



2ème année Cycle Ingénieur Génie Informatique

Mini - projet:

Conception d'une application de gestion de l'UFR

Module:

Programmation Java

Réalisé par :

LAGHLID Ayoub 30 MANBER Chaïmae 34

Professeur:

Pr. HAFIDI Imad

ENSA Khouribga - GI2 - (2021 - 2022)

Table des matières

IN	TRODUCTION	5
SF	PECIFICATIONS :	6
	Analyse du besoin :	6
9	Spécification des besoins fonctionnels :	6
:	Spécification des besoins non fonctionnels :	7
	dentification des acteurs :	
	Cas d'Utilisation général :	
(Cas d'Utilisation détaillés et descriptions textuelles :	
	Cas d'utilisation « S'authentifier » :	. 11
	Cas d'utilisation « Gérer des formations » :	. 12
	Cas d'utilisation « Gérer personnel » :	. 14
	Cas d'utilisation « Gérer charge horaire » :	. 17
	Cas d'utilisation « Gérer Doctorats » :	. 19
	Cas d'utilisation « Gérer soutenances » :	. 20
	Cas d'utilisation « Gérer parcours scolaire » :	. 21
	Cas d'utilisation « Gérer publications » :	. 23
	Cas d'utilisation « Gérer modules » :	. 26
	Cas d'utilisation « Consulter résultats et notes » :	. 29
LE	S MAQUETTES :	. 30
l	nterface d'authentification :	30
ı	nterface Responsable UFR :	31
	nterface Étudiant :	36
	nterface Responsable de formation :	
	nterface responsable de module :	
	nterface Professeur :	
CC	DNCEPTION:	. 39
I	Dictionnaire de données :	39
	Modèle du domaine :	
RÉ	ALISATION:	. 41
	Présentation du langage JAVA:	41

Utilisation d'Eclipse :	41
Connexion à la base de données :	41
Le modèle MVC :	42
Hiérarchie du projet :	43
Préparation de la base de données :	45
Interfaces :	51
CONCLUSION	53

Remerciement:

Nous tenons bien évidemment à vous remercier, Monsieur Hafidi Imad, pour votre bienveillance, votre disponibilité pour répondre à nos nombreuses questions et pour tout l'effort que vous avez fourni lors des cours en ligne et présentiels. Nos premiers pas dans la programmation Java étaient grâce à vous.

Votre gracieuseté ainsi que le travail méticuleux que vous fournissiez sont contagieux ! C'est avec un réel plaisir que nous sommes assistés à vos cours en ligne et présentiels.

Nous avons essayé de fournir un travail particulièrement soigneux et minutieux, en espérant qu'il retiendra toute votre attention et appréciation!

INTRODUCTION

Notre travail consiste à concevoir et implémenter un nouveau service, il s'agit d'un logiciel de gestion des UFR, au profit des professeurs, doctorants, responsables, ainsi qu'étudiants. Ce système devra être en mesure de résoudre les problèmes liés à la lenteur et à l'ambigüité de traitement des tâches redondantes pour chaque année, au niveau des services administratifs de l'UFR. En effet, le service des affaires universitaires gère chaque année un grand nombre de doctorats, soutenances, formations etc... et l'absence d'un mécanisme de gestion informatique de ces dossiers rend la tâche de traitement très difficile et ennuyeuse.

Le présent rapport a pour objectif de présenter d'une façon détaillée les étapes d'élaboration de ce projet. Il s'articule autour les chapitres suivants : analyse et conception et réalisation.

SPECIFICATIONS:

Analyse du besoin :

- Le stockage des données dans des fichiers Excel augmente le risque de perte d'informations qui, à l'absence de moyens de sauvegarde et de sécurité, sont exposées aux virus informatiques, fautes de frappes et à la suppression.
- L'information est décentralisée et dispersée sur plusieurs fichiers et qui cause le problème de réplication et de redondance
- Problème de redondance : le traitement des dossiers dans plusieurs machines et par plusieurs utilisateurs cause le problème de réplication et de redondance, d'où on remarque qu'un même dossier peut être ré-saisit plusieurs fois.
- Perte de temps : la ré-saisie des données provoque une perte de temps, et pour garder l'intégrité de données, toute modification dans l'un des fichiers doit être établie et contrôlée manuellement dans les autres.
- Le responsable UFR devra être en mesure de maîtriser l'outil Excel, sinon il sera incapable de dégager les erreurs de redondance et d'exploiter les fonctionnalités qui aident à mieux gérer les données.
- Problème de confidentialité : l'absence de mécanisme de sécurité expose les données au risque de mal utilisation et nuit à la confidentialité des informations personnelles des professeurs, doctorants, étudiants, etc.

Spécification des besoins fonctionnels :

Ce projet doit satisfaire certains besoins et doit préparer l'application contenant les modules nécessaires pour l'utilisateur final. Ces tâches à effectuer sont définies comme suit :

- Authentification des utilisateurs à travers un login et un mot de passe introduit au système.
- Gestion des formations (Ajout, modification, suppression, recherche)
- Gestion du personnel (Ajout, modification, suppression, recherche).
- Gestion de la charge horaire (Affectation des modules, calcul de la charge horaire, calcul des heures sup)

- Gestion des Doctorats (Création et affectation des encadrants)
- Gestion des soutenances (Validation des soutenances, archivage des thèses)
- Gestion du parcours scolaire des étudiants (Calcul des notes, validation des modules, passage des semestres)
- Gestion des modules (Ajout, modification, suppression, recherche, affectation aux semestres d'une formation)
- Gestion des publications (Ajout, modification, suppression, recherche)
- Consultation des résultats et notes
- Partie statistiques (Visualisation des statistiques des publications, visualisation du taux de réussite par formation ou par module)

Spécification des besoins non fonctionnels :

Les besoins non fonctionnels se sont les exigences qui caractérisent un système.

Ces besoins peuvent être énoncés suivant des plans de classifications :

• Ergonomie:

Les interfaces doivent être simples, conviviales, clairs et ne doivent pas nécessiter de connaissances assez poussées des utilisateurs pour leur permettre de se sentir à l'aise et familiarisés avec l'environnement.

Robustesse:

L'application doit permettre le stockage des informations des utilisateurs inscrits, en assurant une bonne gestion d'erreurs.

• Sécurité:

L'application doit garantir à l'utilisateur connecté l'intégrité et la confidentialité de ses données. L'accès à la partie administration doit être sécurisé et authentifié.

• Performance:

Un temps de réponse raisonnable au moment de l'identification et des traitements des données. Les utilisateurs non administrateur, ne doivent pas s'apercevoir des données sécurisées.

Identification des acteurs :

Avant de réaliser les diagrammes des cas d'utilisation, il serait de bonne pratique d'identifier les acteurs en premiers. Un acteur est une entité externe qui peut être un utilisateur humain, un dispositif matériel ou un autre système.

Dans notre système nous avons identifié les acteurs suivants :

Le responsable UFR:

- Gestion des formations (Ajout, modification, suppression, recherche)
- Gestion du personnel (Ajout, modification, suppression, recherche)
- Gestion de la charge horaire (Affectation des modules, calcul de la charge horaire, calcul des heures sup)
- Gestion des Doctorats (Création et affectation des encadrants)
- Gestion des soutenances (Validation des soutenances, archivage des thèses)
- Visualisation des statistiques

Le responsable de la formation :

• Gestion des modules (Ajout, modification, suppression, recherche, affectation aux semestres d'une formation)

Le responsable du module :

• Gestion du parcours scolaire des étudiants (Calcul des notes, validation des modules, passage des semestres)

<u>Professeur/Doctorant</u>:

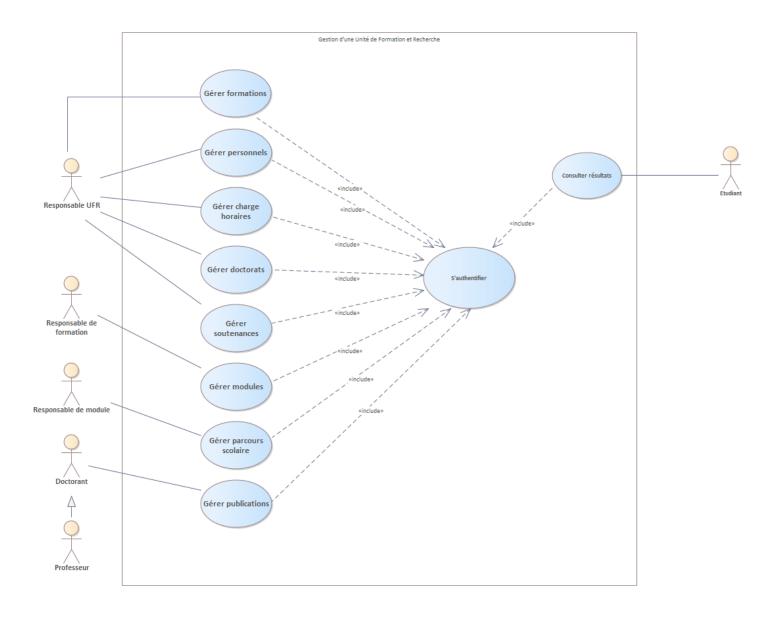
• Gestion des publications (Ajout, modification, suppression, recherche)

Étudiant :

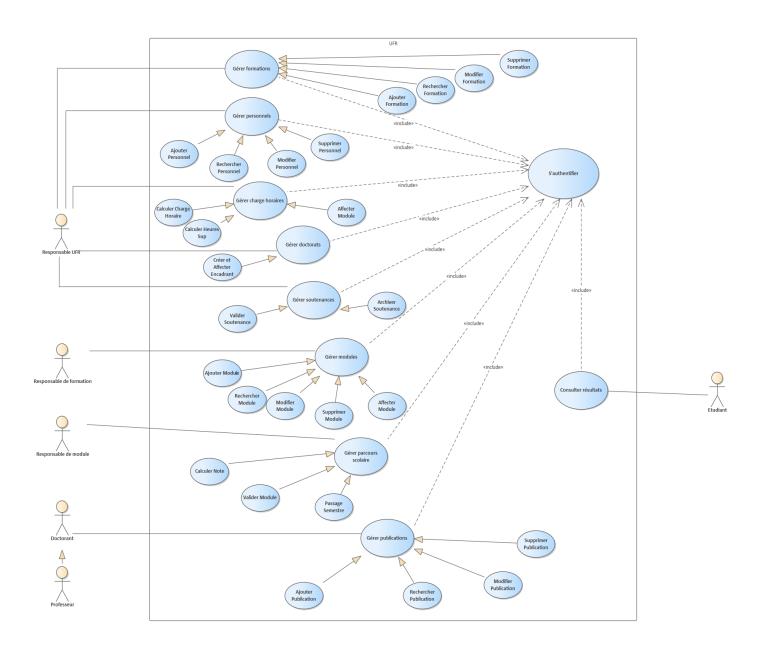
• Consultation des résultats et notes

Cas d'Utilisation général :

L'identification des diagrammes des cas d'utilisation s'avère une étape cruciale. Ces derniers décrivent d'une manière précise les besoins du client final est spécifient le comportement attendu par le système à développer.



