



N° d'ordre :

UNIVERSITE DE Mohammed Boudiaf M'sila
FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE
Département d'Informatique

MEMOIRE de fin d'étude
Présenté pour l'obtention du diplôme de MASTER
Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Systèmes d'Informations Avancés
Par : TOUAMA Hakim

SUJET

**Système d'information collaboratif pour
Les transferts inter-facultés**

Soutenu publiquement le : 15 / 06 /2013 devant le jury composé de :

Nom et prénom Enseignant

Mr. Amroune Nasser

Mr. MEHENNI Tahar

Mr. Guerna Abderrahim

Université de M'sila

Université de M'sila

Université de M'sila

Président

Rapporteur

Examineur

Promotion : 2014/2015

Remerciements

*Nous tenons à exprimer nos vifs remerciements à **ALLAH** qui nous a donné la volonté et la patience afin de finir notre étude malgré les difficultés rencontrées.*

Nous tenons à remercier par ce travail tous ceux qui nous ont aidés de loin ou de près à réaliser ce mémoire de fin d'étude :

*Nous remercions en particulier **MEHENNI Tahar**, notre encadreur qui a bien voulu accepter l'encadrement de ce mémoire et nous a apporté des conseils précieux durant toutes les étapes de ce travail et Nos vifs remerciements s'adressent également à **M^R. ben Lakhdar Djamel** et **M^R. Mokhtari** pour leurs aides.*

Nous souhaitons également remercier les membres du jury pour nous avoir fait l'honneur d'accepter de juger et d'évaluer notre travail.

Nous remercions tous nos enseignants pour toutes les connaissances.

Qu'ils nous ont inculquées tout au Lang des cinq années.

Table des Matières

Introduction Général

CHAPITRE 1 : Système d'information collaboratif(SIC)

1. introduction	01
2. Système d'information.....	01
2.1. Définition de système d'information.....	01
2.2. Une approche systémique d'un système d'information.....	02
2.3. Une approche fonctionnelle d'un système d'information	03
2.4. Une approche structurelle d'un système d'information	03
2.5. Les composantes d'un SI	04
2.6. Rôle d'un système d'information.....	05
3. Architecture de système d'information	05
3.1. Architecture fonctionnelle.....	06
3.2. Architecture logique.....	06
3.3.Architecture informatique (ou physique).	06
4. La collaboration.....	07
4.1. Concept de Système d'Information Collaboratif	07
4.2. Le Système d'Information Collaboratif	08
4.3. Architecture de base du Système d'information Collaboratif	09
4.4. Gestion de processus collaboratif	09
4.5.Définition des partenaires de la collaboration	10
5. Les modèles	10
5.1.Le modèle	10
5.2. Le métamodèle	11
5.3. Le métamétamodèle	11
6. Classification des applications web	12
6.1.Les applications web de diffusion (Web1.0)	12
6.2. Le web collaboratif (Web2.0 et plus)	13
7. conclusion.....	14

CHAPITRE 2 : Conception d'un SIC Pour les Transfert Inter-faculté

1. Introduction.....	15
2. les scénarios de Transfert.....	15
3. UML.....	18
3.1. Présentation d'UML.....	18
4.Processus de développement suivi	20
5. Diagrammes des cas d'utilisation	20
5.1. Description textuelle des cas d'utilisation	23
6. Regroupement des cas d'utilisation en paquetages.....	29
6.1. Digramme de Paquetage	29
7. Diagramme de classes	29
8. Les Diagramme d'interaction	32
8.1. Diagramme de collaboration	32
8.2. Diagramme de séquence.....	34
9.Conception de la base de données.....	38
9.1. Introduction sur la base de données.....	38
9.2. Importance de clé relationnel (schéma relationnel).....	38
9.3. le schéma relationnel.....	39
10. Conclusion.....	43

CHAPITRE 3 : Implémentation

1.Introduction.....	44
2. Les outils de développement	44
2.1. Python	44
2.1.1. Python	44
2.1.2. Domaines d'application	45
2.2. Framework Django	46
2.2.1. Django.....	46
2.2.2. Pourquoi utiliser ce Framework	47

2.3. Présentation de l'architecture MVC	48
2.3.1. Avantages du MVC	48
2.3.2. Inconvénients du MVC	49
2.3.3.MVC.....	49
2.3.4. La spécificité de Django	50
2.4. Présentation de PostgreSQL	51
2.4.1. Caractéristiques de PostgreSQL.....	52
2.5. serveur web	53
2.5.1. Présentation Apache.....	53
3. Les Différents tables que nous avons créent avec postgresql.....	53
3.1. Présentation de la base de données	53
4. Présentation de l 'application web.....	55
4.1. Authentification	55
4.2. Espace Administrateur.....	55
4.3. Espace d'ajouter demande	56
4.4. Consulter les demandes	56
4.5. Création fiche de Transfert	57
4.5. Consulter fiche de Transfert	58
4.5. Décision final	59
4.5. Consulter Décision	60
5. Conclusion.....	60

Liste des figures

Figure 1.1 :	Vue systémique d'un système d'information .	2
Figure 1.2 :	Une approche structurale d'un système d'information.	4
Figure 1.3 :	Système d'information collaboratif.	8
Figure 1.3 :	Système d'information Collaboratif (médiateur)	9
Figure 1.4 :	Classification des modelés	11
Figure 2.1 :	Schéma décrit Scenario 1	16
Figure 2.2 :	Schéma décrit Scenario 2	17
Figure 2.3 :	Schéma décrit Scenario 3	18
Figure 2.4 :	les vues d'UML	19
Figure 2.5 :	processus de développement	20
Figure 2.6:	diagramme de cas d'utilisation <User faculté + User département>	22
Figure 2.7 :	diagramme de cas d'utilisation<Admin>	23
Figure 2.8 :	diagramme de paquetage	29
Figure 2.9 :	diagramme de classe <paquetage donnée>	31
Figure 2.10 :	diagramme de collaboration <Authentification>	32
Figure 2.11 :	diagramme de collaboration <Demande de transfert>	33
Figure 2.12 :	diagramme de collaboration <Décision de transfert>	33
Figure 2.13 :	diagramme de collaboration	34
Figure 2.14:	diagramme de séquence <Authentification>	35
Figure 2.15:	diagramme de séquence < Demande a son département >	35
Figure 2.16:	diagramme de séquence < Consulter la Demande >	36
Figure 2.17 :	diagramme de séquence < creation Fiche de Transfert >	36
Figure 2.18:	diagramme de séquence < Consulter fiche de Transfert >	37
Figure 2.19 :	diagramme de séquence <décision finale >	37
Figure 2.20 :	diagramme de séquence < Consulter la décision finale >	38

