhaite écraser le fichier ou ajouter :"(1)" au nom du fichier.

Module Étudiant



Dans ce diagramme, on montre les fonctionnalités du **module étudiant**, il sera donc possible pour l'utilisateur de **créer un groupe** directement en passant par le logiciel (création d'un groupe vide, les étudiants étant ajoutés lors de la création d'un étudiant) ou en important un fichier Excel (les groupes étant automatiquement détecté dans le fichier Excel). On pourra également modifier les groupes existant en modifiant les informations du groupe sélectionné. Enfin, l'utilisateur pourra visualiser l'ensemble des élèves des groupes qu'il aura sélectionné.

Concernant les modifications apportées depuis l'étude préalable, lors de la modification d'un groupe, le diagramme laissait entendre que l'on pouvait directement ajouter des étudiants à celui-ci, or ce n'est pas le cas, nous avons donc supprimé cette bulle et modifier également la bulle de modification des informations qui avant permettait de modifier les informations d'un étudiant. Également, sur l'ancien diagramme il était possible de créer un groupe en important un fichier Excel ce qui n'est pas le cas, cela a donc été corrigé. D'autres ajustements ont été effectués sur ce diagramme.

Toutes ces modifications ont été effectuées à cause d'une refonte totale de ce module lors de la conception du projet.

Description textuelle du cas d'utilisation importer une liste d'étudiant

Précondition :

L'utilisateur se trouve dans l'onglet du module étudiant.

L'utilisateur possède un fichier Excel conforme.

PostCondition :

Une fois la liste importée sans erreur, les groupes trouvés dans celle-ci sont alors ajoutés à la catégorie choisie.

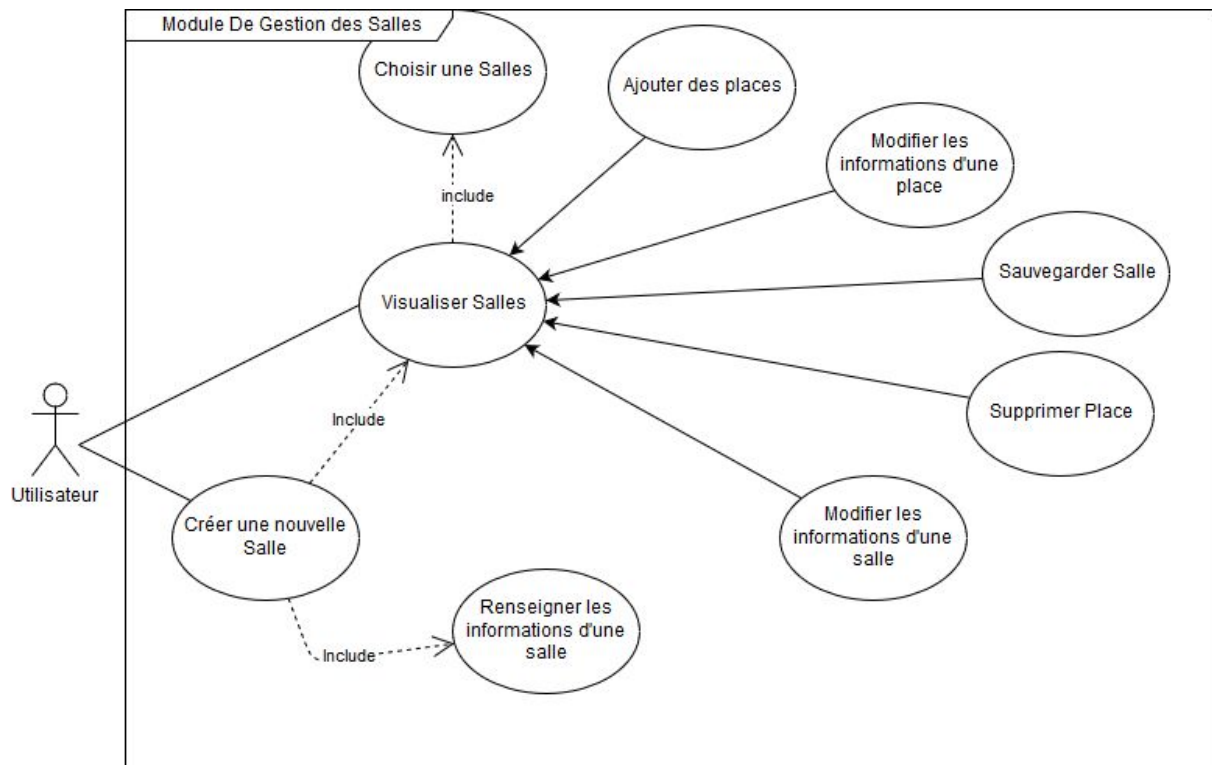
Déroulement normal :

- (1) L'utilisateur se rend dans l'onglet "Étudiants" du logiciel
- (2) L'utilisateur clique sur le bouton permettant d'importer un fichier excel
- (3) L'utilisateur choisi dans quelle catégorie il souhaite ranger les groupes contenus dans le fichier excel
- (4) L'utilisateur sélectionne le fichier excel désiré
- (5) L'utilisateur confirme l'importation
- (6) Le système détecte les groupe contenu dans le fichier et créer les groupes correspondant puis les ajoutes au logiciel.

