

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université ABDERAHMANE MIRA de BEJAIA  
Faculté des Sciences Exactes  
Département D'Informatique



جامعة بجاية  
Tasdawit n Bgayet  
Université de Béjaïa

# Mémoire de fin de cycle

*En vue de l'obtention du diplôme de Master en Informatique*

**Option : Administration et Sécurité des Réseaux**

## *Thème*

**Conception d'une application informatique  
de gestion des examens**

**Proposé et dirigé par :**

M<sup>r</sup> BOUZIDI L' hadi

**Jury composé de :**

**Président :** M<sup>r</sup> ALOUI Abdelouahab

**Examineur :** M<sup>r</sup> FARAH Zoubeyr

**Présenté par :**

- M<sup>elle</sup> BENMAMMAR Fahima
- M<sup>elle</sup> HADJI Wissam

Promotion 2015

# Remerciements

*Quelques lignes ne pourront jamais exprimer la reconnaissance que nous éprouvons envers tous ceux qui, de près ou de loin, ont contribué, par leurs conseils, leurs encouragements ou leurs amitiés à l'aboutissement de ce travail.*

*Nous remercions le bon **Dieu** de nous avoir donné le courage d'aller au bout de nos objectifs.*

*Nos vifs remerciements accompagnés de toute notre gratitude vont tout d'abord à notre promoteur **M Bouzidi L' hadi**, pour nous avoir proposé ce sujet et nous avoir guidé pour réaliser ce projet.*

*Egalement aux :*

*Président de jury de nous avoir honoré de sa présence, et aux jurys d'avoir accepté d'examiner notre travail.*

*Nous tenons aussi à témoigner notre reconnaissance à notre cousine : **El djida et Slimane**, d'avoir prêté main forte pour la réalisation d'une partie de notre travail.*

*Notre reconnaissance va également à tous nos enseignants.*

***Merci à tous***

# Dédicaces

*Je dédie ce travail :*

*A mes très chers parents qui n'ont jamais cessé de me témoigner leur affection et de m'apporter leurs soutient depuis toujours, en témoignage d'affection et de profondes reconnaissances envers eux, et aux quels je ne rendrai jamais assez.*

*A la mémoire de mes grandes mères Baya et saltana.*

*A ma sœur Zoulikha et son marie Nabil et leur petite fille assile.*

*A ma petite sœur cylvia.*

*A mes très chers frères Athemen et yougarithen (youyou)*

*A ma cousine El djida et son marie Mouloud et leur petite fille asma.*

*A ma grande mère tasaadit (yama lala).*

*A ma cousine Allaoua et leurs femmes Hania (ma tente) et leurs enfants*

*Naima, Nassima, Saida, Dalila, Fariza, Hakim, Larbi, Rayane, Yani.*

*A ma tente fahima et son marie ouali et leur enfants warda et massin.*

*A mes amies Hanane, Souad, Souhila, Drifa, Sonia.*

*A tous mes enseignants du primaire à l'université.*

*Je le dédie également à mon binôme de travail Wissam et sa famille*

*Fahima*

*Je dédie ce travail :*

*À mes très chers parents qui n'ont jamais cessé de me témoigner leur affection*

*et de m'apporter leurs soutiens depuis toujours*

*À la mémoire de mes grands parents*

*À mes chers frères et sœurs, neveux et nièces et mes amis*

*À tous mes enseignants du primaire à l'université.*

*Je le dédie également à mon binôme de travail Fahima et sa famille*

*Wissam*

# Table des matières

Liste des figures.....	vi
Liste des tableaux .....	vii
Liste des abréviations .....	viii
Introduction .....	1
Chapitre.1 – Concepts théoriques abordés .....	3
1.1    Quelle démarche pour passer des besoins au code ? .....	3
1.2    Outil de modélisation UML.....	5
Chapitre.2 –Cahier de charge .....	7
Chapitre.3 – Spécification des besoins - Cas d'utilisation et IHM .....	14
3.1    Identification des acteurs.....	14
3.2    Cas d'utilisation.....	14
3.3    Prototypage de l'Interface Homme/Machine .....	21
3.4    Quelques Exemples de maquettes IHM .....	22
Chapitre.4 – Spécification détaillée des besoins .....	34
Chapitre.5 - Analyse.....	50
5.1    Identification des concepts du domaine .....	51
5.2    Identification des associations.....	52
5.3    Diagramme de classes métiers.....	53
5.4    Modèle relationnel.....	54
Chapitre.6 – Conception objet préliminaire .....	57
6.1    Démarche.....	58
6.2    Diagrammes de séquences détaillées .....	59
Chapitre.7 – Conception objet détaillée et implémentation .....	70
7.1    Solution technologique.....	71
7.2    Le modèle MVC et CodeIgniter .....	71
7.3    Le Framework CodeIgniter .....	72
7.4    Structure des fichiers de GEST-EXAM .....	73
7.5    Grocery CRUD.....	73
7.6    Quelques exemples des codes .....	74
Conclusion (bilan et perspectives).....	84
Bibliographie.....	85

## Liste des figures

FIGURE 1 BESOINS AU CODE .....	03
FIGURE 2 METHODE EMPLOYEE.....	04
FIGURE 3 PROCESSUS DE DEVELOPPEMENT .....	04
FIGURE 4 DIFFERENTS DIAGRAMMES UML.....	06
FIGURE 5 SCHEMA SYNOPTIQUE DE L'APPLICATION ATTENDUE.....	08
FIGURE 6 PROCESSUS D'ÉVALUATION .....	10
FIGURE 7 EDITION DES PREUVES .....	11
FIGURE 8 CAS D'UTILISATIONS D'ADMINISTRATEUR .....	17
FIGURE 9 CAS D'UTILISATIONS DE CHEF DE DEPARTEMENT .....	18
FIGURE 10 CAS D'UTILISATION D'ENSEIGNANT EN LIGNE.....	19
FIGURE 11 CAS D'UTILISATION D'ENSEIGNANT HORS LIGNE .....	20
FIGURE 12 CAS D'UTILISATION D'ÉTUDIANT.....	21
FIGURE 13 DSS1- CAS D'UTILISATION : AUTHENTIFICATION.....	35
FIGURE 14 DSS2- CAS D'UTILISATION : CONFIGURER GESTEXAM-WEB(ADMINISTRATEUR).....	36
FIGURE 15 DSS3- CAS D'UTILISATION : GERER FACULTES (ADMINISTRATEUR) .....	37
FIGURE 16 DSS4- CAS D'UTILISATION : AJOUTER FACULTE (ADMINISTRATEUR).....	38
FIGURE 17 DSS5- CAS D'UTILISATION : MODIFIER FACULTE(ADMINISTRATEUR) .....	39
FIGURE 18 DSS6- CAS D'UTILISATION : SUPPRIMER FACULTE(ADMINISTRATEUR) .....	40
FIGURE 19 DSS7- CAS D'UTILISATION : GERER PROMOTIONS ( CHEF DE DEPARTEMENT ).....	41
FIGURE 20 DSS8- CAS D'UTILISATION: AJOUTER PROMOTION ( CHEF DE DEPARTEMENT ) .....	42
FIGURE 21 DSS9- CAS D'UTILISATION : MODIFIER PROMOTION (CHEF DE DEPARTEMENT ) .....	43
FIGURE 22 DSS10- CAS D'UTILISATION : SUPPRIMER PROMOTION (CHEF DE DEPARTEMENT) .....	44
FIGURE 23 DSS11 - CAS D'UTILISATION : GERER PLANNING EXAMENS (CHEF DE DEPARTEMENT) .....	45
FIGURE 24 DSS12- CAS D'UTILISATION : CONSULTER PLANNING EXAMEN ( ENSEIGNANT ).....	46
FIGURE 25 DSS13- CAS D'UTILISATION : CORRIGER COPIES ( ENSEIGNANT ) .....	47
FIGURE 26 DSS14-CAS D'UTILISATEION : CONSULTER PLANNING EXAMEN ( ETUDIANT ) .....	48
FIGURE 27 DSS 15- CAS D'UTILISATION : RECLAMER ( ETUDIANT ) .....	49
FIGURE 28 DIAGRAMME DE CLASSE METIERS .....	53
FIGURE 29 DCP1 CAS D'UTILISATION : AUTHENTIFICATION.....	55
FIGURE 30 DCP2 CAS D'UTILISATION : GERER FACULTE (ADMINISTRATEUR) .....	56
FIGURE 31 DCP3 CAS D'UTILISATION : GERER PROMOTION (CHEF DE DEPARTEMENT) .....	56
FIGURE 32 PASSAGE DE L'ANALYSE A LA CONCEPTION PRELIMINAIRE .....	58
FIGURE 33 DSD1- CAS D'UTILISATION : AUTHENTIFICATION .....	59
FIGURE 34 DSD2- CAS D'UTILISATION : CONFIGURER GEST-EXAM (ADMINISTRATEUR).....	60
FIGURE 35 DSD3- CAS D'UTILISATION : AJOUTER FACULTE (ADMINISTRATEUR).....	61
FIGURE 36 DSD 4- CAS D'UTILISATION : MODIFIER FACULTE (ADMINISTRATEUR) .....	62
FIGURE 37 DSD 5- CAS D'UTILISATION : SUPPRIME FACULTE (ADMINISTRATEUR) .....	63
FIGURE 38 DSD 6 - CAS D'UTILISATION : AJOUTER PROMOTION (CHEF DE DEPARTEMENT) .....	64
FIGURE 39 DSD 7- CAS D'UTILISATION : MODIFIER PROMOTION (CHEF DE DEPARTEMENT).....	65
FIGURE 40 DSD 8- CAS D'UTILISATION : SUPPRIMER PROMOTION (CHEF DE DEPARTEMENT) .....	66
FIGURE 41 DSD 9- CAS D'UTILISATION : CONSULTER PLANNING EXAMEN(ENSEIGNANT).....	67
FIGURE 42 DSD 10 - CAS D'UTILISATION : CONSULTER PLANNING EXAMENS (ETUDIANT) .....	68
FIGURE 43 DSD 11 - CAS D'UTILISATION : RECLAMER(ETUDIANT) .....	69
FIGURE 44 - INTERACTION ENTRE LE NAVIGATEUR WEB ET L'APPLICATION .....	71
FIGURE 45 – ARCHITECTEUR MVC .....	72
FIGURE 46 – FONCTIONNEMENT DE CODEIGNITER .....	72

FIGURE 47 – L'ARBORESCENCE DU DOSSIER APPLICATION AVEC LES EXPLICATIONS DES DOSSIERS.....	73
FIGURE 48 – CODE PRESENTANT LE CONTROLEUR.....	74
FIGURE 49 – CODE PRESENTANT LE VUE .....	74
FIGURE 50 – CODE PRESENTANT LE MODEL.....	75
FIGURE 51 – L'INTERFACE OBTENUE LA PAGE D'ACCUEIL.....	76
FIGURE 52 – CODE PRESENTANT LE CONTROLEUR D'ENSEIGNANT .....	76
FIGURE 53 – CODE PRESENTANT LE VUE D'ENSEIGNANT.....	77
FIGURE 54 – CODE PRESENTANT LE MODEL D'ENSEIGNANT .....	78
FIGURE 55 – L'INTERFACE OBTENUE LA PAGE D'ENSEIGNANT (COPIES).....	79
FIGURE 56 – CODE PRESENTANT LE CONTROLEUR D'ETUDIANT.....	80
FIGURE 57 – CODE PRESENTANT LE VUE D'ETUDIANT .....	81
FIGURE 58 – CODE PRESENTANT LE MODEL D'ETUDIANT .....	82
FIGURE 59 – L'INTERFACE OBTENUE LA PAGE D'ETUDIANT (CONSULTER COPIES).....	83

## Liste des tableaux

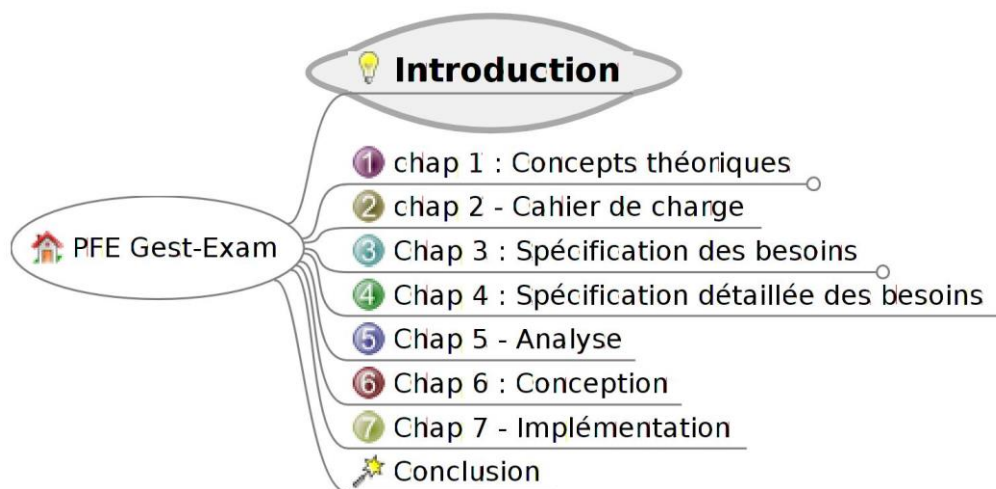
TABLEAU 1 CAS D'UTILISATION .....	16
TABLEAU 2 LA PAGE D'ACCUEIL.....	22
TABLEAU 3 IHM DE L'AMINISTRATEUR (CONFIGURATION).....	23
TABLEAU 4 IHM DE L'AMINISTRATEUR (GESTION DES FACULTES) .....	24
TABLEAU 5 IHM DE L'AMINISTRATEUR (GESTION DES MODULES).....	25
TABLEAU 6 IHM DE L'AMINISTRATEUR (GESTION DES SECTIONS) .....	26
TABLEAU 7 IHM DE CHEF DU DEPARTEMENT (GESTION DES EXAMENS) .....	27
TABLEAU 8 IHM DE L'ENSEIGNANT (GESTION DE LA LISTE DES ENSEIGNANTS).....	28
TABLEAU 9 IHM DE L'ENSEIGNANT (CORRIGER DES COPIES).....	29
TABLEAU 10 IHM DE L'ENSEIGNANT (GESTION DES RECLAMATIONS DES ETUDIANTS).....	30
TABLEAU 11 IHM DE L'ENSEIGNANT (LANCER UNE EVALUATION) .....	31
TABLEAU 12 IHM DE L'ENSEIGNANT (PREVISULIATION DU SUJET) .....	32
TABLEAU 13 IHM DE L'ETUDIANT (CONSULTATION DES COPIES).....	33
TABLEAU 14 LISTE DES CONCEPTS DECOULES DE LA SPECIFICATION DES BESOINS ET DE L'IHM .....	51

## Liste des abréviations

<b>Abréviation</b>	<b>Signification</b>
UML	Unified modeling language ou langage de modélisation unifié
DS	Diagramme de séquence
DSS	Diagramme de séquence système
DSD	Diagramme de séquence détaillé
UC	Use case ou cas d'utilisation
BDD	Base de données
GEST-EXAM	Gestion des examens
DCP	Diagrammes de classes participantes
IHM	Interface homme machine
PHP	Php : hypertext preprocessor
MVC	Modèle vue contrôleur
URL	Uniform resource locator
CSV	Comma-separated values
SGBD	Système de gestion de bases de données
CSS	Cascading style sheet
SQL	Structured Query Language
HTML	Hypertext markup language
UP	Unified Process (processus unifié)
OCR	Optical Character Recognition
XML	Extensible Markup Language



# Introduction



La gestion des examens est l'une des tâches les plus délicates dans le processus éducatif. Il s'agit d'un processus laborieux mettant en jeu plusieurs acteurs. Elle exige une bonne organisation de l'établissement et un bon système de communication et de gestion. Les examens doivent être planifiés à temps, l'information doit parvenir aux différents acteurs impliqués rapidement, des changements peuvent survenir et doivent être pris en compte efficacement, divers documents sont produits et échangés et des enjeux importants sont induits, notamment, pour les étudiants. Ceci exprime clairement la complexité d'un tel système. L'Université de Bejaia, à travers sa section chargée du e-learning, nous a chargé de concevoir une solution informatique à la gestion des examens tout en nous fixant, au préalable, des exigences fonctionnelles et techniques [0]. Ce projet a pour objectif de concevoir et de réaliser une application Web pour :

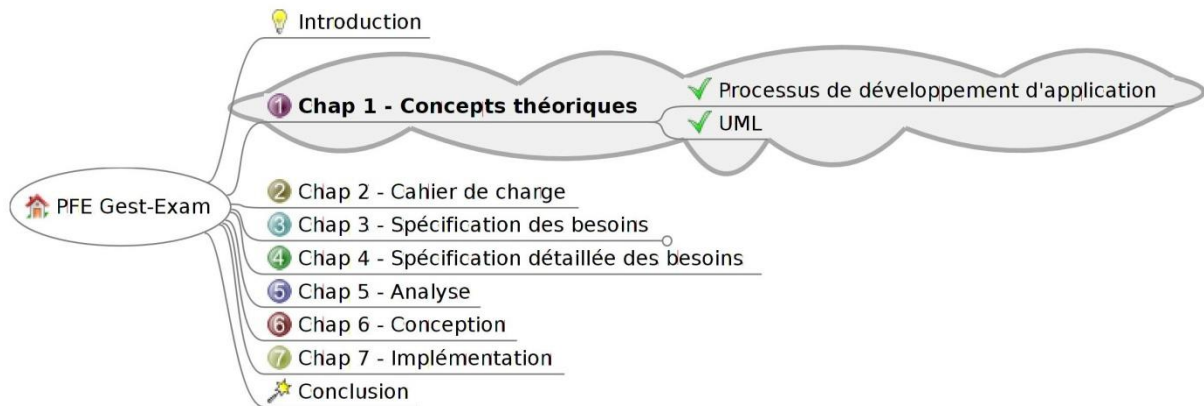
- La gestion des examens par les départements (gestion des modules, des promotions, des enseignants, planification des examens, etc.),
- la saisie des informations concernant les utilisateurs, les facultés et les départements
- la gestion des examens par les enseignants
- l'accès aux copies et aux notes d'examen par les étudiants

En plus de l'application Web, une application hors ligne est prévue uniquement pour les enseignants afin de leur permettre de numériser et de corriger les copies sur ordinateur.

D'un point de vue méthodologique, nous nous sommes inspirés de la méthode de développement proposée par Pascal Roques [1]. Il s'agit d'une méthode de conception simplifiée et orientée vers le développement d'application Web tout en se basant sur le langage de modélisation UML2.

Nous présentons, dans ce rapport, la démarche entreprise et les résultats obtenus : Dans un premier chapitre nous présentons quelques concepts théoriques que nous avons jugés importants vis-à-vis de ce projet : Le processus de développement d'applications informatiques et le langage UML. Le second chapitre, traite de l'expression initiale des besoins. Dans le troisième chapitre nous abordons la spécification des besoins d'après les cas d'utilisation (identification des acteurs et des cas d'utilisation, structuration en packages, relations entre les cas d'utilisation). Nous poursuivons dans ce chapitre par la spécification des besoins d'après un prototypage de l'IHM. Dans le chapitre 4, nous présentons la spécification détaillée des besoins en élaborant, pour chaque cas d'utilisation, un diagramme de séquence système (DSS). Dans les chapitres 2, 3 et 4, nous faisons une description fonctionnelle de notre application. Dans le chapitre 5 en revanche, nous allons présenter une description structurelle du système à réaliser. En particulier, nous passons à une conception objet en réalisant, pour chaque cas d'utilisation, des classes d'analyse (identification des concepts du domaine, identification des associations et des attributs, élaboration des diagrammes de classes participantes (DCP)). Dans le chapitre 6, nous présentons la conception objet préliminaire au travers des diagrammes de séquence détaillés (DSD). Il s'agit d'attribuer des responsabilités (comportements, services ou méthodes) aux classes d'analyse identifiées au chapitre précédent. Nous finirons ce travail par le chapitre 7 où nous présentons l'implémentation de notre application.

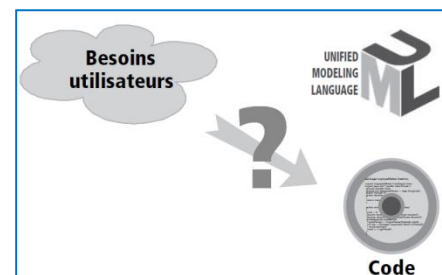
## Chapitre.1 – Concepts théoriques abordés



*Avant de commencer tout projet de développement d'application informatique, il est important d'esquisser les concepts théoriques permettant d'assoir la méthodologie à suivre et les outils à utiliser afin d'arriver à la solution souhaitée par les utilisateurs. Sachant que notre choix méthodologique s'est porté sur le processus de développement proposé par Pascal Roques [Roques, 2006] et l'outil de modélisation UML, nous présenterons ces deux outils.*

### 1.1 Quelle démarche pour passer des besoins au code ?

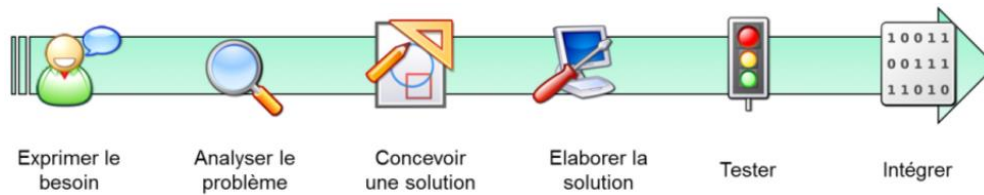
Afin de mener à bien un processus de développement permettant, à partir des besoins, réaliser le code de l'application, il est nécessaire de formaliser (par des documents en utilisant des outils) : les processus, les besoins, les spécifications logicielles, l'architecture logicielle et les tests. A cet effet plusieurs méthodes sont utilisées : Développement rapide d'applications, *eXtremeProgramming*, UP<sup>1</sup>, Merise, etc.



**Figure 1– Besoins au code**

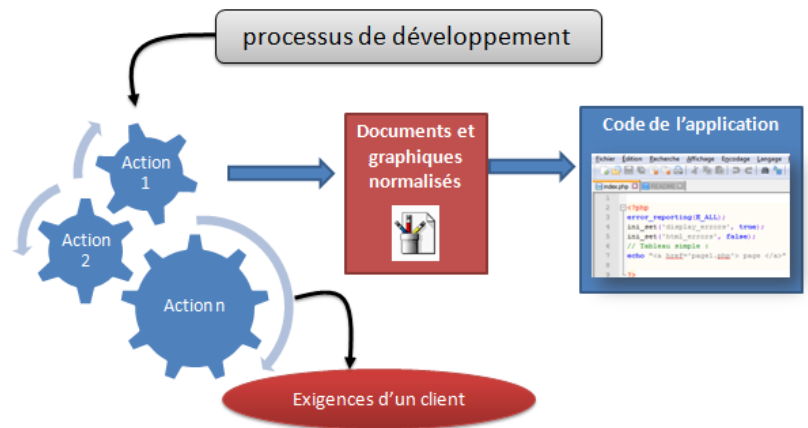
Quelle que soit la méthode employée, on retrouve toujours les activités ci-dessous dans un projet informatique. Les différences résident dans leur degré de formalisation et leur orchestration : itérative, séquentielle, incrémentale, pilotée par les modèles, pilotée par les tests, etc.

<sup>1</sup> Unified Process (processus unifié)



**Figure 2- Méthode employée**

Dans notre travail, nous allons nous intéresser uniquement à celles qui sont en liaison directe avec le contexte de notre projet. Ainsi, nous mettrons l'accent sur la conception d'application Web. De façon générale, un processus de développement est un enchaînement d'actions que devrait suivre un concepteur d'application



**Figure 3– Processus de développement**

afin de répondre le mieux possible aux exigences d'un client (commanditaire) en proposant des documents et des graphiques normalisés permettant d'aboutir de manière cohérente et logique au code de l'application. [2]

Dans le cadre de ce projet, nous allons utiliser un processus de développement conduit par les cas d'utilisation, léger, mais ne négligeant pas les activités d'analyse et de conception et fondé sur l'utilisation d'un sous-ensemble des diagrammes du langage UML. Nous suivrons la démarche proposée par Pascal Roques [1] :

1. Dans un premier temps, les besoins vont être modélisés au moyen des cas d'utilisation UML. Ils seront représentés de façon plus concrète par une maquette d'IHM (Interface Homme-Machine).
2. Chaque cas d'utilisation est décrit textuellement de façon détaillée et donne lieu à un diagramme de séquence simple représentant graphiquement la chronologie des interactions entre les acteurs et le système vu comme une boîte noire. Nous appellerons ce diagramme : «Diagramme de Séquence Système» ou DSS.
3. Les diagrammes de séquence système et l'IHM vont permettre de trouver le diagramme de classe de conception.
4. Le diagramme de classe de conception d'un côté et l'IHM de l'autre vont permettre de trouver les diagrammes de classes participantes. Pour chaque cas d'utilisation, les trois principaux

types de classes d'analyse (les dialogues, les contrôles et les entités) et leurs relations vont être décrits.

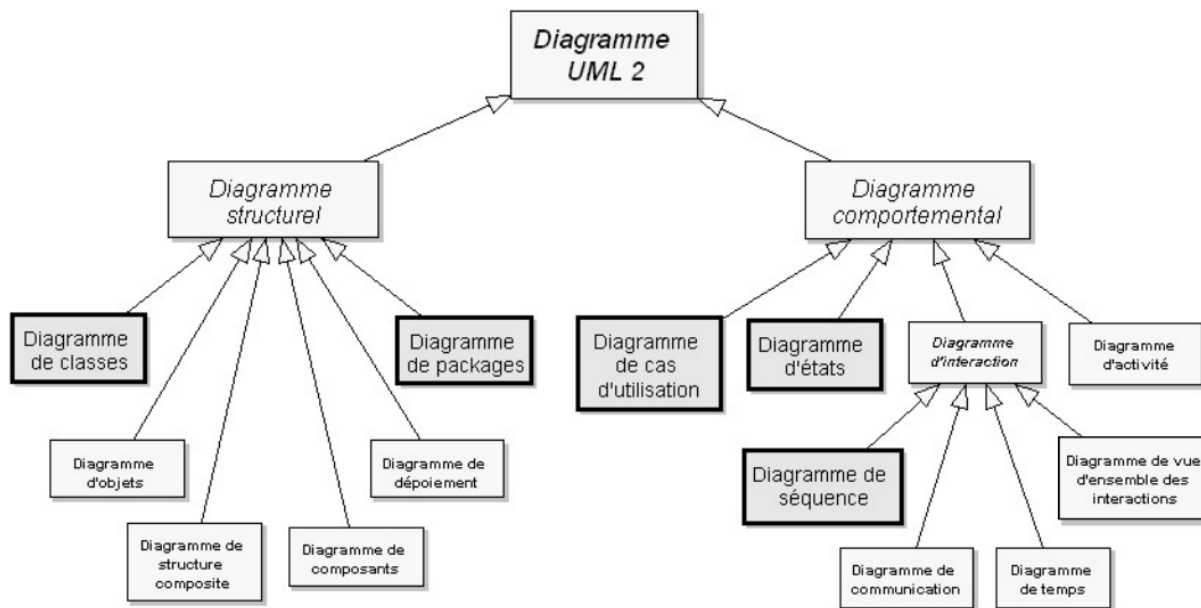
5. Par la suite, en remplaçant le système vu comme une boîte noire par un ensemble choisi d'objets de conception, nous décrirons l'attribution des responsabilités dynamiques, tout en conservant une forte traçabilité avec les cas d'utilisation. Ceci donne lieu au diagramme de séquence détaillé.
6. Les diagrammes de classe participantes et les diagrammes de séquences détaillées vont servir pour écrire le code de l'application. A ce propos, nous allons directement exploiter les diagramme de classes participantes et les diagrammes de séquences détaillées pour implémenter le modèle MVC qui est un modèle de conception permettant de distinguer les données ou bases de données (Modèle), les dialogues (Vue) et les codes métiers (Contrôleurs).

## 1.2 Outil de modélisation UML

UML<sup>2</sup> [3] est un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, à spécifier et documenter des systèmes, à esquisser des architectures logicielles, à concevoir des solutions et communiquer des points de vue. Pascal Roque, dans son ouvrage intitulé « UML2, Modéliser une application Web » [1] explique que ce langage unifie à la fois les notions et les concepts orientés objet et précise qu'il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots du langage naturel. Cet outil unifie les notations et les concepts orienté objet. Il unifie également les notations nécessaires à la description des activités d'un processus de développement allant de l'expression des besoins jusqu'au codage. Il s'articule autour de 13 diagrammes répartis en 6 diagrammes structurels et 7 diagrammes comportementaux (figure.4).

---

<sup>2</sup>Unified Modeling Language

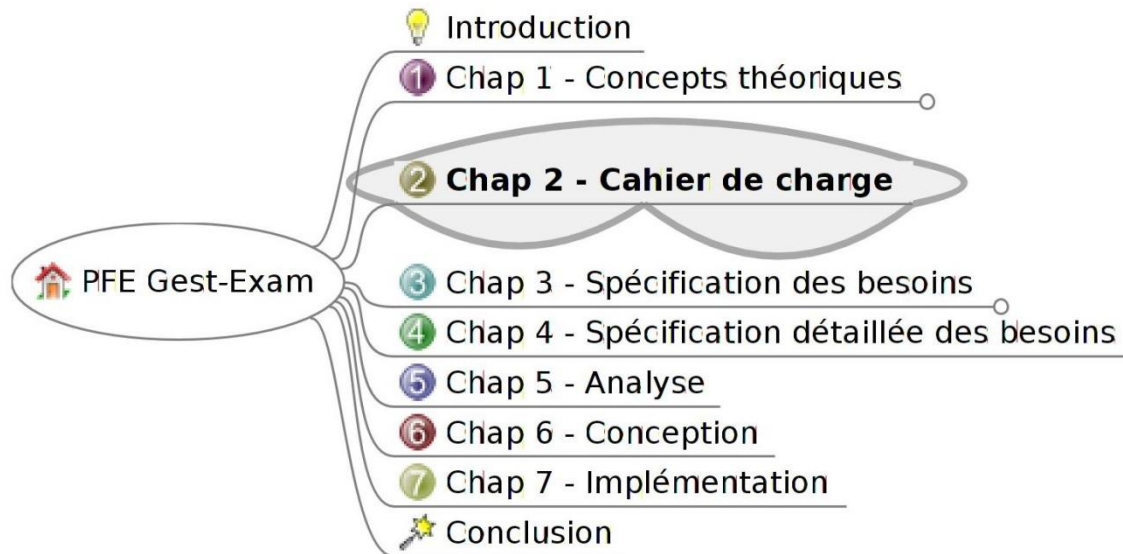


**Figure 4– Différents diagrammes UML**

Voici les diagrammes les plus importants que nous avons utilisés dans notre projet :

- Les diagrammes de cas d'utilisation sont utilisés dans le cadre de la spécification des besoins. Ils montrent l'interaction fonctionnelle entre les acteurs sollicitant des services et le système répondant à ces services.
- Les diagrammes de classe constituent le point focal dans un développement orienté objet. Dans la phase d'analyse, ils ont pour objectif de décrire la structure des entités manipulées par les utilisateurs. Dans la phase de conception, ils représentent les structures des codes orientés objet.
- Les diagrammes de package décrivent l'organisation logique du modèle d'application ; Il indique aussi les relations entre les différents packages. Il permet de structurer les classes d'analyse et de conception, mais aussi les cas d'utilisation.
- Les diagrammes de séquence et les diagrammes de communication sont des diagrammes d'interaction. Ils représentent les échanges de messages entre les différents éléments dans le cadre d'un fonctionnement particulier du système. Dans la phase d'analyse, ils servent à développer les scénarios d'utilisation du système. Dans la phase de conception, ils servent à concevoir les méthodes (comportements) des classes.
- Les diagrammes de déploiement montrent la disposition physique des différents matériels (les nœuds) qui entrent dans la composition d'un système et la répartition des instances de composants, processus et objets qui « vivent » sur ces matériels.

## Chapitre.2 –Cahier de charge

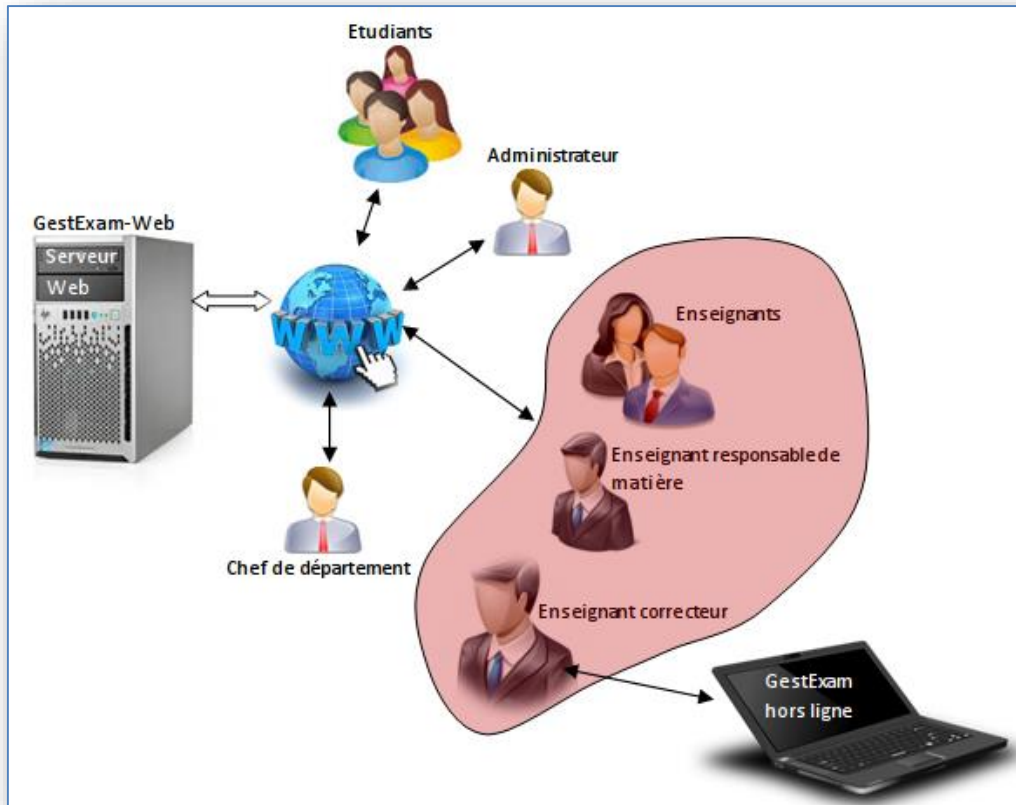


*Dans cette partie, nous présentons les exigences du commanditaire du projet. Ces exigences seront exprimées sous forme textuelles.*

*Le commanditaire de ce projet est la section chargée du e-learning de l'Université de Bejaia. Nous allons présenter dans ce qui suit, ce que nous avons consigné textuellement à l'issue de nos différentes entrevus.*

La gestion des examens est l'une des tâches les plus délicates dans le processus éducatif. Le but de ce projet est de mieux gérer ce processus en profitant des possibilités qu'offre la technologie numérique. Ainsi, nous proposons d'automatiser un processus de gestion des examens. Nous souhaitons que quatre acteurs puissent intervenir dans notre application : l'administrateur, l'enseignant, l'étudiant et le chef de département. [4]

Nous souhaitons mettre en œuvre deux applications distinctes l'une en ligne et l'autre hors ligne. La partie en ligne sera accessible à tous les acteurs. La partie hors ligne, ne sera utilisée que par l'enseignant qui est désigné pour corriger des copies. Dans ce qui suit, nous appellerons la partie en ligne de notre application : « GestExam-Web ». La partie hors ligne sera appelée « GestExam-Hors Ligne ». [4]



**Figure 5– Schéma synoptique de l'application attendue**

Dans ce qui suit, nous formulerons le cahier de charge en termes de spécification des fonctionnalités des deux applications que nous envisageons de réaliser. Ainsi, nous décrirons, pour chaque utilisateur, ce que notre système devrait offrir comme services.