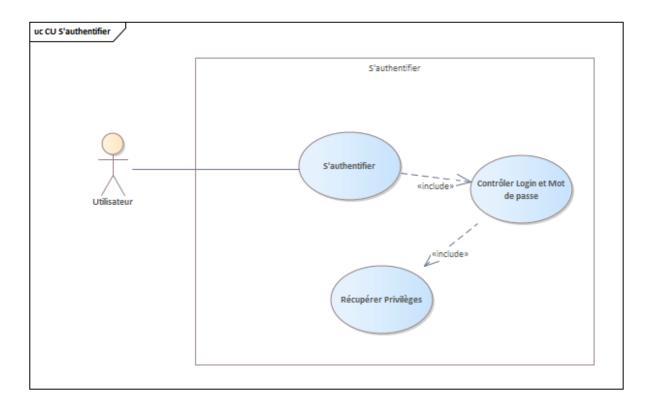
Nous avons pensé à raffiner le diagramme des cas d'utilisation encore plus. Nous obtenons le diagramme des cas d'utilisation ci-dessous :



Cas d'Utilisation détaillés et descriptions textuelles :

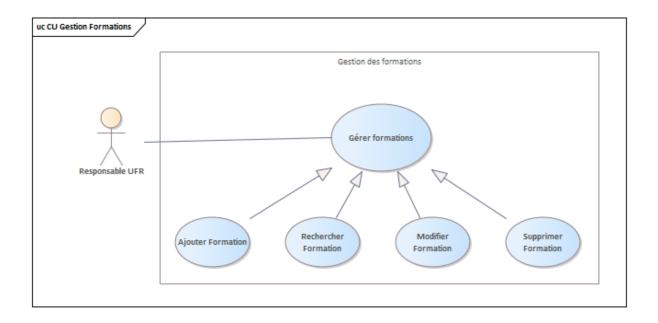
Dans ce qui suit, nous allons présenter chaque cas d'utilisation avec sa description textuelle bien détaillée.

Cas d'utilisation « S'authentifier » :



Cas d'utilisation	S'authentifier
Résumé	Ce cas d'utilisation permet au responsable UFR, responsable du module, professeur et encadrant, responsable de la formation, ainsi que l'étudiant de s'authentifier
Acteurs	Le responsable UFR, Responsable du module, Professeur et encadrant, Responsable de la formation, étudiant
Pré-condition	L'acteur doit exister dans la base de données
Post-condition	- L'acteur s'est authentifié - L'acteur est dirigé vers la page qui lui convient
Scénario nominal	 Le système affiche l'interface d'authentification. L'acteur saisit les informations de connexion (login et mot de passe). Le système vérifie les coordonnées saisies. Le système affiche l'espace propre à l'acteur L'acteur accède aux pages dont il a besoin.
Scénario alternatif	Le nom d'utilisateur ou le mot de passe erroné 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige à l'étape 2 du scenario nominal.
Scénario d'erreur	Utilisateur non existant dans la base de données.

Cas d'utilisation « Gérer des formations » :



> Description textuelle du cas d'utilisation « Ajouter formation » :

Cas d'utilisation	Ajouter formation
Résumé	Ce cas d'utilisation permet au responsable UFR d'ajouter une formation
Acteurs	Le responsable UFR
	Le responsable UFR authentifié.
Pré-condition	Formation non existante.
Post-condition	Formation ajoutée
	1. Le responsable UFR clique sur le bouton « ajouter une formation »
Scénario nominal	Le système affiche l'interface d'ajout. Le responsable UFR rempli les champs et confirme l'ajout
	4. Le système enregistre la nouvelle formation.
	Le responsable UFR a oublié de remplir tous les champs
Scénario alternatif	3.1. Le système affiche un message d'erreur.3.2. Le système le redirige au scénario 2 du scénario nominal.
	3.2. Le système le reunige au scenario 2 du scenario nominal.

➤ Description textuelle du cas d'utilisation « Modifier formation » :

Cas d'utilisation	Modifier formation
Résumé	Ce cas d'utilisation permet au responsable UFR de modifier une formation
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié. Formation existante.
Post-condition	Formation modifiée
Scénario nominal	 Le responsable UFR clique sur le bouton « modifier » Le système affiche l'interface de la modification. Le système affiche la liste des formations. Le responsable choisit la formation à modifier. Le responsable UFR modifie les informations et confirme les modifications. Le système affiche un message de confirmation. Le responsable UFR confirme la modification. Le système enregistre les nouvelles modifications et affiche un message de succès.

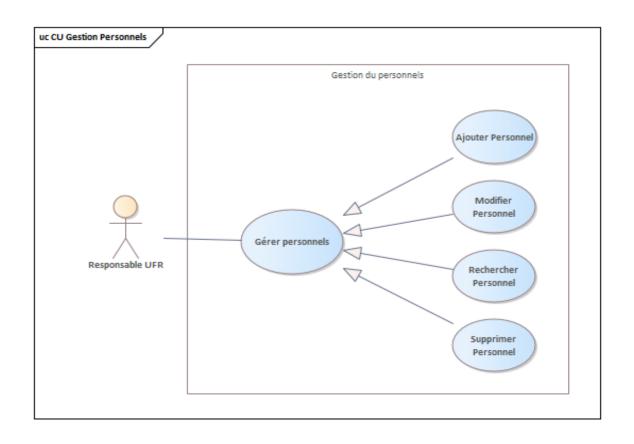
> Description textuelle du cas d'utilisation « Supprimer formation » :

Cas d'utilisation	Supprimer formation
Résumé	Ce cas d'utilisation permet au responsable UFR de supprimer une formation
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié. Formation existante.
Post-condition	Formation supprimée
Scénario nominal	 Le responsable UFR clique sur le bouton « supprimer une formation » Le système affiche l'interface de suppression des formations. Le système affiche la liste des formations. Le responsable choisit la formation à supprimer. Le système affiche un message de confirmation. Le responsable UFR confirme la suppression Le système supprime la formation.

> Description textuelle du cas d'utilisation « Rechercher formation » :

Cas d'utilisation	Rechercher formation
Résumé	Ce cas d'utilisation permet au responsable UFR de rechercher une formation
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié. Formation existante.
Post-condition	La recherche aboutit aux résultats souhaités par le responsable UFR.
Scénario nominal	 Le responsable UFR clique sur le bouton « chercher une formation » Le système affiche l'interface de la recherche. Le responsable écrit le nom de la formation à chercher dans la zone de recherche Le système cherche la formation dans la base de données. Le système affiche le résultat de la recherche.

Cas d'utilisation « Gérer personnel » :



Description textuelle du cas d'utilisation « Ajouter personnel » :

Cas d'utilisation	Ajouter personnel
Résumé	Ce cas d'utilisation permet au responsable UFR d'ajouter le personnel
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié.
Post-condition	Professeur ou doctorant ajoutés
Scénario nominal	 Le responsable UFR clique sur le bouton « gérer personnel » Le système affiche l'interface de l'ajout du personnel. Le responsable UFR rempli les champs et confirme l'ajout Le système enregistre le nouveau professeur ou doctorant.
Scénario alternatif	Le responsable UFR a oublié de remplir tous les champs 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 2 du scénario nominal.

> Description textuelle du cas d'utilisation « Modifier personnel » :

Cas d'utilisation	Modifier personnel
Résumé	Ce cas d'utilisation permet au responsable UFR de modifier les informations du personnel
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié. Professeur ou doctorant existants
Post-condition	Personnel modifié
Scénario nominal	 Le responsable UFR clique sur le bouton « gérer personnel » Le système affiche l'interface de modification du personnel. Le système affiche la liste des professeurs ou doctorants. Le responsable choisit le professeur ou doctorant à modifier. Le responsable UFR modifie les informations et confirme les modifications. Le système affiche un message de confirmation. Le responsable UFR confirme la modification. Le système enregistre les nouvelles modifications et affiche un message de succès.

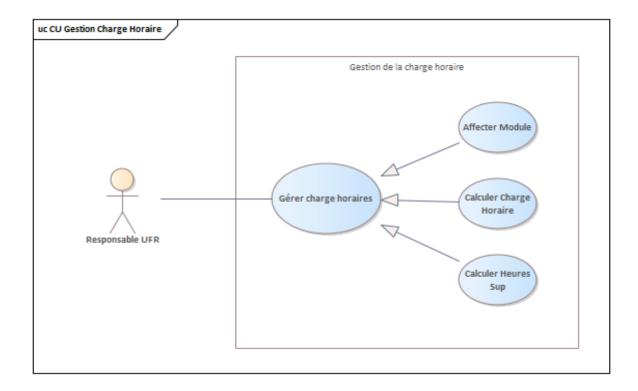
➤ Description textuelle du cas d'utilisation « Supprimer personnel » :

Cas d'utilisation	Supprimer personnel
Résumé	Ce cas d'utilisation permet au responsable UFR de supprimer le personnel
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié. Professeur ou doctorant existants
Post-condition	Personnel supprimé
Scénario nominal	 Le responsable UFR clique sur le bouton « gérer personnel » Le système affiche l'interface de suppression du personnel. Le système affiche la liste des professeurs ou doctorants. Le responsable choisit le professeur ou doctorant à supprimer. Le système affiche un message de confirmation. Le responsable UFR confirme la suppression Le système supprime la formation.