Variantes :

- (A) Lors de l'étape (4) l'utilisateur peut choisir un fichier excel n'ayant pas la mise en forme des données souhaité, dans ce cas on le signale à l'utilisateur.

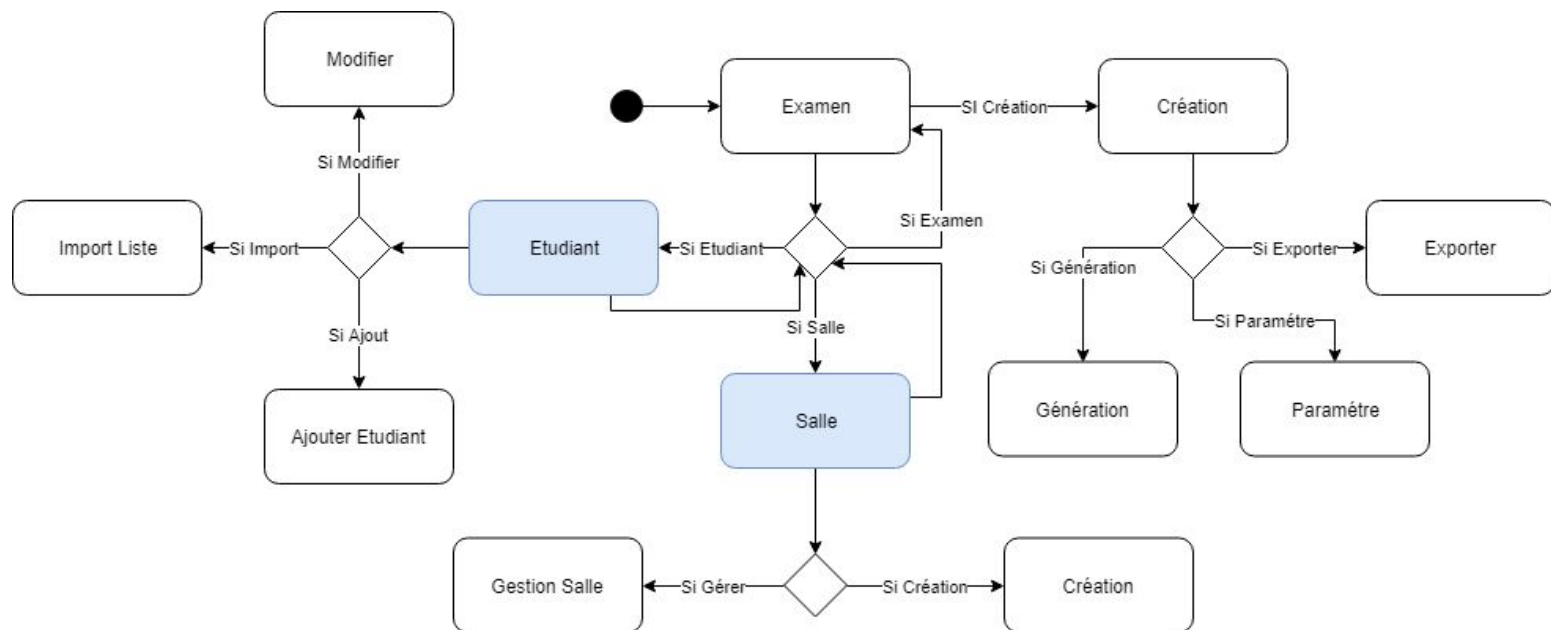
Module Salle



Enfin, ce dernier diagramme de cas d'utilisation présente le fonctionnement du **module salle**. Il est possible dans ce module de **créer des salles et de les modifier**, on pourra donc modifier **le type d'une place** (place, couloir, place inutilisable), **ajouter une place, supprimer la salle ou encore modifier ses informations**.

Ce Diagramme a subi quelques modifications depuis l'étude préalable, certaines bulles étant inutile nous les avons supprimés. Nous avons aussi constaté que certains textes n'étaient pas assez explicites ou plus en accord nous les avons donc modifiés.

B. Diagramme cas d'activité



Ce diagramme d'activité représente l'interface général, on peut voir les pages principales du logiciel, **Examen**, **Salle** et **Étudiant**.

Les pages **Étudiant** et **Salle** seront détaillées plus précisément ci-dessous.