Avant de décrire notre cahier, nous tenons à clarifier les concepts d'examen et d'évaluation. Ainsi, nous considérerons l'examen comme un évènement qui va réunir des étudiants dans un lieu pendant une durée déterminée afin qu'ils puissent montrer leur compétences et ou connaissances pour un module (matière) donné. Les examens que l'on considérera ici sont surveillés. Par contre, nous considérerons une évaluation comme un processus plus général permettant à l'enseignant de juger le niveau d'acquisition des connaissances et/ou compétences des étudiants pour une matière donnée. Ainsi, toutes les étapes, à commencer par l'édition d'un énoncé (ou sujet) d'examen jusqu'à la correction et la consultation des copies par les étudiants seront considéré ici comme faisant partie du processus d'évaluation.

### **Fonctionnalités à mettre à la disposition de l'administrateur :**

L'administrateur pourra configurer « Gest-exam » en ligne (Indiquer le nom de l'institution, ses coordonnées et un logo). Il doit être capable de créer et de gérer les facultés et les départements, les comptes des utilisateurs et les comptes des chefs de départements.



Remarques:

- Un département appartient à une et une seule faculté
- Un département est géré par un seul chef de département
- Les comptes des utilisateurs peuvent être créés de façon massive à partir d'un fichier CSV.

**Fonctionnalités à mettre à la disposition du chef de département :**

Pour une année universitaire et un semestre donnés, le chef de département pourra :

- Gérer les utilisateurs affectés au département (Ajout, modification et suppression d'enseignants, de promotions, de sections et de groupes)
- Gérer (Créer, modifier et supprimer) des matières (modules)
- Gérer les examens :
  - Indiquer : le nom du module concerné, le type (EMD, Rattrapage, etc.), la date et l'heure de déroulement, les lieux de déroulement par groupe d'étudiants, le responsable du module et une liste de surveillants.
  - Editer un rapport de déroulement par lieu d'examen. Ce rapport doit indiquer la date et l'heure effective de l'examen, le nombre d'inscrits, le nombre de présents, le nombre de copies, les noms des surveillants et un rapport d'incidents
- Consulter et télécharger les notes, le rapport de déroulement en classe, le sujet ou l'énoncé, et le corrigée d'un examen concernant les étudiants du département.

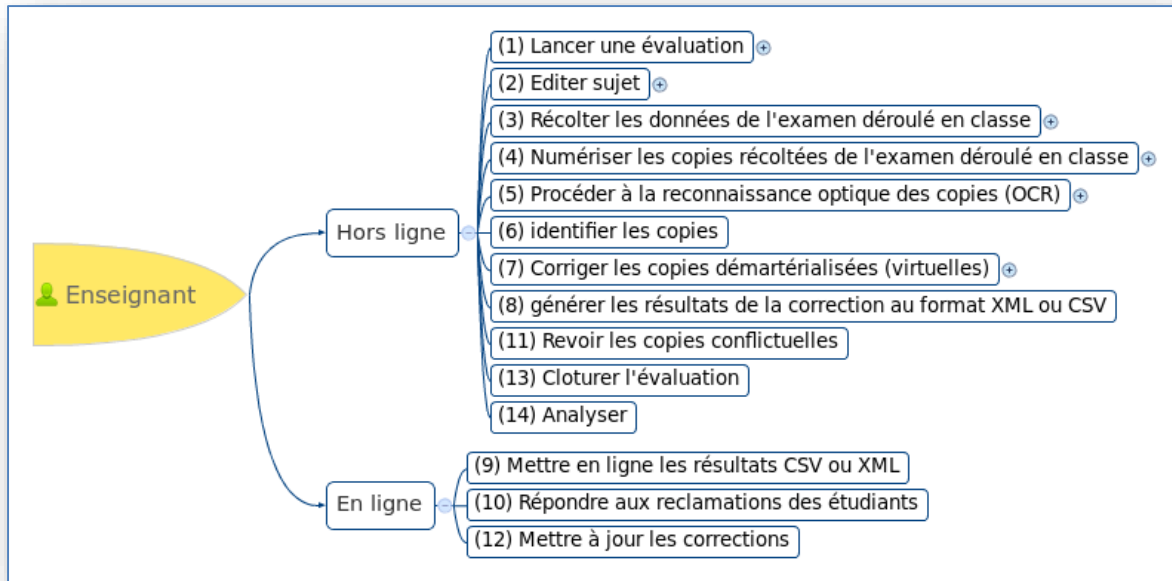
**Fonctionnalités à mettre au service de l'enseignant**

Nous rappelons que les examens étant planifiés par le chef de département. Ce dernier indique, en ligne et pour chaque matière, un responsable et une liste de surveillants parmi les enseignants.

L'enseignant, concerné par la surveillance d'un examen donné, pourra consulter le planning des examens en ligne (pour une année universitaire et un semestre donnés).

L'enseignant en sa qualité de responsable de matière, aura la possibilité, pour un examen planifié, d'indiquer les enseignants qui vont procéder à sa correction. Il doit indiquer, pour chaque groupe l'enseignant-correcteur.

Un enseignant s'il est désigné pour corriger les copies d'un ou de plusieurs groupes d'un examen donné, aura la possibilité de télécharger, au format CSV, les données concernant cet examen (toutes les informations d'identification de l'examen et la liste des étudiants concernés par la correction de cet enseignant). Par ailleurs, il devra aussi récupérer (physiquement), chez le responsable de matière, les feuilles d'examen qui le concernent. A partir de ce moment, l'enseignant doit suivre un processus d'évaluation composé de 14 étapes comme indiqué sur la figure suivante :



**Figure 6– Processus d'évaluation**

**Etape 1 :** Lancer un processus d'évaluation : On suppose que l'examen a été créé et planifié par le chef de département en ligne. L'enseignant, s'il est désigné pour corriger une partie des feuilles d'examens, pourra télécharger les données concernant cet examen et la liste des étudiants pour lesquels il doit corriger les feuilles d'examen. Ainsi, ces données, au format CSV, doivent être introduites dans l'application « GestExam-Hors ligne» afin de lancer le processus d'évaluation. C'est cette étape qui va préparer le terrain pour enclencher le processus d'évaluation en mettant dans la base de données hors ligne les informations concernant l'examen et surtout la liste des étudiants dont les copies seront corrigées par l'enseignant en question.

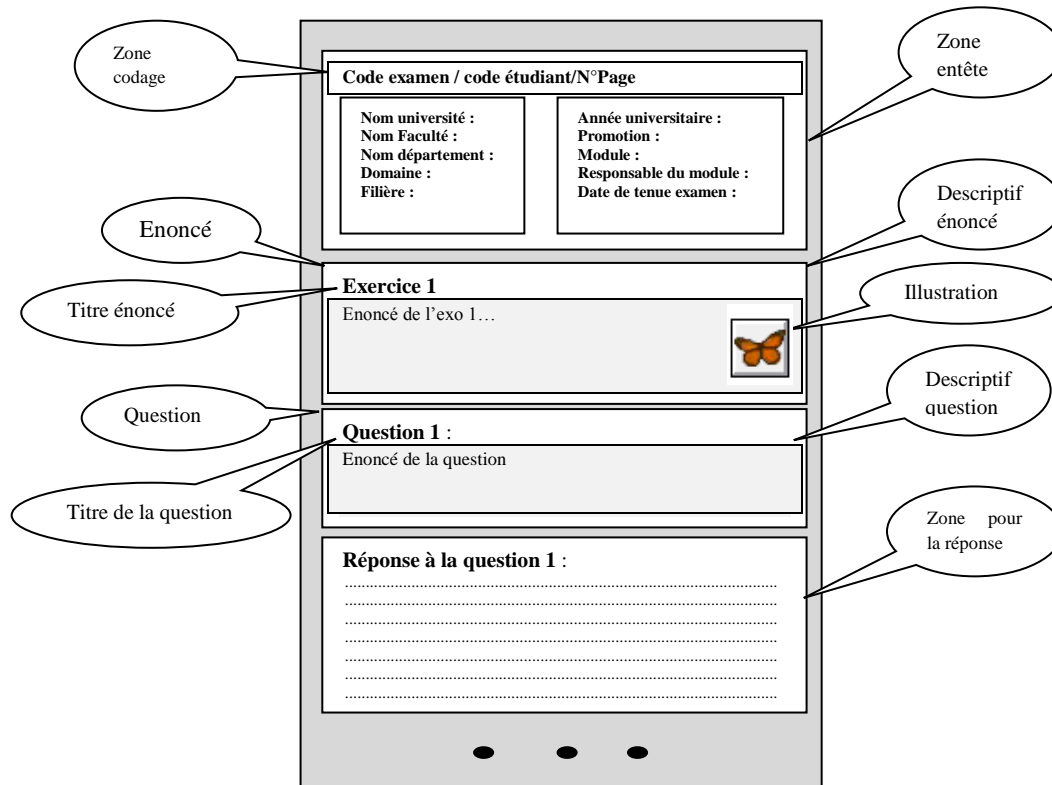
**Etape 2 :** Rédaction du sujet (énoncé) d'examen : Cette étape est composée de deux phases :

Phase1 – Spécification du sujet (énoncé) d'un examen : Chaque sujet doit correspondre à un et un seul examen (évaluation). On doit offrir à l'enseignant deux possibilités : Créer l'énoncé de l'examen à l'aide d'un outil externe et de générer au format PDF ou créer l'examen à l'aide de notre outil logiciel.

Dans le cas de la création de l'énoncé avec un outil externe, il faut indiquer le barème de l'examen. En d'autres termes, pour chaque question (identifier par un nom) indiquer une note maximale, un coefficient et un pas de notation. Pour l'examen, il faut indiquer la note maximale et la solution au format PDF.

Dans le cas où l'énoncé va être créé par notre application, il faut pouvoir créer le sujet entièrement. Ainsi, on supposera qu'un sujet est composé d'une ou de plusieurs page. Chaque page doit comporter un numéro de page. Le sujet peut contenir 0 à  $n$  questions pouvant être alternées par des énoncés. Un énoncé peut contenir, un titre, un texte descriptif et éventuellement une illustration (image). Chaque question doit posséder un *titre*, une *description*, une *note maximale*, un *coefficient* (ou poids qui est par défaut de 1) et un *pas de notation*. En plus de cela, on doit indiquer le *nombre de lignes* approximatif que va occuper la réponse correspondante à la question (cela va servir pour l'édition des pré-imprimés). Pour chaque question, l'enseignant doit être capable d'indiquer la solution sous forme d'un *texte* et éventuellement d'une *illustration*.

**Phase 2 – Edition des pré-imprimés :** Les pré-imprimés de l'épreuve doivent comporter un entête indiquant, tout à fait en haut de la page, un code identifiant l'épreuve, l'étudiant et le numéro de page. Ils doivent aussi comporter des informations textuelles identifiant le type d'examen (EMD, Interrogation, Rattrapage, ...), l'établissement, la faculté, le département, l'enseignant, le module, la promotion, la durée, la date de déroulement, l'année universitaire et le semestre. Ces imprimés doivent comporter des zones de textes (énoncés) et des zones où va répondre l'étudiant. On doit pouvoir imprimer le sujet à part sans inclure les zones de réponse et les informations concernant les étudiants.



**Figure 7- Edition des preuves**

**Etape 3 :** Récupérer, depuis « GestExam-Web » en ligne, un rapport sur le déroulement de l'examen sur table : Dans cette étape, on suppose que l'examen a déjà eu lieu en classe. Il s'agit, donc, de consigner un rapport du déroulement de cet examen : date, lieux et durée. On doit aussi rédiger un compte rendu du déroulement de l'examen et indiquer les listes des étudiants présents et absents.

**Etape 4 :** Numériser les copies d'examens : Cette tâche peut être réalisée en dehors de GestExam, l'essentiel est d'obtenir un document zippé des copies au format d'image courant (TIFF, GIF ou JPEG). Cependant, on doit prévoir, dans notre application, cette possibilité de sorte à offrir à l'enseignant un environnement intégré. Il est fortement recommandé de se servir d'un scanner à chargeur automatique afin de ne pas surcharger l'enseignant.

**Etape 5 :** Procéder à la reconnaissance optique des copies d'examens (OCR<sup>3</sup>) : Cette procédure doit aboutir à la génération des descripteurs des copies d'examens en produisant un fichier texte. Dans ce fichier texte, chaque ligne doit correspondre à une copie et doit comporter quatre informations : le nom du fichier « image » de la copie, le code de l'épreuve, le code de l'étudiant et le numéro de page. Il

<sup>3</sup> OCR : *Optical Character Recognition*

faut noter qu'on n'est pas obligé de traiter toute l'image de la copie pour la reconnaissance de caractère. En effet, nous connaissons à l'avance ce que le logiciel OCR doit chercher dans l'image et le lieu où il doit chercher. De ce fait, afin d'assurer une plus grande rapidité de la reconnaissance de caractère, il faut traiter uniquement la zone d'entête des copies. Par ailleurs, il faut garantir une très grande fiabilité de la reconnaissance de caractère en prenant des précautions sur les choix des polices de caractères et leurs tailles. En effet, certaines polices et tailles de caractères sont plus indiquées pour la reconnaissance optique (ceci à prévoir donc dans le module d'impressions des pré-imprimés). Enfin, il faut limiter au maximum l'alphabet des textes à reconnaître. Ceci va avoir un impact positif sur la rapidité du processus de reconnaissance. Pour cette raison, nous avons imposé que les codes des pages d'une copie soient composés uniquement de chiffres (dans le cas de la reconnaissance optique de caractères). Ainsi, tout à fait en haut de l'entête de chaque page, nous aurons trois informations chiffrées et séparées par des tirés : « *Code de l'épreuve – Code de l'étudiant – Numéro de page dans la copie* ».

**Etape 6 :** Identification des copies. Cette étape permet de vérifier si les auteurs de ces copies existent bel et bien dans la base de données. Ici, il se peut qu'on trouve des copies qui ne correspondent pas aux informations de la base de données. Dans ce cas, l'enseignant doit procéder à une identification manuelle (c'est-à-dire pour chaque image représentant une copie il faut lui associer trois informations : code de l'épreuve, code ou nom de l'étudiant et numéro de page).

**Etape 7 :** Correction assistée par ordinateur : Si, les copies ont été numérisées par un outil externe à *GestExam*, elles doivent être récupérées et mise dans un dossier créer en fonction des informations liées à l'examen (faculté, département, enseignant, code de l'examen). Pour mener à bien le processus de correction, l'enseignant se sert des barèmes (note maximale et pas de notation pour chaque question) et de la solution (pour chaque question) de l'examen déjà définis lors de l'édition du sujet de l'épreuve.

**Etape 8 :** Générer les résultats de la correction au format CSV ou XML. Ainsi, un fichier CSV ou XML sera généré afin de pouvoir être mis en ligne dans la partie Web de *GestExam*. *GestExam* doit signaler (à l'enseignant) durant cette phase s'il y a des copies qui n'ont pas été corrigées. L'enseignant doit indiquer une période de la consultation (en ligne) des copies corrigées.

**Etape 9 :** Mise en ligne de la correction. Il s'agit juste de déposer dans l'application *GestExam-Web* en ligne le résultat de la correction (fichier CVS ou XML). Notez-bien qu'en plus des copies et de la correction, dans le dossier zippé qui sera déposé en ligne, doit figurer l'énoncé au format PDF et éventuellement le corrigé dans ce même format.

**Etape 10 :** L'enseignant doit traiter les messages de désaccords à propos de son évaluation en provenance des étudiants. Il peut revoir les copies conflictuelles et en répondre par un message d'accord (donnant lieu à une reconsidération de la correction) ou de désaccord (maintenant la correction).

**Etape 11 :** Reprises de la correction pour certaines copies conflictuelles.

**Etape 12 :** Mettre à jours en ligne les résultats de la correction après consultation. Ici l'enseignant va régénérer les résultats de sa correction au format CSV (ou XML) et va le déposer en ligne (*GestExam-Web* en ligne).

**Etape 13 :** La clôture de l'évaluation permet d'indiquer que le processus d'évaluation est terminé et que les notes obtenues sont définitives. Cela veut dire aussi qu'il n'y a plus possibilité de réclamation.

Cette phase est activée dès que l'enseignant le décide. Cependant, la clôture ne pourra pas avoir lieu si toutes les copies n'ont pas été corrigées. Par ailleurs, l'enseignant doit être capable de rouvrir l'évaluation pour reprendre des corrections ou prendre en compte certaines réclamations qui n'auraient pas été prise en compte auparavant.

**Etape 14 :** L'analyse de l'épreuve n'est possible que si une évaluation est clôturée. Elle n'est accessible que pour l'enseignant. Elle lui donne accès à deux services : une analyse docimologique et une analyse statistique. Les rapports d'analyse docimologiques ne sera pas traitée dans le cadre de ce projet, mais doivent être prévue.

### **Fonctionnalité à mettre à la disposition des étudiants :**

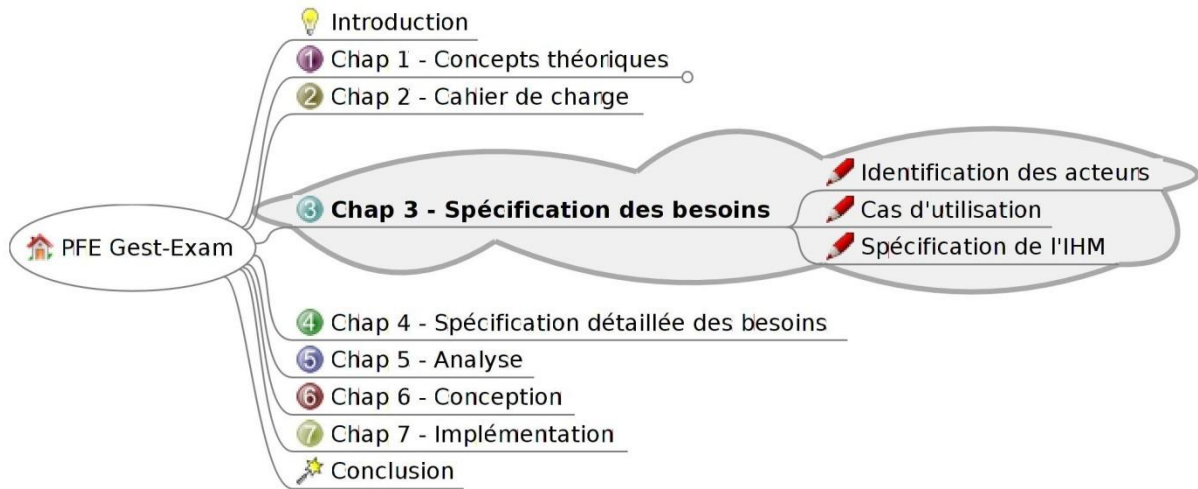
Les étudiants ne pourront accéder qu'à la partie en ligne de notre application (*GestExam-Web*). Ils devront s'authentifier et pourront :

- Consulter le planning (date, lieu notamment) des examens les concernant.
- Consulter les copies corrigées qui les concerne. Ainsi, pour un examen donné, dès le dépôt des résultats de la correction en ligne (étape 10 du processus mené par l'enseignant), les étudiants seront notifiés par mail qu'ils pourront consulter leurs copies et qu'ils ont un délai bien précis pour éventuellement formuler des réclamations. Ils seront aussi informer qu'ils doivent impérativement exprimer leur accord ou désaccord à propos de l'évaluation.
- Poster des réclamations pour une évaluation données. Ces réclamations sont formulées sous forme textuelles et mise à la disposition de l'enseignant. Elles servent à signaler le désaccord de l'étudiant relativement à la correction de l'enseignant. L'enseignant peut répondre favorablement ou défavorablement à la réclamation d'un étudiant. Il pourra aussi être amené à réévaluer la copie de l'étudiant.

Nous avons aussi convenu sur quelques besoins non fonctionnels :

- ✓ **Sécurité :** La sécurité de notre application touche en premier lieu, les niveaux d'accès des utilisateurs au système. Une connexion à un compte doit s'adapter à des conditions du nom d'utilisateur et du mot de passe (type de caractères correspondant dans les deux champs, longueur précise). La vigilance contre les éventuelles attaques et failles.
- ✓ **Simplicité du code source :** Le code doit être simple à assimiler pour des éventuelles améliorations.
- ✓ **L'ergonomie :** L'application offre une interface conviviale et facile à utiliser.
- ✓ **Maintenance :** Faciliter la détection d'anomalies, la correction des erreurs et les mises à jour du code.

## Chapitre.3 – Spécification des besoins - Cas d'utilisation et IHM



*Dans ce chapitre, nous allons procéder à la modélisation des cas d'utilisation permettant ainsi la spécification fonctionnelle des besoins. Nous allons en particulier élaborer le diagramme de cas d'utilisation mettant en évidence les acteurs, les cas d'utilisation et les relations entre ces cas d'utilisation. Pour appuyer notre spécification, nous présenterons le maquetage (IHM) de notre application.*

### 3.1 Identification des acteurs

Un acteur est une entité externe qui agit sur le système, pour notre application les acteurs qui interagiront avec elle sont :

- **Enseignant** : est une personne qui assure un cours au profil d'une classe ou d'une section. Il peut être corrigé un examen, configurer, consulter, et répondre aux réclamations des étudiants.
- **Etudiant** : est une personne qui passe un examen et peut consulter ces copies et formuler une réclamation.
- **Administrateur** : est une personne qui fixe tous les paramètres de GEST-EXAM (non de l'institution, logo). Il a aussi à charge de gérer les utilisateurs, les établissements, les facultés, les départements, les promotions d'étudiants, leurs sections et leurs groupes.
- **Chef de département** : est une personne qui assure la gestion des utilisateurs affectés à son département ainsi que les matières enseignées et les examens affectés au niveau de son département.

### 3.2 Cas d'utilisation

Les cas d'utilisation sont des outils formels utilisés pour identifier et modéliser les besoins des utilisateurs [5]. Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de

l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service. [5].

Voici les cas d'utilisation de notre application :

Acteurs	Cas d'utilisation
Administrateur	Configurer Gest-exam : Définir l'institution (Nom, coordonnées, logo).
	Gérer les facultés : Ajouter/ Modifier / Supprimer une faculté.
	Gérer les départements : Ajouter/ Modifier / Supprimer un département d'une faculté déjà existante.
	Gérer les comptes des utilisateurs <ul style="list-style-type: none"> <li>• Ajouter / Modifier /Supprimer un utilisateur.</li> <li>• Importer une liste d'utilisateurs à partir d'un fichier CSV.</li> </ul>
	Désigner, parmi les utilisateurs créés, celui qui est le chef d'un département données Désaffecter un chef de département (supprimer juste l'affectation d'un utilisateur au département en qualité de chef de département).

Acteurs	Cas d'utilisation
Chef de département	Gérer les utilisateurs affectés au département (pour un semestre d'une année universitaire précise) : <ul style="list-style-type: none"> <li>• Gérer (Créer, Modifier, Supprimer) les enseignants.</li> <li>• Gérer (Créer, Modifier, Supprimer) les promotions.</li> <li>• Gérer (Créer, Modifier, Supprimer) les sections d'une promotion donnée.</li> <li>• Gérer (Créer, Modifier, Supprimer) les groupes d'une section.</li> </ul>
	Gérer (Créer, Modifier, Supprimer) les Modules (pour un semestre d'une année universitaire précise).
	Gérer les examens (pour un semestre d'une année universitaire précise) : <ul style="list-style-type: none"> <li>• Créer, Modifier, Supprimer un examen.</li> <li>• Créer, Modifier, Supprimer un planning d'examens.</li> <li>• Editer un rapport de déroulement par lieu d'examen.</li> </ul>
	Consulter et télécharger les informations d'un examen déroulé en classe.

Acteurs	Cas d'utilisation
Enseignant	Consulter le planning des examens pour une année et un semestre (En ligne).
	Lancer un processus d'examen (Hors ligne) : <ul style="list-style-type: none"> <li>• Déposer les données concernant cet examen et la liste des étudiants au format CSV (à télécharger depuis GestExam-Web).</li> <li>• Sauvegarde dans la base de données les informations et la liste des étudiants.</li> </ul>
	Configurer un examen (Hors ligne) : <ul style="list-style-type: none"> <li>• Ajouter / Modifier /Supprimer l'énoncé d'un examen.</li> <li>• Ajouter / Modifier / Supprimer le pré-imprimé d'un examen.</li> <li>• Ajouter / Modifier / Supprimer la solution type d'un examen en englobant le barème.</li> </ul>
	Gérer les copies d'examen (Hors ligne) : <ul style="list-style-type: none"> <li>• Numérisation.</li> <li>• Reconnaissance optique (générer des descripteurs des copies dans un fichier texte).</li> <li>• Identification (s'assurer que les copies trouvée par la reconnaissance optique de caractères correspondent bien aux étudiants existants dans la base de données).</li> </ul>
	Corriger les copies numériques d'un examen. (Hors ligne) : <ul style="list-style-type: none"> <li>• Corriger et sauvegarder la correction (notes par questions) pour chaque copie d'un examen.</li> <li>• Générer les résultats de la correction au format CSV ou XML.</li> <li>• Mise en ligne de la correction.</li> </ul>
	Gérer les réclamations (En ligne) : <ul style="list-style-type: none"> <li>• Traiter les messages de désaccords à propos d'un examen en provenance des étudiants.</li> <li>• Recorriger les copies conflictuelles correspondant à une réclamation approuvée par l'enseignant.</li> <li>• Mettre à jours les résultats en ligne en redéposant les nouvelles corrections (notes par questions pour chaque copie).</li> </ul>
	Clôturer la correction d'un examen. (En ligne).
	Effectuer une analyse docimologique et une analyse statistique d'un examen. (En ligne).

Acteurs	Cas d'utilisation
Etudiant	Consulter le planning des examens pour une année et un semestre.
	Consulter ça copie corrigée pour un examen donné.
	Formuler une réclamation pour une correction donnée.

**Tableau 1– Cas d'utilisation**



Afin de garantir la clarté les diagrammes, nous présenterons les cas d'utilisation selon les acteurs.

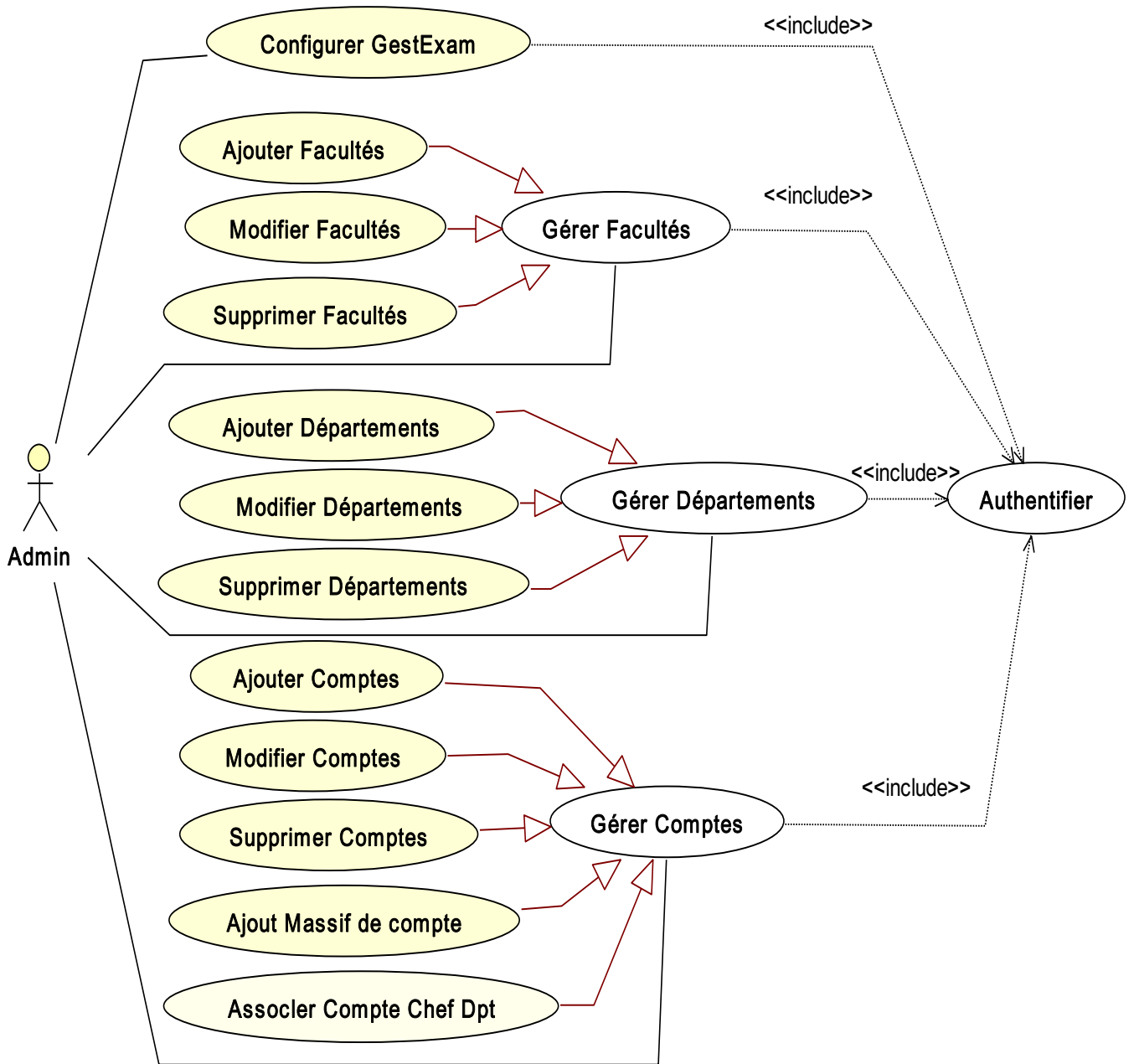


Figure 8-Cas d'utilisations pour un administrateur

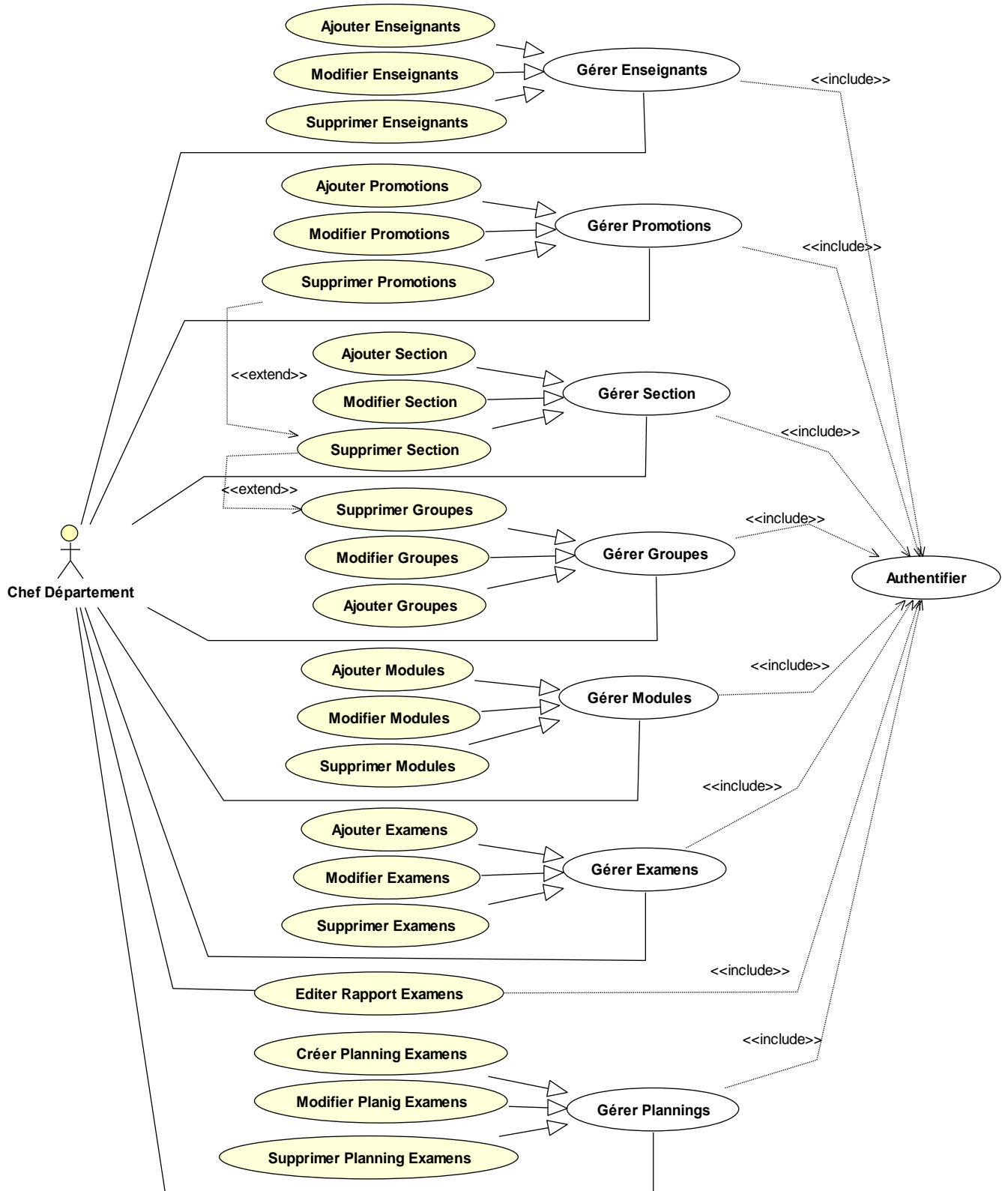


Figure 9-Cas d'utilisations pour un chef de département

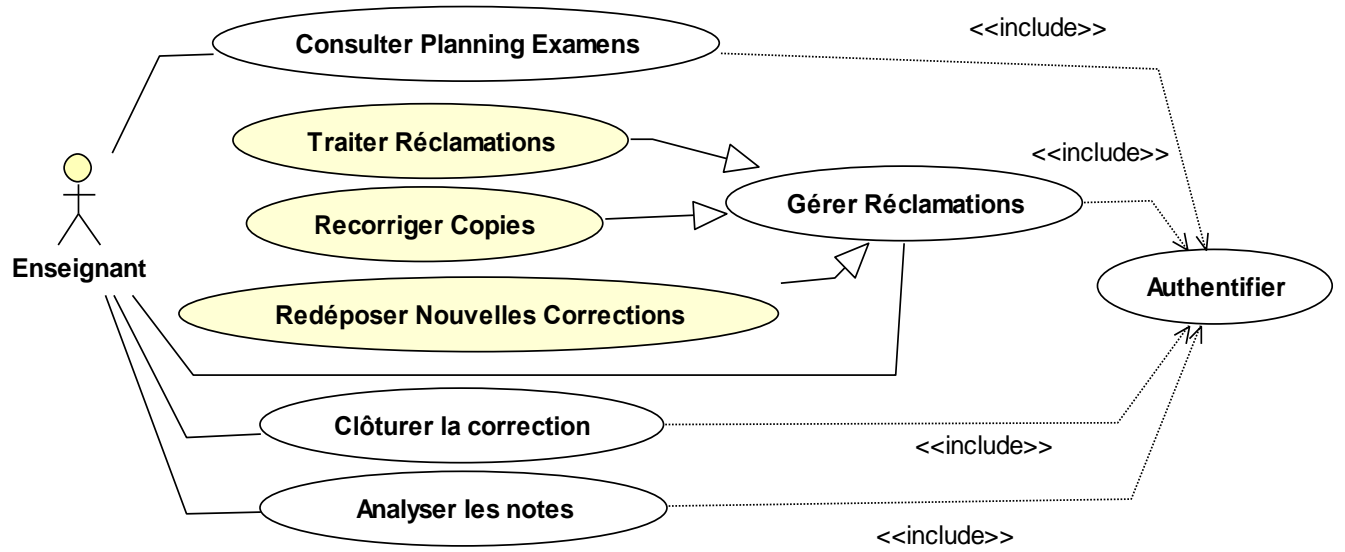


Figure 10– Cas d'utilisations pour un enseignant (en ligne)