➤ Description textuelle du cas d'utilisation « Rechercher personnel » :

Cas d'utilisation	Rechercher personnel
Résumé	Ce cas d'utilisation permet au responsable UFR de rechercher le personnel
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié. Professeur ou doctorant existants
Post-condition	La recherche aboutit aux résultats souhaités par le responsable UFR.
Scénario nominal	 Le responsable UFR clique sur le bouton « gérer personnel » Le système affiche l'interface de la recherche. Le responsable écrit le nom de la formation à chercher dans la zone de recherche Le système cherche la formation dans la base de données. Le système affiche le résultat de la recherche.

Cas d'utilisation « Gérer charge horaire » :



Description textuelle du cas d'utilisation « Affecter des modules » :

Cas d'utilisation	Affecter des modules
Résumé	Ce cas d'utilisation permet au responsable UFR d'affecter des modules aux professeurs
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié. Les modules déjà existants.
Post-condition	Module affecté au professeur
Scénario nominal	 Le responsable UFR clique sur le bouton « gérer personnel » Le système affiche l'interface de l'affectation des modules. Le responsable sélectionne un professeur et choisit un module Le responsable clique sur affecter. Le système enregistre l'affectation et affiche un message de succès.
Scénario alternatif	Aucun module sélectionné 3.1. Le système affiche un message d'erreur 3.2. Retour à l'étape 2 du scénario nominal

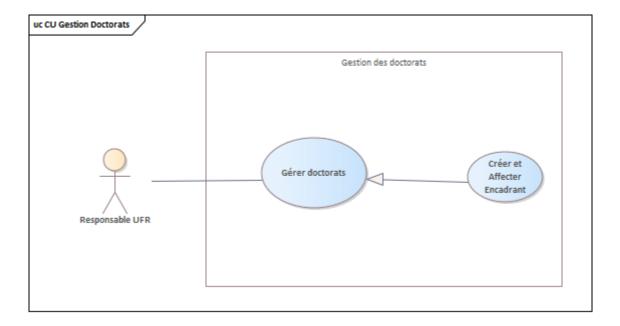
> Description textuelle du cas d'utilisation « Calculer charge horaire » :

Cas d'utilisation	Calculer charge horaire
Résumé	Ce cas d'utilisation permet au responsable UFR de calculer la charge horaire des professeurs
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié.
Post-condition	Charge horaire calculée
Scénario nominal	 Le responsable UFR clique sur le bouton « Gestion d'horaire » Le système affiche l'interface de calcul de la charge horaire. Le responsable UFR rempli les champs et clique sur calculer. Le système affiche le résultat de calcul Le responsable UFR clique sur Enregistrer. Le système enregistre et affiche un message de succès.
Scénario alternatif	Le responsable UFR a oublié de remplir tous les champs 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 2 du scénario nominal.

> Description textuelle du cas d'utilisation « Calculer heures supplémentaires » :

Cas d'utilisation	Calculer heures supplémentaires
Résumé	Ce cas d'utilisation permet au responsable UFR de calculer les heures supplémentaires des professeurs
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié.
Post-condition	Heures supplémentaires calculées
Scénario nominal	 Le responsable UFR clique sur le bouton « Gestion d'horaire » Le système affiche l'interface de calcul des heures supplémentaires. Le responsable UFR rempli les champs et clique sur calculer. Le système affiche le résultat de calcul Le responsable UFR clique sur Enregistrer. Le système enregistre et affiche un message de succès.
Scénario alternatif	Le responsable UFR a oublié de remplir tous les champs 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 3 du scénario nominal.

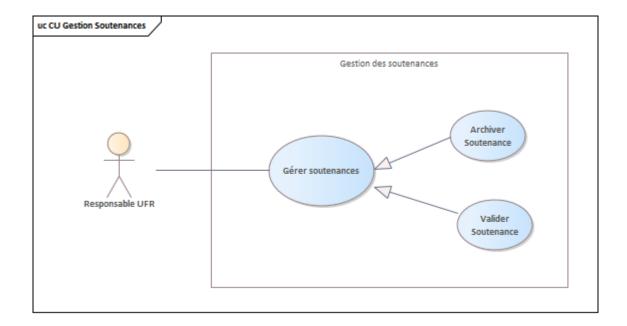
Cas d'utilisation « Gérer Doctorats » :



> Description textuelle du cas d'utilisation « Affecter des encadrants » :

Cas d'utilisation	Affecter des encadrants
Résumé	Ce cas d'utilisation permet au responsable UFR d'affecter des encadrants aux doctorants
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié.
Post-condition	Encadrant affecté
Scénario nominal	 Le responsable UFR clique sur le bouton « affectation » Le système affiche l'interface de l'affectation. Le responsable UFR demande la liste des encadrants Le système affiche la liste des encadrants. Le responsable choisit un encadrant Le système affiche la liste des doctorants. Le responsable UFR choisit un doctorant et valide. Le système enregistre l'affectation et affiche un message de succès.
Scénario alternatif	Aucun encadrant sélectionné 7.1. Le système affiche un message d'erreur 7.2. Retour à l'étape 4 du scénario nominal

Cas d'utilisation « Gérer soutenances » :



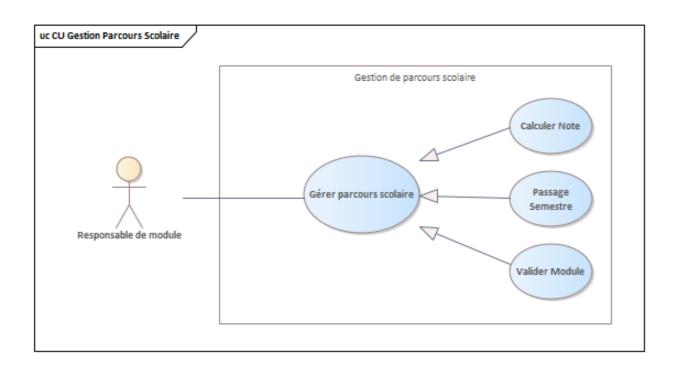
> Description textuelle du cas d'utilisation « Valider une soutenance » :

Cas d'utilisation	Valider une soutenance
Résumé	Ce cas d'utilisation permet au responsable UFR d'attribuer une décision aux soutenances
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié.
Post-condition	Validation de la soutenance
Scénario nominal	 Le responsable UFR clique sur le bouton « valider soutenance » Le système affiche l'interface relative à la validation des soutenances. Le système affiche la liste des doctorants. Le responsable UFR choisit un doctorant Le responsable UFR remplit le formulaire de validation et valide l'action Le système enregistre la validation dans la base de données
Scénario alternatif	Aucun doctorant sélectionné 4.1. Le système affiche un message d'erreur 4.2. Retour à l'étape 3 du scénario nominal Le responsable UFR a oublié de remplir tous les champs 5.1. Le système affiche un message d'erreur. 5.2. Le système le redirige au scénario 3 du scénario nominal.

> Description textuelle du cas d'utilisation « Archiver les thèses » :

Cas d'utilisation	Archiver les thèses
Résumé	Ce cas d'utilisation permet au responsable UFR d'archiver les thèses
Acteurs	Le responsable UFR
Pré-condition	Le responsable UFR authentifié. Thèse existante dans la base de données.
Post-condition	Thèse archivée
Scénario nominal	 Le responsable UFR clique sur le bouton « Archiver thèses » Le système affiche l'interface relative à l'archivage des thèses. Le système affiche la liste des thèses. Le responsable UFR sélectionne une thèse. Le responsable UFR clique sur le bouton « archiver ». Le système archive la thèse et affiche un message de succès
Scénario alternatif	Aucune thèse sélectionnée 4.1. Le système affiche un message d'erreur 4.2. Retour à l'étape 3 du scénario nominal

Cas d'utilisation « Gérer parcours scolaire » :



> Description textuelle du cas d'utilisation « Valider module » :

Cas d'utilisation	Valider module
Résumé	Ce cas d'utilisation permet au responsable du module de valider un module
Acteurs	Le responsable du module
Pré-condition	Le responsable du module authentifié. Module déjà existant
Post-condition	Validation du module
Scénario nominal	 Le responsable du module clique sur le bouton « Gestion parcours scolaire » Le système affiche l'interface de gestion du parcours scolaire. Le responsable du module rempli les champs et attribue une décision au module. Le responsable du module clique sur Enregistrer. Le système enregistre et affiche un message de succès.
Scénario alternatif	Le responsable du module a oublié de remplir tous les champs 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 2 du scénario nominal.

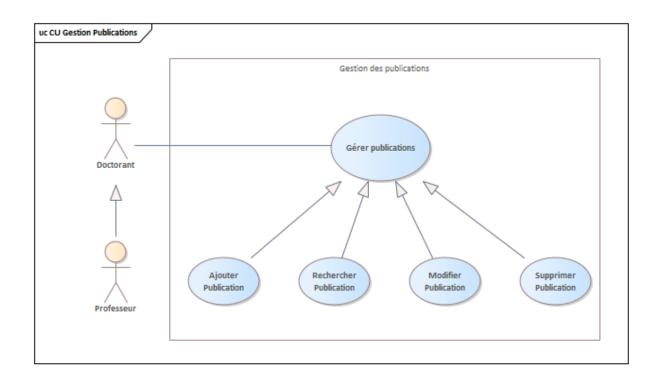
> Description textuelle du cas d'utilisation « Calculer la note » :

Cas d'utilisation	Calculer la note
Résumé	Ce cas d'utilisation permet au responsable du module de calculer la note d'un étudiant
Acteurs	Le responsable du module
Pré-condition	Le responsable du module authentifié. Modules déjà existants
Post-condition	Note calculée
Scénario nominal	 Le responsable du module clique sur le bouton « Gestion parcours scolaire » Le système affiche l'interface de calcul de la note. Le responsable du module rempli les champs des modules. Le système affiche le résultat de calcul Le responsable du module clique sur Enregistrer. Le système enregistre et affiche un message de succès.
Scénario alternatif	Le responsable du module a oublié de remplir tous les champs 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 2 du scénario nominal.