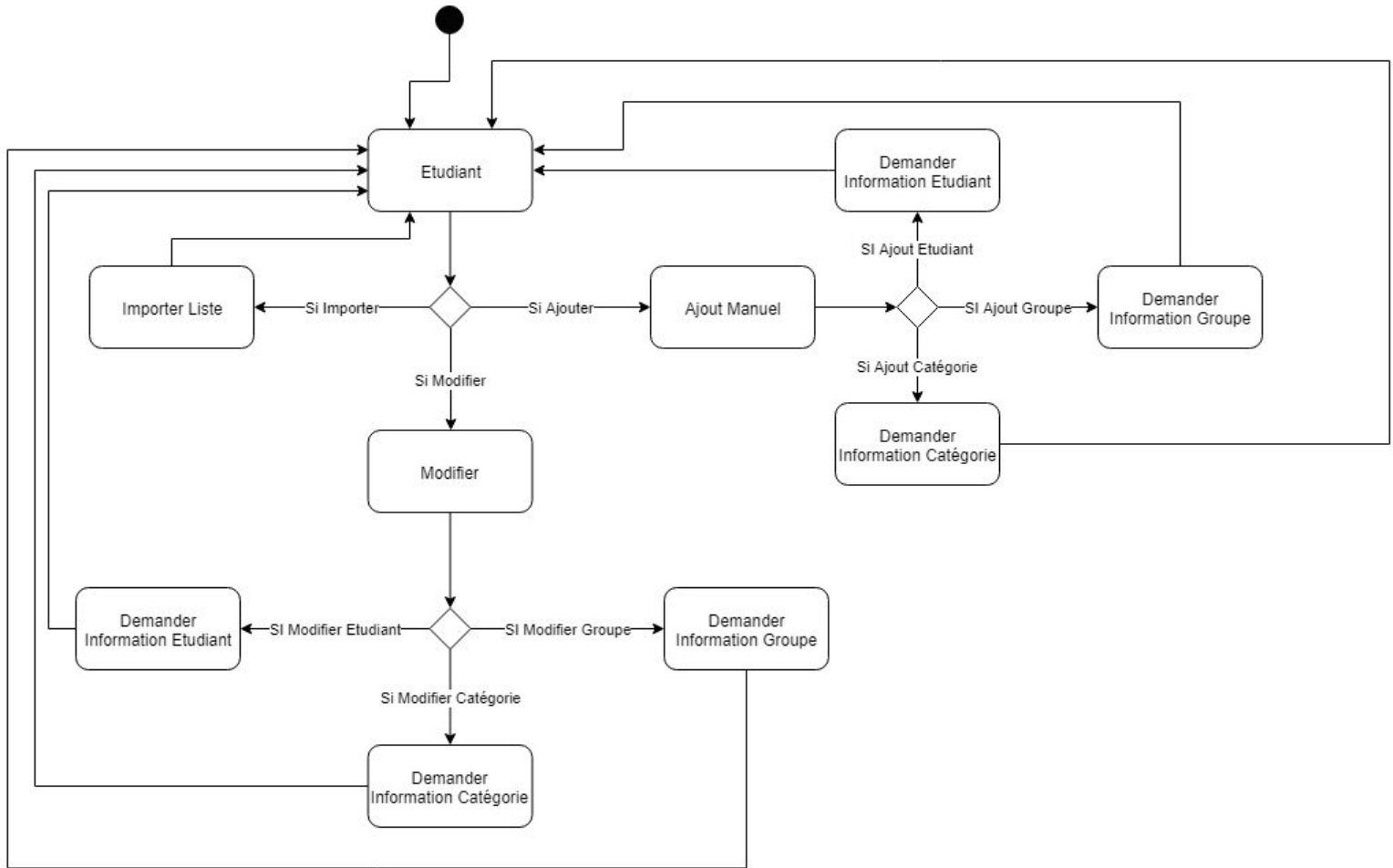
Module Étudiant

Le module étudiant est la suite du diagramme général vu précédemment, on peut y voir ses trois fonctions principales : la modification, l'ajout et l'importation.

L'ajout permet la création d'un Étudiant à partir des informations données, mais aussi la création de Groupe et de catégorie.