Figure 3.1 :	Logo python	45
Figure 3.2 :	Logo Django	46
Figure 3.3 :	Approche MVC	48
Figure 3.4 :	Schéma de l'architecture MVC	50
Figure 3.5 :	Schéma d'exécution d'une requête (MVT)	51
Figure 3.6 :	logo postgresSQL	51
Figure 3.7:	la structure de la base de donnée(postgreSQL)	53
Figure 3.8 :	La Table User	53
Figure 3.9:	La Table demande	54
Figure 3.10 :	La Table fiche de transfert	54
Figure 3.11 :	La Table décision finale	54

Liste des tables

Table 01 :	Structure de la table utilisateur	40
Table 02 :	Structure de la table demande	40
Table 03 :	Structure de la table Fiche de transfert	41
Table 04 :	Structure de la table décision finale	41
Table 05 :	Structure de la table Faculté	42
Table 06 :	Structure de la table département	42

Introduction générale

L'évolution de l'utilisation des systèmes d'informations offre des possibilités considérables et accroît les chances des échanges d'information à travers un temps rapide, Le système d'information permet de regrouper, classifier, traiter et diffuser des informations sur un environnement donné.

Actuellement, le système d'information a besoin de collaboration pour offrir la possibilité de coopération et joue le rôle de système intermédiaire assurant la cohésion entre les partenaires potentiels.

Dans plusieurs domaines on utilise les système d'information collaboratif clairement dans les entreprises et généralement dans les domaines économique, d'autres domaines qui ont besoin des systèmes d'informations collaboratif pour éliminer plusieurs problèmes on essayer de manipulée d'autre avec ce système

On remarque qu'il ya plusieurs problèmes dans les universités qu'on peut les résoudre par la system de collaboration comme c'est le cas de transfert des étudiants habituel ,parmi ces problème on perds beaucoup de temps ,l' absence de coopération au sein de l'université et les opérations de l'assurance de transfert des étudiants .

Notre objectif dans ce travail est de concevoir, implémenter et réaliser un Application web , on essaye de créer et de trouver des solutions aux problèmes de transfert étudiant localement on a créé un système d'information collaboratif pour les transfert étudiants . Dans le but de faciliter l'opération de transfert, on propose plusieurs techniques pour aider l'administration et gère les processus administratif de la meilleure façon pour détruire la méthode traditionnelle de transfert .

Cette application servira à Ajouter les demandes , créer fiche de transfert Qui contient tous les informations de l'étudiant, renvoie les décision de transfert et consulter les demandesetc.

Ce mémoire composée trois chapitres essentielles sont:

- ❖ Le premier chapitre permettra de présenter quelques notions sur le système d'information collaboratif et ses contributions et la Classification des applications web.
- ❖ second chapitre propose quelque scénario de transfert et décri la méthode de l'analyse , la conception et ses différentes phases.
- ❖ Enfin, le troisième chapitre sera consacré à l'exposition des outils de développement et les moyens utilisés pour réaliser notre Application avec quelque interfaces.

CHAPITRE I

Système d'information collaboratif(SIC)

CHAPITRE 1 : Système d'information collaboratif(SIC)

1. Introduction

Ce chapitre montre le contexte de notre travaux de thèse. Nous nous intéressons tout d'abord à traiter et à définir la notion de collaboration : Qu'est-ce qu'un système d'information ? Quelles sont les l'approche de SI ? Quelles sont les composantes d'un SI sont? Comment peut-on caractériser une collaboration ? Existe-t-il des niveaux de collaboration ? Quels sont les modelés de collaboration ? et enfin la classification des application web .

Par la suite, nous présentons les grandes questions liées à nos travaux. Elles font directement référence à une vision « supérieure » et « idéale » de la notion de collaboration nous présentons un concept de « médiateur », qui assure cette interopérabilité. Enfin, Nous nous intéressons également à la spécification d'un tel concept.

2. Systèmes d'information

Dans cette section, nous allons présenter le concept de système d'information en nous basant sur différentes approches traitant de ce sujet. Nous discutons également des architectures d'un tel système.

2.1. Définition de système d'information

- ❖ Reix, 1995 : Ensemble organisé de ressources : matériel, logiciel, personnel, données, procédures ; permettant d'acquérir, traiter, stocker, communiquer des informations (données, textes, images, sons...) dans les organisations.
- ❖ Mason et Mittrof, 1973 : Tout système d'information concerne un individu, pourvu d'un profil psychologique donné, confronté à un problème précis, dans un contexte organisationnel déterminé.
- ❖ Alter, 1991 : Une combinaison de pratiques de travail, d'informations, de personnes et de technologies de l'information organisée pour atteindre des objectifs dans une organisation.

Dans la suite, nous présentons trois approches différentes qui ont caractérisé les systèmes d'information.

2.2. Une approche systémique d'un système d'information

L'approche systémique positionne un système d'information dans une entreprise ou organisation comme un sous-système interagissant avec d'autres sous-systèmes.