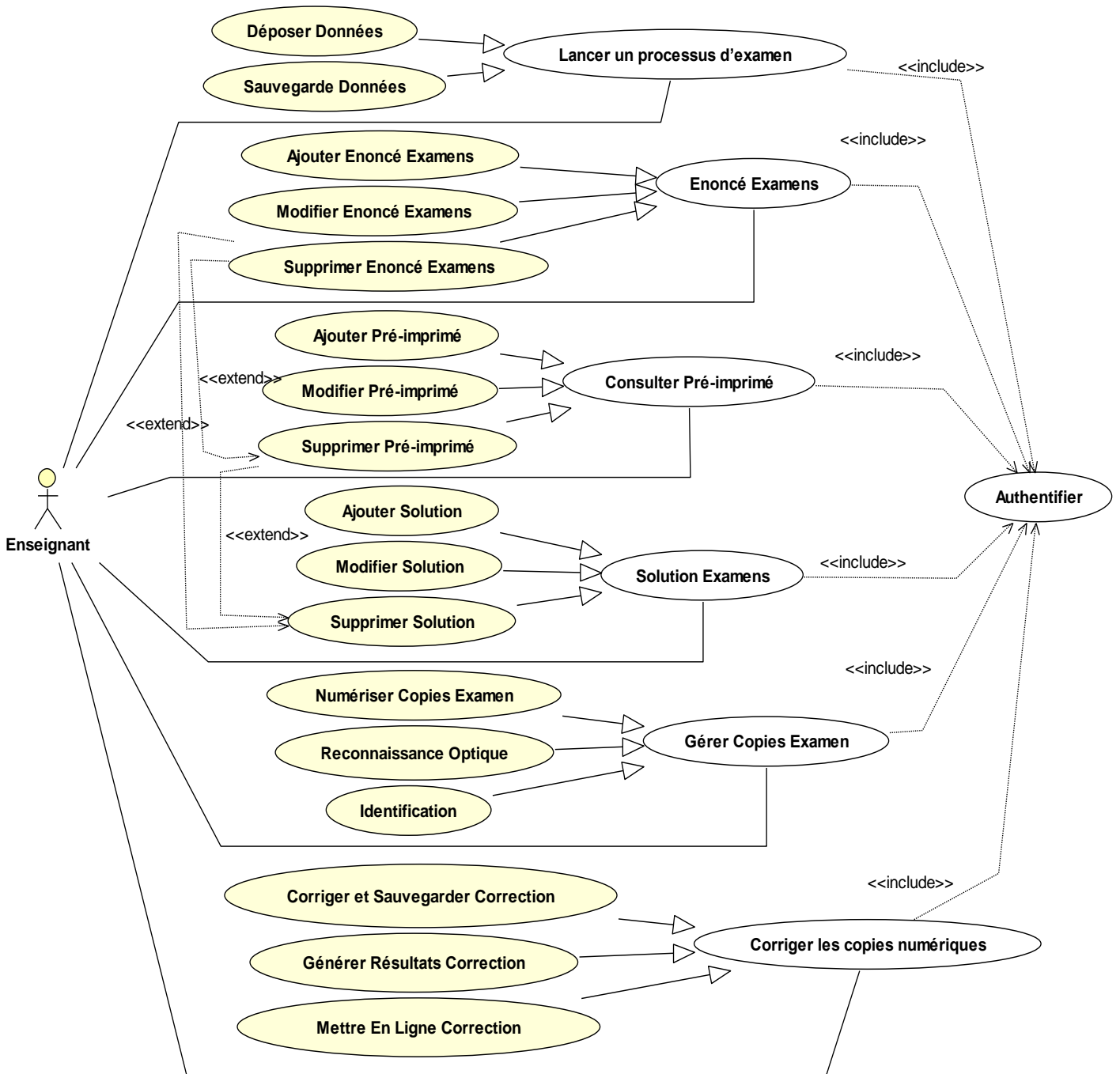


Figure 11– Cas d'utilisations pour un enseignant (hors ligne)

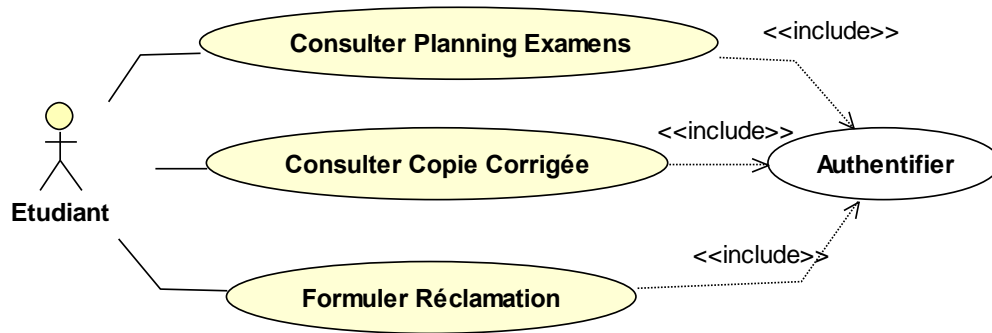


Figure 12– Cas d'utilisations pour un étudiant

### 3.3 Prototypage de l'Interface Homme/Machine

Il nous semble important, en parallèle avec le cahier de charge, d'esquisser les maquettes de l'IHM attendu (ou prototype) par le commanditaire. C'est un point de repère commun qui servira à nous orienter dans notre conception. Dans ce qui suit, nous présenterons les maquettes d'écran pour les trois utilisateurs de notre application : les enseignants, les étudiants et l'administration.

**Méthodologie :** D'après nos discussions avec le commanditaire, nous avons compris que nous devons réaliser deux modules distincts de l'application : *GestExam-Web* hors ligne et *GestExam* en ligne. Nous avons donc esquissé les maquettes d'écran pour ces deux modules et pour chacun des acteurs : Administrateur, Enseignant, Etudiants et Chef de département.

Plus de 30 maquettes IHM ont été discutées avec le commanditaire. Chacune d'elles est décrite en précisant, son numéro, les acteurs concernés, le lieu (en ligne ou hors ligne), les interactions engendrées et le visuel de la maquette.

Vu le nombre important de ces maquettes, nous ne présenterons ici que quelques exemples. Nous renvoyons le lecteur au document annexe spécialement élaboré pour le maquettage IHM.

### 3.4 Quelques Exemples de maquettes IHM

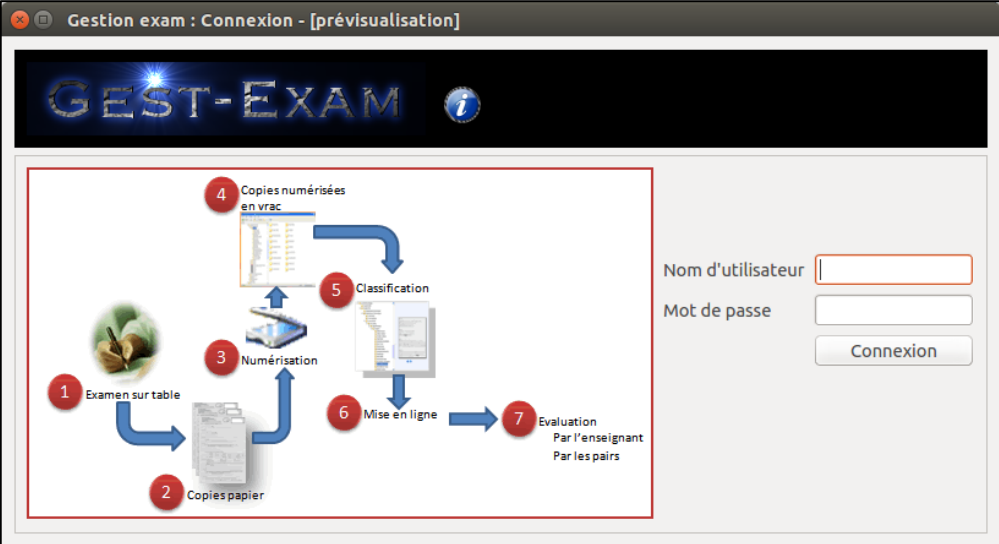
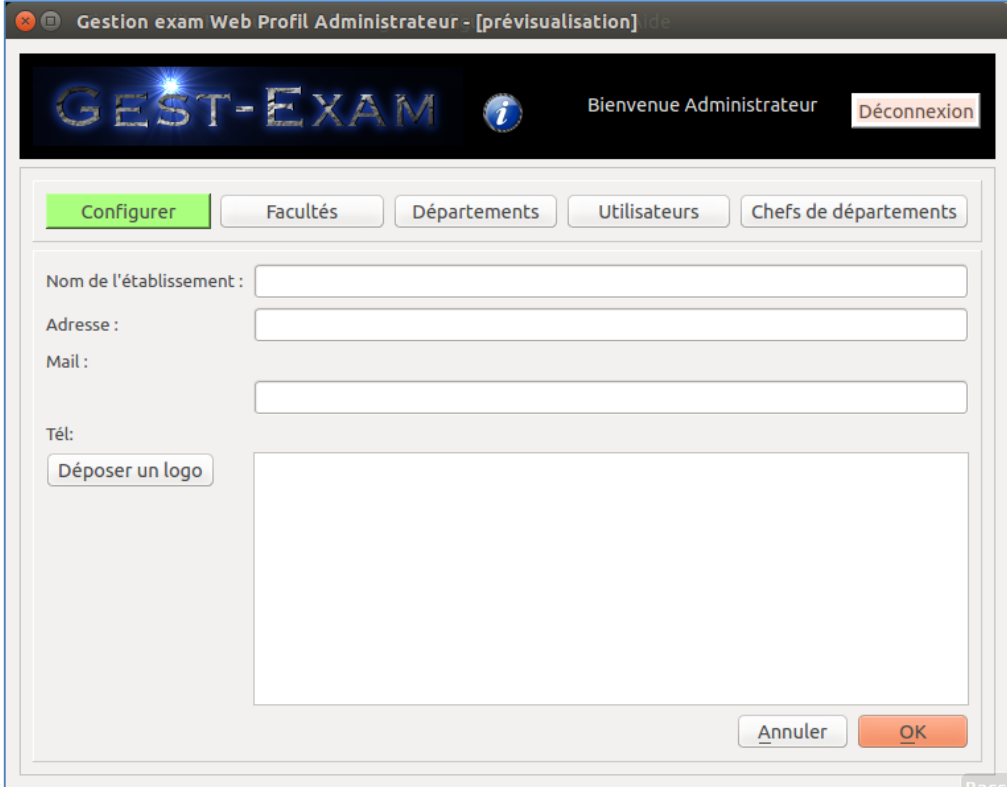
Numéro	IHM - 1
Utilisateurs	Tous les acteurs
Lieu	En ligne
Interaction	<p>Dans la partie en ligne de notre application, nous souhaitons que l'administrateur, les étudiants, les enseignants et le chef de département puissent accéder aux examens. Ainsi, il va falloir prévoir un système d'authentification permettant de reconnaître les uns et les autres afin de leurs associer l'IHM qui leur convient. Donc la première chose qui sera affichée en ligne pour tous les usagers de notre application est un écran comme celui de la figure ci-dessous. Il faut donc que l'internaute introduise un nom d'utilisateur et un mot de passe. Une fois cela fait, notre application vérifie l'identité de la personne et lui affiche l'IHM de l'administrateur, de l'enseignant, de l'étudiant ou du chef de département selon le profil de cette personne.</p> <p>Bien évidemment, il faut prévoir un message d'erreur d'authentification lorsque l'internaute saisie un nom d'utilisateur ou un mot de passe incorrecte.</p>
Maquette	

Tableau 2– La page d'accueil

Numéro	IHM-2
Utilisateurs	Administrateur
Lieu	En ligne
Interaction	L'administrateur pourra configurer notre application en indiquant le nom de l'établissement, son logo et ses coordonnées
Maquette	

**Tableau 3– IHM de l'administrateur (configuration)**

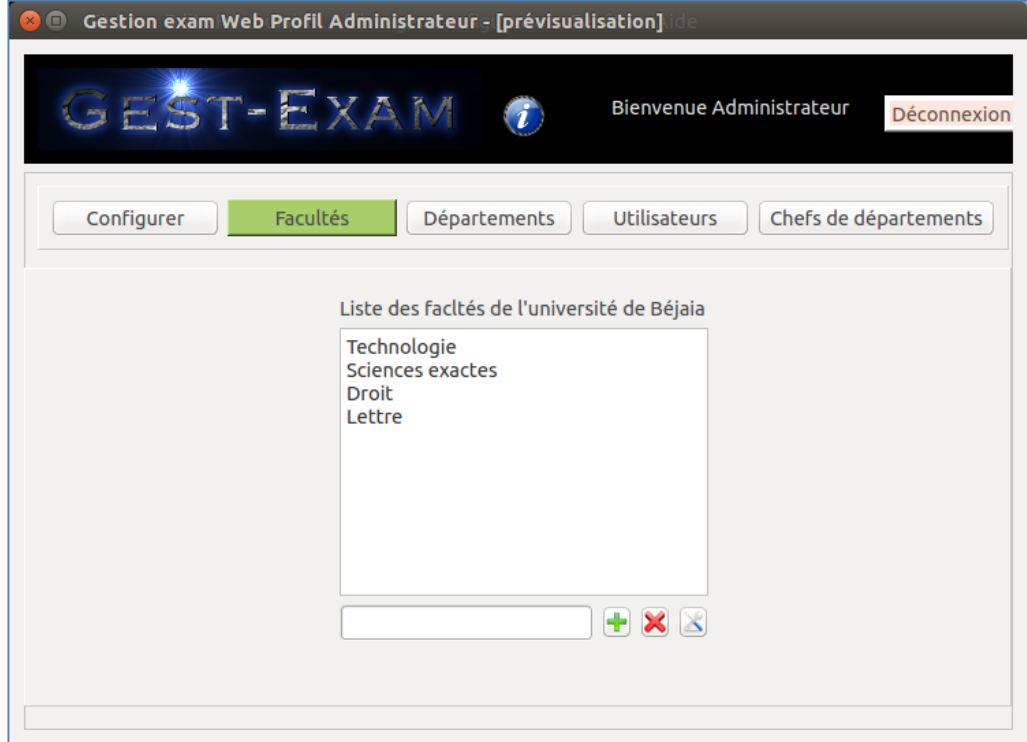
Numéro	IHM-3
Utilisateurs	Administrateur
Lieu	En ligne
Interaction	L'administrateur pourra ajouter modifier ou supprimer des facultés
Remarque	La suppression d'une faculté provoque la suppression de tous les départements qu'elle renferme. A cet effet, il faut prévoir une boîte de dialogue de confirmation de la suppression tout en informant l'administrateur du risque qu'il prend en réalisant cette opération.
Maquette	 <p>The screenshot shows a web application window titled "Gestion exam Web Profil Administrateur - [prévisualisation]". The header features the "GEST-EXAM" logo, an information icon, the text "Bienvenue Administrateur", and a "Déconnexion" button. Below the header is a navigation menu with buttons for "Configurer", "Facultés" (highlighted in green), "Départements", "Utilisateurs", and "Chefs de départements". The main content area displays "Liste des factés de l'université de Béjaia" with a list containing: Technologie, Sciences exactes, Droit, and Lettre. At the bottom of the list, there is an empty input field and three icons: a green plus sign, a red X, and a blue refresh icon.</p>

Tableau 4– IHM de l'administrateur (Gestion des facultés)



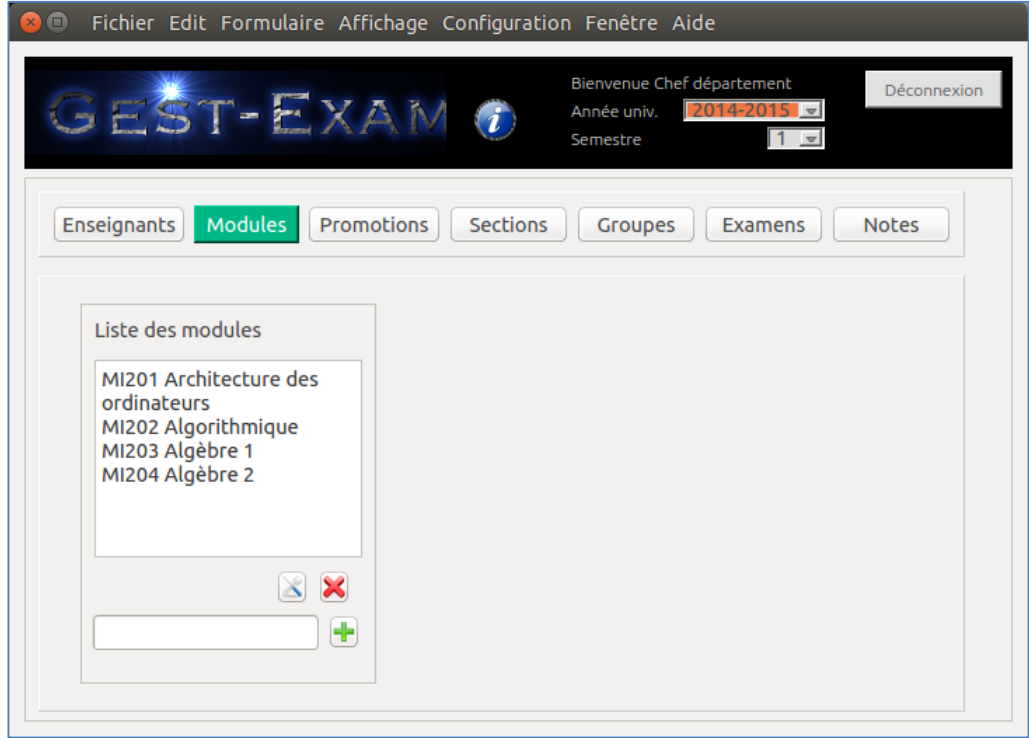
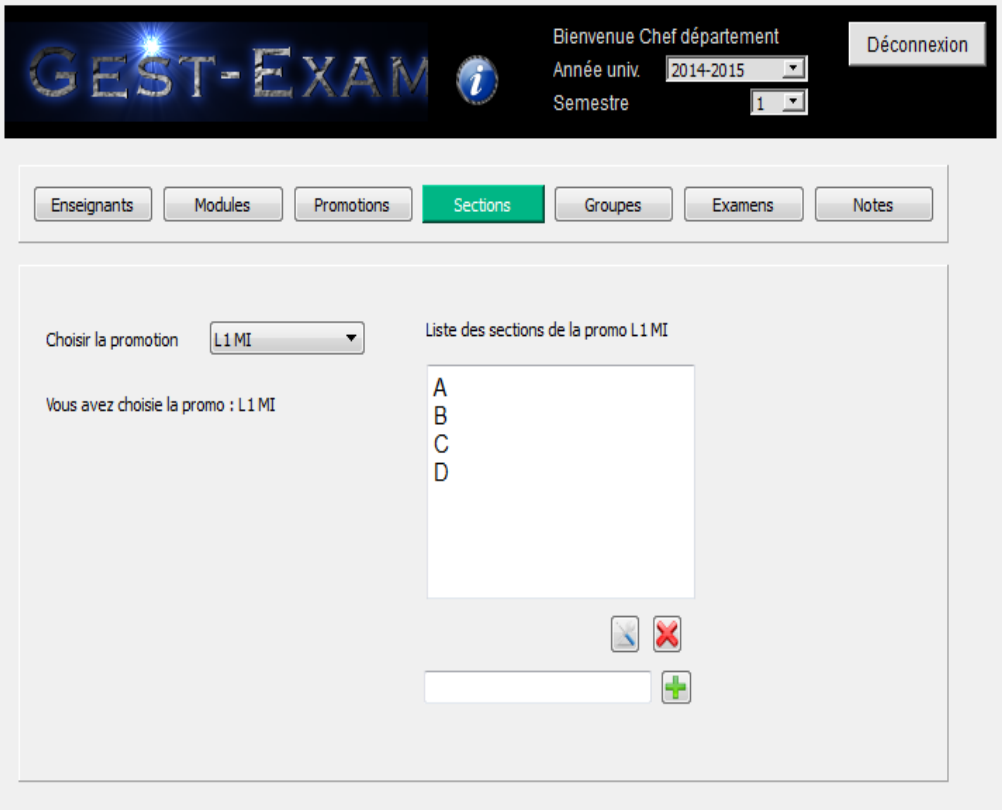
Numéro	IHM-8
Utilisateurs	Chef de département
Lieu	En ligne
Interaction	Le chef de département pourra créer et gérer (modifier ou supprimer) les modules (matières) enseignés dans son département
Remarque	La suppression d'un module n'est pas possible si un examen le concernant a été créé. Il faut d'abord supprimer cet examen.
Maquette	 <p>The screenshot shows a web browser window titled 'GEST-EXAM'. The interface includes a menu bar (Fichier, Edit, Formulaire, Affichage, Configuration, Fenêtre, Aide) and a user header with 'Bienvenue Chef département', 'Année univ. 2014-2015', 'Semestre 1', and a 'Déconnexion' button. Below the header is a navigation bar with tabs for 'Enseignants', 'Modules' (selected), 'Promotions', 'Sections', 'Groupes', 'Examens', and 'Notes'. The main content area displays a 'Liste des modules' with a scrollable list containing: MI201 Architecture des ordinateurs, MI202 Algorithmique, MI203 Algèbre 1, and MI204 Algèbre 2. At the bottom of the list are icons for zooming, deleting, and adding items.</p>

Tableau 5– IHM de l’administrateur (Gestion des modules)

Numéro	IHM-10
Utilisateurs	Chef de département
Lieu	En ligne
Interaction	Le chef de département pourra créer les sections d'une promotion donnée
Remarque	<ul style="list-style-type: none"> <li>➤ La section appartient nécessairement à une promotion. Donc, le chef de département doit d'abord sélectionner une promotion pour pouvoir créer ses sections.</li> <li>➤ La suppression d'une section entraine automatiquement la suppression des groupes quelle renferme. Un message d'alerte et de confirmation doit être affiché en réponse à cette action.</li> </ul>
Maquette	 <p>The screenshot displays the 'Gest-Exam' web application interface. At the top, there is a header with the logo 'GEST-EXAM' and a user welcome message 'Bienvenue Chef département'. Below the header, there are navigation buttons: 'Enseignants', 'Modules', 'Promotions', 'Sections' (highlighted in green), 'Groupes', 'Examens', and 'Notes'. The main content area shows a form for creating sections. It includes a dropdown menu labeled 'Choisir la promotion' with 'L1 MI' selected. Below this, it says 'Vous avez choisie la promo : L1 MI'. To the right, there is a list titled 'Liste des sections de la promo L1 MI' containing the letters 'A', 'B', 'C', and 'D'. At the bottom of the list, there are icons for editing (pencil), deleting (red X), and adding (green plus) sections.</p>

**Tableau 6– IHM de Chef du département (Gestion des sections)**

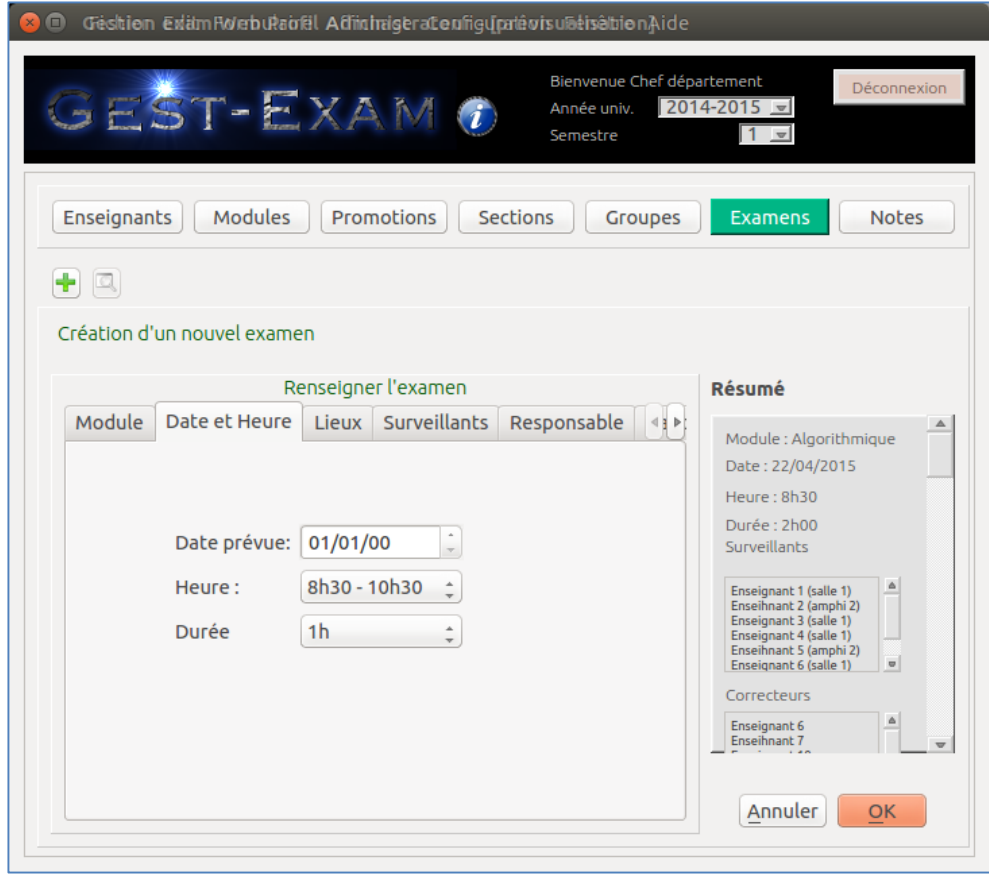
<b>Numéro</b>	<b>IHM-13</b>
Utilisateurs	Chef de département
Lieu	En ligne
Interaction	Le chef de département pourra créer des examens (indiquer la date et l'horaire)
Remarque	
Maquette	 <p>The screenshot shows a web browser window titled 'Gestion Exam Web'. The page header includes the 'GEST-EXAM' logo, a welcome message 'Bienvenue Chef département', and a 'Déconnexion' button. Below the header is a navigation menu with buttons for 'Enseignants', 'Modules', 'Promotions', 'Sections', 'Groupes', 'Examens' (highlighted in green), and 'Notes'. The main content area is titled 'Création d'un nouvel examen' and contains a form with two tabs: 'Renseigner l'examen' and 'Résumé'. The 'Renseigner l'examen' tab is active and shows fields for 'Date prévue' (01/01/00), 'Heure' (8h30 - 10h30), and 'Durée' (1h). The 'Résumé' tab shows a list of exam details: 'Module : Algorithmique', 'Date : 22/04/2015', 'Heure : 8h30', 'Durée : 2h00', and 'Surveillants' (a list of six instructors). At the bottom of the form are 'Annuler' and 'OK' buttons.</p>

Tableau 7– IHM de Chef du département (Gestion des examens)

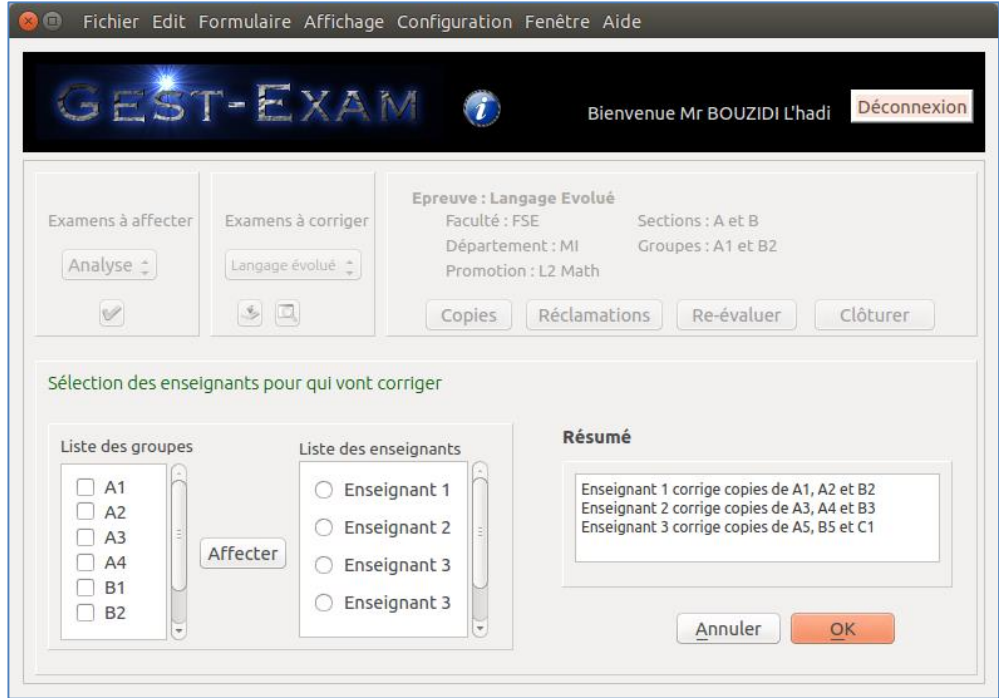
Numéro	IHM- 18
Utilisateurs	Enseignant (Responsable de module)
Lieu	En ligne
Interaction	L’enseignant responsable de module doit indiquer la liste des enseignants qui vont corriger un examen dont il est responsable
Remarque	Les groupes déjà affectés, ne doivent pas apparaître dans la liste des groupes. Ceci va garantir que les copies d’un groupe ne seront corrigées que par un seul enseignant
Maquette	 <p>The screenshot shows a web application window titled 'GEST-EXAM'. The user is logged in as 'Mr BOUZIDI L'hadi'. The interface includes a menu bar (Fichier, Edit, Formulaire, Affichage, Configuration, Fenêtre, Aide) and a header with the application name and a 'Déconnexion' button. The main content area is divided into several sections: 'Examens à affecter' with a dropdown menu set to 'Analyse'; 'Examens à corriger' with a dropdown menu set to 'Langage évolué'; and a summary section for the current exam 'Epreuve : Langage Evolué' showing details like 'Faculté : FSE', 'Département : MI', and 'Promotion : L2 Math'. Below this, there are buttons for 'Copies', 'Réclamations', 'Re-évaluer', and 'Clôturer'. The central part of the interface is titled 'Sélection des enseignants pour qui vont corriger' and contains two lists: 'Liste des groupes' (A1, A2, A3, A4, B1, B2) and 'Liste des enseignants' (Enseignant 1, Enseignant 2, Enseignant 3). An 'Affecter' button is positioned between these lists. To the right, a 'Résumé' box displays the assignment logic: 'Enseignant 1 corrige copies de A1, A2 et B2', 'Enseignant 2 corrige copies de A3, A4 et B3', and 'Enseignant 3 corrige copies de A5, B5 et C1'. At the bottom right, there are 'Annuler' and 'OK' buttons.</p>

Tableau 8– IHM de l’enseignant (Gestion de la liste des enseignants)