> Description textuelle du cas d'utilisation « Passage semestre » :

Cas d'utilisation	Passage semestre
Résumé	Ce cas d'utilisation permet au responsable du module de valider le semestre d'un étudiant
Acteurs	Le responsable du module
Pré-condition	Le responsable du module authentifié.
Post-condition	Validation du semestre
Scénario nominal	 Le responsable du module clique sur le bouton « Gestion parcours scolaire » Le système affiche l'interface de gestion du parcours scolaire. Le responsable du module rempli le champs du résultat du semestre en attribuant une décision. Le responsable du module clique sur Enregistrer. Le système enregistre et affiche un message de succès.
Scénario alternatif	Le responsable du module a oublié de remplir le champs 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 2 du scénario nominal.

Cas d'utilisation « Gérer publications » :



> Description textuelle du cas d'utilisation « Ajouter publication » :

Cas d'utilisation	Ajouter publication
Résumé	Ce cas d'utilisation permet au professeur ou doctorant d'ajouter une publication
Acteurs	Le professeur ou doctorant
Pré-condition	Professeur ou doctorant authentifié.
Post-condition	Publication ajoutée
Scénario nominal	 Le responsable de formation clique sur le bouton « gérer des publications » Le système affiche l'interface de l'ajout des publications. Le professeur ou doctorant importe une publication ou rempli le champs de publication et confirme l'ajout Le système enregistre la nouvelle publication.
Scénario alternatif	Le responsable de formation a oublié d'importer une publication et enregistre 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 2 du scénario nominal.

> Description textuelle du cas d'utilisation « Modifier publication » :

Cas d'utilisation	Modifier publication
Résumé	Ce cas d'utilisation permet au professeur ou doctorant de modifier une publication
Acteurs	Le professeur ou doctorant
Pré-condition	Professeur ou doctorant authentifié. Modules existants
Post-condition	Publication modifiée
Scénario nominal	 Le professeur ou doctorant clique sur le bouton « gérer des publications » Le système affiche l'interface de gestion des publications Le système affiche la liste des publications. Le professeur ou doctorant choisit la publication à modifier. Le professeur ou doctorant modifie les informations et confirme les modifications. Le système affiche un message de confirmation. Le professeur ou doctorant confirme la modification. Le système enregistre les nouvelles modifications et affiche un message de succès.

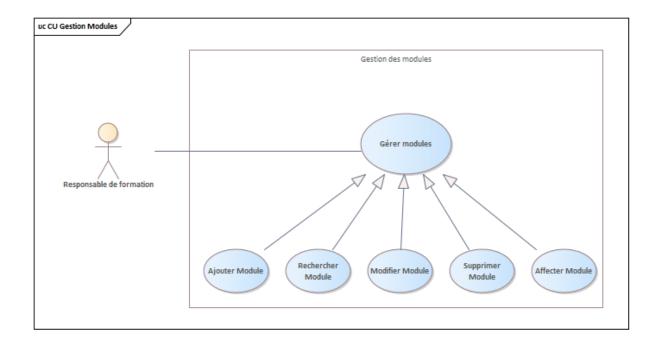
> Description textuelle du cas d'utilisation « Supprimer publication » :

Cas d'utilisation	Supprimer publication
Résumé	Ce cas d'utilisation permet au professeur ou doctorant de supprimer une publication
Acteurs	Le professeur ou doctorant
Pré-condition	Professeur ou doctorant authentifié. Modules existants
Post-condition	Publication supprimée
Scénario nominal	 Le professeur ou doctorant clique sur le bouton « gérer des publications » Le système affiche l'interface de suppression des publications. Le système affiche la liste des publications. Le professeur ou doctorant choisit la publication à supprimer. Le système affiche un message de confirmation. Le professeur ou doctorant confirme la suppression Le système supprime la formation.

> Description textuelle du cas d'utilisation « Rechercher publication » :

Cas d'utilisation	Rechercher publication
Résumé	Ce cas d'utilisation permet au professeur ou doctorant de rechercher une publication
Acteurs	Le professeur ou doctorant
Pré-condition	Professeur ou doctorant authentifié. Modules existants
Post-condition	La recherche aboutit aux résultats souhaités par le responsable UFR.
Scénario nominal	 Le professeur ou doctorant clique sur le bouton « gérer des publications » Le système affiche l'interface de la recherche. Le professeur ou doctorant écrit le nom de la publication à chercher dans la zone de recherche Le système cherche la formation dans la base de données. Le système affiche le résultat de la recherche.

Cas d'utilisation « Gérer modules » :



> Description textuelle du cas d'utilisation « Ajouter module » :

Cas d'utilisation	Ajouter module				
Résumé	Ce cas d'utilisation permet au responsable de formation d'ajouter un module				
Acteurs	Le responsable de formation				
Pré-condition	Le responsable de formation authentifié.				
Post-condition	Module ajouté				
1. Le responsable de formation clique sur le bouton « gérer des module 2. Le système affiche l'interface de l'ajout des modules. 3. Le responsable de formation rempli les champs et confirme l'ajout 4. Le système enregistre le nouveau module.					
Le responsable de formation a oublié de remplir tous les champs 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 2 du scénario nominal.					

> Description textuelle du cas d'utilisation « Modifier module » :

Cas d'utilisation	Modifier module			
Résumé	Ce cas d'utilisation permet au responsable de formation de modifier un module			
Acteurs	Le responsable de formation			
Pré-condition	Le responsable de formation authentifié. Modules existants			
Post-condition	Module modifié			
Scénario nominal	 Le responsable de formation clique sur le bouton « gérer des modules » Le système affiche l'interface de modification des modules. Le système affiche la liste des modules. Le responsable choisit le module à modifier. Le responsable de formation modifie les informations et confirme les modifications. Le système affiche un message de confirmation. Le responsable de formation confirme la modification. Le système enregistre les nouvelles modifications et affiche un message de succès. 			

> Description textuelle du cas d'utilisation « Supprimer module » :

Cas d'utilisation	Supprimer module				
Résumé	Ce cas d'utilisation permet au responsable de formation de supprimer un module				
Acteurs	Le responsable de formation				
Pré-condition	Le responsable de formation authentifié. Modules existants				
Post-condition	Module supprimé				
Scénario nominal	1. Le responsable de formation clique sur le bouton « gérer modules » 2. Le système affiche l'interface de suppression des modèles. 3. Le système affiche la liste des modules. 4. Le responsable choisit le module à supprimer. 5. Le système affiche un message de confirmation. 6. Le responsable de formation confirme la suppression 7. Le système supprime la formation.				

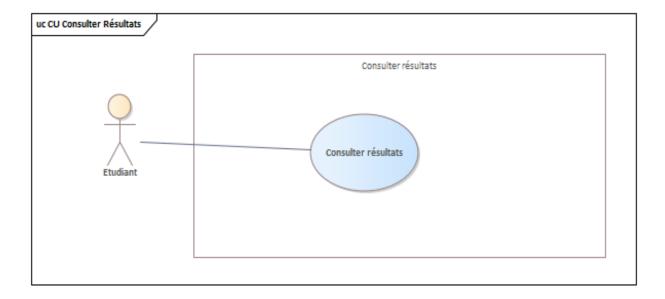
• Description textuelle du cas d'utilisation « Rechercher module » :

Cas d'utilisation	Rechercher module			
Résumé	Ce cas d'utilisation permet au responsable de formation de rechercher un module			
Acteurs	Le responsable de formation			
Pré-condition	Le responsable de formation authentifié. Modules existants			
Post-condition	La recherche aboutit aux résultats souhaités par le responsable UFR.			
1. Le responsable de formation clique sur le bouton « gérer modules » 2. Le système affiche l'interface de la recherche. 3. Le responsable écrit le nom de la formation à chercher dans la zone or recherche 4. Le système cherche la formation dans la base de données. 5. Le système affiche le résultat de la recherche.				

> Description textuelle du cas d'utilisation « Affectation au semestre d'une formation » :

Cas d'utilisation	Affectation au semestre d'une formation				
Résumé	Ce cas d'utilisation permet au responsable de formation d'affecter des modules au semestre d'une formation				
Acteurs	Le responsable de formation				
Pré-condition	Le responsable de formation authentifié. Modules existants Formations existantes				
Post-condition	Module affecté au semestre				
Scénario nominal	 Le responsable de formation clique sur le bouton « gérer des modules » Le système affiche l'interface de gestion des formations. Le responsable de formation rempli les champs et confirme l'affectation Le système enregistre. 				
Le responsable UFR a oublié de remplir tous les champs 3.1. Le système affiche un message d'erreur. 3.2. Le système le redirige au scénario 2 du scénario nominal.					

Cas d'utilisation « Consulter résultats et notes » :



> Description textuelle du cas d'utilisation « Consulter les notes » :

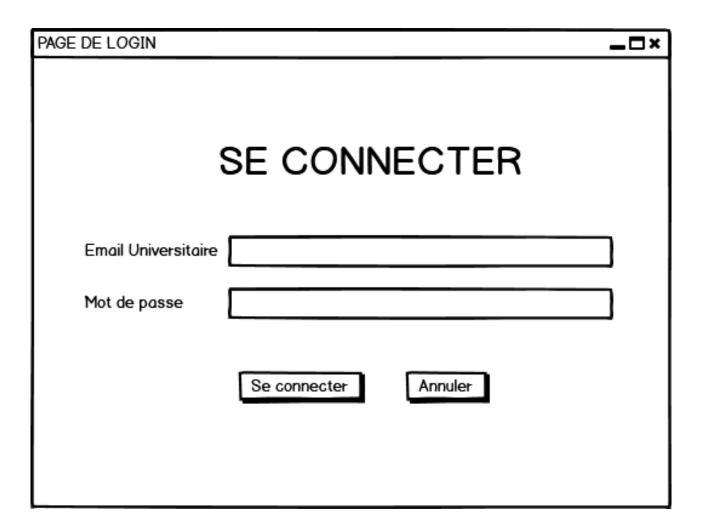
Cas d'utilisation	Consulter les notes				
Résumé	Ce cas d'utilisation permet à l'étudiant de consulter ses notes et résultats				
Acteurs	L'étudiant				
Pré-condition	L'étudiant authentifié				
Post-condition	La liste des notes est affichée sur l'écran				
Scénario nominal	ominal 1. Le système affiche la liste des notes.				
Scénario alternatif Aucun résultat 1.1. Le système affiche un message de type « aucun résultat trouvé »					

LES MAQUETTES:

Nous consacrons cette partie pour exposer notre projet à travers des imprimes d'écran des interfaces de notre application.