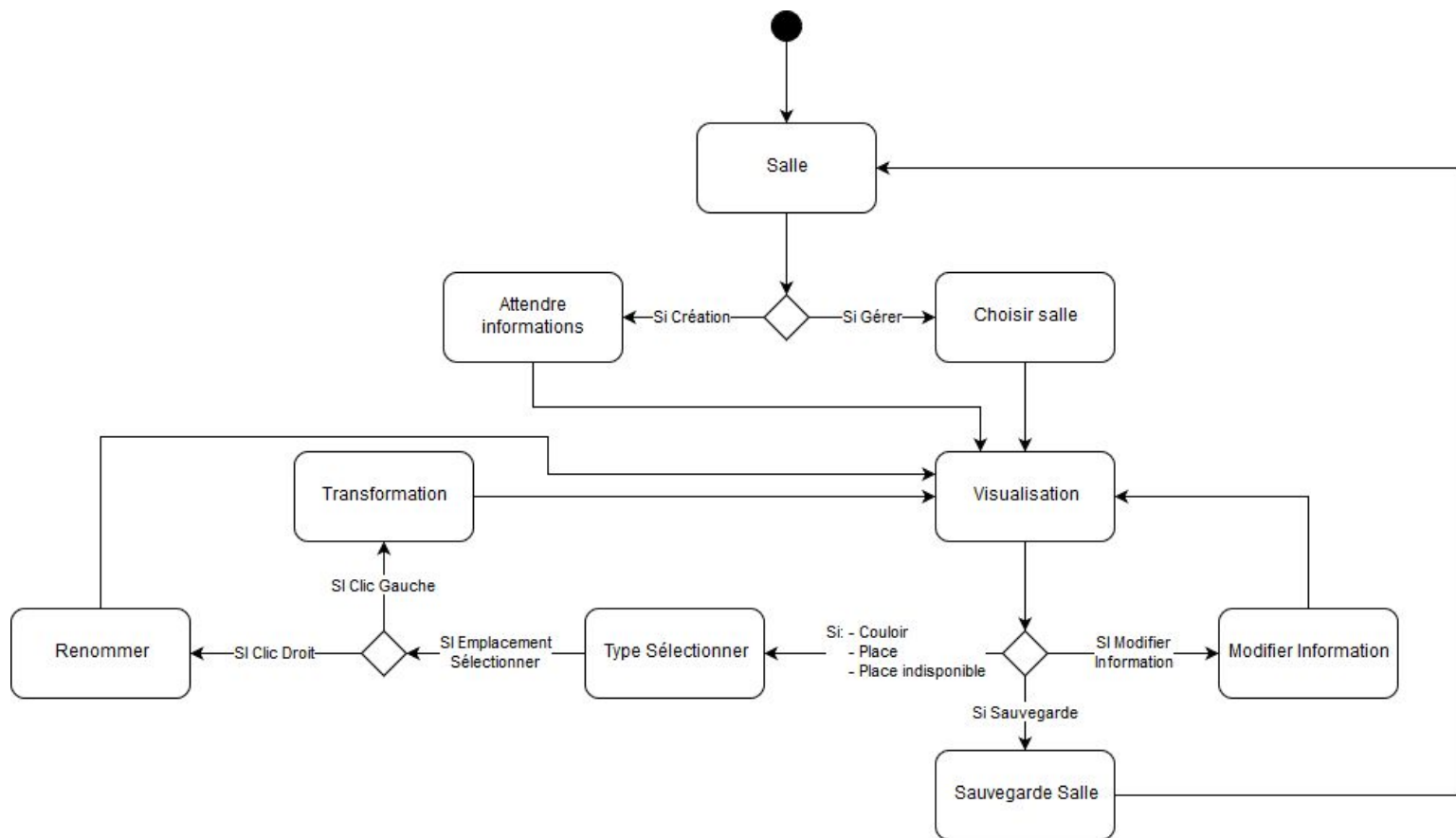
La modification permet de modifier les informations d'un Étudiant, d'un Groupe ou d'une Catégorie.

L'importation consiste à **importer une liste Étudiants en format Excel**.



Avec ce diagramme, on visualise l'activité du système en fonction de ce que souhaite effectuer l'utilisateur avec le logiciel, en effet, on peut voir que le système va demander un certain nombre d'informations à l'utilisateur ainsi que la position où va se retrouver l'utilisateur après avoir fait une action.

Module Salle



Dans le module Salle ,on peut voir la fonction principale : la **Visualisation**.

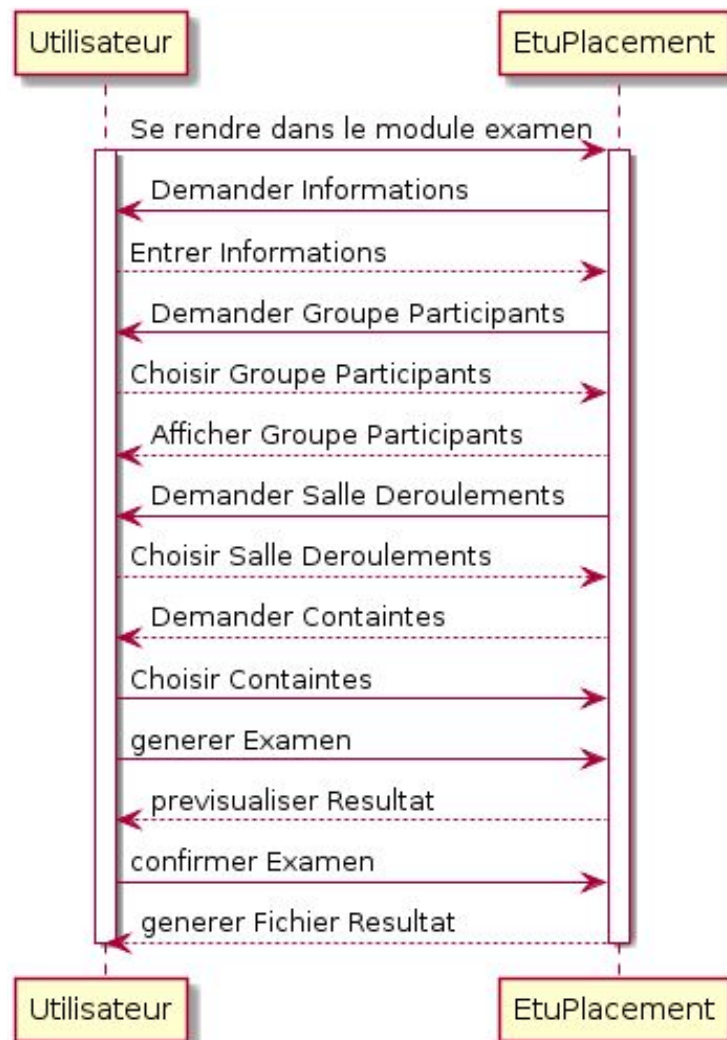
Tout d’abord, la **Création** d’une salle où l’utilisateur devra renseigner les dimensions de la salle (en nombre de places) et le nom de la salle.