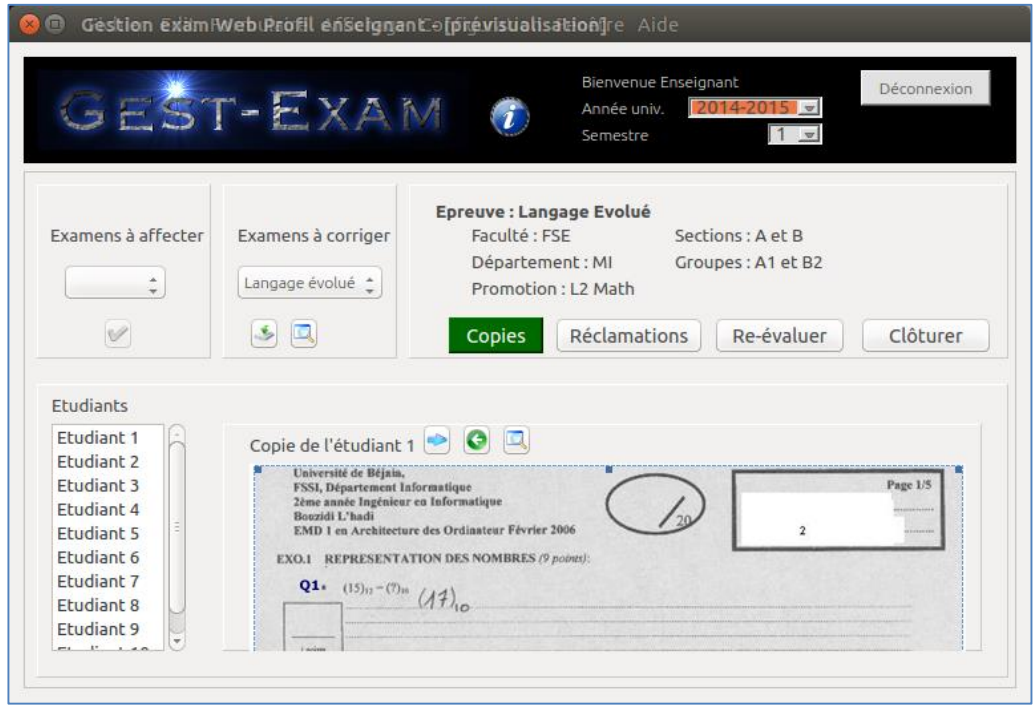
Numéro	IHM - 19
Utilisateurs	Enseignant (Responsable de module)
Lieu	En ligne
Interaction	L'enseignant chargé de corriger des copies peut consulter en ligne les copies numérisées
Remarque	Seules les copies des groupes dont l'enseignant à la charge de correction de l'examen seront affichées Pour chaque étudiant, l'enseignant pourra voir chacune des pages de sa copie d'examen.
Maquette	

Tableau 9– IHM de l'enseignant (corriger des copies)

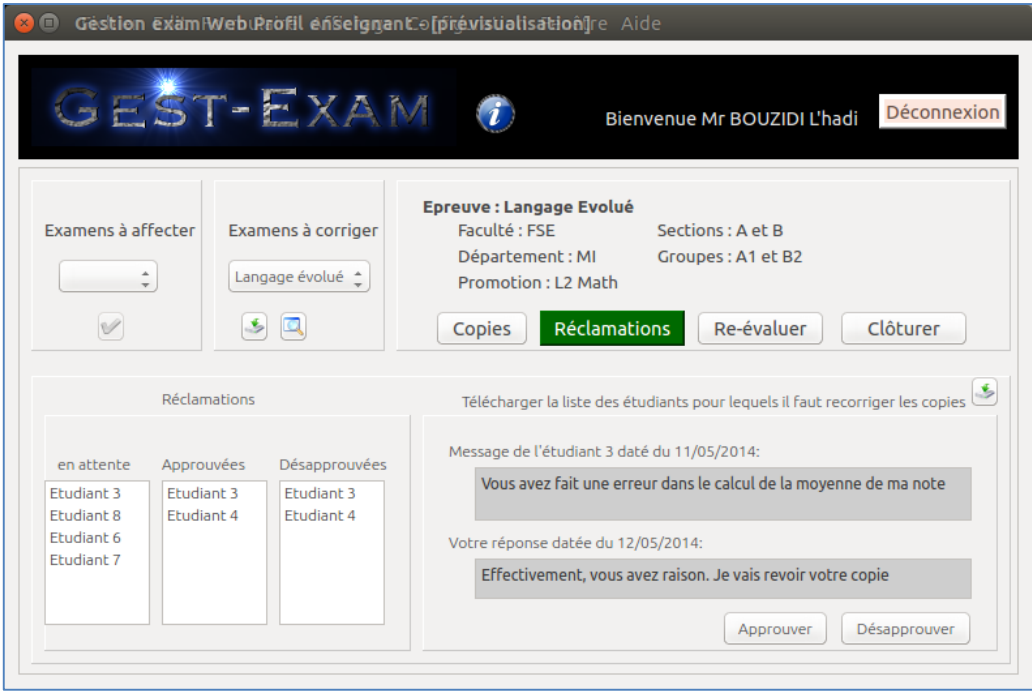

Numéro	IHM - 20
Utilisateurs	Enseignant (Responsable de module)
Lieu	En ligne
Interaction	L'enseignant chargé de corriger des copies peut gérer les réclamations des étudiants. Il peut notamment approuver ou désapprouver des réclamations.
Remarque	<ul style="list-style-type: none"> <li>• Seuls les étudiants qui ont formulé des réclamations seront affichés dans cette IHM</li> <li>• Un étudiant peut poster plusieurs messages.</li> <li>• Dès qu'un message est posté par un étudiant, il sera mis dans la liste des réclamations en attente.</li> <li>• Dès que l'enseignant approuve une réclamation, elle sera retirée de la liste des réclamations en attente pour être mise dans la liste des réclamations approuvées.</li> <li>• Dès que l'enseignant désapprouve une réclamation, elle sera retirée de la liste des réclamations en attente pour être mise dans la liste des réclamations désapprouvées.</li> <li>• Pour chaque message posté par un étudiant, l'enseignant peut répondre ou non.</li> <li>• L'enseignant peut télécharger les noms des étudiants pour lesquels il a approuvé les réclamations. Cette liste servira GestExam hors ligne pour reprendre (hors ligne) la correction de leurs copies.</li> </ul>
Maquette	

Tableau 10– IHM de l'enseignant (Gestion des réclamations des étudiants)

Numéro	IHM - 21
Utilisateurs	Enseignant (Correcteur)
Lieu	Hors ligne
Interaction	L'enseignant pourra lancer une évaluation en téléchargeant les données qui le concerne sur un examen planifié par le chef de département. A cette étape l'examen n'a pas encore eu lieu en classe, mais juste programmé. L'enseignant va ainsi débiter un processus en 14 étape pour mener à bien la correction des copies qui lui sont affectées.
Remarque	Le lancement d'une nouvelle évaluation se fait en cliquant sur le bouton « + » du tableau de bord. Ainsi automatiquement, on se positionne sur la première phase du processus (qui en compte 14). La seule action possible est de déposer les données concernant l'examen à corriger dans ce processus d'évaluation. Ainsi, une boîte de dialogue demandant de d'indiquer un fichier CSV va s'afficher. Ce fichier va être prise en compte et permettant d'avoir toutes les informations utiles pour lancer l'évaluation.
Maquette	

**Tableau 11– IHM de l’enseignant (lancer une évaluation)**

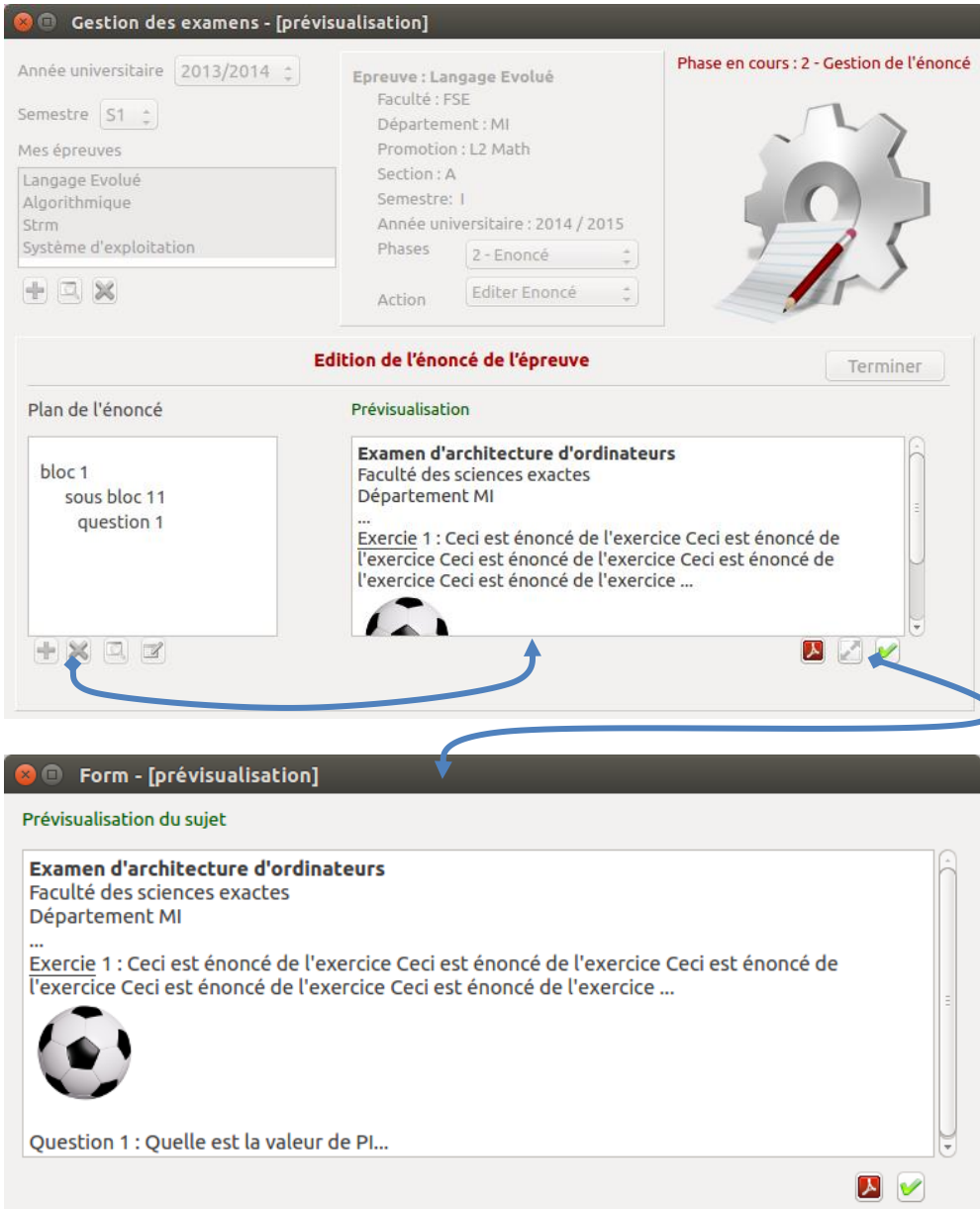
Numéro	IHM - 29
Utilisateurs	Enseignant (Correcteur)
Lieu	Hors ligne
Interaction	
Remarque	<p data-bbox="183 994 312 1028">Maquette</p> 

Tableau 12– IHM de l’enseignant (prévisualisation du sujet)



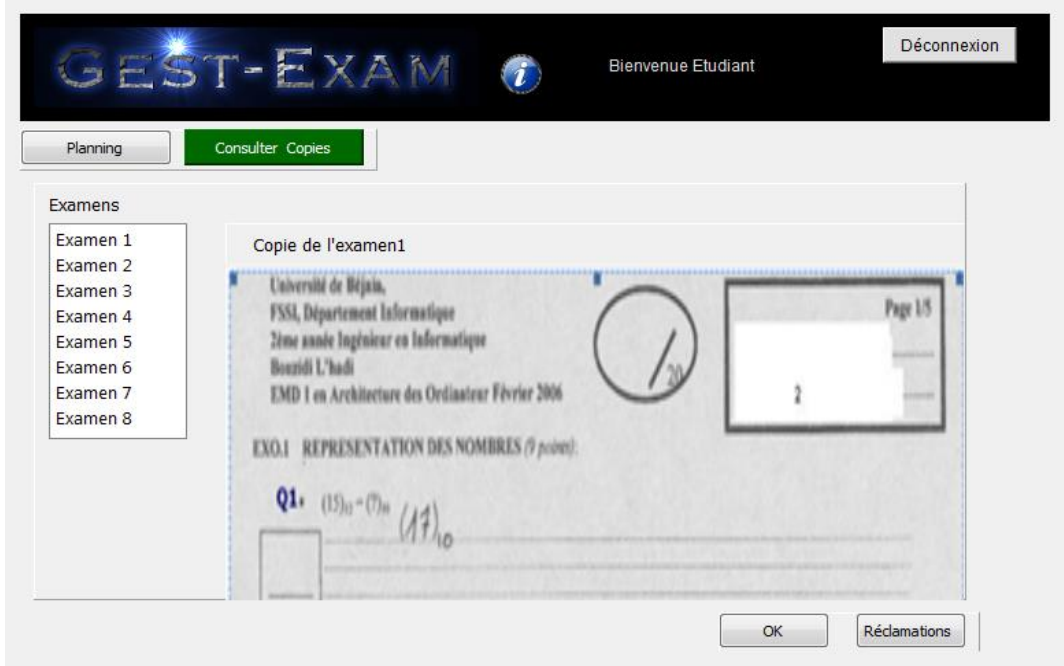
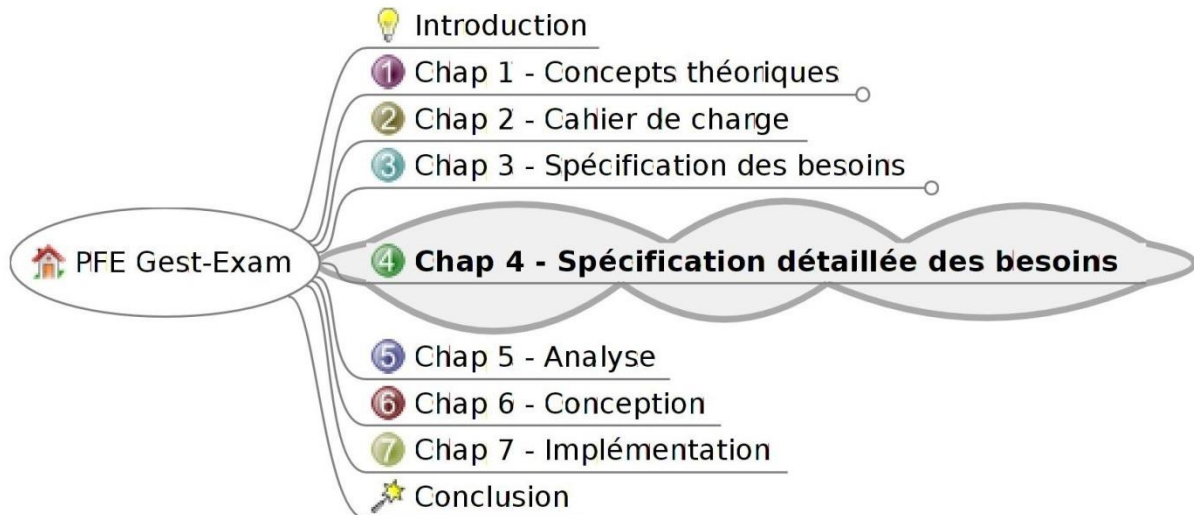
Numéro	IHM- 30
Utilisateurs	Etudiant
Lieu	En ligne
Interaction	L'étudiant doit être consulter ça copie d'examen et de formuler une réclamation.
Remarque	La liste des modules ne doit pas être vide.
Maquette	

Tableau 13– IHM de l'étudiant (consultation des copies)

## Chapitre.4 – Spécification détaillée des besoins

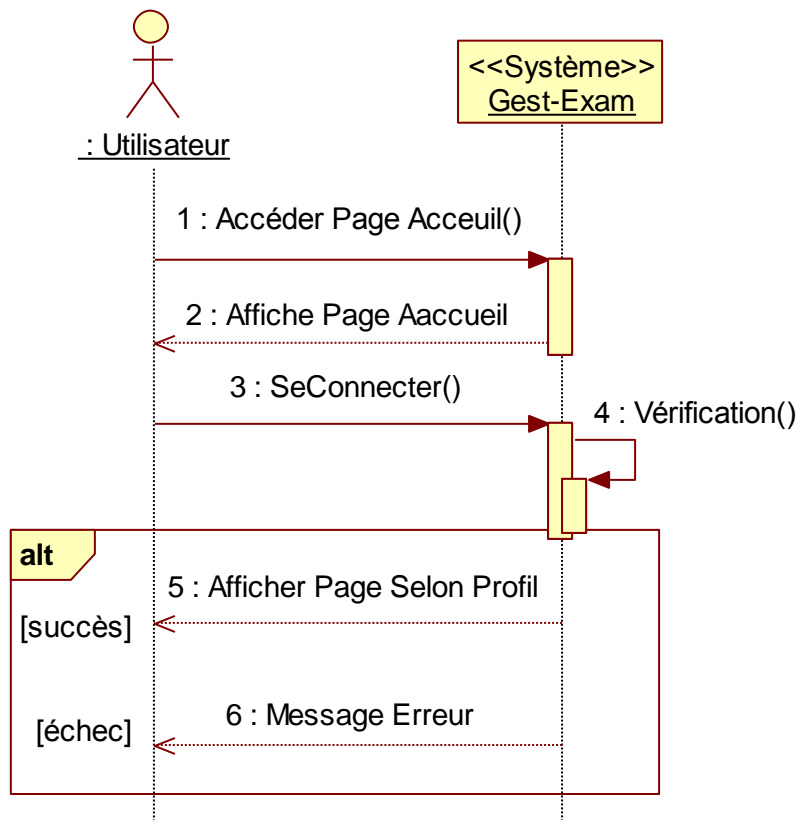


*Les interactions entre les acteurs et le système seront explicitées sous forme textuelle et sous forme graphique au moyen de diagrammes de séquence. Les utilisateurs ont souvent beaucoup de difficultés à exprimer clairement et précisément ce qu'ils attendent du système. L'objectif de cette étape est justement de les aider à formuler et formaliser ces besoins.*

*On va décrire avec les diagrammes de séquence les cas d'utilisation cité précédemment.*

**Diagramme de séquence du cas d'utilisation «Authentification»**

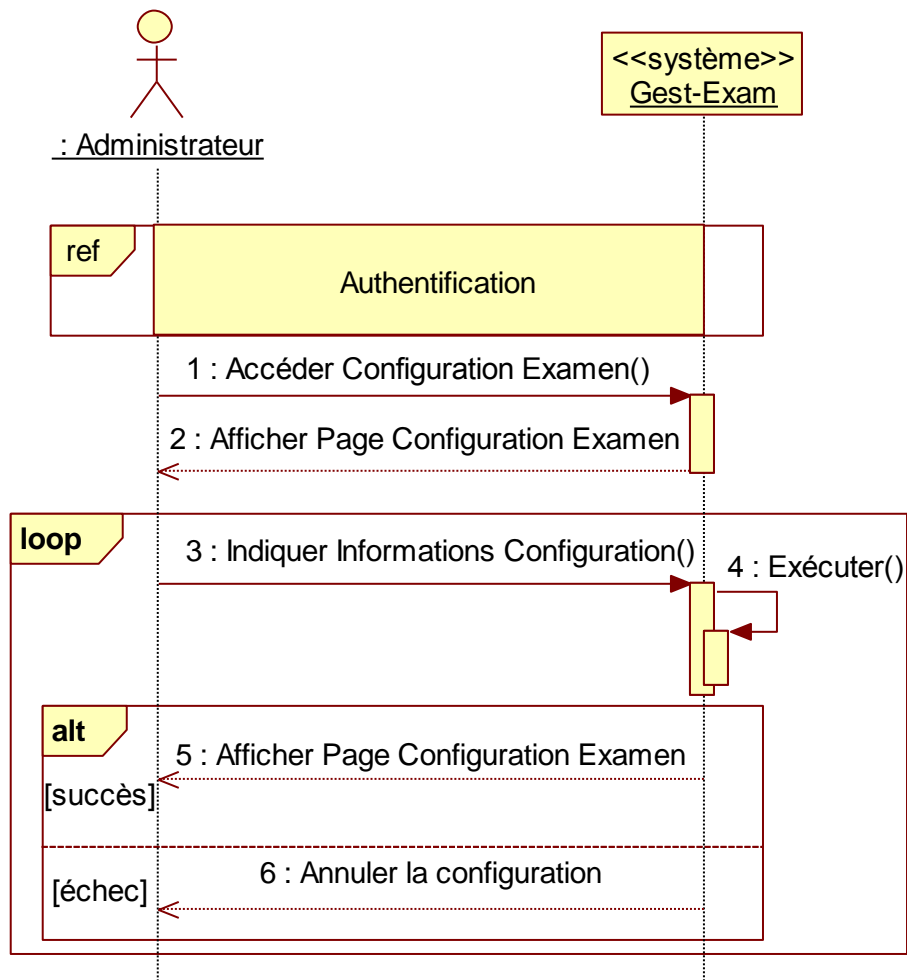
Cas d'utilisation	Authentification
Acteur principal	Utilisateur
Acteur secondaire	Néant
Objectif	Permettre à l'utilisateur de s'authentifier afin de pouvoir accéder aux services de l'application GestExam –Web.
Pré-condition	Accès à la page d'accueil de l'application GestExam –Web.
Post-conditions	Néant
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la page d'accueil de l'application GestExam-Web.</li> <li>2. Le système affiche la page d'accueil de l'application.</li> <li>3. L'utilisateur indique son login et son mot de passe sur la page d'accueil de l'application GestExam-Web et valide sa connexion en appuyant sur un bouton « Connexion ».</li> <li>4. Le système vérifie si l'utilisateur (login et mot) existe dans sa BDD et récupère son profil.</li> <li>5. Selon le profil récupéré, le système affiche la page associée.</li> </ol>
Alternative	<ol style="list-style-type: none"> <li>6. Le système affiche un message d'erreur d'authentification.</li> </ol>



**Figure 13 DSS 1-Cas d'utilisation : Authentification**

**Diagramme de séquence du cas d'utilisation «Configurer GestExam–Web»**

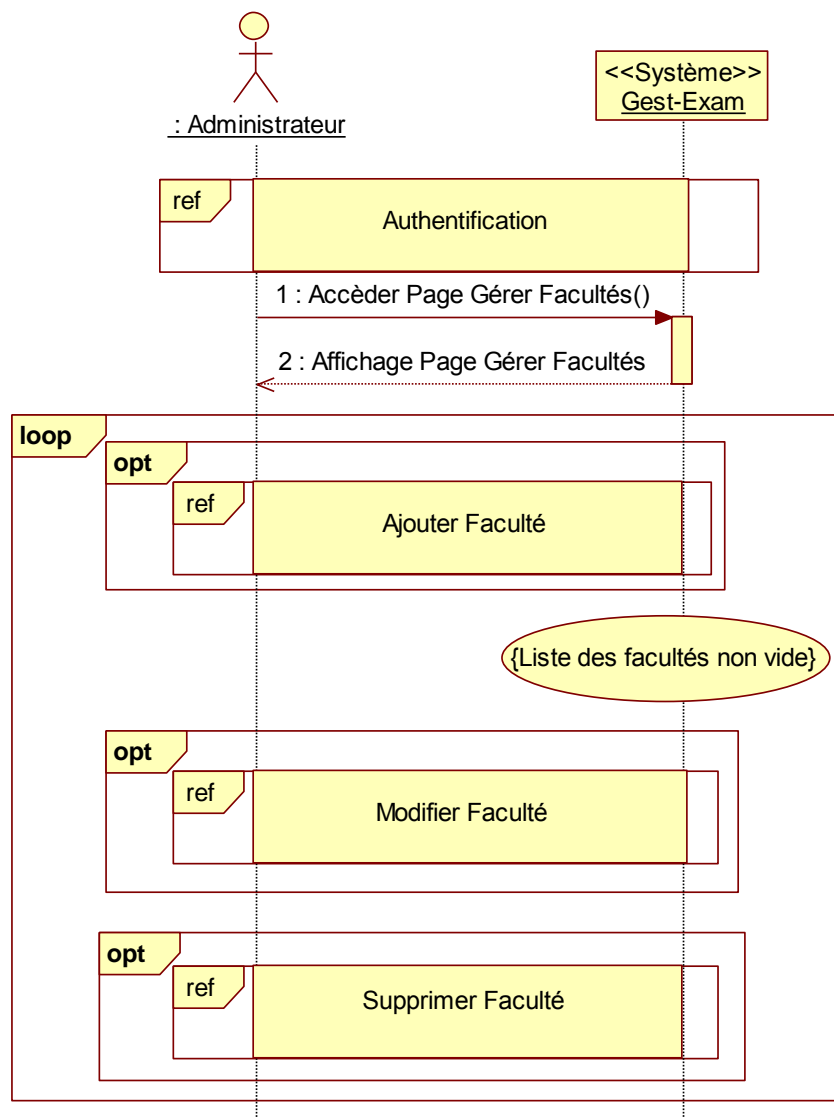
Cas d'utilisation	Configurer GestExam–Web
Acteur principal	Administrateur
Acteur secondaire	Néant
Objectif	Permettre à l'administrateur de configurer GestExam–Web.
Pré-condition	Administrateur connecté (authentification).
Post-conditions	Néant
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur accède à l'espace de « configuration GestExam-Web » en cliquant sur le menu « Configurer » de la barre de navigation.</li> <li>2. Le système affiche la page « Configurer GestExam -Web ».</li> <li>3. L'administrateur indique le nom, l'adresse, le mail, le tél et le logo de l'établissement et valide en cliquant sur le bouton « Valider ».</li> <li>4. Le système exécute la requête en mettant à jour sa base de données.</li> <li>5. Le système retourne la page de configuration « GestExam -Web ».</li> </ol>
Alternative	6. L'administrateur annule la configuration de GestExam -Web en cliquant sur le bouton « Annuler ».



**Figure 14 DSS2-Cas d'utilisation : Configurer GestExam -Web (Administrateur)**

**Diagramme de séquence du cas d'utilisation «Gérer Facultés »**

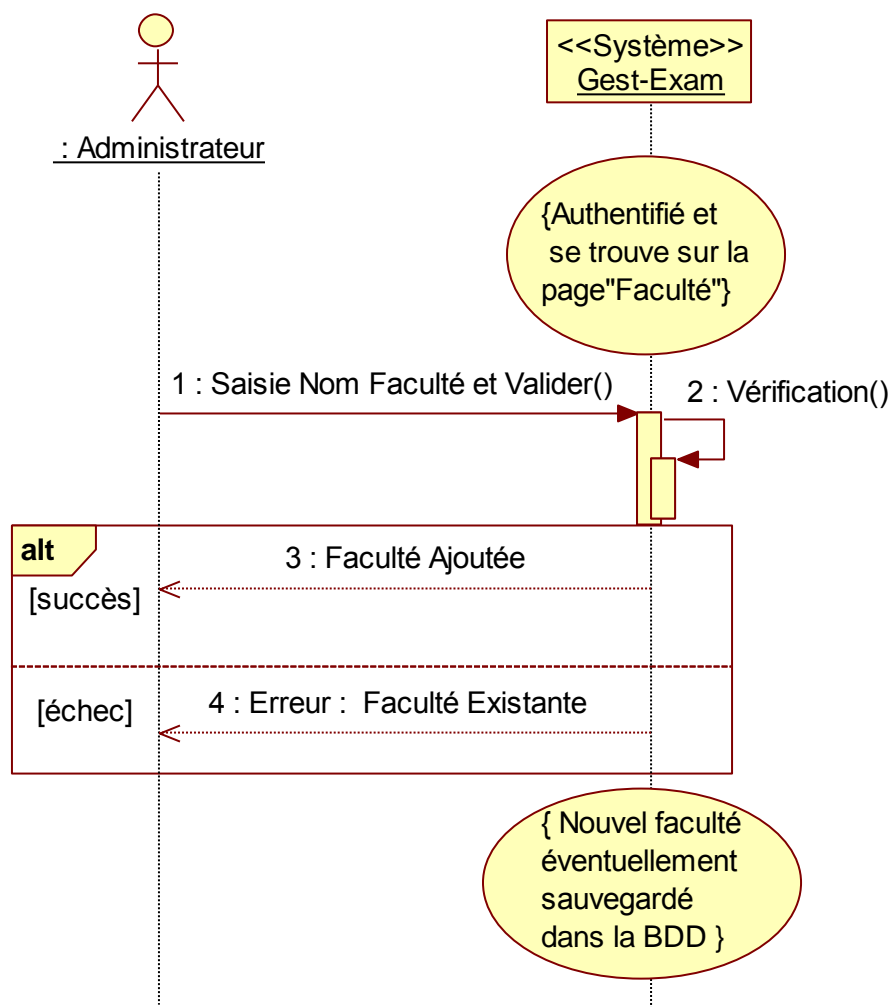
Cas d'utilisation	Gérer Facultés
Acteur principal	Administrateur
Acteur secondaire	Néant
Objectif	Permettre à l'administrateur d'ajouter, supprimer ou modifier des facultés.
Pré-condition	Authentification.
Post-conditions	Néant
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur accède à l'espace « Gérer Facultés » en cliquant sur le menu «Facultés» dans la barre de navigation.</li> <li>2. Le système affiche la liste des facultés créés et formulaire de saisie pour ajouter une faculté.</li> <li>3. L'administrateur choisi l'action à effectuer(ajouter, modifier,supprimer) : <ol style="list-style-type: none"> <li>a. Ajout : voir le cas d'utilisation « Ajouter Faculté ».</li> <li>b. Suppression : voir le cas d'utilisation « Supprimer Faculté ».</li> <li>c. Modification : voir le cas d'utilisation « Modifier Facultés ».</li> </ol> </li> </ol>
Alternative	Néant



**Figure 15 DSS3-Cas d'utilisation : Gérer Facultés(Administrateur)**

**Diagramme de séquence du cas d'utilisation «Ajouter Faculté »**

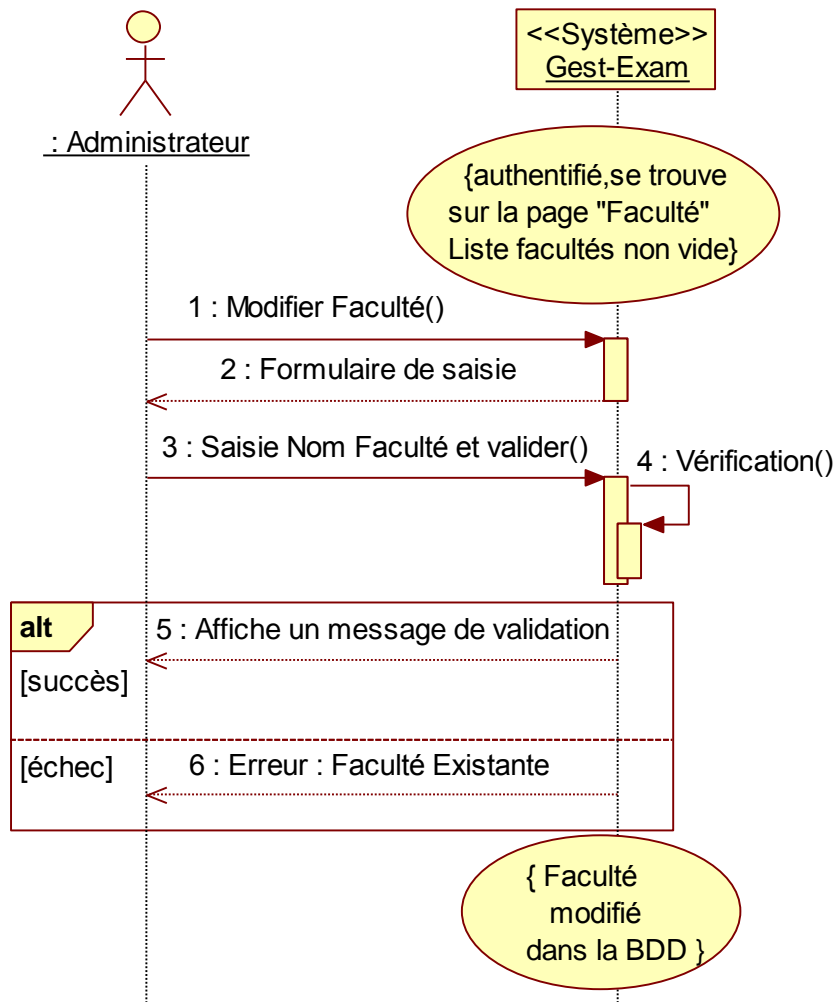
Cas d'utilisation	Ajouter Faculté
Acteur principal	Administrateur
Acteur secondaire	Néant
Objectif	Permettre à l'administrateur d'ajouter une faculté en indiquant son nom.
Pré-condition	Administrateur connecté (authentification) et accède à la page «Facultés».
Post-conditions	La nouvelle faculté devra être rajouté à la base de donnée.
Scénario nominal	1 L'administrateur saisie le nom d'une nouvelle faculté et valide en cliquant sur l'icône d'ajout. 2 Le système vérifie si la faculté n'existe pas déjà. 3 Le système confirme l'ajout.
Alternative	2a. La faculté existe déjà : Le système renvoie un message d'erreur.