Interface d'authentification:

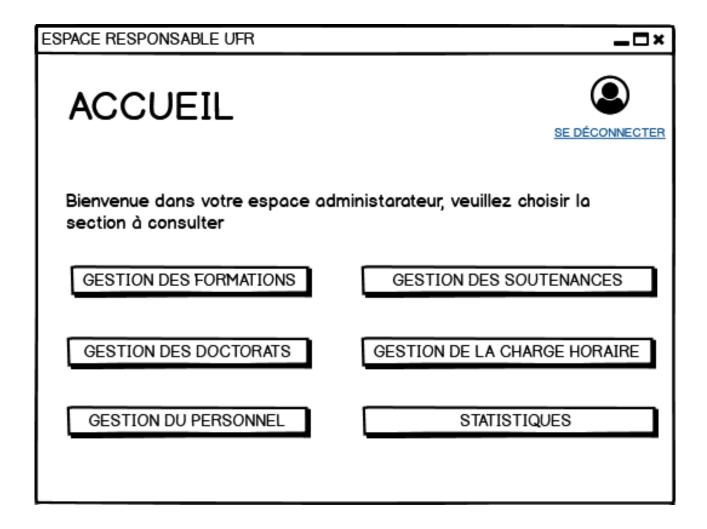
Chaque utilisateur (Responsable UFR, responsable de la formation, responsable du module, Professeur ou doctorant, étudiant), doit s'authentifier avant d'accéder aux déférentes fonctionnalités. L'authentification est assurée par l'interface illustrée par la figure ci-dessous.



Interface Responsable UFR:

Acceuil:

L'interface accueil s'affiche directement après que le responsable UFR s'authentifie. Elle regroupe toutes les opérations de gestion que le responsable peut effectuer. Par exemple, il suffit qu'il appuie sur le bouton « Gestion formations » pour que l'interface relative à la gestion des formations s'affiche.

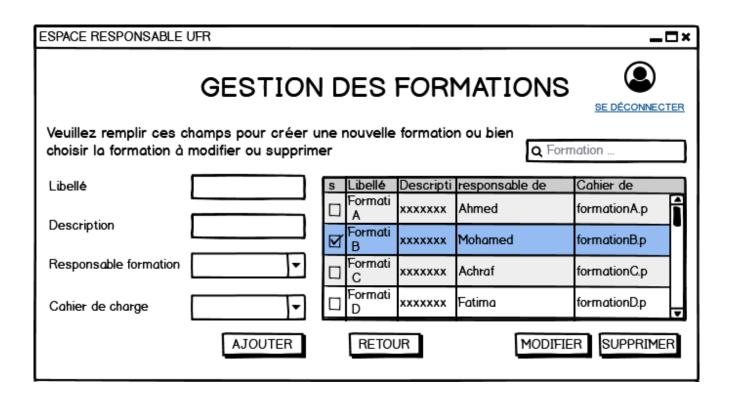


Gestion des formations :

Après avoir cliqué sur le bouton « Gestion des formations », l'interface cidessous s'affiche. Dans cette étape, le responsable UFR a la possibilité de créer, modifier, supprimer ou même rechercher une formation.

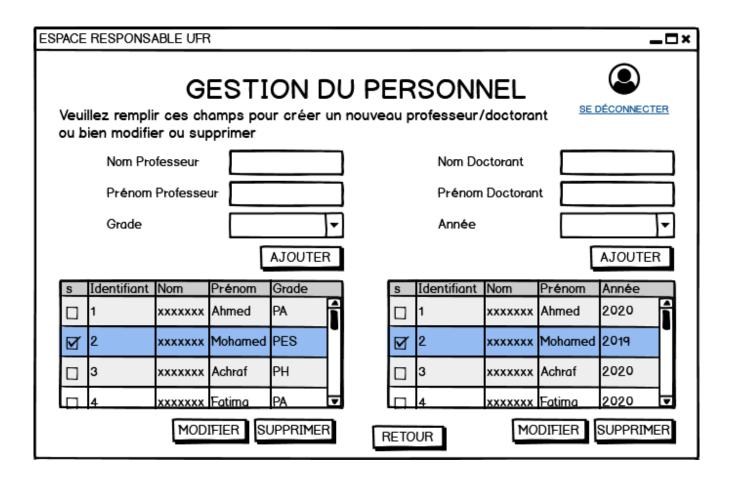
Pour la création d'une formation, le responsable UFR doit remplir tous les champs relatifs à cette nouvelle formation à savoir le libellé, la description, le nom du responsable de la formation ainsi que le cahier de charge. L'appui sur le bouton Ajouter entraîne simultanément l'enregistrement de la formation dans la base de données.

Pour la modification d'une formation, il suffit de sélectionner la formation qu'on souhaite modifier et puis appuyer sur le bouton modifier. La même chose pour la suppression.



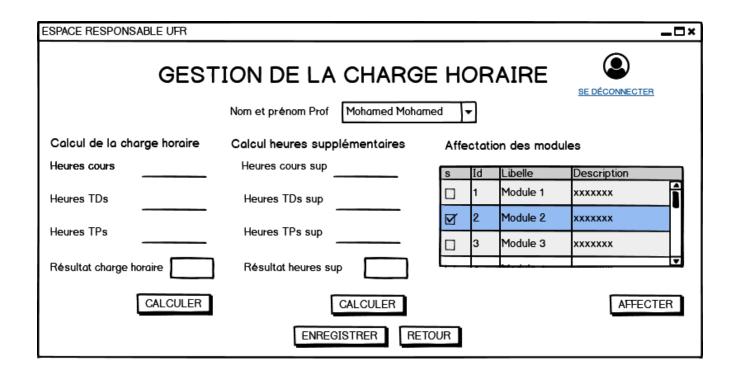
Gestion du personnel

Le responsable UFR obtient cette interface à partir du bouton « gestion du personnel ». Elle lui permet soit de créer un nouveau professeur ou doctorant en remplissant les champs et d'appuyer par la suite sur ajouter afin d'enregistrer, ou bien de choisir un professeur ou doctorant à partir des tableaux affichés sur l'interface et appuyer sur modifier ou supprimer.



Gestion de la charge horaire :

Cette interface est obtenue lorsque le responsable UFR appuie sur le bouton « Gestion de la charge horaire ». Elle permet soit de calculer la charge horaire, les heures supplémentaires, pour un professeur, en remplissant tous les champs appropriés et puis en appuyant sur le bouton « calculer ». Le résultat s'affiche immédiatement. Ou bien, d'affecter des modules à un professeur en sélectionnant un module et puis appuyer sur le bouton affecter.



Gestion des Doctorats:

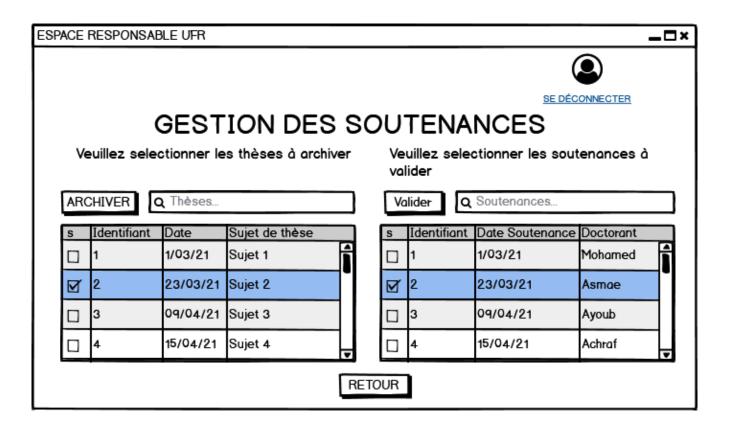
Cette interface est obtenue lorsque le responsable UFR appuie sur le bouton « Gestion des Doctorats ». Il suffit de sélectionner un encadrant depuis la liste des encadrants et puis sélectionner un doctorant pour les affecter.

ESPACE RESPONSAB	BLE UFR			×		
	GESTION DE Veuillez choisir un en		051	DÉCONNECTER		
Q Encadrant		Q Do	ctorant			
S	Nom encadrant	S	Nom doctorant	1		
	Mohamed		Ihssane			
\boxtimes	Said		Malika			
	Asmaa		Soufiane			
	Mounir		Chaimae			
AFFECTER						

Cette interface est obtenue lorsque le responsable UFR appuie sur le bouton « Gestion des Soutenances ». Elle permet d'effectuer deux taches :

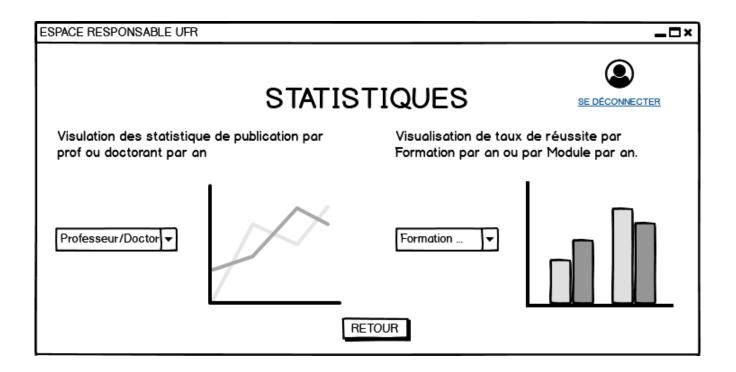
- Archiver des thèses en sélectionnant une thèse depuis la liste des thèses et appuyant sur le bouton « archiver »
- Valider des soutenances en sélectionnant une soutenance depuis la liste des thèses et appuyant sur le bouton « Valider ».