Ensuite, la partie **Modification** consiste à choisir une salle pour la modifier.

Depuis la Visualisation, l’utilisateur pourra modifier les informations de la salle (nom ...), sauvegarder la salle et aussi gérée les places. Pour ce faire, un sélecteur sera présent pour choisir “Place”, “Place indisponible”, “Couloir”. Ce sélecteur aura automatiquement une option sélectionnée. Ensuite, le système pourra modifier ou renommer la place grâce à un clic de souris.

C. Diagrammes Séquences

Création d'un Examen (Module Examen)



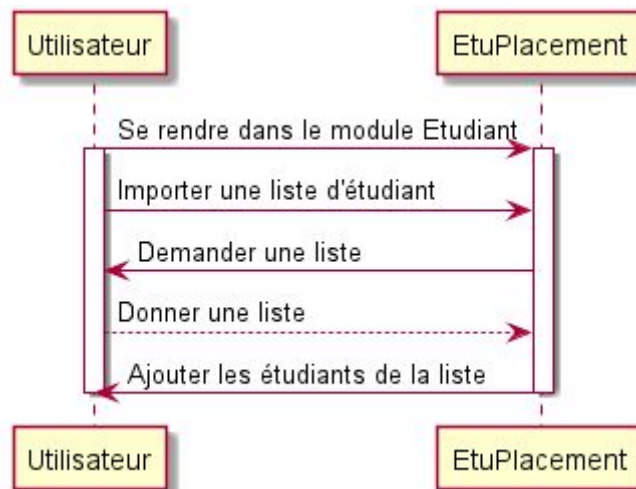
Pour **créer un examen**, il faut que l'utilisateur se rende sur le module "**Examen**" , dans lequel il entrera les **différentes informations** de l'examen comme : **le nom**, **la matière** et **la date**.

Après, il faudra choisir **les groupes participants** à l'examen et **les salles** utilisées pour l'examen.

Pour finir, il faudra cliquer sur "**Créer Examen**" , une **prévisualisation** de celui-ci aura lieu. L'utilisateur cliquera sur "**Confirmer**" pour **générer les fichiers** souhaités.

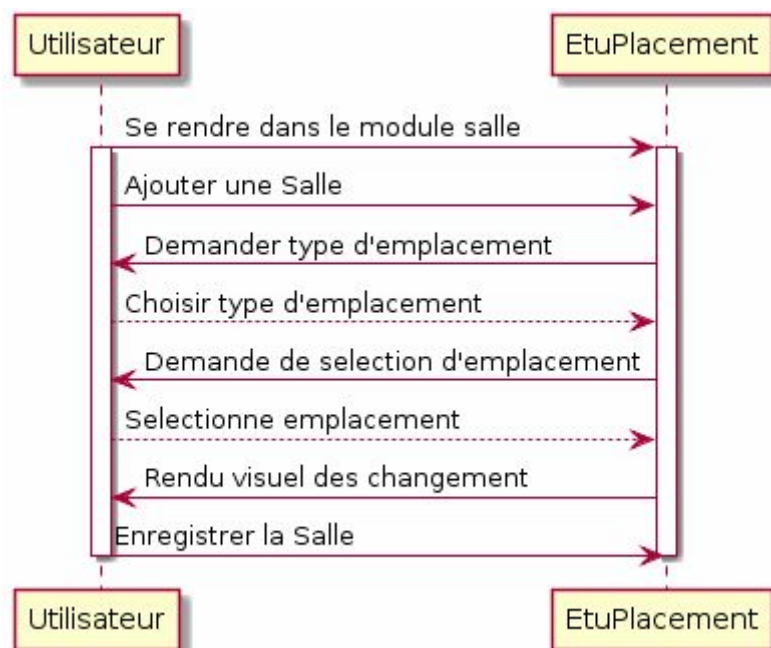
Importation d'une liste d'étudiant (Module Élève)

Ce diagramme décrit les interactions entre l'utilisateur et le système lors d'une importation de liste.



Création d'une salle d'examen (Module Salle)

Ce diagramme montre les interactions entre l'utilisateur et le système lors d'une importation de la création d'une salle d'examen.



Pour **créer une Salle**, il faut que l'utilisateur se rende sur le module "**Salle**" et cliquer sur le bouton "**Ajouter**", ensuite l'utilisateur pourra **sélectionner un "type d'emplacement"** (Comme : une "**Allée**", une "**Place**", une "**Place Inutilisable**") à placer dans la salle (représenter sous forme de quadrillage). Un **rendu visuel** ce fait en temps réel.