**Figure 16 DSS4-Cas d'utilisation : Ajouter Faculté (Administrateur)**

**Diagramme de séquence du cas d'utilisation «Modifier Faculté»**

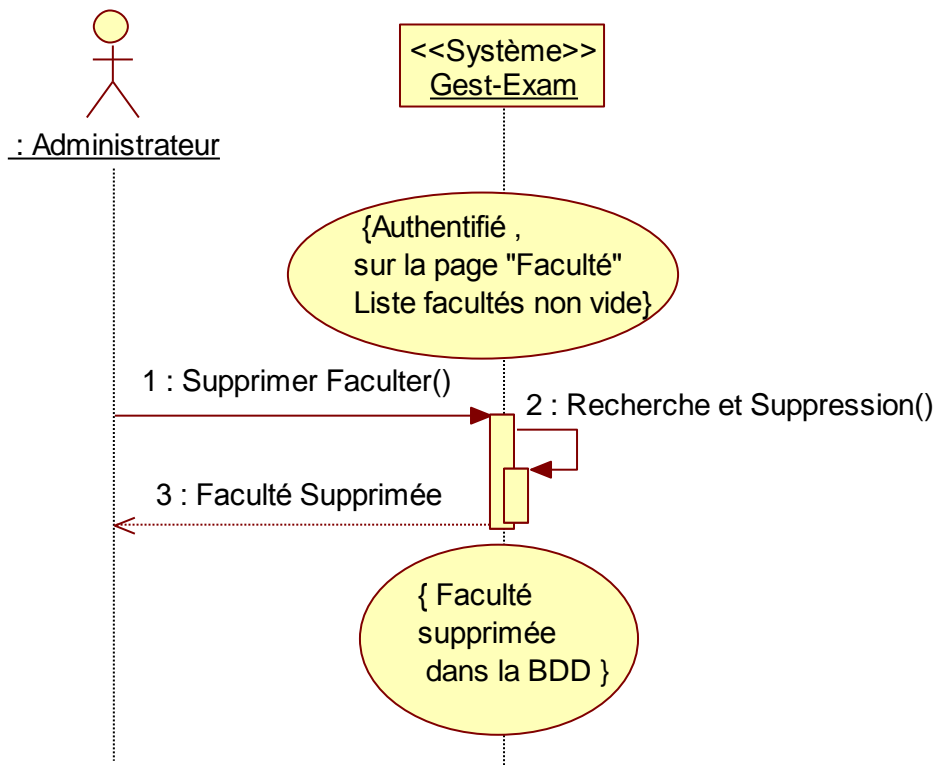
Cas d'utilisation	Modifier Faculté
Acteur principal	Administrateur
Acteur secondaire	Néant
Objectif	Permettre à l'administrateur de modifier les informations concernant une faculté.
Pré-condition	L'administrateur doit être connecté et doit être sur la page «Facultés». La liste des facultés doit être non vide.
Post-conditions	La faculté devra être modifiée dans la base de donnée.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur clique sur l'icône «modifier».</li> <li>2. Le système affiche un formulaire de saisie.</li> <li>3. L'administrateur fait les modifications nécessaires et valide en cliquant sur le bouton «valider».</li> <li>4. Le système vérifie s'il n'existe pas une faculté portant le même nom.</li> <li>5. Le système affiche un message de validation.</li> </ol>
Alternative	4a. La faculté existe déjà : Le système retourne une erreur.



**Figure 17 DSS5-Cas d'utilisation : Modifier Faculté(Administrateur)**

**Diagramme de séquence du cas d'utilisation « Supprimer Faculté »**

Cas d'utilisation	Supprimer Faculté
Acteur principal	Administrateur
Acteur secondaire	Néant
Objectif	Permettre à l'administrateur de supprimer une faculté à tout moment en cliquant sur l'icône «supprimer» .
Pré-condition	L'administrateur doit être connecté et doit être sur la page «Facultés». La liste des facultés doit être non vide.
Post-conditions	La faculté supprimée devra être retirée dans la base de donnée.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur clique sur l'icône «supprimer».</li> <li>2. Le système fait une recherche et supprime la faculté sélectionnée.</li> <li>3. Le système affiche que la faculté à bien été supprimée.</li> </ol>
Alternative	Néant

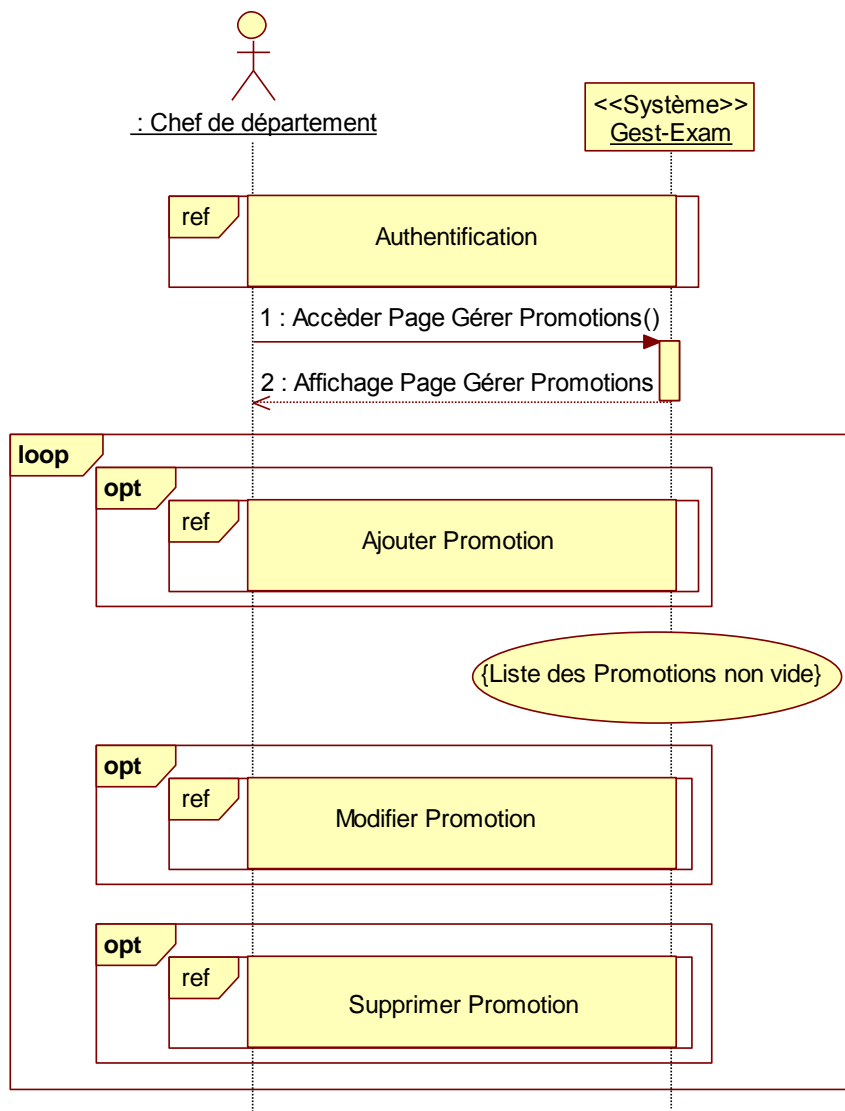


**Figure 18 DSS6-Cas d'utilisation : Supprimer Faculté (Administrateur)**



**Diagramme de séquence du cas d'utilisation «Gérer Promotions»**

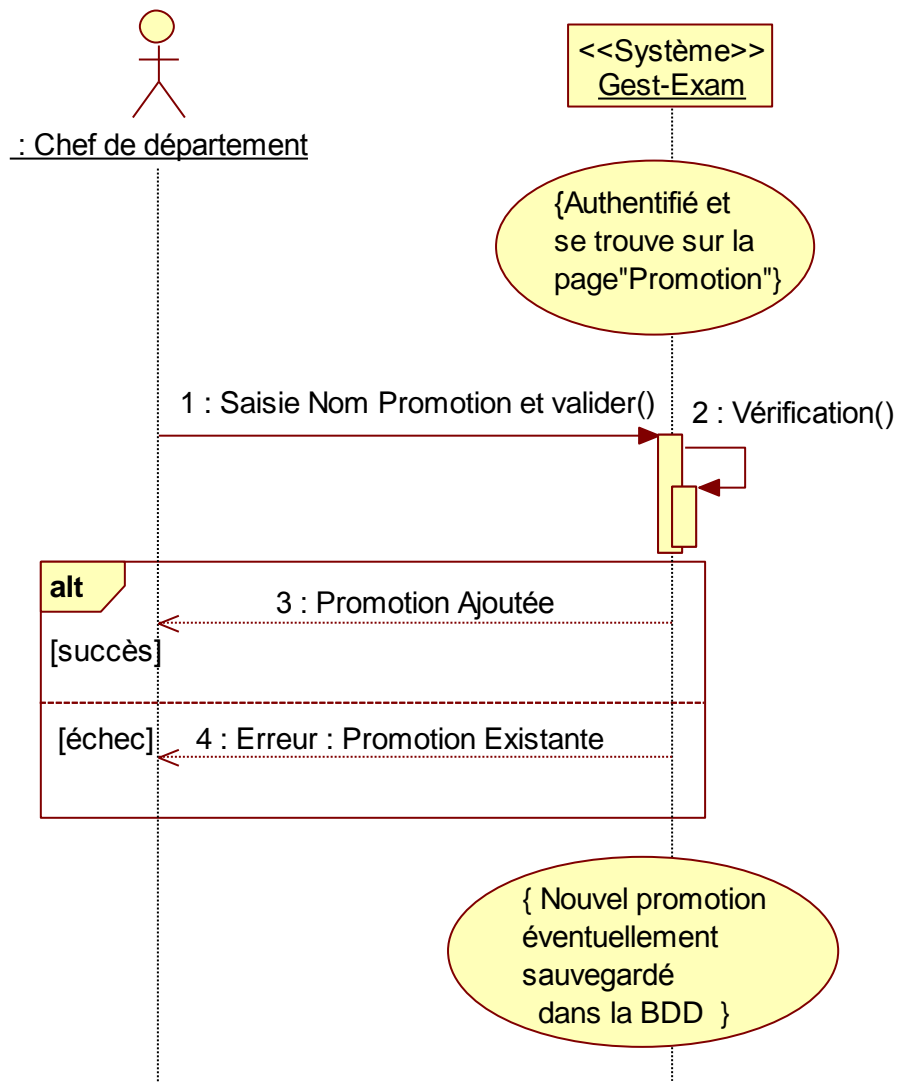
Cas d'utilisation	Gérer Promotions
Acteur principal	Chef de département
Acteur secondaire	Néant
Objectif	Permettre à le chef de département d'ajouter, supprimer ou modifier des promotions.
Pré-condition	Le chef de département doit être connecté.
Post-conditions	Néant.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le chef de département accède à l'espace « Gérer Promotions » en cliquant sur le menu «Promotions» dans la barre de navigation.</li> <li>2. Le système affiche la liste des promotions créés et formulaire de saisie pour ajouter une promotion.</li> <li>3. Le chef de département choisi l'action à effectuer (ajout, suppression, modification) : <ol style="list-style-type: none"> <li>a. Ajout : voir le cas d'utilisation « Ajouter Promotion.».</li> <li>b. Suppression : voir le cas d'utilisation « Supprimer Promotion.».</li> <li>c. Modification : voir le cas d'utilisation « Modifier Promotion. ».</li> </ol> </li> </ol>
Alternative	Néant



**Figure 19 DSS7- Cas d'utilisation : Gérer Promotions (Chef de département)**

**Diagramme de séquence du cas d'utilisation «Ajouter Promotion»**

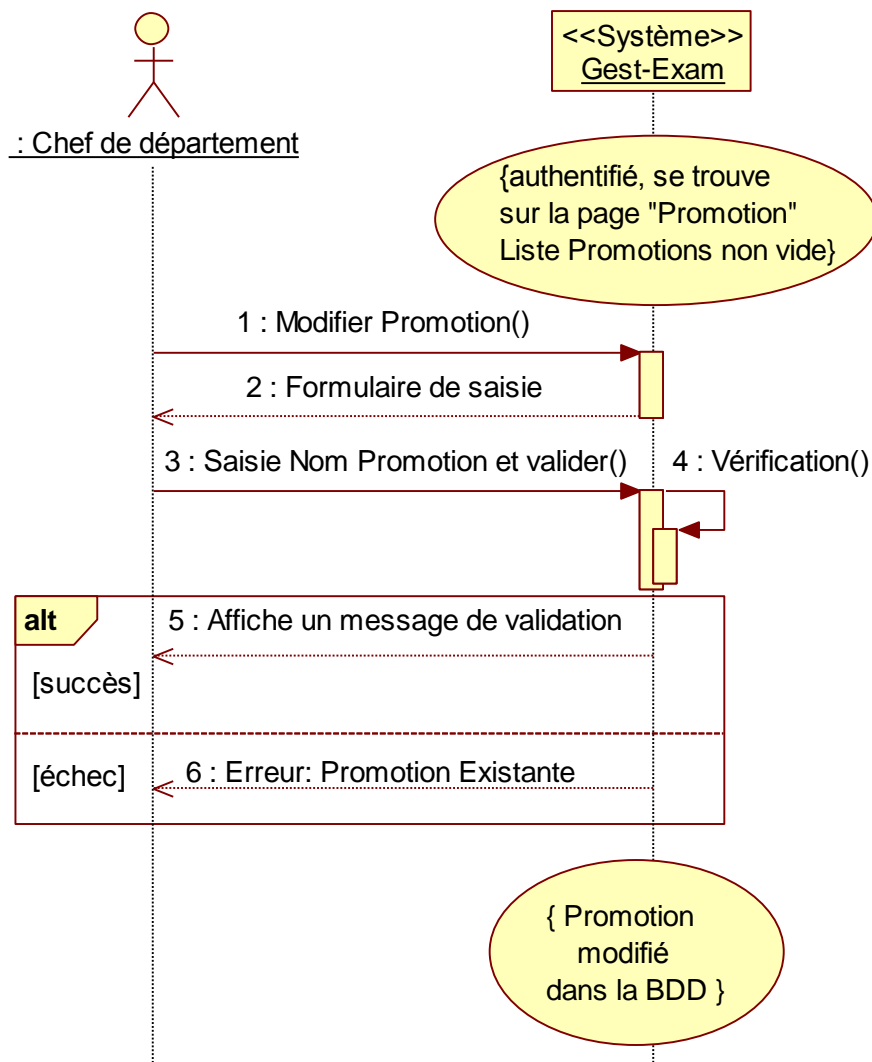
Cas d'utilisation	Ajouter Promotion
Acteur principal	Chef de département
Acteur secondaire	Néant
Objectif	Permettre au chef de département d'ajouter une promotion en indiquant son nom, année universitaire et niveau.
Pré-condition	Le chef de département doit être connecté et doit être sur la page «Promotion».
Post-conditions	La nouvelle promotion devra être rajoutée à la base de donnée.
Scénario nominal	<ol style="list-style-type: none"> <li>Le chef de département saisie le nom d'une nouvelle promotion et valide en cliquant sur l'icône d'ajout.</li> <li>Le système vérifie si la promotion n'existe pas déjà.</li> <li>Le système confirme l'ajout.</li> </ol>
Alternative	2a. La promotions existe déjà, le système renvoie un message d'erreur.



**Figure 20 DSS8 - Cas d'utilisation : Ajouter Promotion (Chef de département)**

**Diagramme de séquence du cas d'utilisation «Modifier Promotion»**

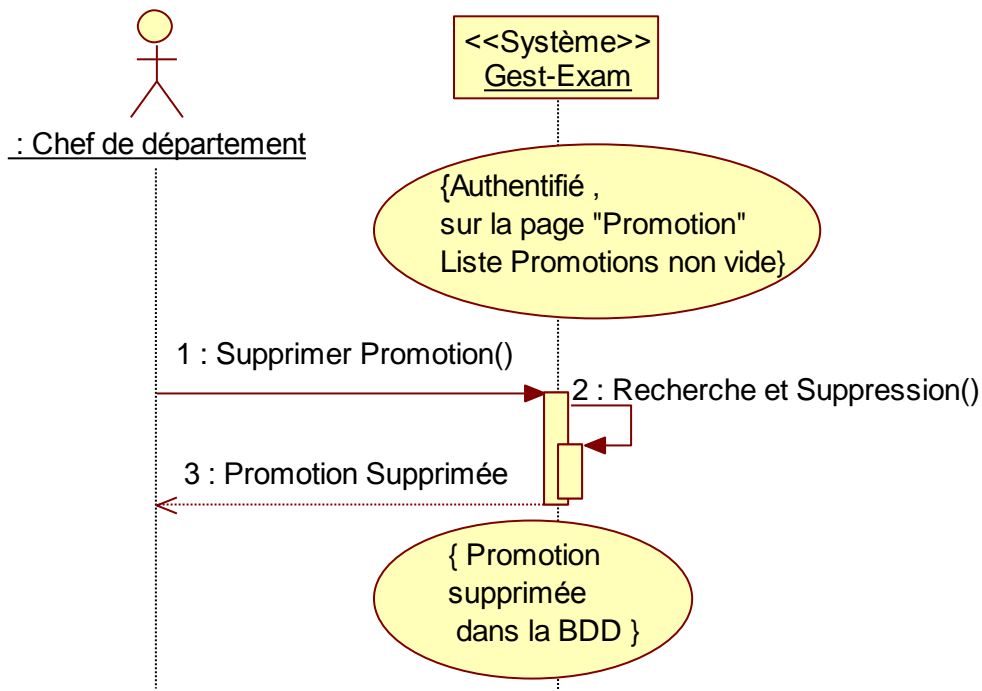
Cas d'utilisation	Modifier Promotion
Acteur principal	Chef de département.
Acteur secondaire	Néant.
Objectif	Permettre au chef de département de modifier les informations concernant une promotion.
Pré-condition	Le chef de département doit être connecté et doit être sur la page «Promotion». La liste des promotions doit être non vide.
Post-conditions	La promotion devra être modifiée dans la base de donnée.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le chef de département clique sur l'icône « modifié ».</li> <li>2. Le système affiche une formulaire de saisie.</li> <li>3. Le chef de département fait les modifications nécessaires et valide en cliquant sur le bouton « valider ».</li> <li>4. Le système vérifie s'il n'existe pas une promotion portant le même nom.</li> <li>5. Le système affiche un message de validation.</li> </ol>
Alternative	4a. La promotion existe déjà, le système retourne une erreur.



**Figure 21 DSS9 - Cas d'utilisation : Modifier Promotion (Chef de département)**

**Diagramme de séquence du cas d'utilisation «Supprimer Promotion»**

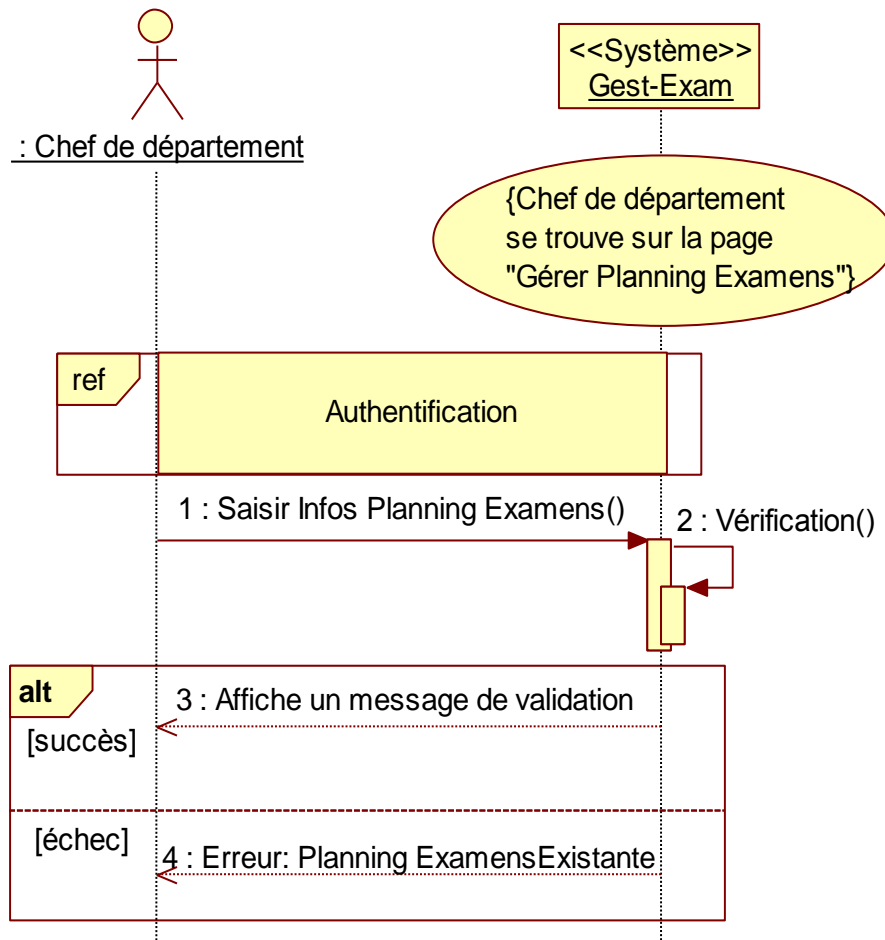
Cas d'utilisation	Supprimer Promotion
Acteur principal	Chef de département
Acteur secondaire	Néant
Objectif	Permettre au chef de département de supprimer une promotion, à tout moment, en cliquant sur l'icône «supprimer».
Pré-condition	Le chef de département doit être connecté et doit être sur la page «Promotion». La liste des promotions doit être non vide.
Post-conditions	La promotion supprimée devra être retirée dans la base de donnée.
Scénario nominal	1. Le chef de département clique sur l'icône «supprimer». 2. Le système fait une recherche et supprime la promotion sélectionnée. 3. Le système confirme la suppression.
Alternative	Néant



**Figure 22 DSS10- Cas d'utilisation : Supprimer Promotion (Chef de département)**

**Diagramme de séquence du cas d'utilisation «Gérer Planning Examens»**

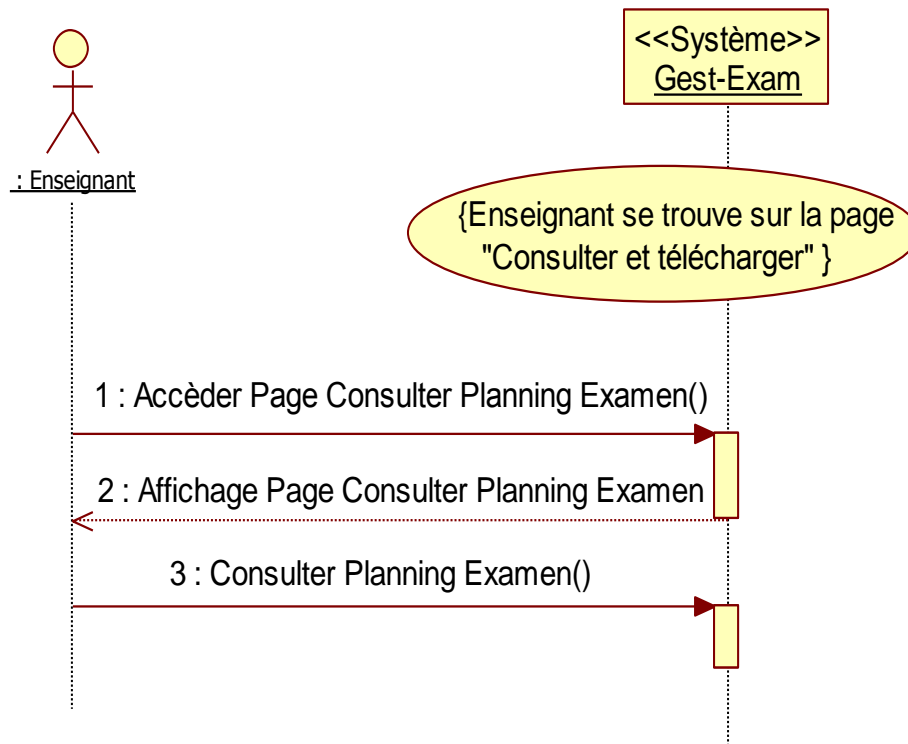
Cas d'utilisation	Gérer Planning Examens
Acteur principal	Chef de département
Acteur secondaire	Néant
Objectif	Permettre à chef de département d'indiquer un planning examens.
Pré-condition	Le chef de département doit être connecté et doit être sur la page «Gérer Planning Examens».
Post-conditions	Néant
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le chef de département saisie les données requises par planning examens et valide en cliquant sur l'icône «OK».</li> <li>2. Le système vérifie si le planning examens n'existe pas déjà.</li> <li>3. Le système envoie un message de validation.</li> </ol>
Alternative	2a. Le planning examens existe déjà, le système renvoie un message d'erreur.



**Figure 23 DSS 11- Cas d'utilisation : Gérer Planning Examens(Chef de département)**

**Diagramme de séquence du cas d'utilisation «Consulter Planning Examen»**

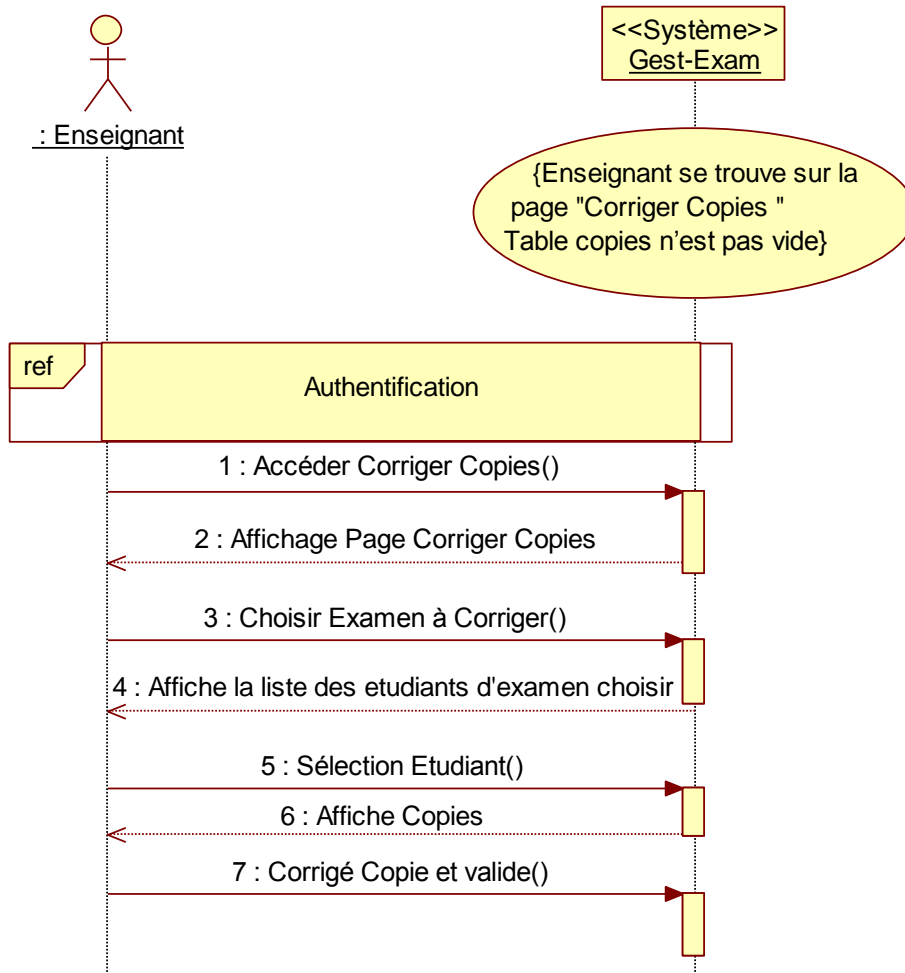
Cas d'utilisation	Consulter Planning Examen
Acteur principal	Enseignant
Acteur secondaire	Néant
Objectif	Permettre à l'enseignant de Consulter le planning des examens.
Pré-condition	Enseignant accède à la page «Consulter Planning Examen ». La table planning examen ne doit pas être vide.
Post-conditions	Néant
Scénario nominal	1. L'enseignant accède à l'espace «Consulter Planning Examen» en cliquant sur le menu « Consulter Planning Examen» dans la barre de navigation. 2. Le système affiche la page « Consulter Planning Examen ». 3. L'enseignant consulte le planning examens.
Alternative	Néant



**Figure 24 DSS 12- Cas d'utilisation : Consulter Planning Examen (Enseignant)**

**Diagramme de séquence du cas d'utilisation «Corriger Copies»**

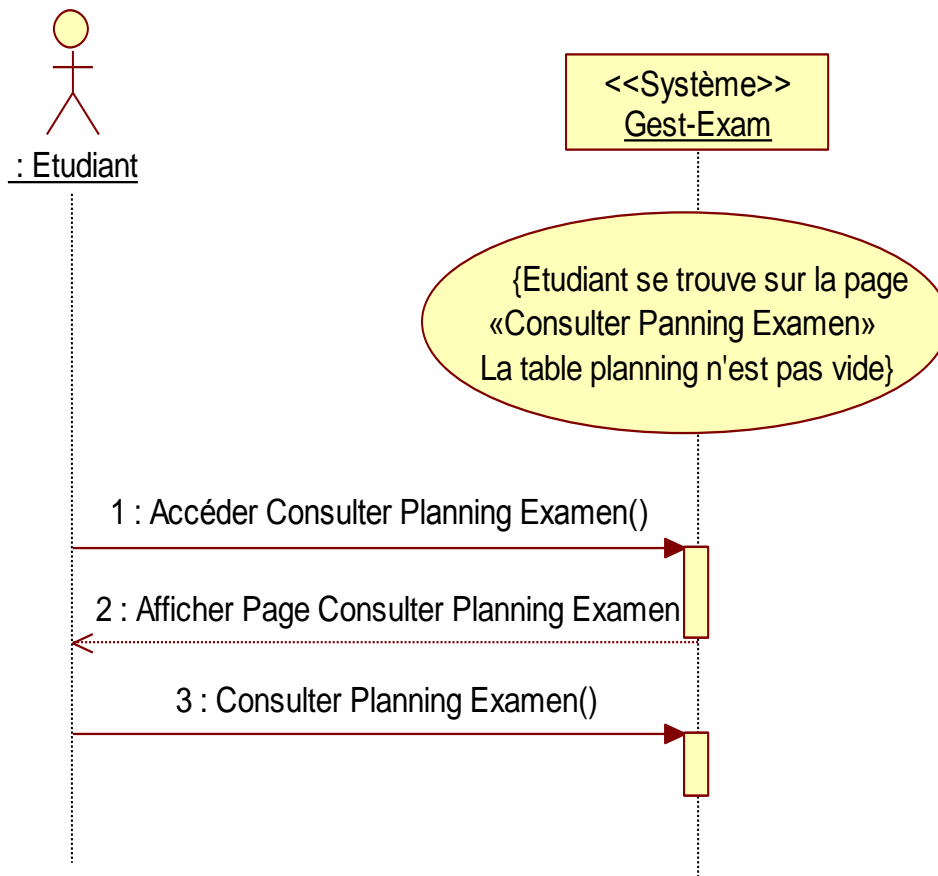
Cas d'utilisation	Corriger Copies
Acteur principal	Enseignant
Acteur secondaire	Néant
Objectif	Permettre à l'enseignant de corriger des copies numérisées.
Pré-condition	Enseignant connecté et accède à la page «Corriger Copies». La table des copies n'est pas vide.
Post-conditions	Néant.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'enseignant accède à l'espace « Corriger des copies » en clique sur le menu « Copies » dans la barre de navigation.</li> <li>2. Le système affiche la page « Corriger des copies ».</li> <li>3. L'enseignant choisi l'examen à corriger.</li> <li>4. Le système affiche la liste des étudiants de l'examen choisi.</li> <li>5. L'enseignant sélection l'un des étudiants.</li> <li>6. Le système affiche la copie de l'étudiant choisi.</li> <li>7. L'enseignant corrige la copie de l'étudiant choisi et valide son travail.</li> </ol>
Alternative	Néant.



**Figure 25 DSS 13- cas d'utilisation : Corriger Copies (Enseignant)**

**Diagramme de séquence du cas d'utilisation «Consulter Planning Examen »**

Cas d'utilisation	Consulter Planning Examen
Acteur principal	Etudiant
Acteur secondaire	Néant
Objectif	Permettre à l'étudiant de consulter le planning des examens.
Pré-condition	Etudiant accède à la page «Consulter Planning Examen». Le planning des examens existe.
Post-conditions	Néant
Scénario nominal	1. L'étudiant accède à l'espace «Consulter Planning Examen» en cliquant sur le menu « Consulter Planning Examen» dans la barre de navigation. 2. Le système affiche la page « consulter planning examens ». 3. L'étudiant consulter planning examens.
Alternative	Néant

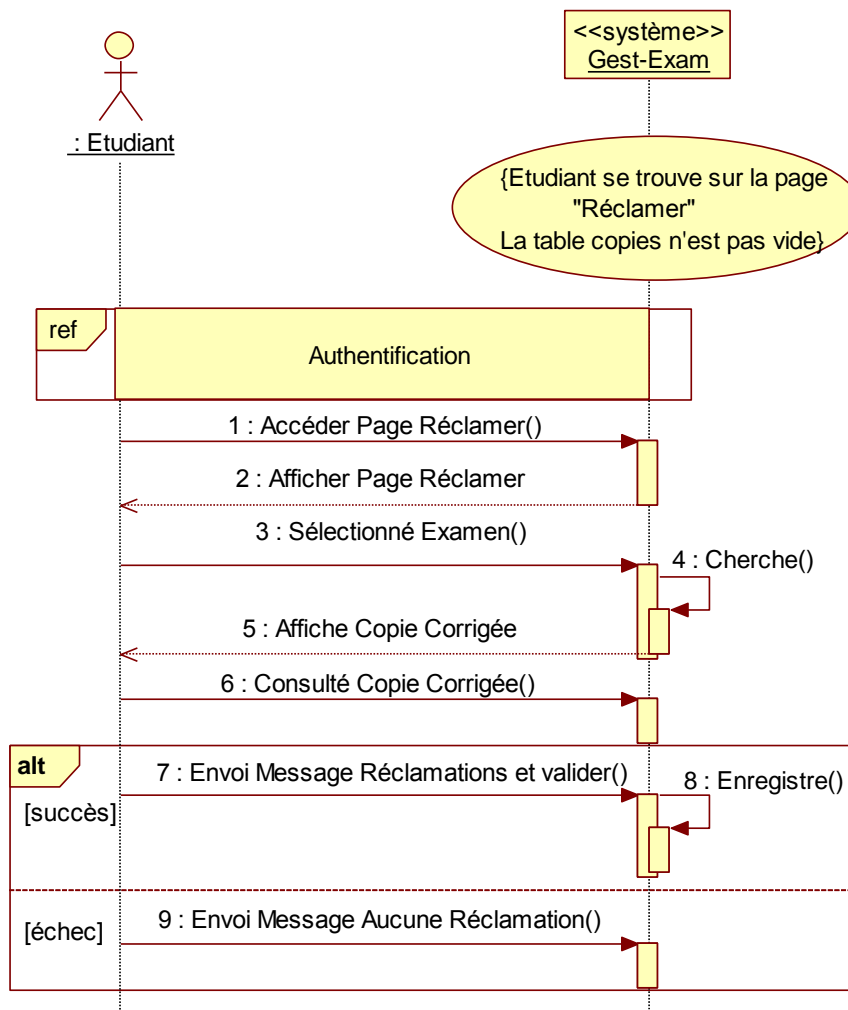


**Figure 26 DSS14- Cas d'utilisation : Consulter Planning Examen (Etudiant)**



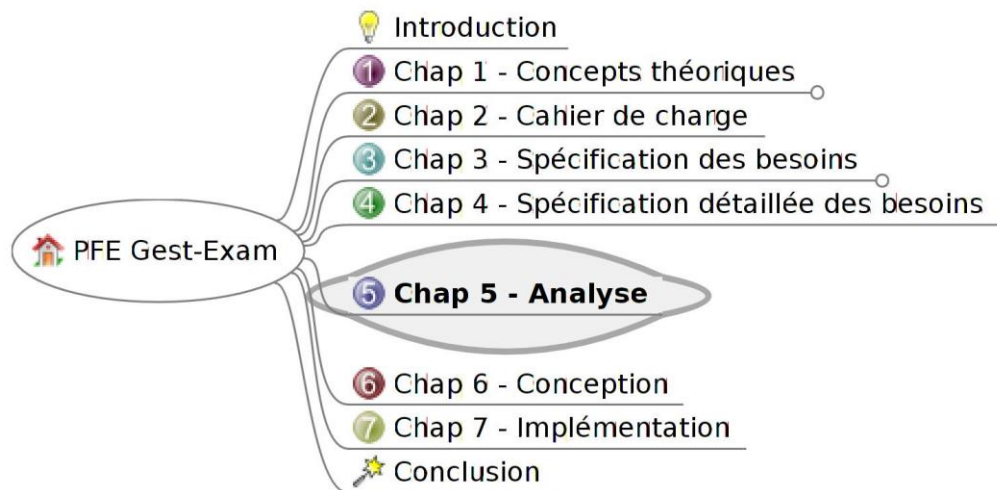
**Diagramme de séquence du cas d'utilisation «Réclamer»**

Cas d'utilisation	Réclamer
Acteur principal	Etudiant
Acteur secondaire	Néant
Objectif	Permettre à l'étudiant de consulter sa copie d'examen et de formuler une réclamation.
Pré-condition	Etudiant connecté et doit être sur la page «Réclamer». La copie de l'étudiant existe.
Post-conditions	La nouvelle réclamation doit être sauvegardées dans la base de donnée.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'étudiant accède à l'espace «Réclamer» en cliquant sur le menu «Réclamer» dans la barre de navigation.</li> <li>2. Le système affiche la page «Réclamer».</li> <li>3. L'étudiant sélectionne un examen.</li> <li>4. Le système cherche la copie corrigée de l'étudiant.</li> <li>5. Le système affiche la copie corrigée de l'étudiant.</li> <li>6. L'étudiant consulte sa copie corrigée.</li> <li>7. L'étudiant envoie le message de réclamation en clique sur le bouton «OK».</li> <li>8. Le système enregistre le message de réclamation dans la base de donnée.</li> </ol>
Alternative	6a. L'étudiant envoie le message «aucune réclamation».