Toutes les soutenances et thèses sont recueillies dans un tableau.



Statistiques

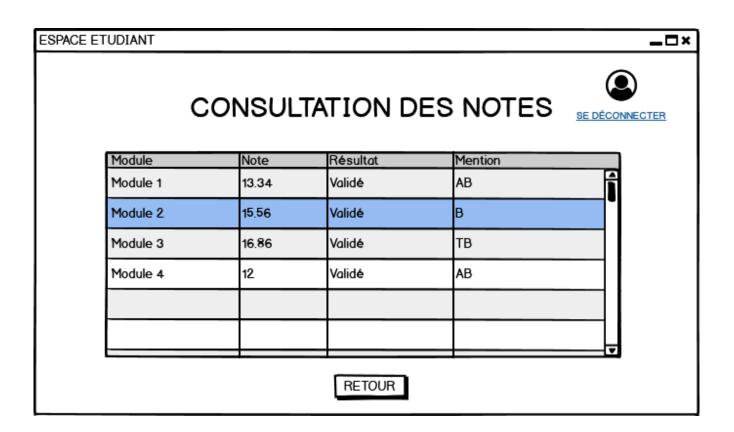
Cette interface est obtenue lorsque le responsable UFR appuie sur le bouton « Statistiques ». Elle comporte deux sections : Une section pour la visualisation des statistiques des publications par prof ou doctorant par an, et une autre pour la visualisation du taux de réussite par formation par an ou par module par an.



Interface Étudiant :

Consultation des notes :

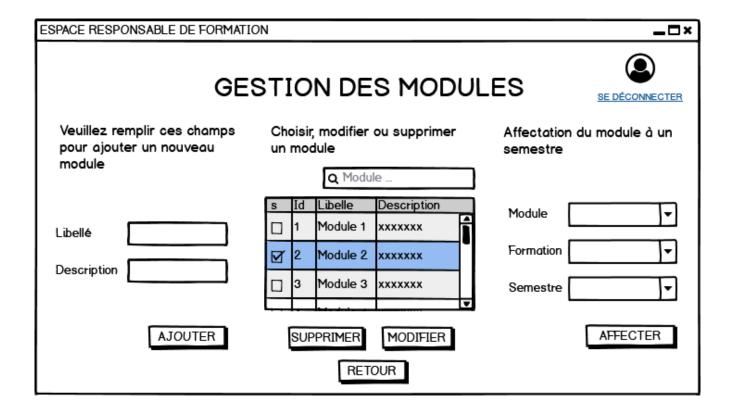
L'interface « Consultation des notes » s'affiche directement après que l'étudiant s'authentifie. Elle permet aux étudiants de consulter toutes leurs notes.



Interface Responsable de formation :

Gestion des modules :

L'interface « Gestion des notes » s'affiche directement après que le responsable de formation s'authentifie. Elle lui permet d'ajouter, modifier ou supprimer un module, ainsi que d'affecter un module à un semestre au moyen du bouton affecter.

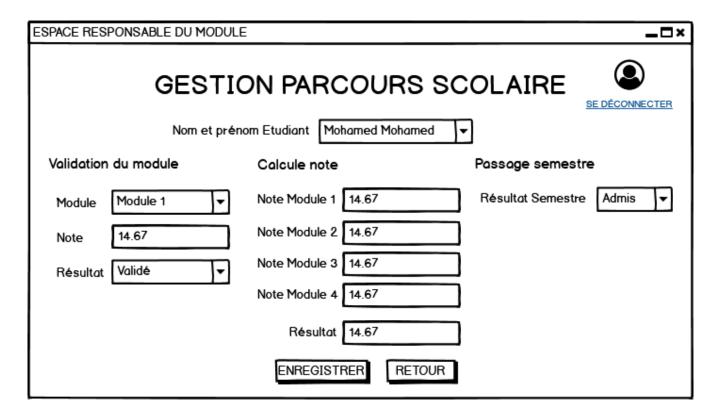


Interface responsable de module :

Gestion parcours scolaire:

Cette interface est obtenue lorsque le responsable de formation s'authentifie.

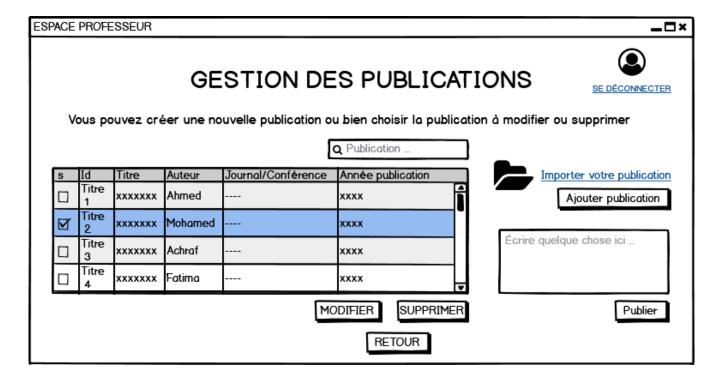
Elle lui permet de valider un module, calculer la note, et indiquer le résultat du semestre pour un étudiant choisi.



Interface Professeur:

Gestion publications:

L'interface « Gestion des publications » s'affiche directement après que le professeur ou encadrant s'authentifie. La liste regroupe toutes les publications. Le professeur ou encadrant a la possibilité soit de modifier ou supprimer la publication sélectionnée. Il peut aussi ajouter une publication en l'important, ou bien de l'écrire dans la zone de texte appropriée et par la suite appuyer sur le bouton publier.

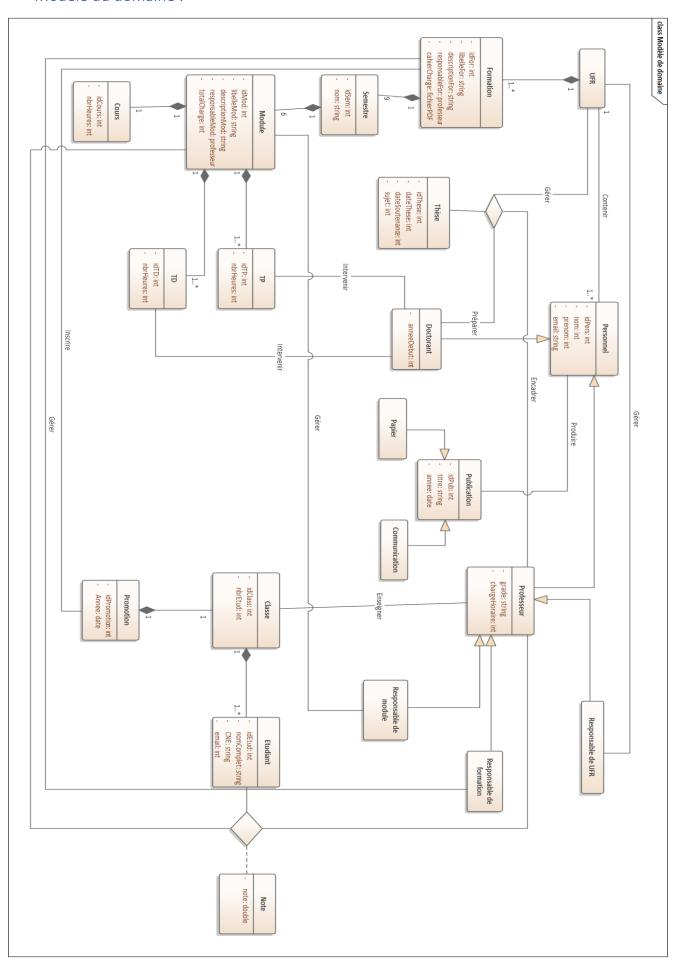


CONCEPTION:

Dictionnaire de données :

Nom de donnée	Format	Long ueur	Type	Règle de calcule	Table
idFor libelleFor descriptionFor responsableFor cahierCharge	Numérique Alphabétique Alphabétique Professeur Fichierpdf	5 10 60	Elémentaire Elémentaire Elémentaire Elémentaire Elémentaire	- - - -	Formation
idSem nom	Numérique Alphabétique	5 2	Elémentaire Elémentaire	- -	Semestre
idMod libelleMod descriptionMod responsableMod totalCharge	Numérique Alphabétique Alphabétique Professeur Numérique	5 10 60 2 5	Elémentaire Elémentaire Elémentaire Elémentaire Calculable	- - - heuresCours x 1.5 + heuresTD + heures TP x 0,75	Module
idCours nbrHeures	Numérique Numérique	5 2	Elémentaire Elémentaire	- -	Cours
idCours nbrHeures	Numérique Numérique	5 2	Elémentaire Elémentaire	- -	TP
idCours nbrHeures	Numérique Numérique	5 2	Elémentaire Elémentaire		TD
note	Numérique	2	Elémentaire	-	Note
idPromotion annee	Numérique Date	1	Elémentaire Elémentaire		Promotion
idPers nom prenom email	Numérique Alphabétique Alphabétique Alphanumérique	5 20 20 20	Elémentaire Elémentaire Elémentaire Elémentaire	- - -	Personnel
grade chargeHoraire	Alphabétique Numérique	2 3	Elémentaire Calculable	heuresCours x 1.5 + heuresTD + heures TP x 0,75	Professeur
idClass nbrEtud	Numérique Numérique	2 2	Elémentaire Elémentaire		Classe
idEtud nomComplet CNE email	Numérique Alphabétique Alphanumérique Alphanumérique	4 20 10 20	Elémentaire Elémentaire Elémentaire Elémentaire	- - -	Etudiant
idThese dateThese dateSoutenance sujet	Numérique Date Date Alphabétique	4 30	Elémentaire Elémentaire Elémentaire Elémentaire	- - -	Thèse
anneeDebut	Date		Elémentaire	-	Doctorant

Modèle du domaine :



RÉALISATION:

Présentation du langage JAVA:

Développé par Sun Microsystems depuis la fin des années 1980, JAVA est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Il existe 2 types de programmes en JAVA : Les applets et les applications. Une application autonome (stand alone program) est une application qui s'exécute sous le contrôle direct du système d'exploitation. Un applet est une application qui est chargée par un navigateur et qui est exécutée sous le contrôle d'un plug in de ce dernier. Dans notre projet, nous allons nous intéresser à la réalisation d'une application autonome.