Définit en effet une entreprise comme la composition de trois sous-systèmes :

- **le système opérant** : transforme des intrants (entrées) en extrants (sorties) en fonction d'une finalité donnée. Ce système transforme des ressources ou des flux primaires (flux d'information, flux financier, etc.) en d'autres flux ou ressources.
- **Le système de décision** : (appelé aussi système de pilotage) élabore des commandes (décisions d'action) en fonction d'informations de suivi. C'est le système qui pilote l'entreprise en fonction des objectifs choisis et des stratégies adoptées.
- **Le système d'information** : relie les deux systèmes cités précédemment. Il a pour rôle d'acquérir, mémoriser et transmettre les informations nécessaires au bon fonctionnement de l'entreprise :
 - le comportement du système opérant vers le système de décision,
 - les actions à réaliser par le système opérant, définies par le système de décision. [1]

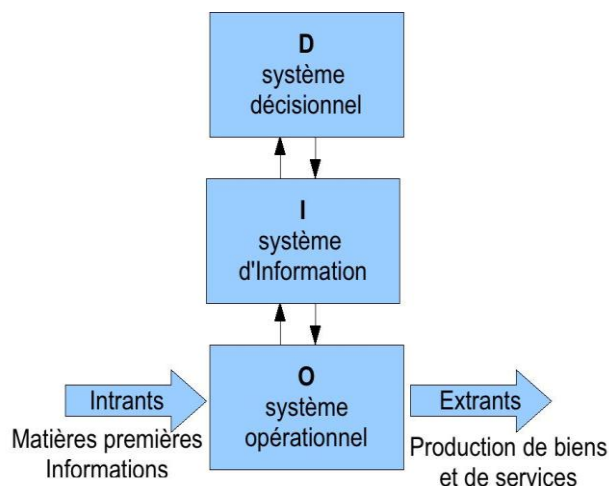


Figure 1.1: Vue systémique d'un système d'information[9]

2.3. Une approche fonctionnelle d'un système d'information

Dans cette approche, on s'intéresse aux fonctions qu'un système d'information propose à ses utilisateurs. Considèrent qu'un « SI doit garantir que la bonne information est disponible au bon endroit et au bon moment ». Le système d'information a naturellement pour objet principal l'information. C'est l'échange d'information qui permet aux utilisateurs de travailler ensemble . Pour que ce travail soit optimal et efficace, la bonne information doit être disponible au bon endroit (bon utilisateur) et au bon moment. C'est la valeur ajoutée d'un système d'information dans l'entreprise. détaille davantage ce qu'offre un système d'information à un utilisateur : un système d'information est un ensemble organisé de ressources :

- matériels (ordinateurs, papier, etc.),
- personnel (clients, responsables, etc.)
- connaissances (modèles, règles, etc.)
- logiciels et procédures (applications informatiques, méthodes de travail, etc.)
qui permet de acquérir des informations, traiter (organiser) des informations,
- stocker (mémoriser) des informations , communiquer des informations. [1]

2.4. Une approche structurelle d'un système d'information

La définition d'un système d'information de (C. Morley) propose de voir un SI comme la composition de deux sous-systèmes :

Le système de traitement de l'information (comprenant les acteurs, les données et les processus) et le système informatique (comprenant les ressources matérielles et logicielles, les bases de données et les fonctions). L'originalité de cette définition du système d'information est qu'elle met l'accent sur les liens forts entre processus et système d'information. En effet, dans le système de traitement de l'information qui est composé d'acteurs. [1]

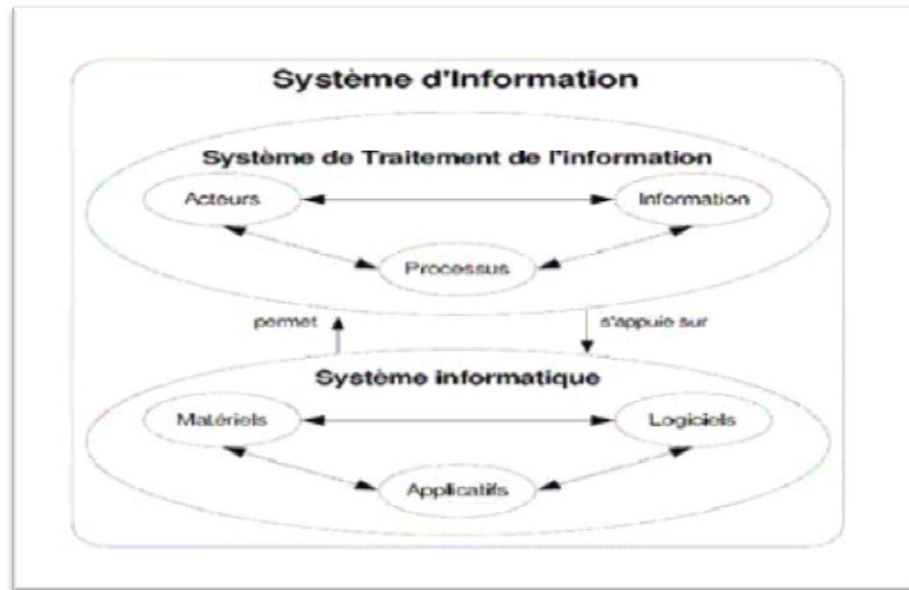


Figure 1.2: Une approche structurée d'un système d'information[8]

2.5. Les composantes d'un SI

- Informations :
 - Circulation des informations/événements,
 - Communications entre les acteurs,
 - Langages de communications.
- Communications:
 - Supports d'informations.
 - Formalisation de la circulation de l'information.
 - Règles de mise en forme et d'interprétation de l'information.
- Organisation :
 - Définir la démarche est indispensable à la réalisation des objectifs.
- L'informatique :
 - Traitement automatique et rationnel de l'information.
 - Enregistrer.
 - Distribuer.
 - Structurer.
 - Analyser.
 - Restituer. [3]

2.6. RÔLE D'UN SYSTÈME D'INFORMATION

Le Système d'Information regroupe l'ensemble des documents, Textes, notes, brouillons, publications, graphiques, photos, enregistrements...Qu'une entreprise ou qu'un organisme utilise dans son fonctionnement quotidien.

Le Système d'Information est géré par les Technologies de l'Information (TI) mais il est important de ne pas les confondre.

Un premier moyen de les distinguer est de dire que le système d'information est un contenu et une logique d'organisation et que le système informatique est un contenant technologique, doté d'automatismes.

3. Architecture de système d'information

L'architecture des systèmes d'information est une discipline multiforme et transversale qui traite de la démarche, des modèles, des outils et des langages permettant de concevoir et construire un système d'information. Elle vise à décrire la structuration d'un système d'information en termes de constituants et d'organisation de ces constituants en tenant compte des multiples niveaux de conception, des aspects de modélisation, mais aussi du degré de généricité du système.