Pour finir, l'utilisateur clique sur "**Enregistrer**" ce qui va enregistrer la salle dans le logiciel.

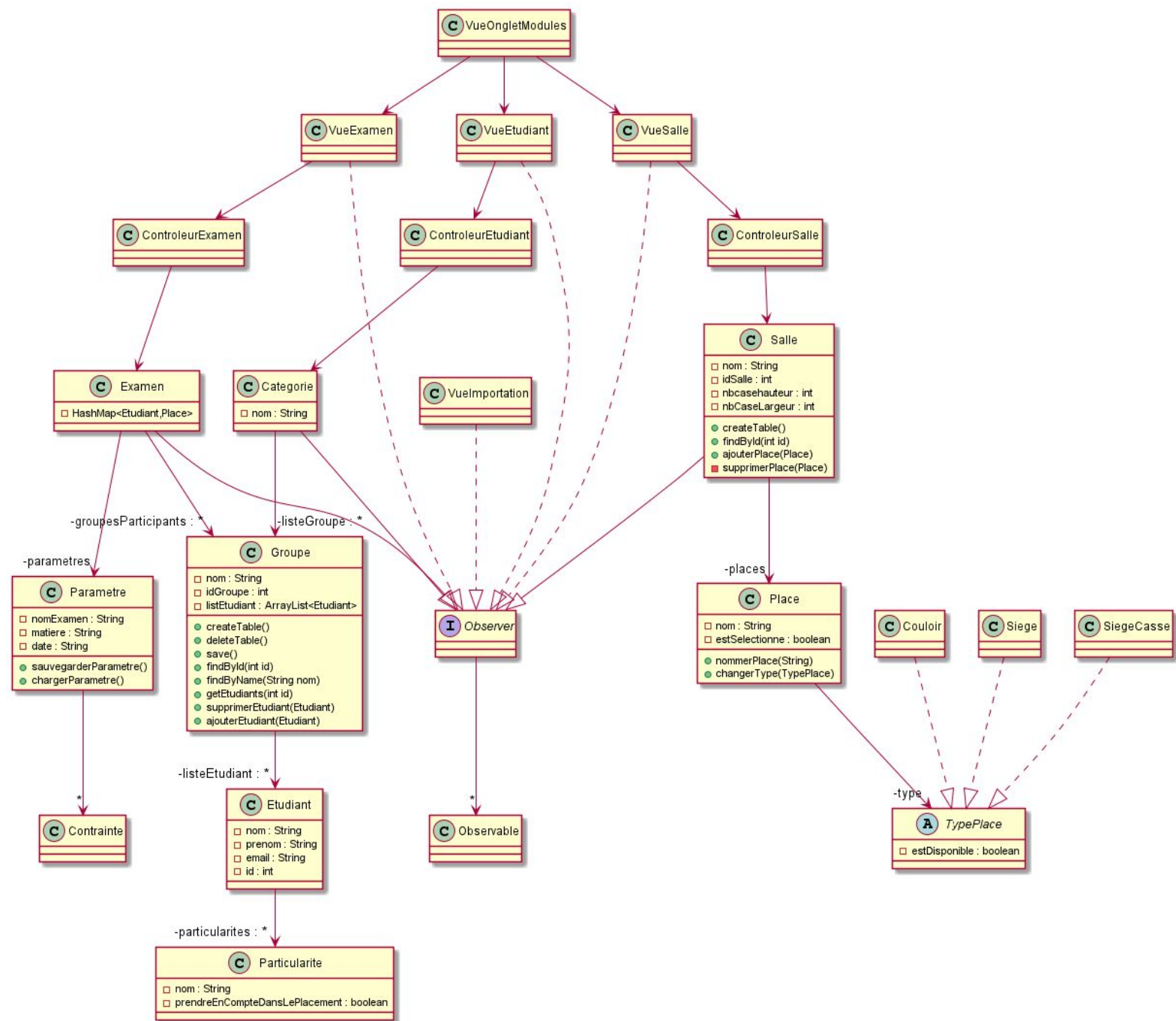
D. Diagramme de classe

Voici le dernier diagramme, il s'agit d'un diagramme de classe qui va permettre de décrire le fonctionnement interne du système.

La classe VueOngletModule correspond à une vue avec des onglets affichant les vues Examen, Étudiant et Salle.

Il s'agit d'une version minimaliste du diagramme de classe, un diagramme complet n'aurait pas pu tenir.