**Figure 27 DSS 15- Cas d'utilisation : Réclamer (Etudiant)**

## Chapitre.5 - Analyse



*L'activité d'analyse est à mi-chemin entre la spécification des besoins et la conception de l'application. Nous allons commencer par l'analyse du domaine, c'est-à-dire l'analyse des concepts manipulés par les experts du domaine (analyse métier) pour identifier des classes candidates. Pour passer en douceur à la conception, nous allons identifier les principales classes d'IHM ainsi que celles qui décrivent la cinématique de l'application (les contrôles). Nous allons dégager, pour chaque cas d'utilisation un diagramme de classes participantes mettant en évidence une distinction nette entre les classes d'interface (IHM), les classe de contrôles et les classe du domaine (les entités).*

## 5.1 Identification des concepts du domaine

Voici la liste des concepts que nous pouvons déduire de la spécification des besoins et de l'IHM

Selon les cas d'utilisation	Concepts impliqués
Authentification	Utilisateur, chef de département, administrateur, enseignants, étudiants.
Configurer Gest-exam	administrateur, établissement.
Gérer Facultés (ajouter, modifier, supprimer)	administrateur, faculté.
Gérer Départements (ajouter, modifier, supprimer)	administrateur, faculté, départements.
Gérer Comptes Utilisateurs (ajouter, modifier, supprimer)	administrateur, utilisateurs.
Associer Compte Chef Département	administrateur, départements, chefs de départements.
Gérer Enseignants (ajouter, modifier, supprimer)	Chef de département, enseignants
Gérer Modules (ajouter, modifier, supprimer)	chef de département, module.
Gérer Promotions (ajouter, modifier, supprimer)	chef de département, promotion.
Gérer Sections (ajouter, modifier, supprimer)	chef de département, promotion, section.
Gérer Groupes (ajouter, modifier, supprimer)	chef de département, promotion, section, groupe.
Gérer Examen	chef de département, examens, module.
Gérer Planning Examens	chef de département, examens, planning examens.
Editer Rapport	chef de département, examens, rapport examens.
Consulter Planning Examens	Enseignant, étudiant, planning examens.
Lancer Examens	Enseignant (responsable module), groupe, enseignants correcteur.
Configurer Examen (Outil externe, Gest exam)	Enseignant, examen.
Enoncé Examen	Enseignant, examen, question.
Gérer Réclamations	Enseignant (responsable module), examen, étudiant.
Consulter Copies Examen	Etudiant, examen, copies corrigé.

**Tableau 14-** Liste des concepts découlés de la spécification des besoins et de l'IHM

## 5.2 Identification des associations

Les associations peuvent être déduites de l'analyse des règles existantes entre les différents concepts :

- Un utilisateur est soit un administrateur, un étudiant, un enseignant ou un chef de département.
- Une promotion est **composée** d'une ou de plusieurs sections.
- Une section est **composée** d'un ou de plusieurs groupes.
- Un bloc est **composé** d'une ou de plusieurs salles.
- Un enseignant, un étudiant et un chef de département **appartiennent** à un département d'affiliation.
- Un examen peut **avoir** plusieurs questions.
- Une question fait **partie** d'un seul examen.
- Un examen **concerne** une et une seule promotion.
- Pour une promotion on peut **avoir** plusieurs examens.
- Un examen **concerne** à un et un seul module.
- Un groupe **contient** plusieurs étudiants.
- Une section **contient** plusieurs groupes.
- Un administrateur peut **gérer** plusieurs départements.

### 5.3 Diagramme de classes métiers

Voici le diagramme de classe permettant de modéliser les concepts du domaine avec leurs relations :

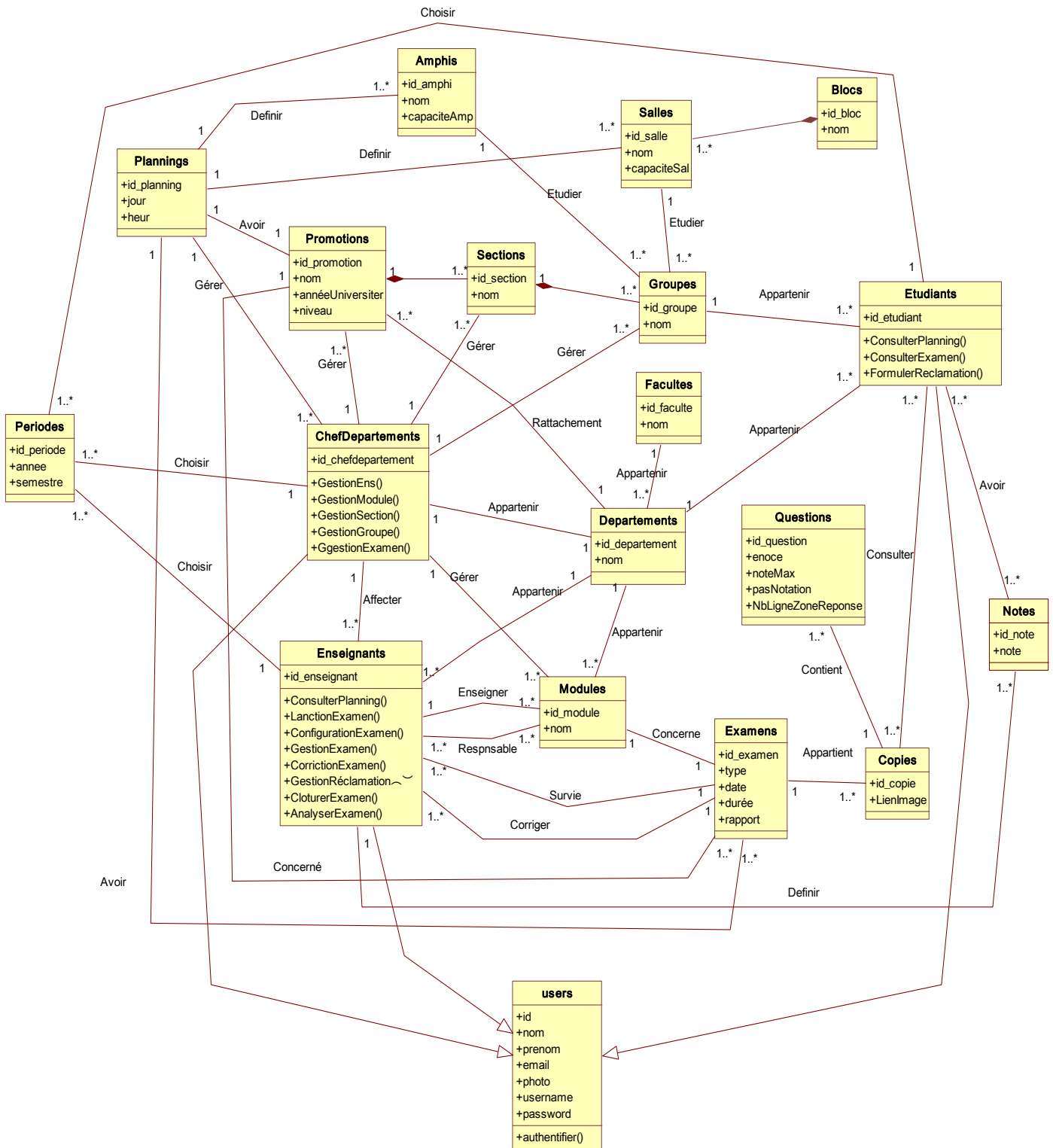


Figure 28– Diagramme de classes métiers

## 5.4 Modèle relationnel

ChefDepartement (**id\_chefDepartement**, nom, prenom, email, photo, username, password, #id\_plannig)  
 Enseignants (**id\_enseignant**, nom, prenom, email, photo, username, password, #id\_chefDepartement, #id\_departement, #id\_module, #id\_examen)  
 Etudiants (**id\_etudiant**, nom, prenom, email, photo, username, password, #id\_groupe, #id\_departement, #id\_copie, #id\_note)  
 Départements (**id\_departement**, nom, #id\_faculte)  
 Plannings (**id\_planning**, jour, heure)  
 Facultés (**id\_faculte**, nom)  
 Perodes (**id\_periode**, annee, semestre, #id\_etudiant, #id\_chefDepartement, #id\_enseignant)  
 Promotions (**id\_promotion**, nom, annéeUniversiter, niveau, #id\_chefDepartement, #id\_departement)  
 Sections (**id\_section**, nom, #id\_promotion, #id\_chefDepartement)  
 Groupes (**id\_groupe**, nom, #id\_salle, #id\_amphi, #id\_section, #id\_chefDepartement)  
 Modules (**id\_module**, nom, #id\_chefDepartement, #id\_departement, #id\_enseignant)  
 Examens (**id\_examen**, type, date, durée, rapport, #id\_promotion, #id\_planning)  
 Questions (**id\_question**, enoce, noteMax, pasNotation, NbrLigneZoneReponse, #id\_copie)  
 Copies (**id\_copie**, LienImage, #id\_etudiant, #id\_examen )  
 Notes (**id\_note**, note, #id\_etudiant, #id\_enseignant)  
 Salles (**id\_salle**, nom, capaciteSal, #id\_plannig, #id\_bloc)  
 Blocs (**id\_bloc**, nom)  
 Amphis (**id\_amphi**, nom, capaciteAmp, #id\_plannig)

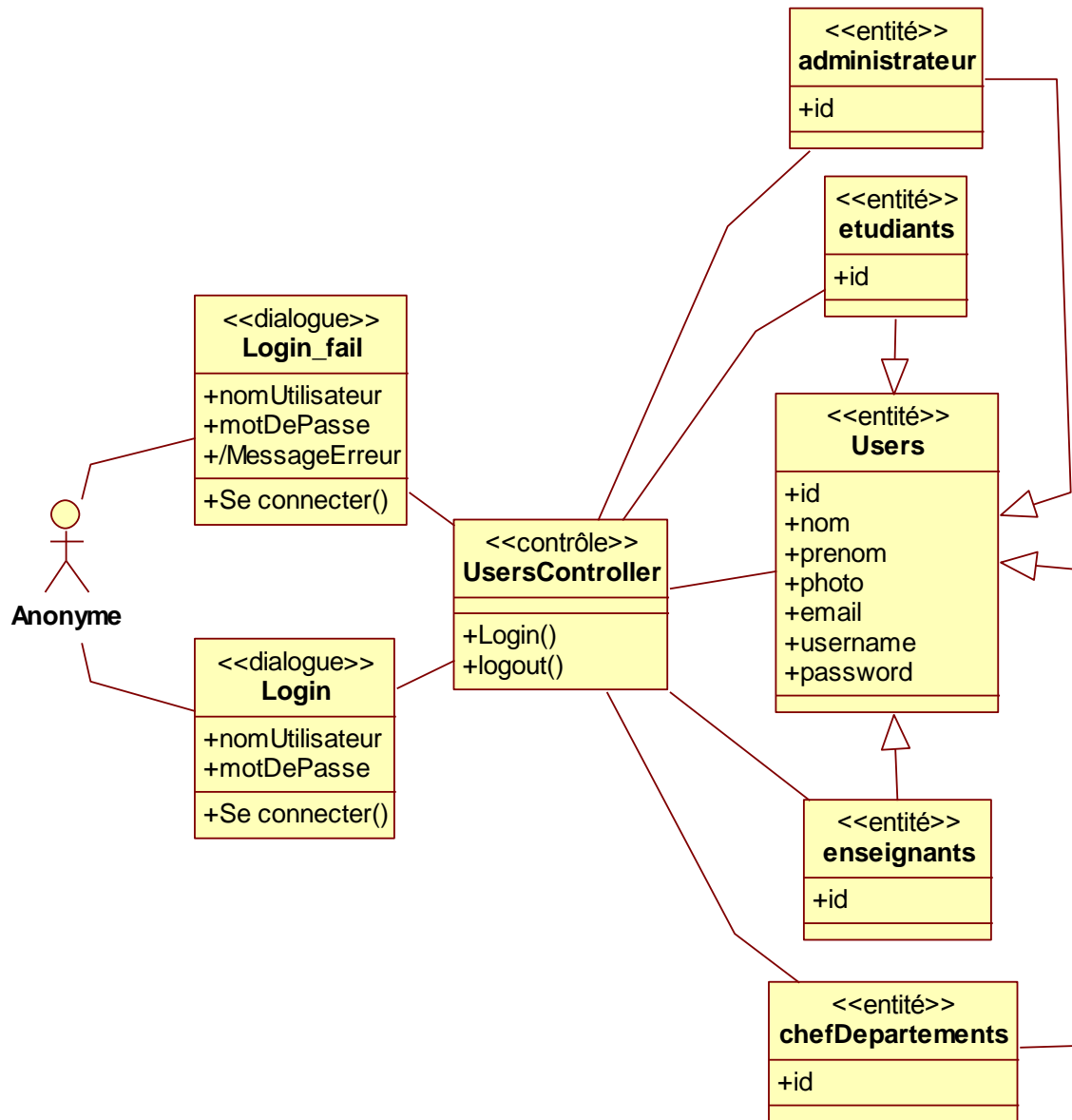


Figure 29 DCP1- Cas d'utilisation : Authentification

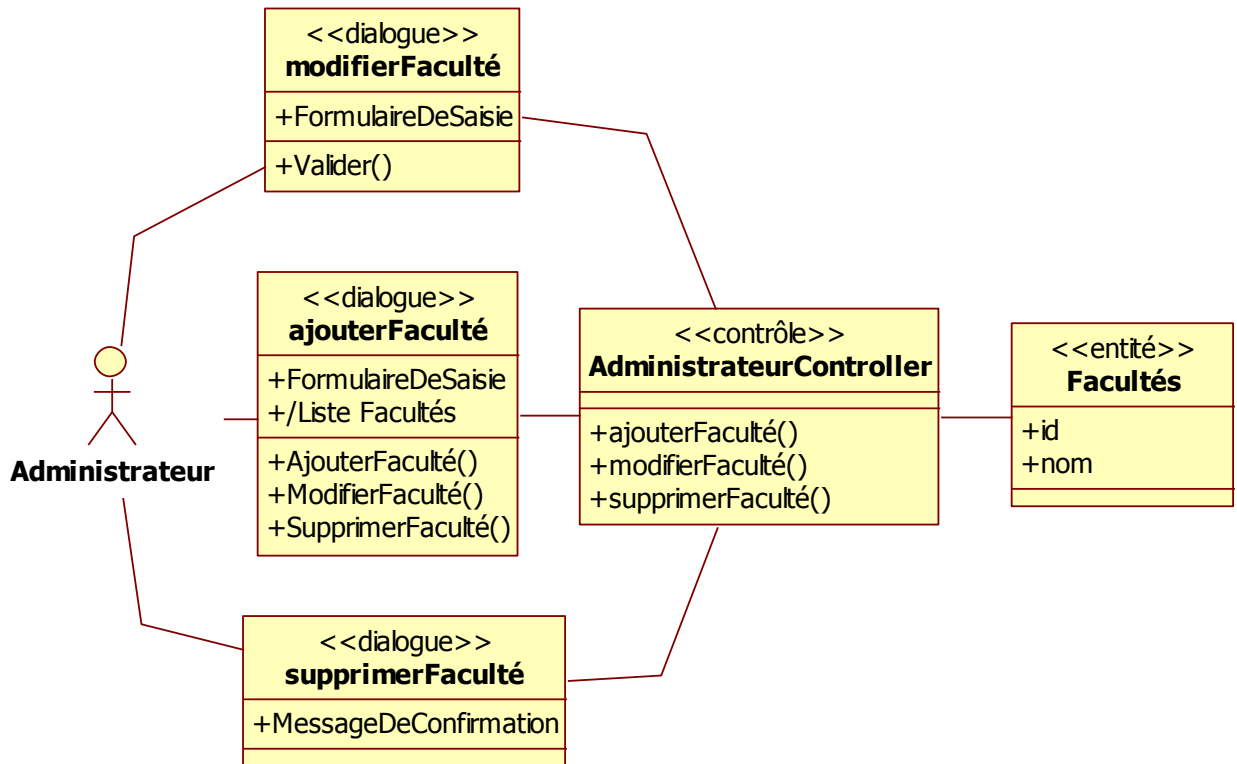


Figure 30 DCP2- Cas d'utilisation : Gérer Faculté (Administrateur)

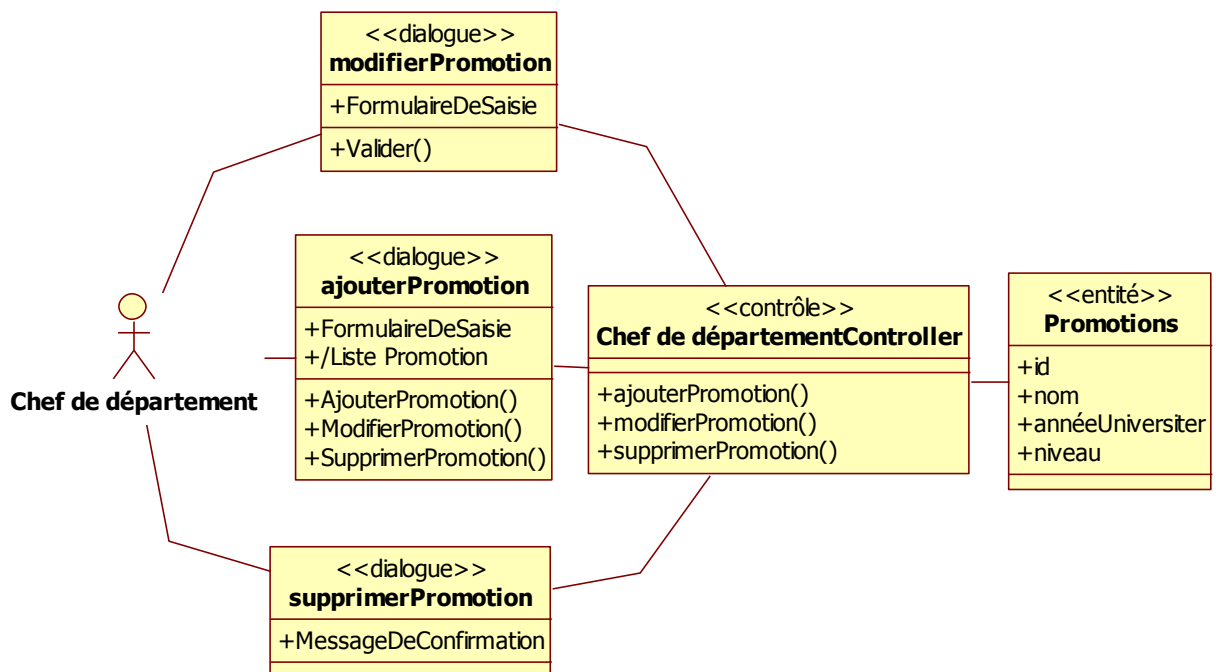
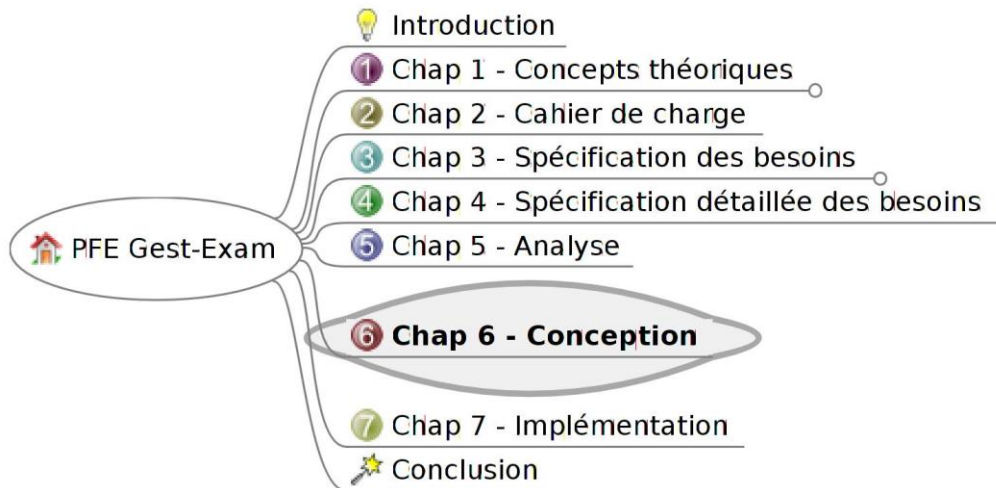


Figure 31 DCP3- Cas d'utilisation : Gérer Promotion (Chef de département)



## Chapitre.6 – Conception objet préliminaire



*Dans cette étape, nous allons faire la synthèse de notre analyse en élaborant un diagramme de séquence détaillé mettant en évidence toute la cinématique de notre application.*

---

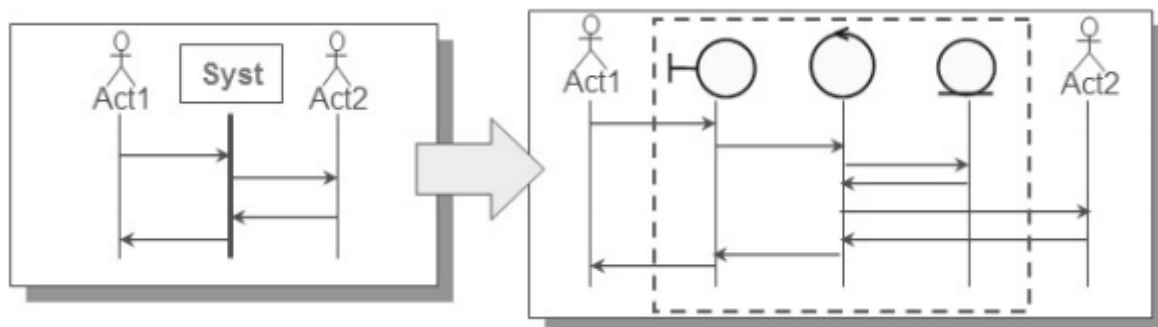
## 6.1 Démarche

Rappelons le positionnement de cette phase de conception par rapport à l'ensemble du processus entrepris dans ce projet. Nous avons identifié les cas d'utilisation au (chapitre 3) et nous les avons décrits en détail au (chapitre 4), grâce en particulier aux diagrammes de séquence système. Nous avons également réalisé un modèle d'analyse en se servant des diagrammes de classes participantes (chapitre 5). Pour passer maintenant vraiment à la conception, nous allons répartir tout le comportement du système entre ces classes d'analyse et nous décrivons les interactions correspondantes. Nous allons donc utiliser les diagrammes de séquence détaillé qui sont particulièrement utiles pour représenter graphiquement les décisions d'allocation de responsabilités. Chaque diagramme va ainsi représenter un ensemble d'objets de classes différentes collaborant dans le cadre d'un scénario d'exécution du système. Dans ce genre de diagramme, les objets communiquent en s'envoyant des messages qui invoquent des opérations (ou méthodes) sur les objets récepteurs. Il est ainsi possible de suivre visuellement les interactions dynamiques entre objets, et les traitements réalisés par chacun.[1]

Par rapport aux diagrammes de séquence système, nous allons remplacer le système vu comme une boîte noire par un ensemble d'objets en interaction. Pour cela, nous utiliserons trois types de classes d'analyse, à savoir les dialogues, les contrôles et les entités. Nous respecterons également les règles que nous avons fixées sur les relations entre classes d'analyse, mais en nous intéressant cette fois-ci aux interactions dynamiques entre objets :

- Les acteurs ne peuvent interagir (envoyer des messages) qu'avec les dialogues.
- Les dialogues peuvent interagir avec les contrôles.
- Les contrôles peuvent interagir avec les dialogues, les entités, ou d'autres contrôles.
- Les entités ne peuvent interagir qu'entre elles.

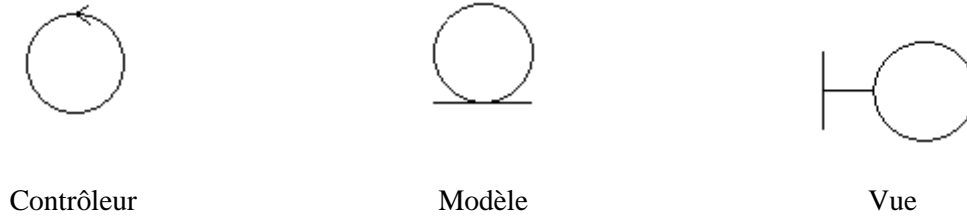
Le changement de niveau d'abstraction par rapport au diagramme de séquence système peut ainsi se représenter comme sur la figure suivante.



**Figure 32– Passage de l'analyse à la conception préliminaire**

## 6.2 Diagrammes de séquences détaillées

Dans les diagrammes qui vont suivre, nous utiliserons trois stéréotypes pour représenter les objets système correspondant aux dialogues, entités et contrôleurs :



Pour chaque cas d'utilisation, nous présenterons un diagramme de séquence détaillé.

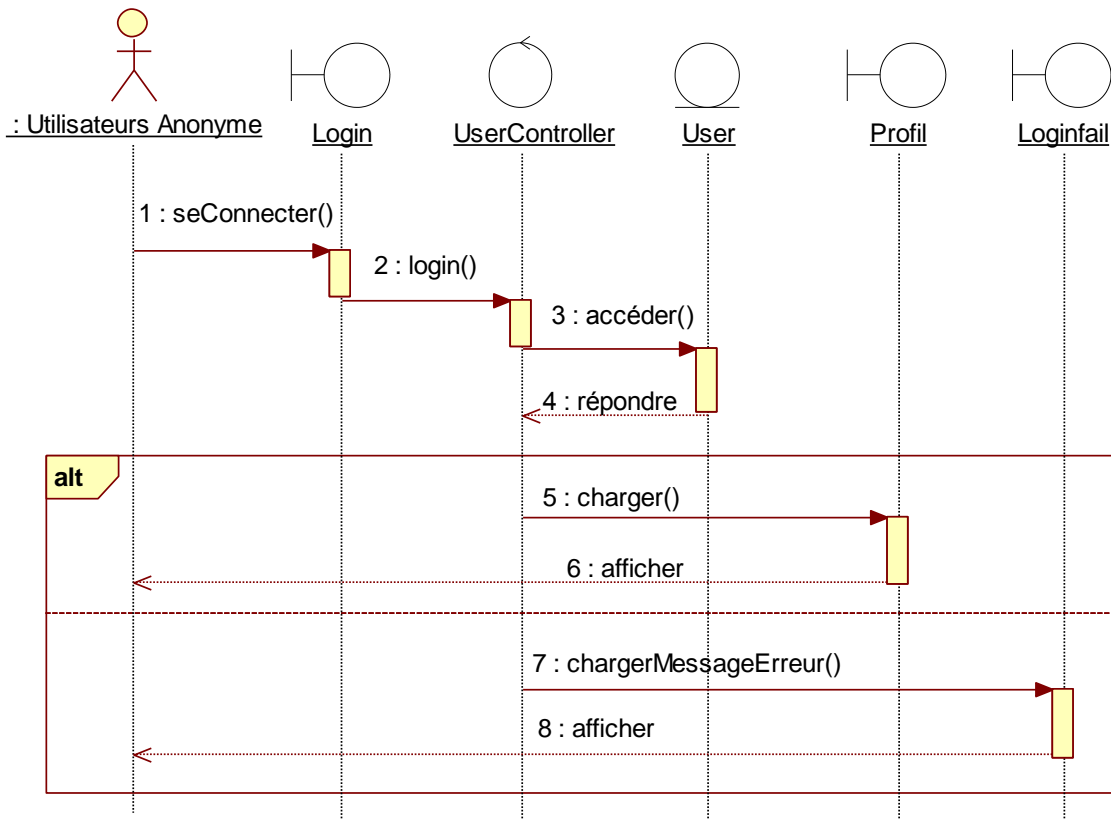


Figure 33 DSD 1- Cas d'utilisation : Authentification

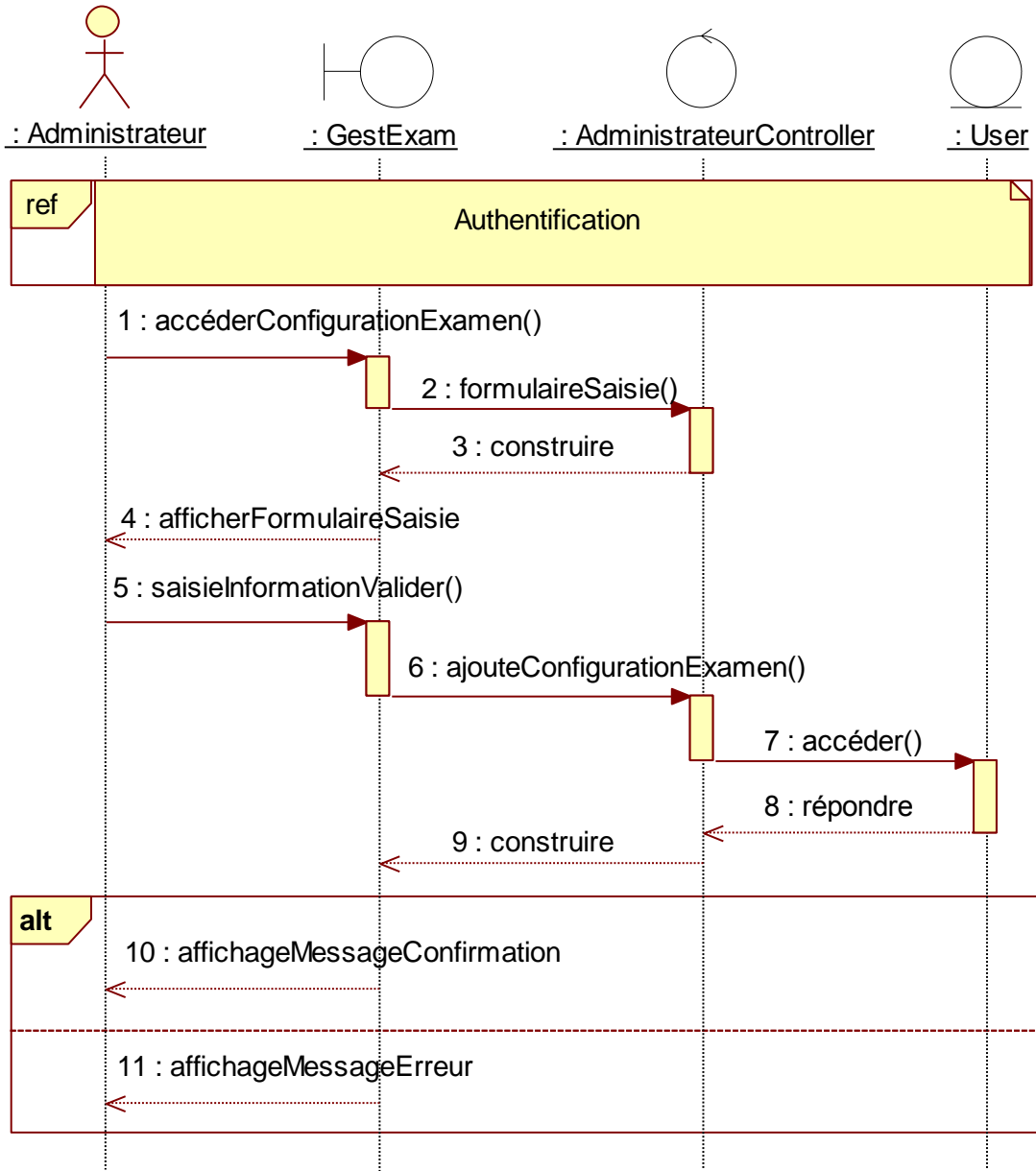


Figure 34 DSD 2- Cas d'utilisation : Configurer Gest-exam (Administrateur)