L'architecture représente alors un enjeu fondamental en vue de permettre une certaine maîtrise de la conception et l'évolution des systèmes d'information. La maîtrise de la conception passe par une description détaillée des composants, niveaux et organisations mais également par différents modèles de conception d'un système d'information. [1]

Il existe une typologie riche d'architectures de système d'information. Ces architectures présentent des niveaux d'abstraction différents d'un système d'information.

Nous distinguons :

3.1. Architecture fonctionnelle

permet de spécifier les besoins fonctionnels des utilisateurs de système d'information. Ainsi, un système d'information est présenté en termes de fonctions, Nous parlons aussi d'urbanisme de système d'information, qui consiste à découper le SI en modules autonomes faisant référence aux fonctions de SI. Cette architecture intervient directement pour faire évoluer ou refondre un système d'information.

3.2. Architecture logique

permet d'identifier la structuration d'un système d'information en adoptant une logique indépendante des considérations techniques. Les architectures logiques les plus connues sont : l'architecture orientée composants et l'architecture orientée services (SOA).

Les composants ou les services sont caractérisés par le fait qu'ils sont des entités autonomes, réagissent par échange de messages et sont définis sur différents niveaux de granularité. La différence principale entre un service et un composant réside dans le fait qu'une architecture orientée services se concentre sur les exigences déterminées au niveau de la stratégie et du processus métier , alors qu'une architecture orientée composants se concentre sur les composants d'un logiciel .

Ces composants peuvent être utilisés pour livrer des services. Un service expose dans ce cas un composant.

3.3. Architecture informatique (ou physique)

Décrit la structuration d'un système informatique en termes d'organisation de fonctions et des constituants qui le composent.

Cette architecture définit d'autres « sous-architectures » :

- **Architecture logicielle :** l'agencement et l'interaction des composants logiciels. Ces derniers peuvent être définis sur plusieurs couches (couche présentation, couche applicative, couche données).
- **Architecture matérielle :** l'agencement et l'interaction des composants physiques (disque dur, unité centrale, etc.).
- **Architecture d'intergiciel (Middleware) :** l'agencement et l'interaction des composants servant à faire communiquer plusieurs applications entre elles.
- **Architecture réseau :** l'architecture permettant la communication au sein d'un système d'information.

4. La Collaboration

Le dictionnaire Larousse définit une collaboration comme suit :

Action de collaborer, de participer à une œuvre avec d'autres. Cette définition sous-entend en général l'action de personnes et se concrétise par l'exécution de certains processus pour produire de la valeur ajoutée. Par exemple, des personnes peuvent collaborer par l'échange de documents, suivant un cycle de vie bien identifié.

Dans le cadre informatique, une collaboration se fait aussi bien entre humains qu'entre humains et machines ou entre machines. On alors d'acteurs de la collaboration, acteurs qui peuvent être aussi bien des humains que des machines.

Ainsi, une collaboration est l'échange entre deux ou plusieurs entités actives afin d'atteindre un objectif bien déterminé. Une entité peut prendre plusieurs formes à différentes échelles. Elle peut être à un niveau macroscopique, par exemple une entreprise, une organisation ou un pays, ou microscopique, par exemple, une personne.

Pour une entreprise nous distinguons la collaboration interne, entre les entités qui la composent, ou externe, avec les autres entreprises, fournisseurs ou clients.

Dire qu'il existe une collaboration entre les entreprises n'est qu'un point de vue à une échelle élevée.

Cela est équivalent à dire qu'il existe une collaboration entre des entités qui appartiennent à des entités englobant es différentes. [2]

4.1. Concept de Système d'Information Collaboratif(SIC)

Ce SIC constitue un système d'information inter-organisationnel susceptible de supporter les processus collaboratifs des partenaires. Le principe de cette option est de faire jouer au SIC le rôle de système intermédiaire assurant la cohésion entre les partenaires potentiels. Le SIC devient alors le garant de l'interopérabilité des SI des partenaires. [1]

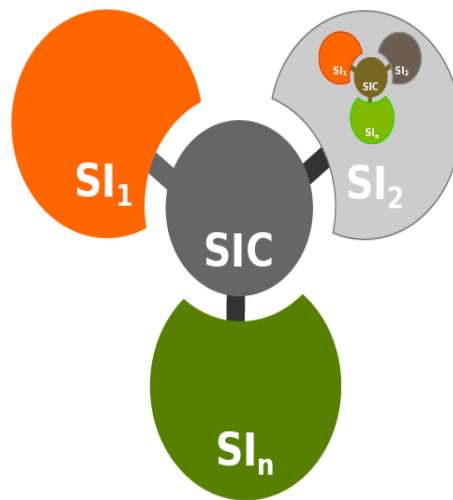


Figure 1.3: Système d'information collaboratif

4.2. Système d'Information Collaboratif (SIC)

Le SIC peut être vu plus largement comme le système qui inclut les parties publiques des systèmes d'information des entreprises et le médiateur. Le médiateur présente le cœur du SIC.

Il permet de faire interopérer les différents systèmes d'information en utilisant des connecteurs directement liés à leurs parties publiques.