Utilisation d'Eclipse:

Nous avons choisi de développer ce projet sur Eclipse car nous l'avions utilisé durant nos réalisations des Tps et souhaitions en poursuivre la découverte.

L'auto complétion, la validation du code en cours de frappe, ainsi que l'exécution en mode débogage nous ont permis de suivre finement l'exécution du programme.

Connexion à la base de données :

Lorsqu'il s'agit de stocker les données dans une base de données, nous avons besoin de connecter l'application avec MySQL Database.

Pour ce faire, nous devons suivre ces 5 étapes suivantes contenues dans la classe **Connexion.java** de notre projet, qui est dans le package **persistance** :

- driverClassName: Le Driver Class Name pour la base de données MySQL est com.mysql.jdbc.Driver.
- ConnectionUrl: l'URL qui contient le chemin vers la base de données est jdbc:mysql://localhost:3308/projet_laghlid_manber. Jdbc est l'API, mysql est la base de données, localhost est le nom de serveur sur lequel s'exécute mysql, on peut aussi utiliser l'adresse IP, 3308 est le numéro de port et projet laghlid manber est le nom de la base de données.
- dbUser: Le username par défaut est root.
- dbPwd: Nous avons laissé ce champs vide.

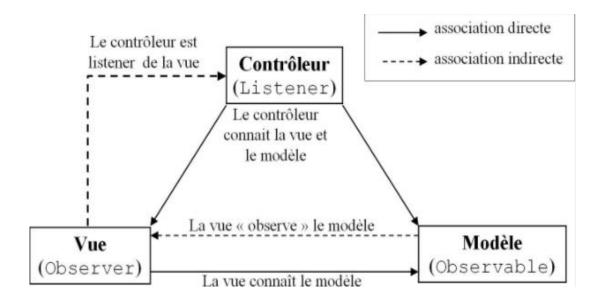
```
☑ Connexion.java 

☒
  package Persistance;
  3⊕ import java.sql.Connection; ...
     public class Connexion {
         final static String driverClassName="com.mysql.jdbc.Driver";
    final static String connectionUrl="jdbc:mysql://localhost:3308/projet_laghlid_manber";
    final static String dbUser="root";
    final static String dbPwd="";
10
11
13Θ
             public Connexion(){
14
             public static Connection getConnexion(){
                  Connection conn=null;
18
19
                     conn=DriverManager.getConnection(connectionUrl,dbUser,dbPwd);
                } catch (SQLException e) {
    // TODO Auto-generated catch block
                     e.printStackTrace();
                   return conn:
```

Le modèle MVC:

Pour que l'application ne soit pas complexe, ainsi que les interfaces correspondantes, nous avons pensé qu'il serait souhaitable de séparer les classes en :

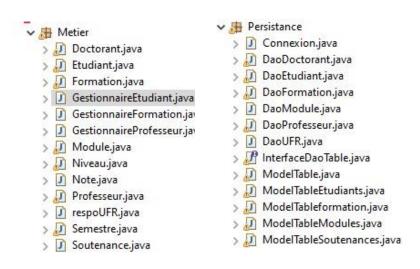
- Classes des interfaces graphiques : elles contiennent la description de l'interface graphique avec ses différents composants ; c'est la vue.
- Classes de l'application elles-mêmes : elles contiennent tous les attributs et méthodes qui doivent participer à la tâche principale à accomplir : c'est le modèle.
- Classes de contrôle qui contiennent l'ensemble des listeners et qui, lorsque des événements parviennent via la vue, préviennent le modèle en conséquence : c'est le contrôleur.



Hiérarchie du projet :

Notre projet est composé de 3 packages à savoir Metier, Persistance et Présentation. Le modèle Dao nous permet de lier les données aux objets ainsi que de séparer la couche métier de la couche d'accès aux données. Voici quelques exemples de classes tirés de notre projet :

- Metier : Regroupe toutes les classes métiers et gestionnaires. Exemples de classes : Etudiant, Doctorant, Enseignant, Formation ... etc.
- Persistance: Contient les classes DAO tel que chaque classe contient les méthodes suivantes: add, delete, selectById, update, getAll, isExiste... etc.
- Présentation : Contient le contrôleur et les classes des interfaces. Par exemple : Login, FenetreDoctorant, FenetreModule, FenetreFormation...



Voici une schématisation simple expliquant le fonctionnement du modèle DAO :



Exemple classe metier Etudiant:

Exemple classe metier Etudiant:

```
package Metier;
pimport Persistance.DaoEtudiant;

import java.util.ArrayList;

public class GestionnaireEtudiant {
    static DaoEtudiant etudiant;

public GestionnaireEtudiant() {
    etudiant = new DaoEtudiant();
}

public static boolean delete(Etudiant table) {
    return etudiant.delete(table);
}

public static boolean add(Etudiant table) {
    return etudiant.add(table);
}

public static boolean isExist(Etudiant table) {
    return etudiant.isExiste(table);
}

public static boolean isExist(Etudiant table) {
    return etudiant.isExiste(table);
}

public static Etudiant searchById(String o) {
    return (Etudiant) etudiant.searchById(o);
}

public static Etudiant) etudiant.searchById(o);
}

public static ArrayListcEtudiant> getAll() {
    return etudiant.getAll();
}
```

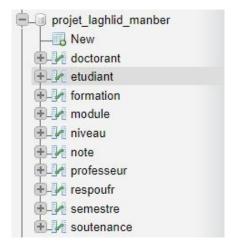
Préparation de la base de données :

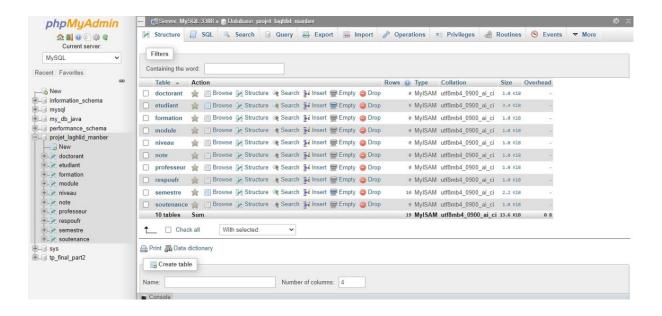
Une base de données bien conçue garantit la cohérence de l'entrée et de l'extraction des données et réduit les doublons dans les tables de base de données.

Les tables d'une base de données relationnelle sont utilisées conjointement pour permettre d'accéder aux données appropriées lorsque nous en avons besoin. Il nous est avéré judicieux de commencer par préparer la base de données en premier.

Les tables nécessaires à la réalisation de notre projet :

- Etudiant
- Professeur
- respoUFR
- niveau
- module
- formation
- semestre
- doctorant
- soutenance
- note





Création des tables :

```
Create table etudiant (
    nummerEtudiant INT NOT NULL,
    idNiveau INT NOT NULL,
   nom VARCHAR(20) NOT NULL,
    email VARCHAR(20) NOT NULL,
   PRIMARY KEY (nummerEtudiant),
    FOREIGN KEY (idNiveau) REFERENCES niveau(idNiveau)
);
Create table professeur (
    nummerSomme INT NOT NULL,
    grade VARCHAR(3) NOT NULL,
    nom VARCHAR(20) NOT NULL,
   email VARCHAR(20) NOT NULL,
   PRIMARY KEY (nummerSomme)
);
Create table respoUFR (
    idResspoUFR INT NOT NULL,
    nummerSomme INT NOT NULL,
   PRIMARY KEY (idResspoUFR),
   FOREIGN KEY (nummerSomme) REFERENCES professeur(nummerSomme)
);
--- niveaux : API1 - API2 .... -----
Create Table niveau (
    idNiveau INT NOT NULL,
    libelle VARCHAR(20),
    idFormation INT NOT NULL,
   PRIMARY KEY (idNiveau),
    FOREIGN KEY (idFormation) REFERENCES Formation(idFormation)
);
Create Table module (
    idModule INT NOT NULL,
    idSemestre VARCHAR(3) NOT NULL,
    idProf INT NOT NULL,
    idNiveau INT NOT NULL,
    libelle VARCHAR(20),
    nhCours INT NOT NULL,
   nhTD INT NOT NULL,
   nhTP INT NOT NULL,
    coeff INT NOT NULL,
    PRIMARY KEY (idModule),
    FOREIGN KEY (idNiveau) REFERENCES niveau(idNiveau),
    FOREIGN KEY (idSemestre) REFERENCES semestre(idSemestre),
   FOREIGN KEY (idProf) REFERENCES professeur(nummerSomme)
);
```

```
-- Check grade != PA dans java
Create Table formation (
    idFormation INT NOT NULL,
    idRespoFormation INT NOT NULL,
    libelle VARCHAR(20),
    description VARCHAR(100),
    PRIMARY KEY (idFormation),
    FOREIGN KEY (idRespoFormation) REFERENCES professeur(nummerSomme)
);
Create Table semestre (
    idSemestre VARCHAR(3) NOT NULL,
    PRIMARY KEY (idSemestre)
);
Create Table doctorant (
    idDoc INT NOT NULL,
    nummerEtudiant INT NOT NULL,
    idEncadrant INT NOT NULL,
    sujetThese VARCHAR(20),
    dateInscription DATE,
    PRIMARY KEY (idDoc),
    FOREIGN KEY (nummerEtudiant) REFERENCES etudiant(nummerEtudiant),
    FOREIGN KEY (idEncadrant) REFERENCES professeur(nummerSomme)
);
Create Table soutenance (
    idSoutnance INT NOT NULL,
    idDoc INT NOT NULL,
    jury VARCHAR(50) NOT NULL,
    dateSoutenance DATE,
    PRIMARY KEY (idSoutnance),
    FOREIGN KEY (idDoc) REFERENCES doctorant(idDoc),
);
Create Table note (
    idNote INT NOT NULL AUTO_INCREMENT,
    nummerEtudiant INT NOT NULL,
    nummerSomme INT NOT NULL,
    idModule INT NOT NULL,
    note FLOAT CHECK ( note BETWEEN 0 AND 20 ),
    PRIMARY KEY (idNote),
    FOREIGN KEY (nummerEtudiant) REFERENCES etudiant(nummerEtudiant),
    FOREIGN KEY (nummerSomme) REFERENCES professeur(nummerSomme),
    FOREIGN KEY (idModule) REFERENCES module(idModule),
);
```