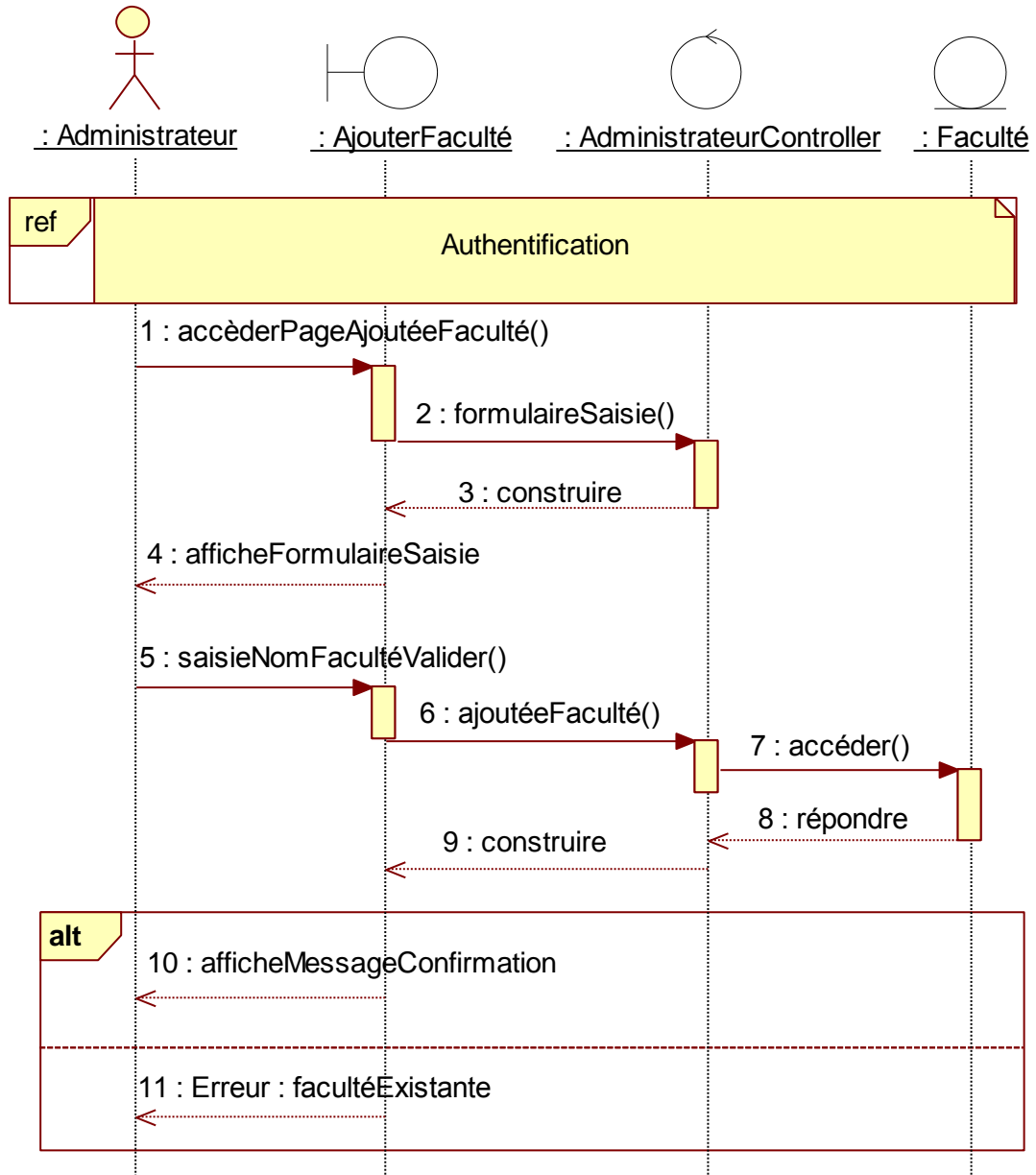


Figure 35 DSD 3- Cas d'utilisation : Ajouter Faculté (Administrateur)

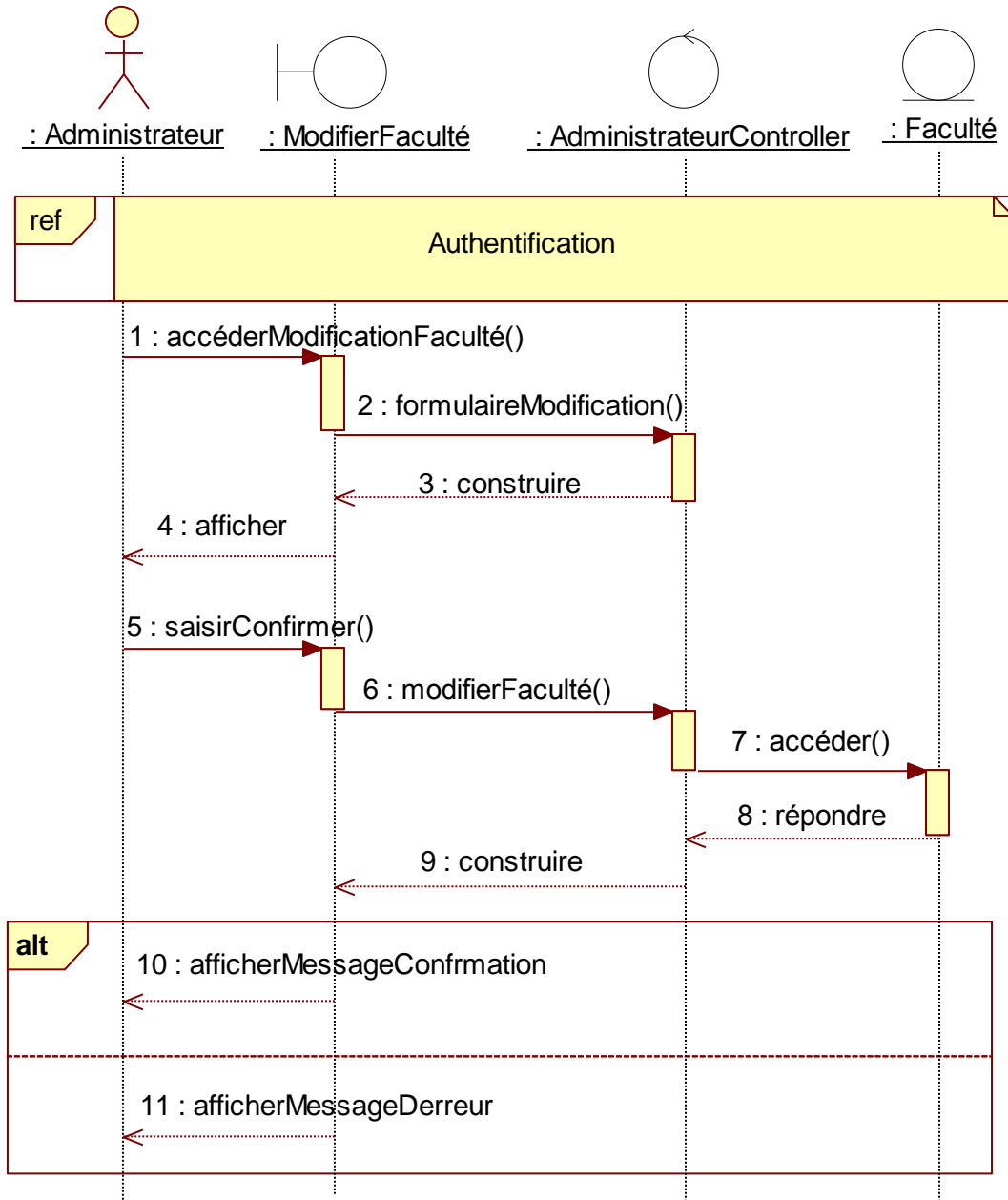


Figure 36 DSD4- Cas d'utilisation : Modifier Faculté (Administrateur)

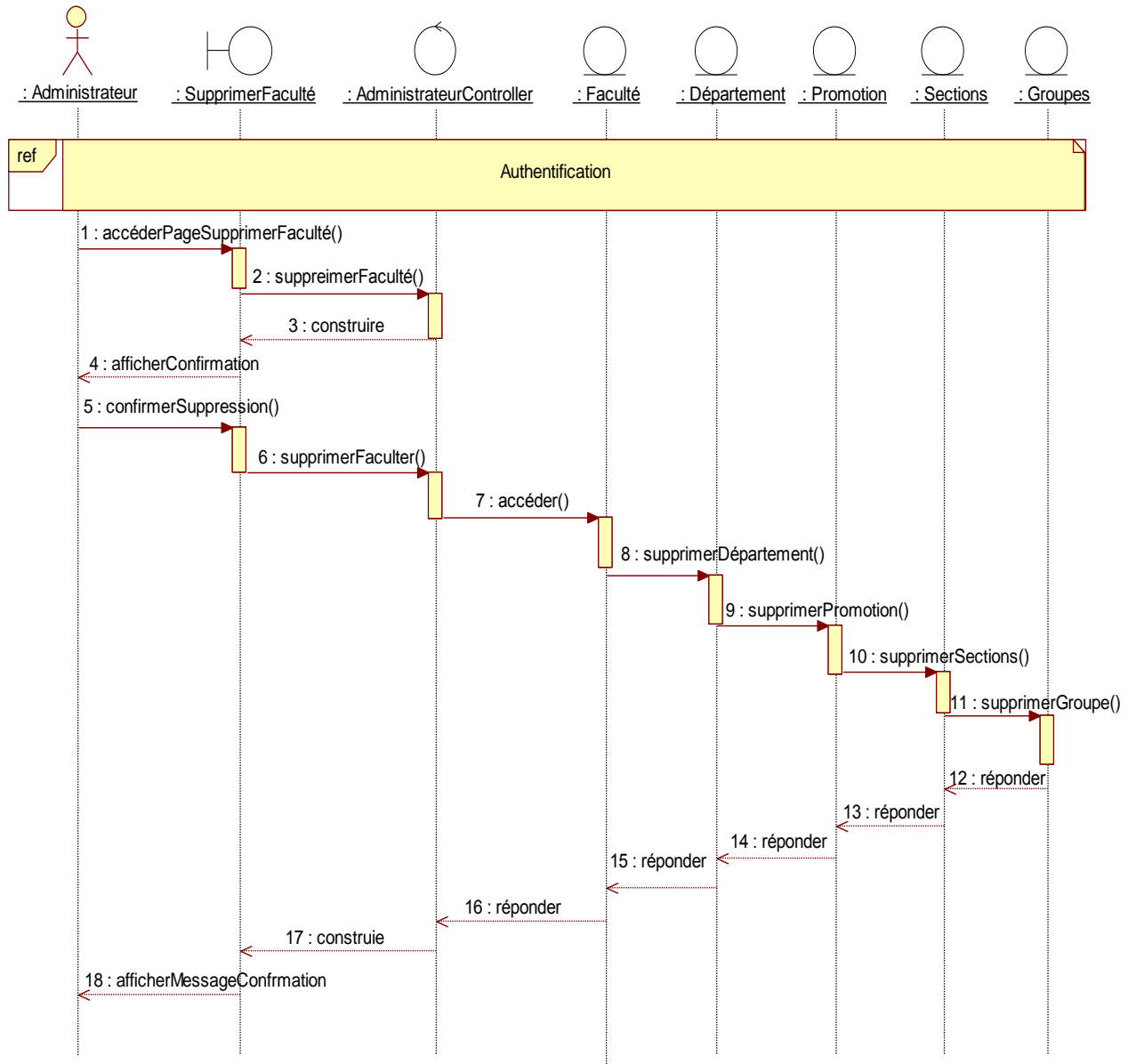


Figure 37 DSD5- Cas d'utilisation : Supprime Faculté (Administrateur)

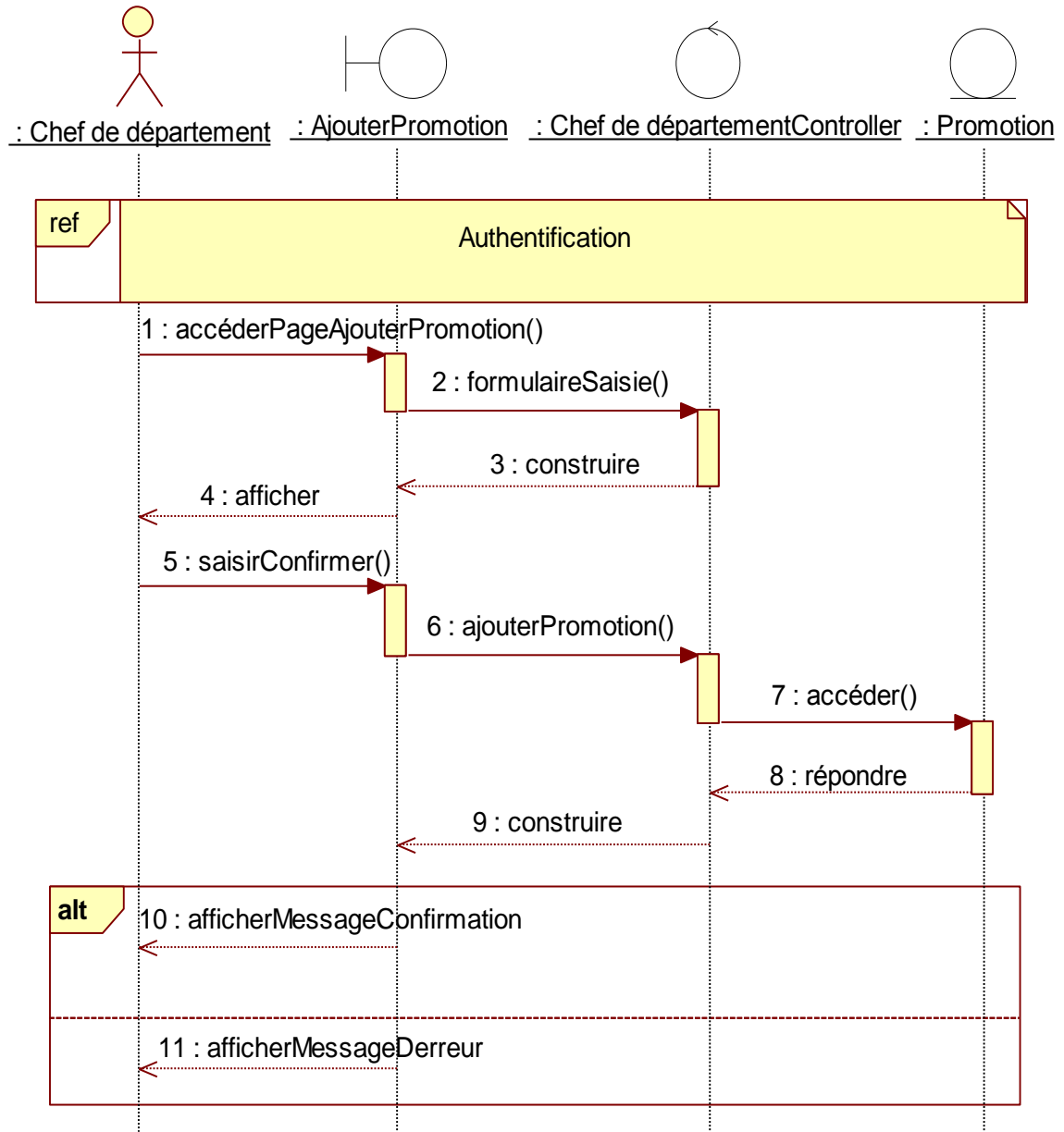


Figure 38 DSD 6 - Cas d'utilisation : Ajouter Promotion (Chef de département)



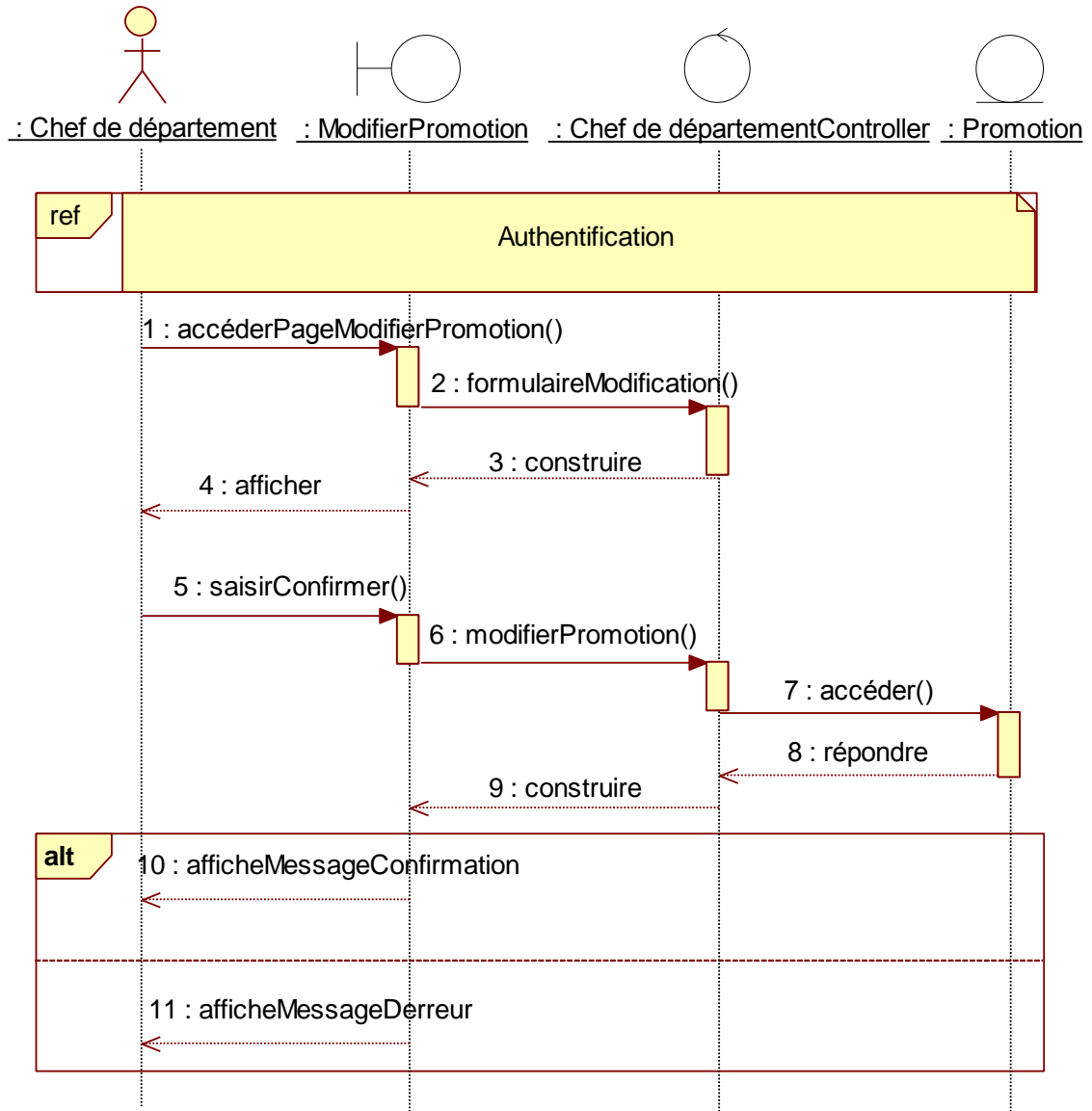


Figure 39 DSD 7- Cas d'utilisation : Modifier Promotion (Chef de département)

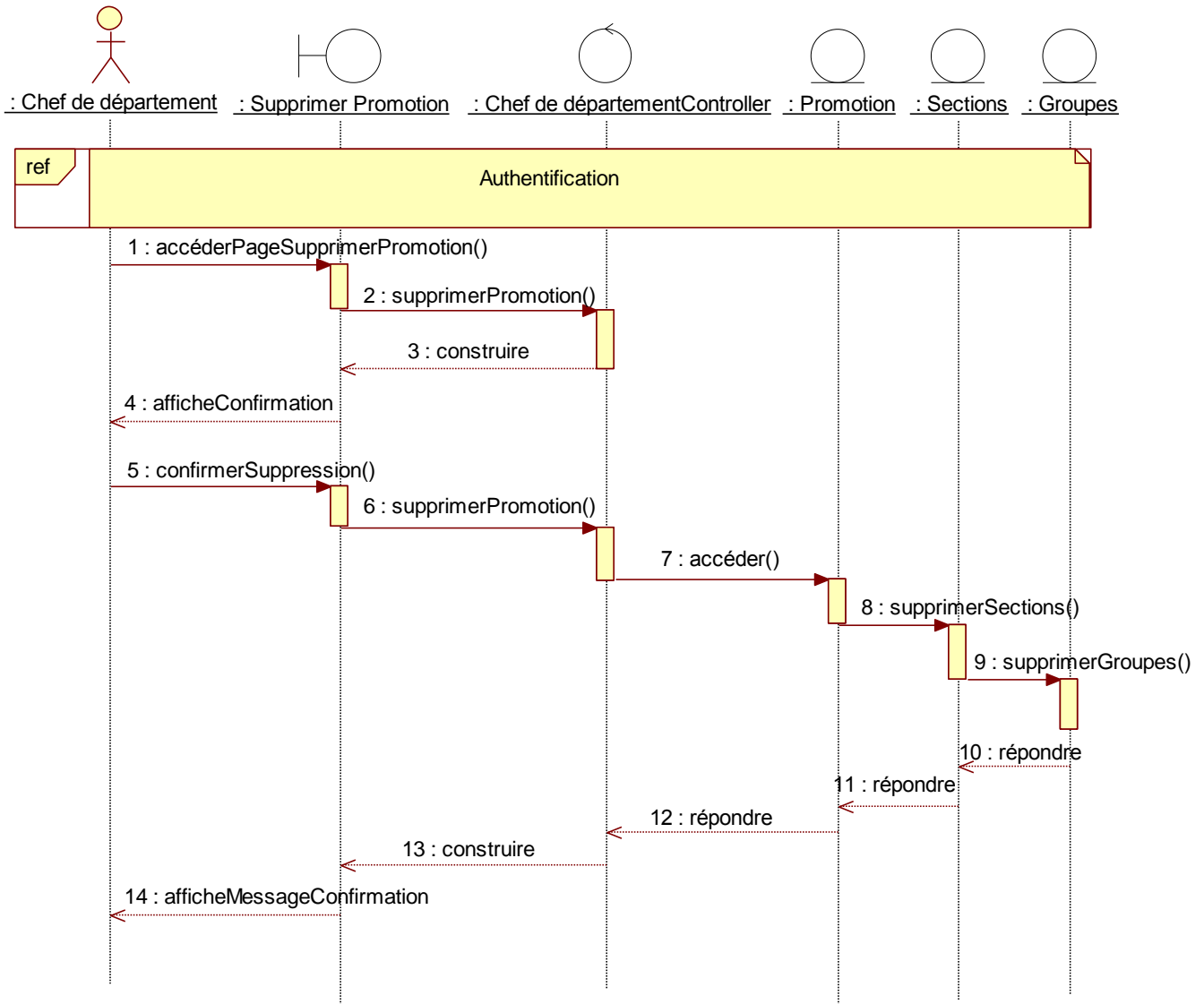


Figure 40 DSD 8- Cas d'utilisation : Supprimer Promotion (Chef de département)

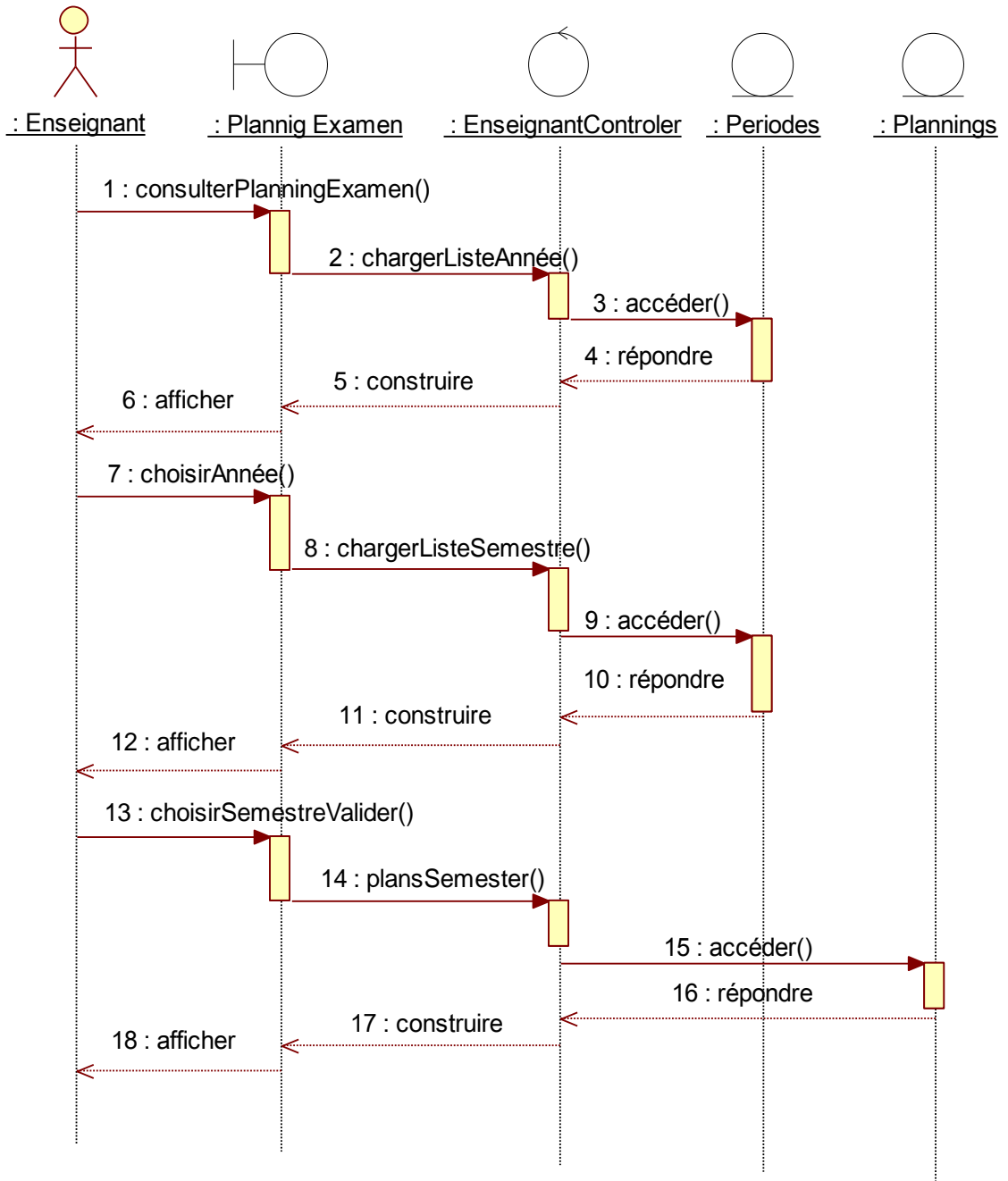


Figure 41 DSD 9 - Cas d'utilisation : Consulter Planning Examen(Enseignant)

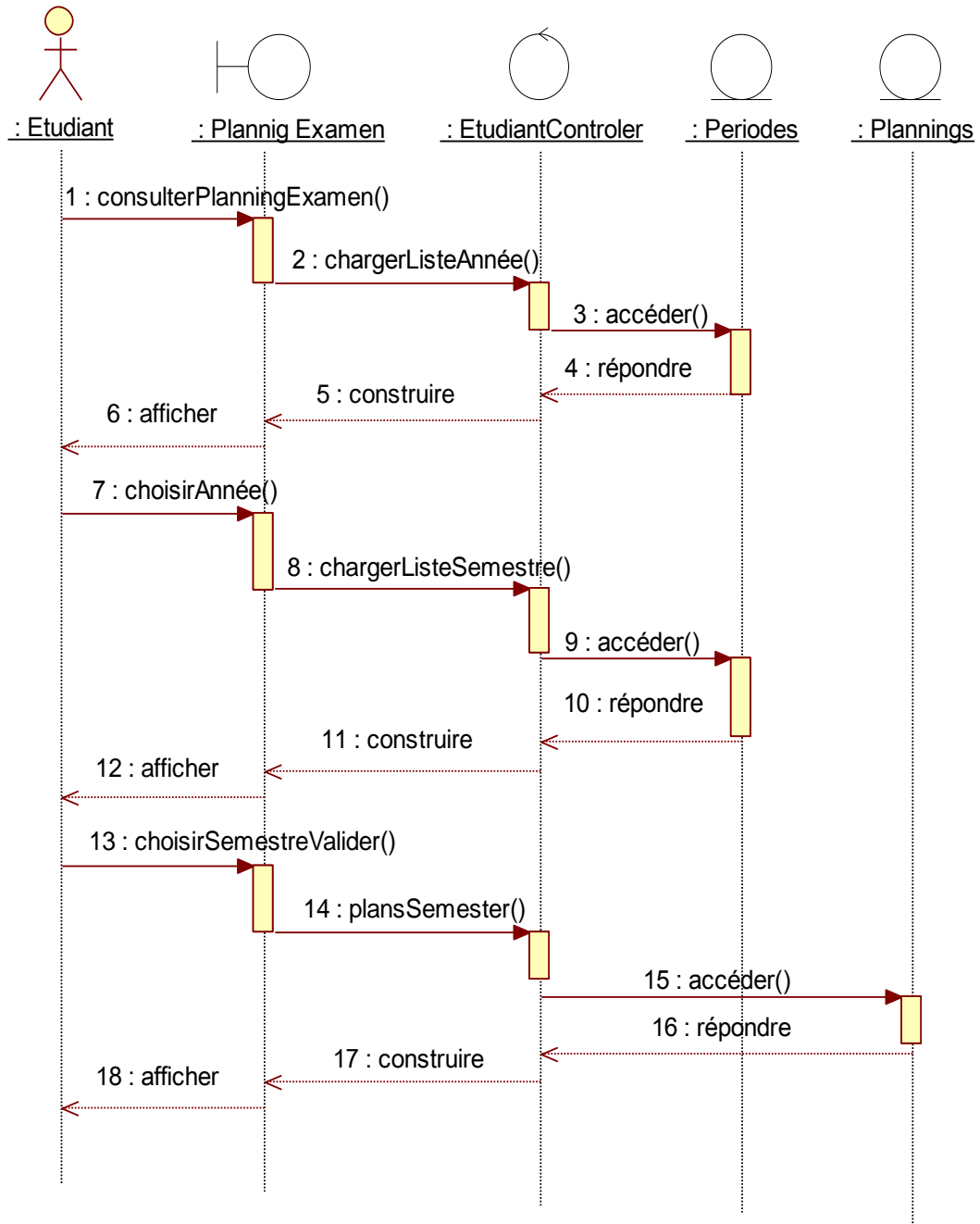


Figure 42 DSD 10 - Cas d'utilisation : Consulter Planning Examen (Etudiant)

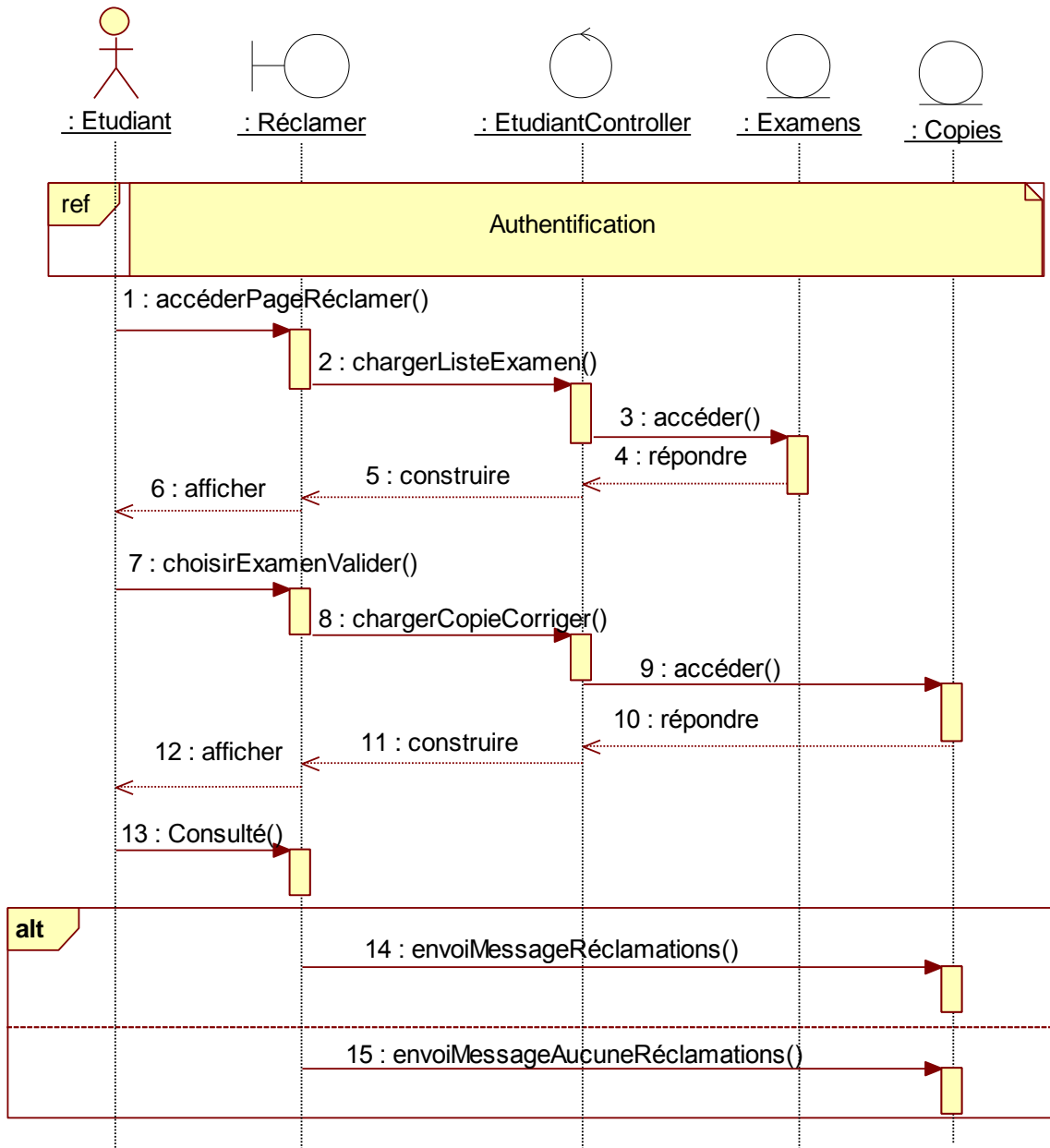
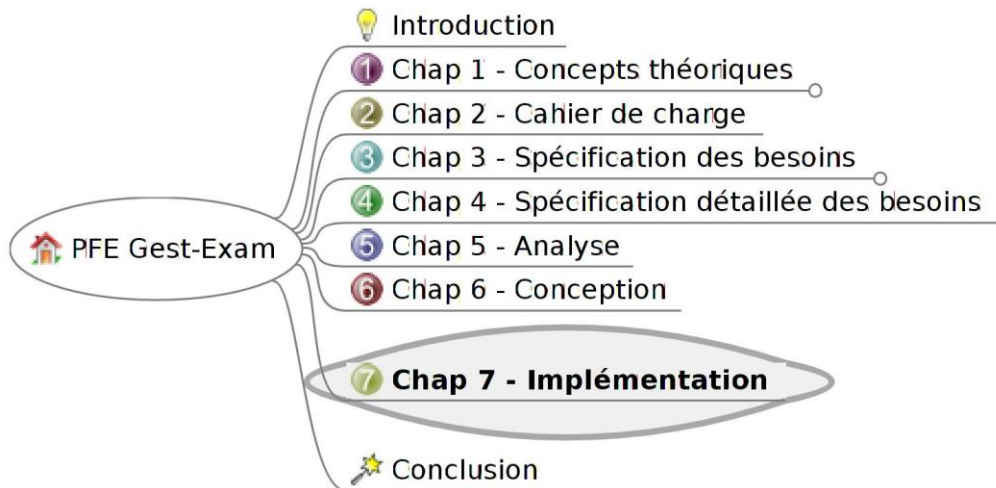


Figure 43 DSD 11 - Cas d'utilisation : Réclamer (Etudiant)

## Chapitre.7 – Conception objet détaillée et implémentation



*Dans cette étape, nous allons faire des choix sur le codage. Nous sommes arrivés pratiquement à la fin du processus de développement. Nous devons prendre des décisions concernant les outils de développement, et les langages de programmation. Bien évidemment, les classes conceptuelles élaborées lors des phases précédentes seront traduites dans les langages de programmation cible.*

---

## 7.1 Solution technologique

L'application que nous avons conçue est un site web dynamique. Donc, elle est accessible depuis un navigateur Web et son code est hébergé sur un serveur. Ce dernier est doté de trois applications serveur : un serveur web (nous avons opté pour Apache), un serveur de base de données (nous avons opté pour MySQL) et un interpréteur de script coté serveur (nous avons opté pour PHP). [6]

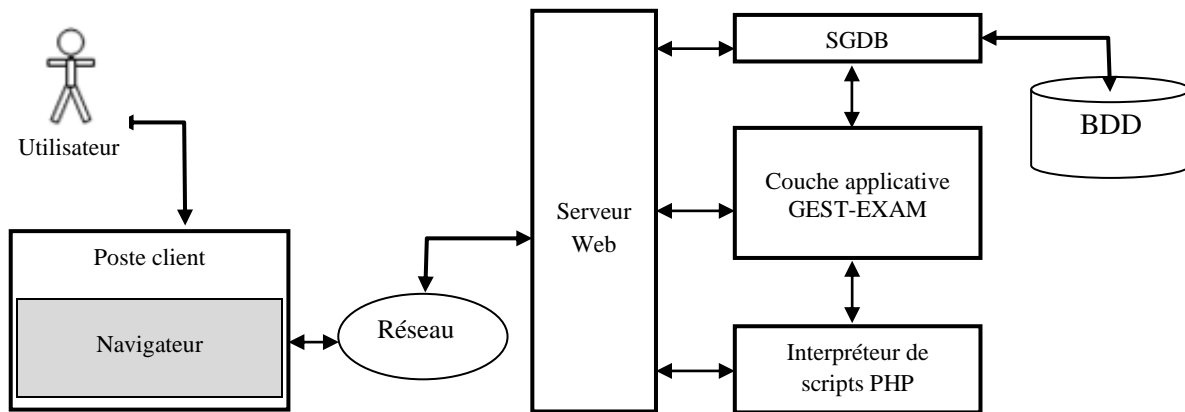


Figure 44 – Interactions entre le navigateur Web et l'application

## 7.2 Le modèle MVC et CodeIgniter

Pour mener à bien le développement du code de notre application, en plus de notre choix du langage de script PHP, nous avons trouvé pertinent de travailler avec un Framework. Ces derniers sont, en majorité, bâtis sur la base de l'architecture MVC (**Modèle Vue Contrôleur**) en anglais design pattern, qui est un modèle de conception logicielle très répandu et fort utile. Créée dans les années 80 par Xerox PARC pour Smalltalk-80. [6]

Le MVC est une méthode d'architecture impose la séparation entre les données, les traitements et la présentation des interfaces Homme-Machine (IHM) d'une application logicielle, ce qui donne trois parties fondamentales dans l'application finale. [6]

- Les données (Modèle).
- L'interface home-machine (Vue).
- La logique de contrôle (Contrôleur).