Les connecteurs permettent d'assurer les échanges d'information entre le médiateur et les parties publiques des SI. Enfin, le médiateur intervient sur les trois couches :

données, applications et processus d'un système d'information. Il gère les différents niveaux de l'interopérabilité des SI que nous avons identifiés : métier, sémantique et technique. Alimenté d'une définition de processus collaboratif, le médiateur assure une exécution de ce processus en échangeant les différentes données entre SI et en se connectant à leurs différentes applications. [2]

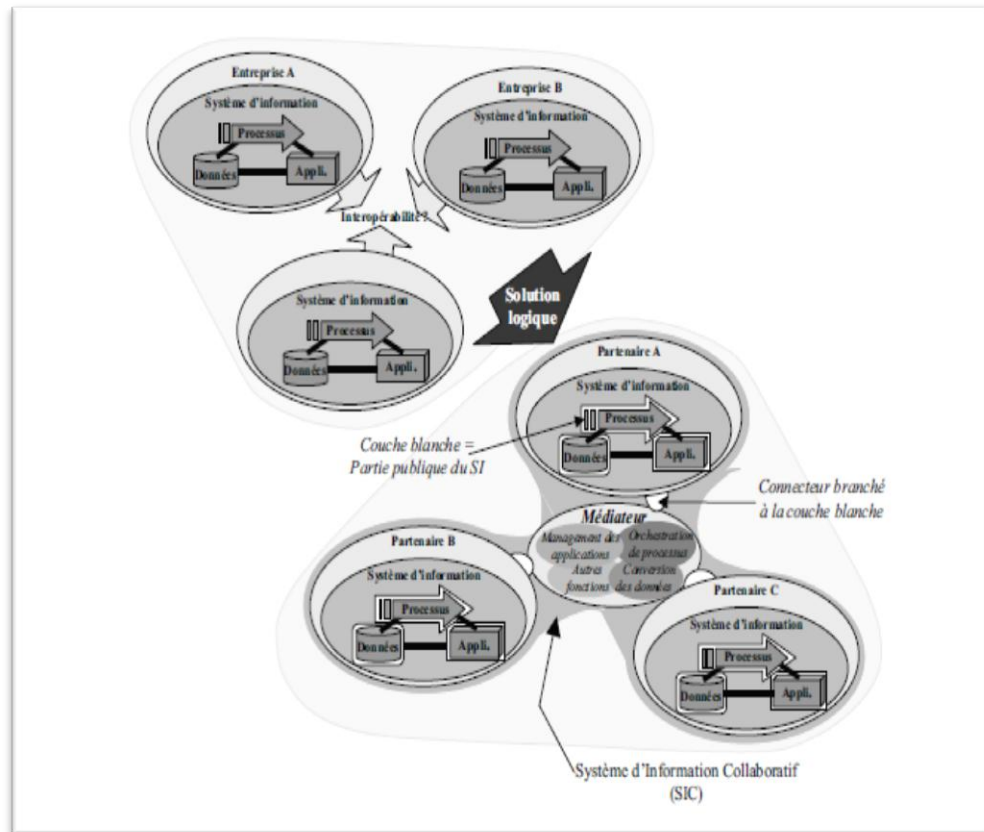


Figure 1.4 : Système d'information Collaboratif (médiateur)[1]

4.3. Architecture de base du Système d'information Collaboratif

Le SIC est composé d'un ensemble de modules interagissant. Chaque module s'occupe d'un niveau de problème de l'interopérabilité des systèmes d'information. Si nous pouvons stigmatiser l'intervention du SIC sur les trois couches d'un système d'information : processus, applications et données, il existe également d'autres modules qui traitent des aspects secondaires, mais complémentaires, pour optimiser la qualité de l'interopérabilité entre SI (sécurité des transactions, gestion des droits, etc.).

4.4. Gestion de processus collaboratif

Un processus collaboratif présente les coordinations, interactions et échanges entre les différents partenaires. Il définit clairement les acteurs impliqués, les données et les fonctionnalités à mettre en œuvre dans la collaboration. Certaines recherches actuelles s'occupent de la flexibilité des processus métiers en la présentant comme un critère primordial pour l'agilité des échanges dans un environnement industriel en perpétuelle évolution.

Dans nos travaux, nous nous concentrons sur la gestion de processus collaboratifs stabilisés. Ce choix de collaboration rigide semble justifiée, à notre avis, par le fait qu'une collaboration demeure aujourd'hui (sans que ce constat soit pour autant voué à demeurer valable très longtemps) une certaine prise de risque pour les partenaires qui n'ont pas d'autres moyens concrets de se rassurer que de s'appuyer sur une définition ferme de la dynamique de la collaboration (qui va faire quoi ?). Néanmoins, notons également que la « réputation » des partenaires et l'enjeu financier constituent également des critères de persuasion conséquents.. Le Système d'Information Collaboratif doit gérer l'exécution du processus collaboratif en synchronisant ce dernier avec les processus internes des partenaires.

Une spécificité de notre vision du processus collaboratif réside dans le fait que le SIC lui-même peut participer à la collaboration en proposant des services métiers. Ces services de support ou à valeur ajoutée participent pleinement aux objectifs du processus global et justifient également le choix d'un système médiateur (porteur de ces compléments, tels que la mise à disposition d'un algorithme de choix performant ou le contrôle des opérations de paiement électronique, etc.). [1]

4.5. Définition des partenaires de la collaboration

Un partenaire de la collaboration, comme nous l'avons déjà expliqué, est considéré comme un ensemble de services variés, proposés par l'entreprise dans la collaboration. Les services encapsulent les applications ou les processus internes d'un partenaire. Les relations entre applications, processus (internes et collaboratifs) et services de l'entreprise sont très dépendante.

5. Les Modèles

Dans cette section nous détaillons les différents types de modèle ,Nous pouvons trouver trois types de modèles :

5.1. modèle

Qui peut être défini comme un ensemble d'informations ou de connaissances en relation avec un domaine donné. Cet ensemble représente un point de vue d'un système réel à un niveau abstrait. Un modèle peut être vu comme une simplification du système réel permettant de répondre à toutes les questions Sous-jacentes à un point de vue donné du système.

Par exemple un modèle BPMN (Business Process Model and Notation) est un modèle qui permet de décrire les processus métier. [2]

5.2. Métamodèle

est un modèle permettant de décrire le modèle. On dit, dans ce cas, qu'un modèle est conforme à un métamodèle. En parallèle avec la notion de langage informatique, le métamodèle est la grammaire et le modèle est le programme.

Par exemple l'ensemble des définitions des concepts de BPMN forme un métamodèle.

5.3. Métamétamodèle

est un modèle permettant de décrire les métamodèles. C'est donc aussi un métamodèle. Afin d'éviter d'avoir un autre métamodèle pour décrire le métamétamodèle, ce dernier se décrit avec lui-même. Dans le parallèle avec la notion de langage, le métamétamodèle est au métamodèle ce que la description BNF (Backus Naur Form) est à une grammaire, le métamodèle est au modèle ce que la grammaire est à un langage.