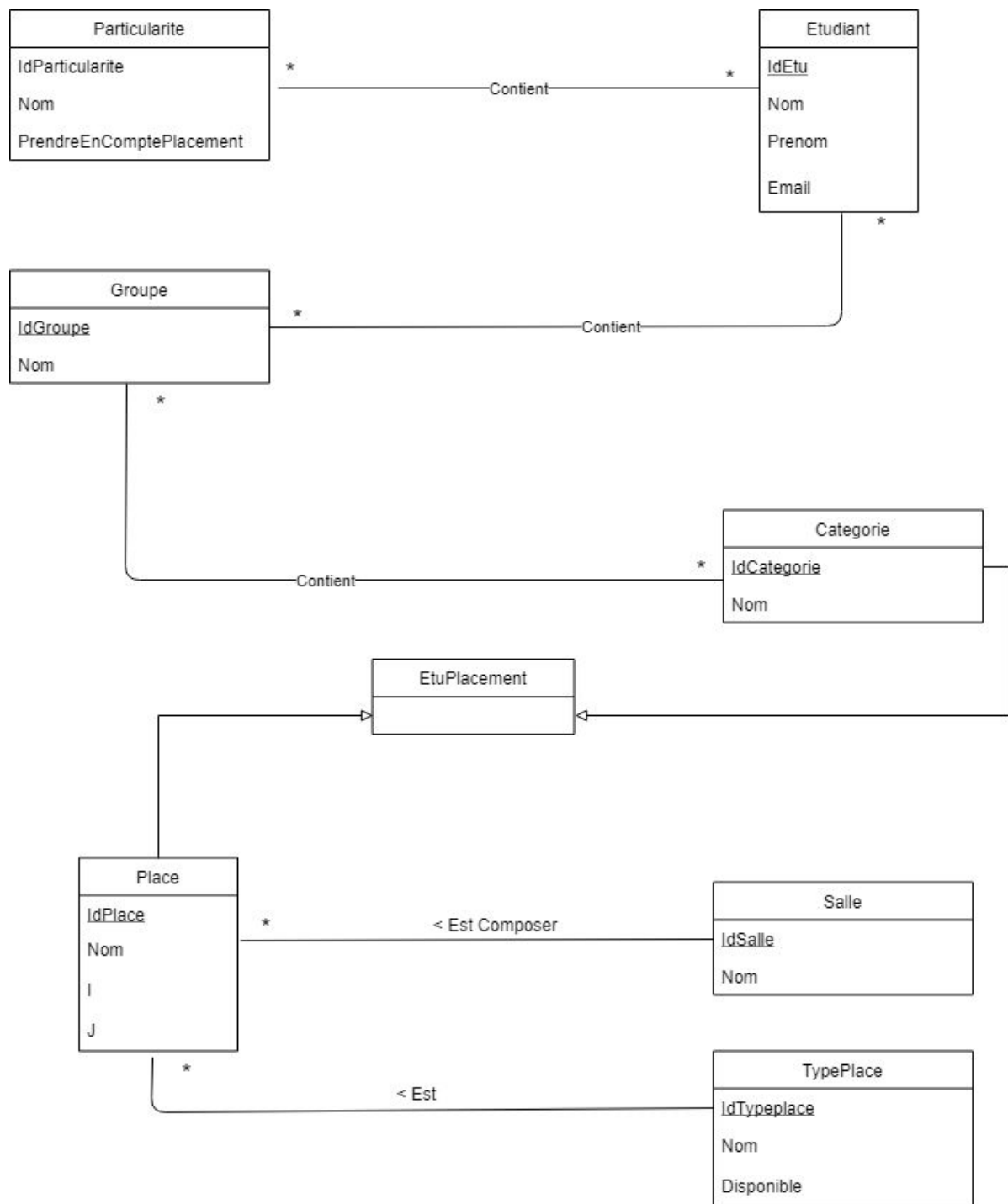
Diagramme de classe complet



On observe donc le fonctionnement interne de chaque module, de la gestion des places est de leurs types, de la sauvegarde des derniers paramètres renseignés lors de la création du dernier examen ainsi que les particularités des étudiants qui n'est pas limité à seulement 2 ou 3.

Concernant les patrons de conception utilisée, nous utilisons l'itérateur pour pouvoir naviguer de façon simple dans les salles lors de la génération du placement et le patron Active Record pour la gestion de la base de données.

Diagramme de classe Modélisation des données



Modèle relationnel

Particularite(<u>idParticularite</u> , nom, prendreEnComptePlacement) ParticulariteEtudiant(<u>#idParticularite</u> , <u>#idEtu</u>) Etudiant(<u>idEtu</u> , nom, prenom, email) EtudiantGroupe(<u>#idEtu</u> , <u>#idGroupe</u>) Groupe(<u>idGroupe</u> , nom)	GroupeCategorie(<u>#idGroupe</u> , <u>#idCategorie</u>) Categorie(<u>idCategorie</u> , nom) Place(<u>idPlace</u> , nom, <u>#idTypePlace</u> , i, j, <u>#idSalle</u> , nomColonne, nomRangee) Salle(<u>idSalle</u> , nom) TypePlace(<u>idTypePlace</u> , disponible, nom)
---	---

La partie analyse se conclut donc sur ce diagramme, concernant les évolutions depuis l'étude préalable de décembre, nous avons notifié les principaux changements lors de la présentation de chaque diagrammes, mais dans l'ensemble nous avons assez bien respecter ce que nous avons établis au départ, seul le diagramme de classe n'est pas complet et à subit de très nombreuses modifications aussi dû aux choix d'implémenter une base de données au cours de l'itération 1. Une grande partie des modifications ont concerné le module Étudiant suite à une refonte totale de ce dernier à cause d'un problème interne à l'équipe.