Jeux d'insertions :

```
------ insertions etudiant ------
INSERT into etudiant VALUES (1, 1, "said fouad", "said@gmail.com");
INSERT into etudiant VALUES (2, 1, "manber chaimae", "manber@gmail.com");
INSERT into etudiant VALUES (3, 2, "laghlid ayoub", "laghlid@gmail.com");
INSERT into etudiant VALUES (4, 3, "ahmed mohamed", "ahmed@gmail.com");
INSERT into etudiant VALUES (5, 6, "farih hanae", "farih@gmail.com");
INSERT into etudiant VALUES (6, 6, "salimi moaad", "salimi@gmail.com");
INSERT into etudiant VALUES (7, 9, "sellak hanane", "sellak@gmail.com");
INSERT into etudiant VALUES (8, 9, "samir sami", "samir@gmail.com");
INSERT into etudiant VALUES (9, 15, "salimi samia", "salimi@gmail.com");
INSERT into etudiant VALUES (10, 8, "fatima imane", "fatima@gmail.com");
INSERT into etudiant VALUES (11, 7, "benani hamza", "benani@gmail.com");
----- insertions professeur ------
INSERT into professeur VALUES (1, "PH" , "denani salim", "denani@gmail.com");
INSERT into professeur VALUES (2, "PA", "asiri khalid", "asiri@gmail.com");
INSERT into professeur VALUES (3, "PA", "farihi meryem", "farihi@gmail.com");
INSERT into professeur VALUES (4, "PH", "daafi imane", "daafi@gmail.com");
INSERT into professeur VALUES (5, "PES", "soufi karim", "soufi@gmail.com");
INSERT into professeur VALUES (6, "PES", "bousalem wafae", "bousalem@gmail.com"
INSERT into professeur VALUES (7, "PA", "khelfi sanae", "khalfi@gmail.com");
INSERT into professeur VALUES (8, "PH", "mekhlouf adam", "mekhlouf@gmail.com");
INSERT into professeur VALUES (9, "PH", "naimi samir", "mekhlouf@gmail.com");
INSERT into professeur VALUES (10, "PA", "benali said", "bousalem@gmail.com");
----- insertions repsoUFR ------
INSERT into respoUFR VALUES (1,1);
INSERT into respoUFR VALUES (2,2);
INSERT into respoUFR VALUES (3,9);
INSERT into respoUFR VALUES (4,10);
----- insertions niveau ------
INSERT into niveau VALUES (1, "API1", 1 );
INSERT into niveau VALUES (2,"API2", 1);
INSERT into niveau VALUES (3,"TC", 2);
INSERT into niveau VALUES (4, "GPEE1", 5);
INSERT into niveau VALUES (5,"IID1", 7);
INSERT into niveau VALUES (6, "GI2", 3);
INSERT into niveau VALUES (7, "GE2", 4);
INSERT into niveau VALUES (8, "GRT2", 6);
INSERT into niveau VALUES (9,"IID2", 7);
INSERT into niveau VALUES (10, "GPEE3",5 );
INSERT into niveau VALUES (11, "GE3", 4);
INSERT into niveau VALUES (12, "GI3", 3);
INSERT into niveau VALUES (13,"IID3", 7);
INSERT into niveau VALUES (14, "GRT3",6 );
INSERT into niveau VALUES (15, "GPEE2", 5);
```

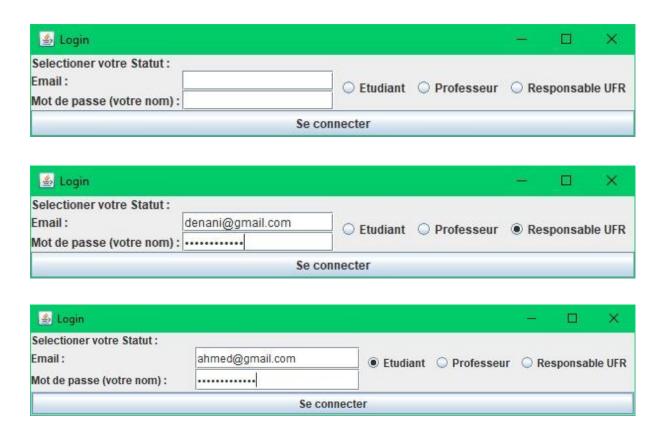
```
----- insertions module ------
INSERT into module VALUES (1,"S1", 1, 1, "Analyse1", 40, 30, 2, 1);
INSERT into module VALUES (2,"S2", 1, 1, "Algebre2", 30, 20, 2, 1);
INSERT into module VALUES (3,"S5", 3, 3, "Probabilité", 35, 20, 2, 1);
INSERT into module VALUES (4, "S6", 4, 3, "Automatique", 40, 20, 4, 1);
INSERT into module VALUES (5, "S7", 6, 6, "UML", 45, 15, 5, 1);
INSERT into module VALUES (6,"S7", 7, 6, "Réseaux avancées", 38, 22, 3, 1);
INSERT into module VALUES (7,"S8", 6, 6, "JAVA", 40, 35, 6, 1);
INSERT into module VALUES (8,"S8", 9, 7, "Electro Puissance", 35, 12, 6, 1);
INSERT into module VALUES (9,"S8", 6, 8, "Oracle", 35, 30, 5, 1);
INSERT into module VALUES (10, "S8", 6, 9, "Prog systeme", 30, 15, 5, 1);
INSERT into module VALUES (11,"S8", 6, 10, "GPI", 25, 10, 2, 1);
INSERT into module VALUES (12,"S8", 6, 5, "TCC", 25, 15, 3, 1);
----- insertions formations ------
INSERT into formation VALUES (1,2,"API", "formation pour API1 et API2");
INSERT into formation VALUES (2,3,"1ereACycleING","formation pour Tronc Commun"
INSERT into formation VALUES (3,6,"GI","formation pour GI2 et GI3");
INSERT into formation VALUES (4,5,"GE","formation pour GE2 et GE3");
INSERT into formation VALUES (5,7,"GPEE", "formation pour GPEE2 et GPEE3");
INSERT into formation VALUES (6,8,"GRT","formation pour GRT2 et GRT3");
INSERT into formation VALUES (7,4,"IID","formation pour IID1, IID2 et IID3");
----- insertions semestre ------
INSERT into semestre VALUES ("S1");
INSERT into semestre VALUES ("S2");
INSERT into semestre VALUES ("S3");
INSERT into semestre VALUES ("S4");
INSERT into semestre VALUES ("S5");
INSERT into semestre VALUES ("S6");
INSERT into semestre VALUES ("S7");
INSERT into semestre VALUES ("S8");
INSERT into semestre VALUES ("S9");
INSERT into semestre VALUES ("S10");
----- insertions doctorant ------
INSERT into doctorant VALUES (1, 1, 1, "Thèse1", 2018-04-12);
INSERT into doctorant VALUES (2, 3, 6, "Thèse2", 2021-05-22);
INSERT into doctorant VALUES (3, 4, 4, "Thèse3", 2021-05-24);
INSERT into doctorant VALUES (4, 7, 6, "Thèse4", 2019-06-17);
INSERT into doctorant VALUES (5, 8, 8, "Thèse5", 2019-04-04);
----- insertions soutenenace ------
INSERT into soutenance VALUES (1, 1, "denani salim, asiri khalid, farihi meryem",
2021-06-27);
INSERT into soutenance VALUES (2, 4, "daafi imane,asiri khalid,khelfi sanae",
INSERT into soutenance VALUES (3, 1, "benali said, mekhlouf adam, soufi karim",
2021-06-29);
```

```
INSERT into note VALUES (1, 2, 6, 5, 16);
INSERT into note VALUES (2, 2, 6, 7, 17);
INSERT into note VALUES (3, 2, 8, 9, 14);
INSERT into note VALUES (4, 2, 9, 10, 18);
INSERT into note VALUES (5, 2, 10, 11, 13);
INSERT into note VALUES (6, 2, 5, 12, 15);
INSERT into note VALUES (7, 3, 6, 5, 17);
INSERT into note VALUES (8, 3, 6, 7, 18);
INSERT into note VALUES (9, 3, 8, 9, 16);
INSERT into note VALUES (10, 3, 9, 10, 15);
INSERT into note VALUES (11, 3, 10, 11, 14);
INSERT into note VALUES (12, 3, 5, 12, 13);
INSERT into note VALUES (13, 4, 1, 1, 15);
```

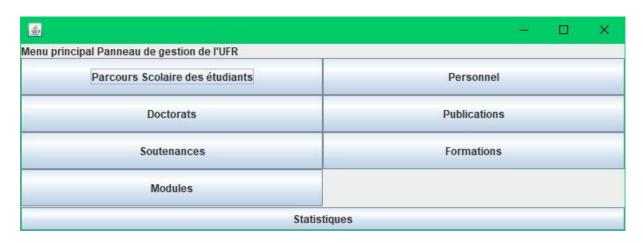
Interfaces:

Login:

Nous avons la possibilité de choisir le type de compte correspondant.



Menu responsable UFR:



CONCLUSION

Ce mini projet s'avère un grand pas dans le monde de la programmation JAVA. En effet, il nous a permis d'illustrer et mettre en application toutes les notions étudiées à savoir : la POO, les JTables.., comprendre différentes architectures pour la réalisation du projet, ainsi que de se familiariser avec le modèle MVC en essayant de l'appliquer dans notre projet. Bien que ce dernier ne soit véritablement complet, il présente des phases importantes en termes de conception et réalisation. Nous espérons avoir l'occasion de parfaire nos connaissances et nos socles de compétences et pouvoir être excellents en la matière.