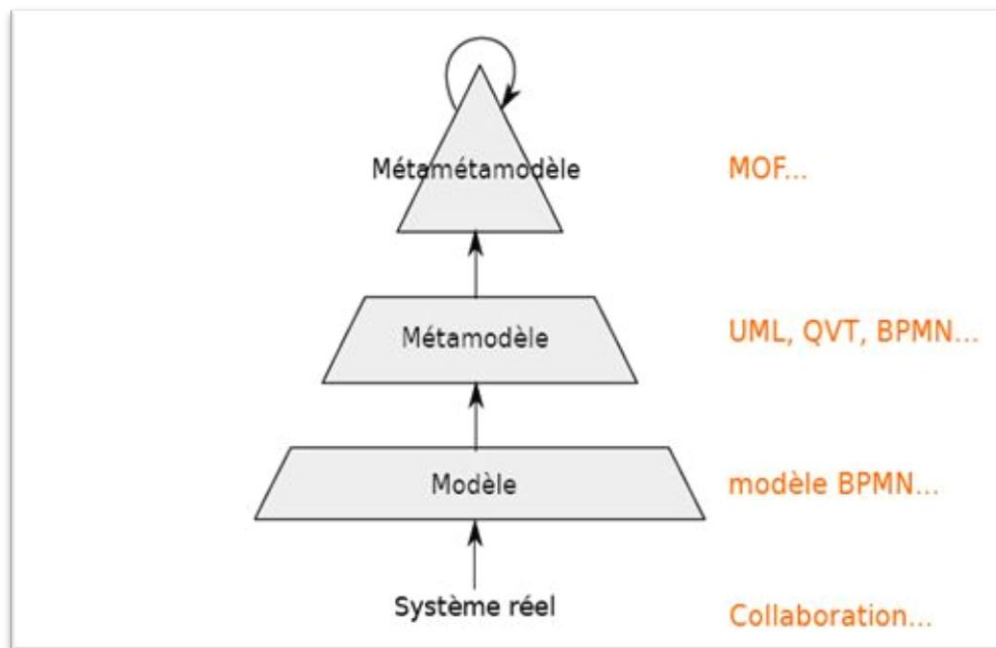


Figure 1.5: Classification des modelés[1]

6. Classification des applications web

Afin de positionner les portails collaboratifs dans le domaine de l'ingénierie des applications web, nous allons tenter de donner une classification des applications web.

Ces applications peuvent être classifiées selon plusieurs critères :

technologie utilisée, architecture, supports, utilité, etc.

Notre travail portant sur les applications collaboratives, nous nous intéressons à l'aspect interactif entre l'application et l'utilisateur. Ainsi, nous distinguons deux types d'application : les applications de diffusion, caractérisées par une faible interactivité, et les applications de collaboration avec une forte interactivité.

Parmi les autres critères nous retenons, aussi, la classification portant sur l'utilité des applications web faite dans cette classification nous distinguons quatre catégories : les sites web de présence sur le Web et les sites web « catalogues » permettant essentiellement de diffuser les données, et les sites web orientés services ainsi que les systèmes d'information basés sur le Web (SIW) caractérisés par leur richesse en fonctionnalités permettant d'agir sur les données. Dans ce qui suit, nous détaillons notre classification. Pour chaque richesse en fonctionnalités permettant d'agir sur les données. [2]

Dans ce qui suit, nous détaillons notre classification. Pour chaque catégorie de cette classification, nous détaillons les catégories d'utilités correspondantes

6.1. Les applications web de diffusion (Web1.0)

Le web a commencé par une présentation statique de l'information sous forme de pages Web reliées par des liens hypertextes. Ces pages étaient écrites par les concepteurs informaticiens, puis déployées sur le serveur pour être enfin interrogées par l'utilisateur. Pour modifier le contenu et la structure de ces pages, les concepteurs informaticiens modifiaient directement le code. Afin d'éviter de modifier le code, les concepteurs informaticiens ont mis en place un système dynamique qui permet de prendre en compte l'évolution des bases de données.

Cependant, les interfaces produites avec ces applications ne permettent pas à l'utilisateur final de l'application d'appliquer des actions métier comme la modification ou la suppression du contenu, ce qui les rend faiblement interactives et limitées à la diffusion de l'information. Ce type de site web est souvent appelé le Web1.0. En parallèle avec la classification faite dans , concernant l'utilité, nous pouvons distinguer, dans la catégorie des applications web de diffusion, deux types d'applications web:

- les sites web de présence sur le Web qui ont comme objectif la mise en ligne d'informations, à des fins publicitaires ou éducatives par exemple.
- les sites web "catalogues" ou à forte densité de données qui sont caractérisées par le fait qu'ils publient une masse importante de données, selon une structure hypertexte complexe, mais pauvre en services. [2]

6.2. Le web collaboratif (Web2.0 et plus)

Le besoin de collaboration et de travail à distance devenant de plus en plus important, les applications du Web1.0 sont devenues inadaptées aux besoins de collaboration. Le passage à un autre type d'application est devenu nécessaire. Ce type d'application est identifié comme le Web 2.0. Cette classe d'application offre à l'utilisateur des interfaces de contrôle lui permettant d'ajouter, modifier ou supprimer du contenu. Avec l'évolution technologique (par exemple AJAX) les applications WEB 2.0 s'approchent de plus en plus des applications bureau classique pour devenir plus dynamiques et plus interactives. Une application bureau classique est une application accessible localement à partir de la machine sur laquelle elle est installée.

avec un excellent rendu de l'interface utilisateur. Le traitement des données a été amélioré et migré vers le navigateur web. Cela permet d'éviter le chargement systématique des pages à chaque action de l'utilisateur. Des micros synchronisations avec le serveur sont faites afin d'assurer la cohérence des données. Ces applications sont identifiées comme les applications web riches. D'autres types d'applications web ont vu le jour afin de s'adapter aux nouvelles technologies. Par exemple les applications web mobiles adaptées aux technologies mobiles comme les téléphones portables. Ces applications permettent d'avoir un rendu plus ergonomique et adapté aux contraintes des technologies mobiles (taille de l'écran, technologie tactile, etc.).