**Le modèle :** Dans le modèle, on trouve à la fois les données et les traitements à appliquer à ces données. Permet de faire **le lien entre le contrôleur et la base de données**. Il décrit les données manipulées par l'application et définit les méthodes d'accès.

**La Vue :** Les vues sont directement responsables de tout ce que l'on va envoyer aux navigateurs. Les vues seront donc composées principalement de HTML. Nous trouverons aussi du PHP qui nous servira pour afficher des variables et faire quelques boucles.

**Le contrôleur :** Le contrôleur se charge d'appeler tous les composants : bibliothèques, helpers, vues, modèles, mais aussi de faire les nombreuses vérifications nécessaires.

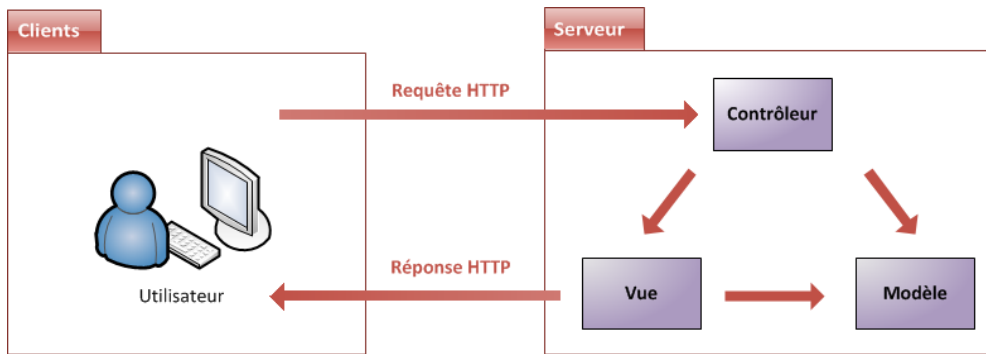


Figure 45 – Architecture MVC

### 7.3 Le Framework CodeIgniter

*CodeIgniter* est un Framework de développement d'application. Il englobe un ensemble d'outils permettant de structurer et de construire des sites Web en utilisant PHP. Son objectif est de permettre le développement de projets beaucoup plus rapidement que si on part de zéro. Ceci est rendu possible grâce aux différentes bibliothèques fournies par ce *Framework*. *CodeIgniter* permet de se concentrer sur la créativité en minimisant la quantité de code nécessaire pour réaliser une tâche donnée.

Voici un schéma qui illustre le fonctionnement de *CodeIgniter*.

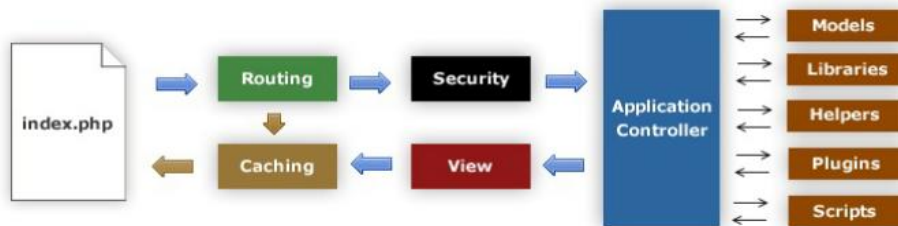


Figure 46 – Fonctionnement de CodeIgniter

Voici quelques différentes étapes de fonctionnement de *CodeIgniter*

- **index.php**: Ce sera toujours le fichier `index.php`, situé à la racine du répertoire, qui sera appelé en premier.
- **Routing**: C'est le routeur. C'est lui qui récupérera l'URL et la décomposera en actions. Il a pour rôle de trouver le contrôleur à appeler.
- **Caching** : Une fois que le routeur a fini son travail, le module de cache va regarder s'il existe des fichiers mis en cache pour cette action. Dans le cas d'une réponse positive, ces fichiers vont être renvoyés au navigateur.
- **Security** : Cette partie va sécuriser toutes les données entrantes : cookies, variables `get`, `post`, etc. C'est la dernière étape avant le contrôleur.
- **ApplicationController** : Ce sera ici que nous commencerons à développer.
- **ModelsLibrariesHelpers** : Comme vous le voyez, le contrôleur fait appel à différents éléments qui vont leur renvoyer des données.
- **View** : Les vues sont les fichiers qui contiennent le code HTML.



## 7.4 Structure des fichiers de GEST-EXAM

L'application GEST-EXAM est basée sur la Framework *CodeIgniter*. L'architecture de ses fichiers est déterminée par celle de *CodeIgniter*. [6] Voici quelques éléments importants que nous jugeons bon de signaler à propos de l'organisation des fichiers de GEST-EXAM :

- Trois dossier importants sont issues directement de la racine de notre application. Il s'agit des dossiers suivants : Application, System et Images.
- Chaque répertoire de notre application contient un fichier index.html permettant d'éviter que l'on puisse lister le contenu de vos répertoires et en déduire la technologie utilisée.

Le dossier « application » contient 9 sous répertoires permettant d'organiser chaque partie de notre application. Voici l'arborescence du dossier application avec les explications des dossiers :

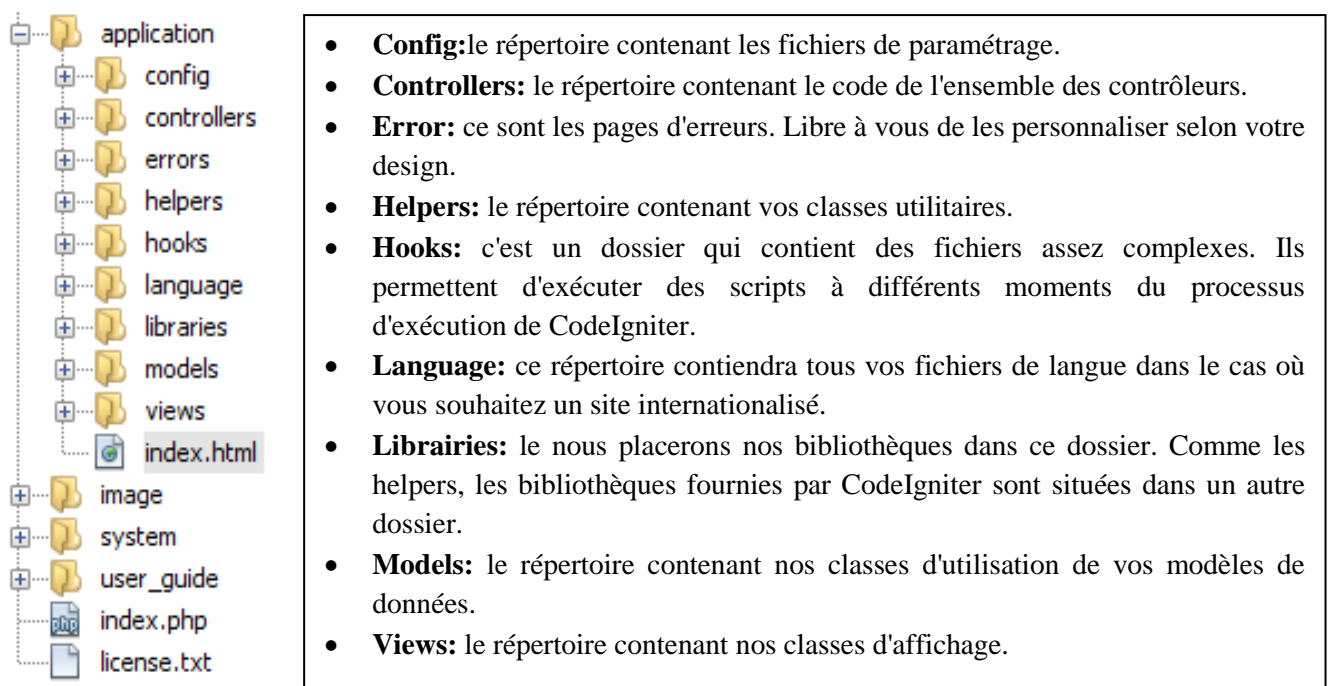


Figure 47– L'arborescence du dossier application avec les explications des dossiers

Le dossier « system » contient tous les codes natifs de *CodeIgniter*. Nous n'allons donc pas modifier les fichiers présents dans ce dossier. [6]

Le dossier « Images » va contenir les fichiers « images » correspondants aux copies des étudiants.

## 7.5 Grocery CRUD

Grocery CRUD est une librairie PHP qui permet de mettre en place dans un site web une gestion d'Ajout, Lecture, Modification et Suppression de vos données.

Si vous écrivez des applications web, vous êtes constamment confronté au même problème : proposer une interface pour mettre en place des tableaux de données avec les opérations de base d'ajout, lecture, modification et suppression. Grocery CRUD est un code prêt à l'emploi qui permet

d'implémenter très simplement ces fonctions de base. Grocery CRUD fonctionne de concert avec CodeIgniter, fameux Framework PHP.

## 7.6 Quelques exemples des codes

Voici un exemple de code présentant le contrôleur Page d'accueil :

```
<?php

if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Login extends Controller {
    function __construct(){
        parent::Controller();
        $this->load->helper(array('form'));
    }

    function index()
    {

        $this->load->view('login_view');
    }
}

?>
```

Figure 48 – Code présentant le contrôleur

Voici un exemple de code présentant la vue Page d'accueil :

```
<?php
echo form_open('verifylogin');
echo "<br>";
echo "<table width = '100%'> <TR><TD align='right'>";
echo form_label('Nom d\'utilisateur :', 'nomuser');
echo "</TD><TD>";

$data = array(
    'name'      => 'username',
    'id'        => 'username',
    'value'     => '',
    'maxlength' => '50',
    'size'      => '25',
    'style'     => 'width:200',
);
echo form_input($data);
echo "</TD><TR><TD align='right'>";
echo form_label('Mot de passe :', 'mdp');
echo "</TD><TD>";
$data = array(
    'name'      => 'password',
    'id'        => 'password',
    'value'     => '',
    'maxlength' => '50',
    'size'      => '25',
    'style'     => 'width:200',
);
echo form_password($data);
echo "</TD></TR></table>";
echo "<P align='center'> ";
echo form_submit('Login', 'Se connecter');
echo "</p>";

?>
```

Figure 49 – Code présentant la vue

Voici un exemple de code présentant la model Page d'accueil :

```

k?php
Class User extends Model
{
    function login($username, $password)
    {
        $this -> db -> select('id, username, password');
        $this -> db -> from('users');
        $this -> db -> where('username', $username);
        $this -> db -> where('password', MD5($password));
        $this -> db -> limit(1);

        $query = $this -> db -> get();

        if($query -> num_rows() == 1)
        {
            return $query->result();
        }
        else
        {
            return false;
        }
    }

    function isAdmin($idd)
    {
        $this -> db -> select('id_admin');
        $this -> db -> from('administrateur');
        $this -> db -> where('id_user_utilisateur', $idd);
        $this -> db -> limit(1);
        $query = $this -> db -> get();

        if($query -> num_rows() == 1)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    function isChef($idd)
    {
        $this -> db -> select('id_chef_departement');
        $this -> db -> from('chef_departement');
        $this -> db -> where('id_user_utilisateur', $idd);
        $this -> db -> limit(1);

        $query = $this -> db -> get();

        if($query -> num_rows() == 1)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    function isEns($idd)
    {
        $this -> db -> select('id_enseignant');
        $this -> db -> from('enseignant');
        $this -> db -> where('id_user_utilisateur', $idd);
        $this -> db -> limit(1);

        $query = $this -> db -> get();

        if($query -> num_rows() == 1)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
?>

```

Figure 50 – Code présentant le model

Voici l'interface obtenue des exemples précédents Page d'accueil :

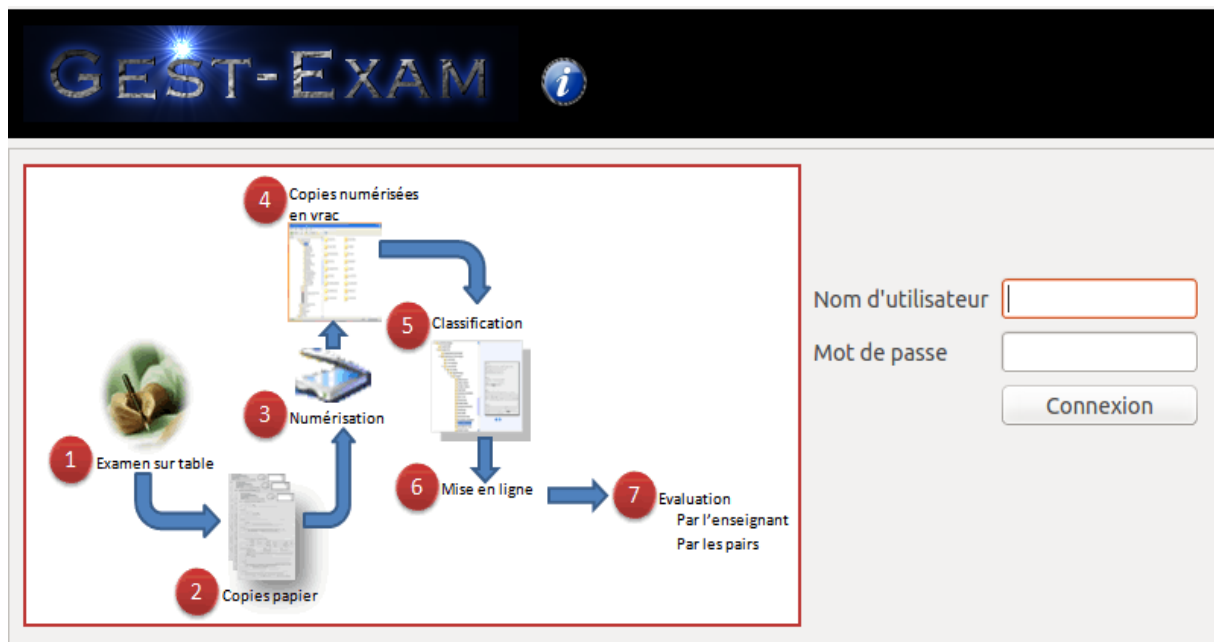


Figure 51– L'interface obtenue la page d'accueil

Voici un exemple de code présentant le contrôleur Page d'enseignant :

```

k?php
class Copies extends Controller
] {
    function __construct()
    ]     {
        parent::__construct();
        $this->load->model('copies_model');
        $this->load->model('etudiant_model');
-     }
    function index()
    ]     {
        $id_etud=0;
    ]     if(!empty($_POST['marry'])) {
        $id_etud = $_POST['marry'];
        $data['copies_list'] = $this->copies_model->get_copies_etud( $id_etud);
    ]     } else {
        $data['copies_list'] = $this->copies_model->get_copies_etud(0);
-     }
        $data['idetud']=$id_etud;
        $data['etudiant_list'] = $this->etudiant_model->get_all_etudiant();
        $this->load->view('copies_list',$data);
-     }
- }
    
```

Figure 52– Code présentant le contrôleur d'enseignant

Voici un exemple de code présentant la vue Page d'enseignant :

```

        <a>Etudiant</a> :
        <select name="marry" id="marry" onchange="this.form.submit()">
            <option value="0">Selection l'etudaint</option>
            <?PHP
foreach($etudiant_list as $row)
    {
    ?>
        <option value="<?php echo $row->id_etudiant; ?>" <?php if($idetud==$row->id_etudiant)
            {echo "selected='1'";}><?php echo $row->nom_etudiant; ?></option>
            <?PHP
    }
    ?>
        </select>
    </form>
    </td>

        <td>
            <br></br>
            <table width="308" height="156" border="0" align="center" bgcolor="#FFFFFF">

                <?PHP
foreach($copies_list as $row)
    {
    ?>
        <tr>

            <td>
                <?php echo ''; ?
            </td>
        </tr>
    <?PHP
    }
    ?>
    
```

Figure 53– Code présentant la vue d'enseignant

Voici un exemple de code présentant la model Page d'enseignant :

```

k?php
class Copies_model extends Model {

    //put your code here
    function __construct() {
        parent::__construct();
    }

    function get_all_copies() {
        $query = $this->db->get('copies');
        return $query->result();
    }

    function get_copies_etud($idetud) {

        $this->db->select ('id_copies,LienImage,id_etudiant_Etudiant,id_module_Module,messagereponde');
        $this->db->from('copies');
        $this->db->where(array('id_etudiant_Etudiant'=>$idetud));
        $query = $this->db->get();
        return $query->result();
    }

    function get_copies_mod($idmod) {

        $this->db->select ('id_copies,LienImage,id_etudiant_Etudiant,id_module_Module,messagereponde');
        $this->db->from('copies');
        $this->db->where(array('id_module_Module'=>$idmod));
        $query = $this->db->get();
        return $query->result();
    }

    function addmessagereponde_new_reclamation($messagereponde,$id_module_Module){
        $data = array(
            'messagereponde'=>$messagereponde,
            'id_module_Module'=>$id_module_Module,
        );
        $this->db->insert('copies',$data);
    }

}

```

Figure 54– Code présentant le model d'enseignant

Voici l'interface obtenue des exemples précédents Page d'enseignant :

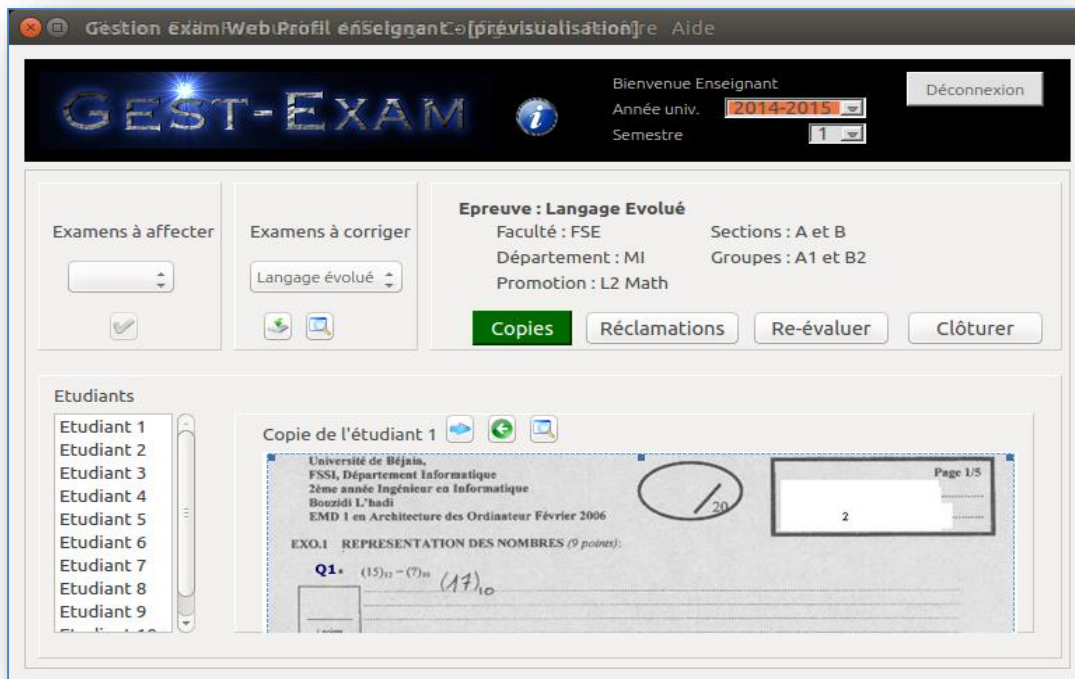


Figure 55– L'interface obtenue la page d'enseignant (copies)

Voici un exemple de code présentant le contrôleur Page d'étudiant :

```

<?php
class Copie extends Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('copies_model');
        $this->load->model('module_model');
        $this->load->model('reclamation_model');
        $this->load->model('etudiant_model');
    }

    function index()
    {
        $id_mod=0;
        if(!empty($_POST['marry'])){
            $id_mod = $_POST['marry'];
            $data['copie_list'] = $this->copies_model->get_copies_mod( $id_mod);
        } else {
            $data['copie_list'] = $this->copies_model->get_copies_mod(0);
        }

        $data['idmod']=$id_mod;
        $data['module_list'] = $this->module_model->get_all_module();
        $this->load->view('copie_list',$data);

        function addmessagerepondesubmit(){
            if(!$_this->input->post('mysubmit'))
                {redirect('copie/addmessagereponde');}

            $messagereponde =$_this->input->post('messagereponde');
            $id_module_Module =$_this->input->post('id_module_Module');
            $this->copies_model->addmessagereponde_new_reclamation($messagereponde,$id_module_Module);
            redirect('reclamation/index');
        }
    }
}

```

Figure 56– Code présentant le contrôleur d'étudiant



Voici un exemple de code présentant la vue Page d'étudiant :

```

<a>Choisir le module</a> :
<br></br> <br></br>
<select name="marry" id="marry" onchange="this.form.submit()">
<option value="0">Selection le module</option>
  <?PHP
    foreach($module_list as $row)
    {
      ?>
      <option value="<?php echo $row->id_module; ?>" <?php if($idmod==$row->id_module)
        {echo "selected='1'";}><?php echo $row->nom_module; ?></option>
      <?PHP
    }
  ?>
</select>
</form>
<br></br>
  <p align="center">Reponse d'un enseignant:</p>
  <table width="200" height="20" border="0" cellpadding="0" cellspacing="0" align="center" bgcolor="#ffffff"
  <tr>
    <td>
      <?PHP
      foreach($copie_list as $row)
      {
        ?>
        <tr>
          <td><?php echo $row-> messagereponde; ?></td>
        </tr>
      <?PHP
    }
  ?> </td>
  </tr>
</table>
</td>
  <td>
    <br></br>
    <table width="308" height="156" border="0" align="center" bgcolor="#FFFFFF">
      <?PHP
        foreach($copie_list as $row)
        {
          ?>
          <tr>
            <td>
              <?php echo ''; ?>
            </td>
          </tr>
        <?PHP
        }
      ?>
    </table>
    <br></br>
  </td>
</center>
<?php
$this->load->helper('form');
echo form_open('reclamation/addmessagereclamationsubmit');
echo form_input(array('name'=>'messagereclamation','value'=>''));
echo form_input(array('name'=>'id_etudiant_Etudiant','value'=> $idmod,'type'=> 'hidden'));
echo form_submit('mysubmit', 'reclame');
echo form_close();
?>

```

Figure 57– Code présentant la vue d'étudiant

Voici un exemple de code présentant la model Page d'étudiant :

```

k?php
class Copies_model extends Model {

    //put your code here
    function __construct() {
        parent::__construct();
    }
    function get_all_copies() {
        $query = $this->db->get('copies');
        return $query->result();
    }
    function get_copies_etud($idetud) {

        $this->db->select ('id_copies,LienImage,id_etudiant_Etudiant,id_module_Module,messagereponde');
        $this->db->from('copies');
        $this->db->where(array('id_etudiant_Etudiant'=>$idetud));
        $query = $this->db->get();
        return $query->result();
    }
    function get_copies_mod($idmod) {

        $this->db->select ('id_copies,LienImage,id_etudiant_Etudiant,id_module_Module,messagereponde');
        $this->db->from('copies');
        $this->db->where(array('id_module_Module'=>$idmod));
        $query = $this->db->get();
        return $query->result();
    }
    function addmessagereponde_new_reclamation($messagereponde,$id_module_Module){
        $data = array(
            'messagereponde'=>$messagereponde,
            'id_module_Module'=>$id_module_Module,
        );
        $this->db->insert('copies',$data);
    }
}
}

```

Figure 58– Code présentant le model d'étudiant

Voici l'interface obtenue des exemples précédents Page d'étudiant :

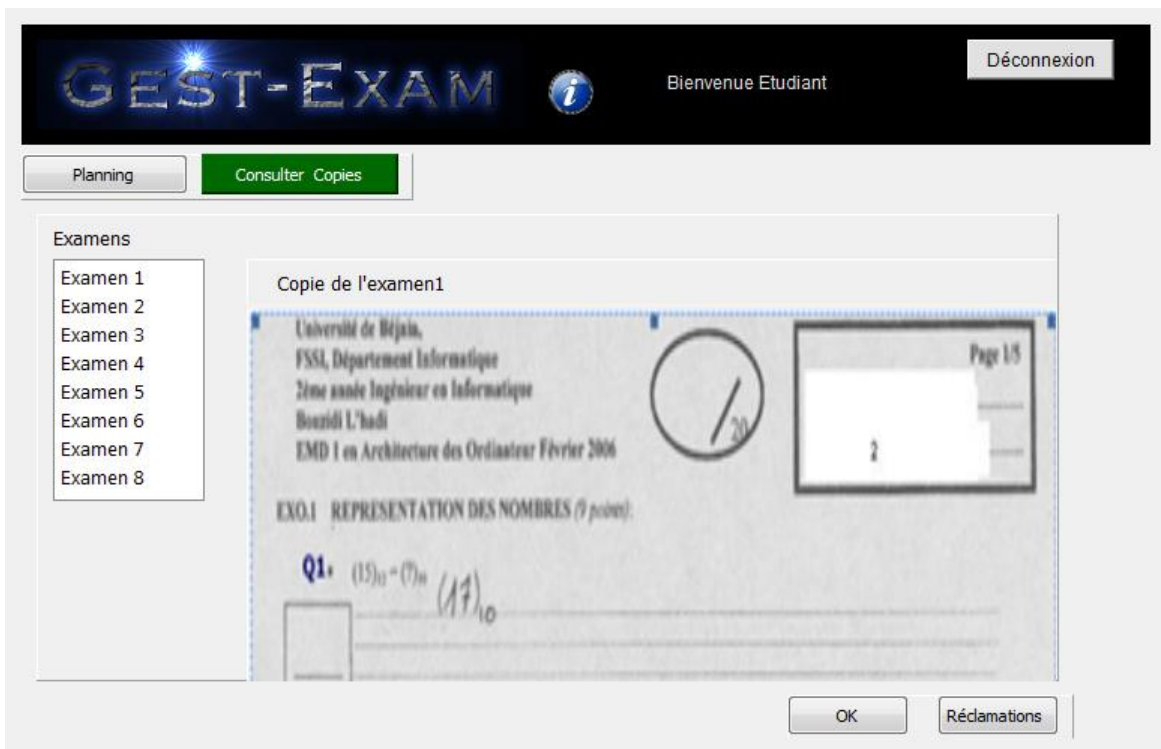


Figure 59– L'interface obtenue la page d'étudiant (Consulter copies)

## Conclusion (bilan et perspectives)

La gestion des examens est un problème majeur pour les institutions éducatives et universitaires. Le défi est de garantir les bonnes conditions d'évaluation des étudiants et de réduire au maximum les risques de stress et d'erreurs. Les méthodes manuelles traditionnelles ont montré leurs limites dans la gestion de grands flux d'informations notamment lorsque le nombre d'étudiants est important. De ce fait, l'intérêt d'automatiser le processus de numérisation, d'archivage et de gestion des examens devient évident. Cette problématique a été le point de départ de notre projet de fin d'étude. On a fixé comme objectif d'atténuer les problèmes qui se posent lors de l'après examen, notamment lors de l'affichage des notes, de la correction et de la consultation des copies corrigées. On a voulu introduire les technologies sans altérer les pratiques habituelles des enseignants. Concernant l'affichage des notes, notre système offre à tous les étudiants la possibilité de consulter leurs notes à tous moment, évitant ainsi les désagréments de la méthode classique où l'affichage n'est pas facilement accessible (bousculade, liste d'affichage illisibles, ...etc.). Contrairement à la pratique traditionnelle des consultations effectuées en classe qui engendre beaucoup de stress pour l'enseignant et l'étudiant, notre projet donne à chaque étudiant les moyens (le corrigé de l'examen, le temps) nécessaires pour consulter sa copie et poster des réclamations éventuelles. D'autre part, avec ce système, l'enseignant aura suffisamment de temps pour traiter chaque réclamation. Au-delà de l'intérêt et du bénéfice que pourraient avoir ce projet pour les institutions éducatives et universitaires, sa réalisation a été, pour nous, une véritable expérience pédagogique et de recherche. En effet, il nous a permis de vivre pratiquement toutes les étapes de développement d'applications logicielles. Dès le début de ce travail, nous nous sommes confronté au choix d'une méthode de conception qui soit à la fois cohérence et simple d'utilisation. Le choix n'a pas été facile mais la lecture de l'ouvrage de Pascal Roques nous a énormément aidés. En effet, nous avons suivi sa démarche qui est pilotée par les cas d'utilisation et le formalisme UML en partant des besoins pour aboutir au code de l'application. L'un des points essentiels cette démarche est d'assurer une cohérence entre les différentes étapes du processus de développement. Arrivé à la fin de la conception, s'est posé, pour nous, le problème de l'implémentation. Là aussi, il a fallu faire des choix et surtout acquérir de nouvelles compétence comme la maîtrise du langage PHP, la programmation orientée objet, l'application du modèle MVC et l'exploitation d'un *Framework*(*CodeIgniter*).

## Bibliographie

[0] <https://elearning.univ-bejaia.dz>.

[1] Pacal Roques, UML 2 : modéliser une application Web, 2 édition Eyrolles, 2006, 236 pages ème

[2] WELLING, Luke. 2009. PHP et MySQL. s.l. : Pearson Education, 2009.

[3] DEBRAUWER, Naouel KARAM - Fien VAN DER HEYDE - Laurent. 2009. *UML2 Maîtrisez le langage de modélisation*. s.l. : ENI, 2009.

[4] Jim Conallen, ‘Concevoir des applications WEB avec UML’, Edition Eyrolles 2000

[5] Laurent AUDIBERT, ‘UML 2’, Institut Universitaire de Technologie de Villetaneuse – Département Informatique, Édition 2007-2008.

[6] TRIVIAK et KITTEL J.« CodeIgniter, le Framework au service des Zéros », 08 Janvier 2013, In [<https://openclassrooms.com/courses/codeigniter-le-framework-au-service-des-zeros>]

[7] Lin, S. et al. ‘Web Based Peer Assessment: Attitude and chievement’, IEEE Transactions on Education, Vol 9, No 4, 2001.

[8] Piéron, H. et al. (1934). , Études docimologiques sur le perfectionnement des examens et concours.. Paris : CNAM.