3. La réalisation

Au moment de passer à la réalisation nous nous sommes posées plusieurs questions, notamment sur le langage que nous allions utiliser, s'il nous fallait certaines librairies en particulier, mais aussi comment gérer la base de données.

Le logiciel étant destiné à être utilisé par le secrétariat, nous avons dû procéder à différents tests pour pouvoir rendre utilisable le logiciel.

Enfin, nous aborderons les différents problèmes que nous avons rencontrés tout au long de la réalisation.

Architecture logicielle

Pour le langage de programmation, nous avons choisi le langage JAVA en version 8 car c'est le langage que nous étudions et donc connaissons le plus, ainsi l'intégration de la base de données serait plus rapide et le modèle MVC également, nous avons donc pensé que d'étudier un nouveau langage nous ferait perdre beaucoup de temps malgré que cela nous aurait apporté une expérience en plus, nous avons aussi choisi un logiciel et non une application WEB, car nous avons pensé qu'un problème avec le réseau pourrait se produire et par conséquent rendre inutilisable l'outil. Pour les librairies, nous avons choisi Apache POI pour permettre la gestion des fichiers Excel afin d'importer des étudiants dans l'application ou encore exporter un examen. Pour la base de données, nous avons utilisé le logiciel Laragon pour le développement et Laragon portable pour l'installation au secrétariat, il s'agit d'un logiciel du même type que XAMPP. Enfin, malgré le fait que l'arrivée de la base de données soit tardive dans le projet, nous avons décidé de l'intégrer, car celle-ci offre beaucoup d'avantages comparés à de simples fichiers sauvegardés. Tout d'abord, on évite les suppressions accidentelles et surtout, on conserve l'intégrité des

données ce qui est important si le logiciel plante durant un processus, là où des fichiers aurait été corrompu et donc rendu inutilisable.

Tests de validation

Afin de garantir le bon fonctionnement du logiciel, nous avons procédé à des tests Java JUnit sur les modèles et la gestion des fichiers Excel. La majeure partie du code étant porté sur l'interface et les contrôleurs, il y a en réalité peu de tests à effectuer en dehors des modèles. La plupart de ces tests ont porté sur l'accès aux données depuis la base de données.

Nous avons aussi pris un rendez-vous au secrétariat du département informatique afin de tester le logiciel en condition réelle. Il s'agissait surtout de tester l'installation de la base de données sur des ordinateurs sans droit administratif ce qui a été un succès, cependant un conflit entre les versions Java s'est révélé être un problème que nous aborderons dans la partie des "Problèmes rencontrés".

Problèmes rencontrés

Dans la conception du projet nous avons rencontré plusieurs problèmes, tout d'abord l'intégration d'une base de données dans le projet à été la cause de beaucoup de problèmes, en effet, nous étions au début partis sur des fichiers pour la sauvegarde des étudiants, examens et salles c'est ce qui à été notre plus grande erreur lors de l'analyse du projet. Nous nous somme vite rendu compte que cela allait être difficile notamment pour conserver l'intégrité des données, mais aussi stocker les fichiers. Nous avons donc dû supprimer une grande partie des modèles déjà établis ainsi que tout le code reposant sur des fichiers et implémenté la base de données. L'implémentation fut fastidieuse est pris des semaines au lieu de quelques jours.

Le problème suivant concerne la fin du projet vers le début de la troisième itération, nous avons commencé le projet par une équipe de quatre personnes, nous le terminons à trois, en effet, un membre de l'équipe n'a pas pu finir le projet pour des raisons personnelles, nous avons dû nous adapter et reprendre le travail en cours et procéder à une nouvelle répartition des tâches, cela à été la cause de la refonte totale du module étudiant ainsi que des finitions qui n'ont pas pu être accomplies dans les temps.

Enfin, lors du test au secrétariat du département informatique, nous nous sommes rendu compte d'un problème de compatibilité entre Java 8 (version utilisée pour le développement) et Java 7 (version installée sur les ordinateurs du secrétariat). Nous avons donc dû modifier une grande partie du code et une grande partie des fonctionnalités n'ont pas pu être présentées lors de ce rendez-vous. Cependant, les problèmes rencontrés sont maintenant corrigés.

4. Conclusion

L'objectif du logiciel étant de placer des étudiants dans une salle d'examen, on peut en conclure que celui-ci est entièrement réalisé. Avec en plus la possibilité de gérer l'ensemble des salles et étudiants. Nous avons essayé de s'adapter au mieux avec les outils utilisés par le secrétariat pour faire le placement manuel, c'est pour cela que l'on peut importer des fichiers Excel et que l'examen est en fichier Excel également.

Cela a été pour nous une expérience intéressante notamment dans la gestion d'une équipe, savoir gérer les désaccords entre certains membres... Cela nous a également permis de s'améliorer en programmation JAVA et d'explorer certains aspects que l'on ne voit pas forcément dans les cours, mais aussi de mettre en application l'ensemble des connaissances acquises au sein d'un même projet.

Actuellement, nous ne pensons pas que le logiciel est apte à être repris et poursuivi l'année prochaine.