Afin d'améliorer l'exploitation et l'exploration des données présentes sur le web, les applications web sémantiques proposent de rendre toutes les données (transactions, documents, etc.) présentes sur le web accessible par la machine. Ce qui n'est pas le cas pour les applications web classiques où la majorité des données ne sont compréhensibles que par l'humain. Cela sert, par exemple, pour la recherche des données sur le web. Les résultats d'une recherche, dans ce cas, sont plus pertinents.

Le web collaboratif est un terme applicable sur toute application web permettant de collaborer autour de processus bien définis. Que le type de l'application web soit riche ou mobile, sémantique ou classique, ne fait qu'augmenter et améliorer les possibilités d'interaction. [2]

7. Conclusion

Dans ce chapitre, on a présenté quelques notions sur le système d'information et la collaboration ainsi qu'une étude plus ou moins détaillée sur les architectures de système d'information collaboratif

Le Système d'Information Collaboratif, support de l'interopérabilité des systèmes d'information hétérogènes, il doit s'appuyer sur des modèles de système d'information qui détaille les données, les processus de la collaboration , Et en fin sur Classification des applications web (expliquée exactement le web collaboratif)

CHAPITRE II

Conception d'un SIC Pour les Transferts Inter-facultés

CHAPITRE 2 : Conception d'un SIC Pour les Transfert Inter-faculté

1. Introduction

Dans ce chapitre on va voir quelques scénario de transfert étudiant inter-faculté ensuite choisir un scénario et puis l'analyse de notre système en utilisant le langage **UML**, nous allons détailler les trois étapes : tous d'abord, nous commencerons par définir le diagramme de cas d'utilisation, ensuite les cas d'utilisation vont être regrouper dans un diagramme de paquetage et puis construire le diagramme de classe qui décrit la structure statique de notre système, nous terminerons par représenter les diagrammes d'interaction.

2. Les scénarios de Transfert

Dans le cas de transfert inter-faculté il existe plusieurs façons de gère le transfert au niveau de l'université. on définit quelque scénario et puis nous Choisissons un scénario :

❖ scénarios 1 :

Scénario habituel de transfert étudiant Passe par les étapes suivantes :

- *Etudiant* : dépôt la demande de transfert au niveau de faculté réception .
- *Faculté destination* : contacter la faculté origine pour rapporter les renseignements de l'étudiant.
- *Faculté origine* : contacter les départements pour rapporter les renseignements de l'étudiant et renvoie les information sur l'étudiant.
- *Faculté destination* : Après l'arrivée des informations sur étudiant , la faculté réception contacte le département que l'étudiant veut rejoindre(demande avis)
- *Département destination* : envoie la décision a son faculté .

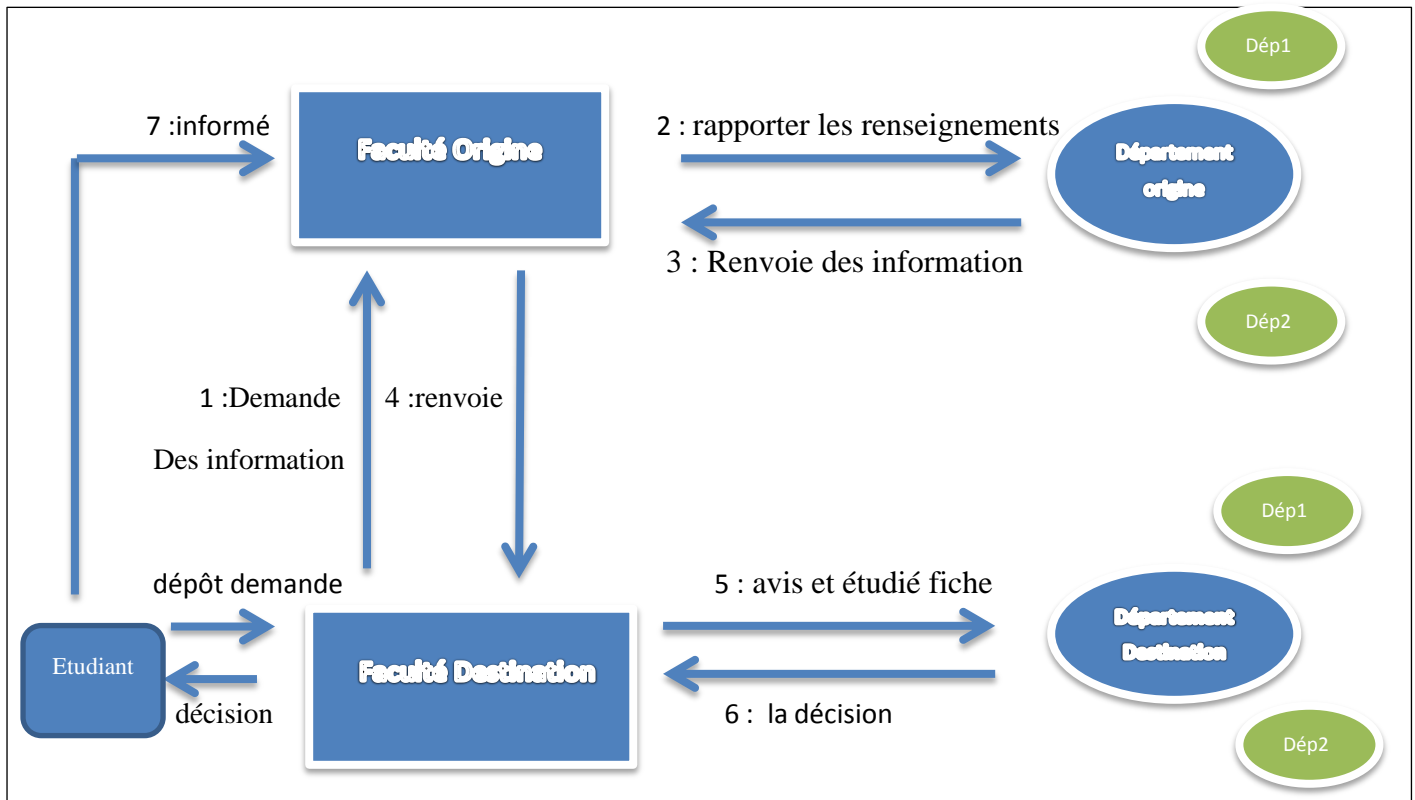


Figure 2.1 : Schéma décrit Scenario 1

Les Avantages :

- ✓ Posée la demande directement au niveau de faculté de réception .

Les inconvénients :

- ✓ La redondance au niveau de chaque faculté.
- ✓ Perdre beaucoup de temps.

❖ scénarios 2 :

on va définir d'autre Scénario de transfert qui Passe par les étapes suivantes :

- *Etudiant* : dépôt la demande de transfert au niveau de faculté .
- *Faculté origine* : contact la faculté destinée pour transférer .
- *Faculté destinée* : contact département pour savoir l'avis d'un transfert étudiant à son département.
- *Département destinée* : demande des informations sur l'étudiant
- *Faculté destinée* : contact la faculté origine pour rapporter les informations concernant l'étudiant.
- *Département origine* : Renvoie fiche d'étudiant vers Faculté destinée .

- *Faculté destinée* : voir la fiche de transfert .
- *Département destinée* :étudiée fiche de étudiant et puis renvoie la décision

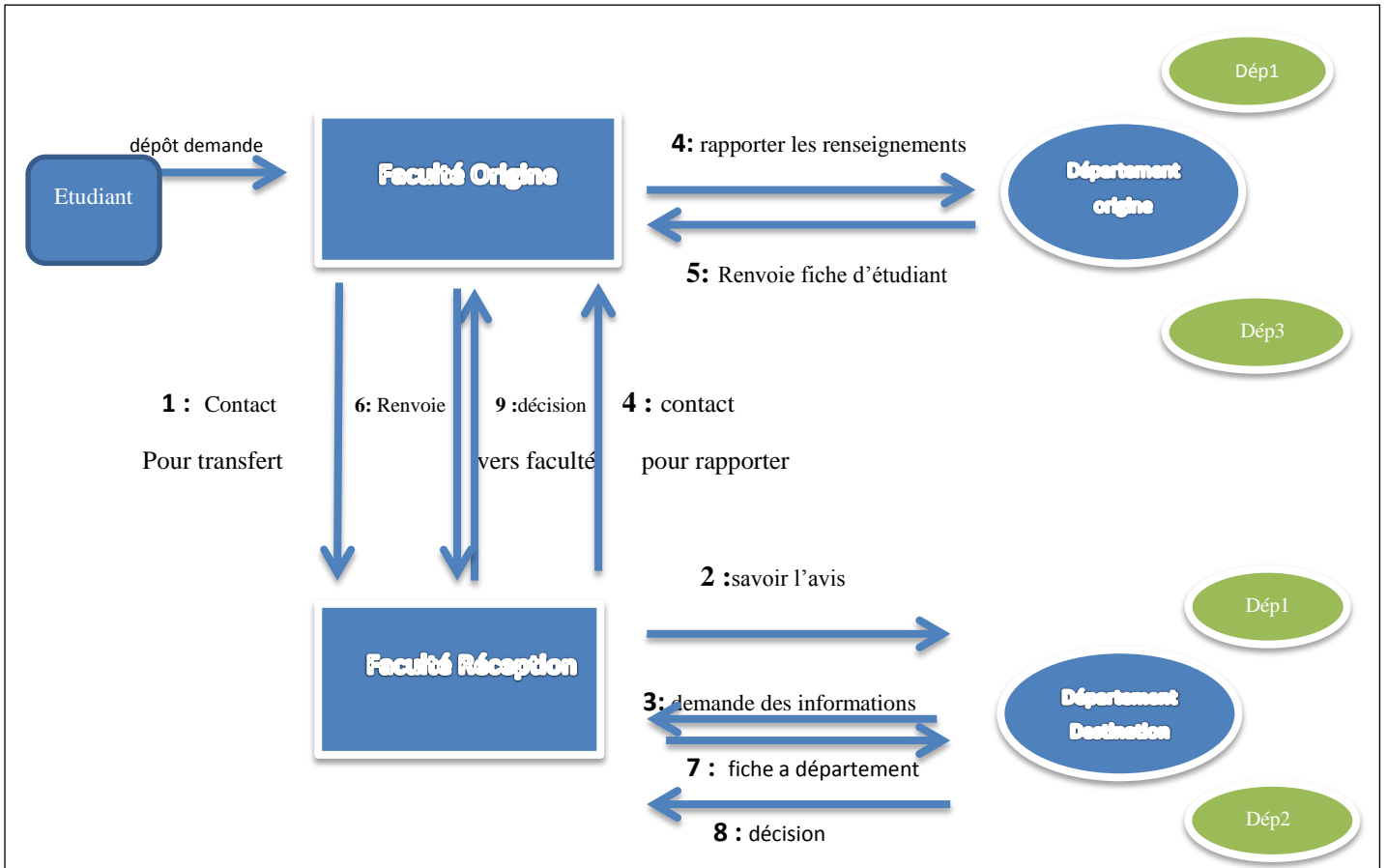


Figure 2.2 : Schéma décrit Scenario 2

Les Avantages :

- ✓ Posée la demande directement au niveau de faculté origine .

Les inconvénients :

- ✓ La redondance au niveau de chaque faculté.
- ✓ Prendre du temps .
- ✓ La complexité du système et Façon de collaboration.

❖ scénarios 3 :

Le dernier scénario est plus claire que le précédent , il Passe par les phases suivantes :

- *Etudiant* : dépôt la demande de transfert au niveau de faculté .
- *Faculté origine* : Ajoute la demande pour consulter par le département spécifique.
- *Département origine* : création fiche de transfert et la faculté origine l'envoie a faculté destinée.
- *Faculté et Département destinée* : consulter la fiche de transfert

- *Département destinée* : étudiée la fiche de transfert et puis envoie la décision finale de transfert.

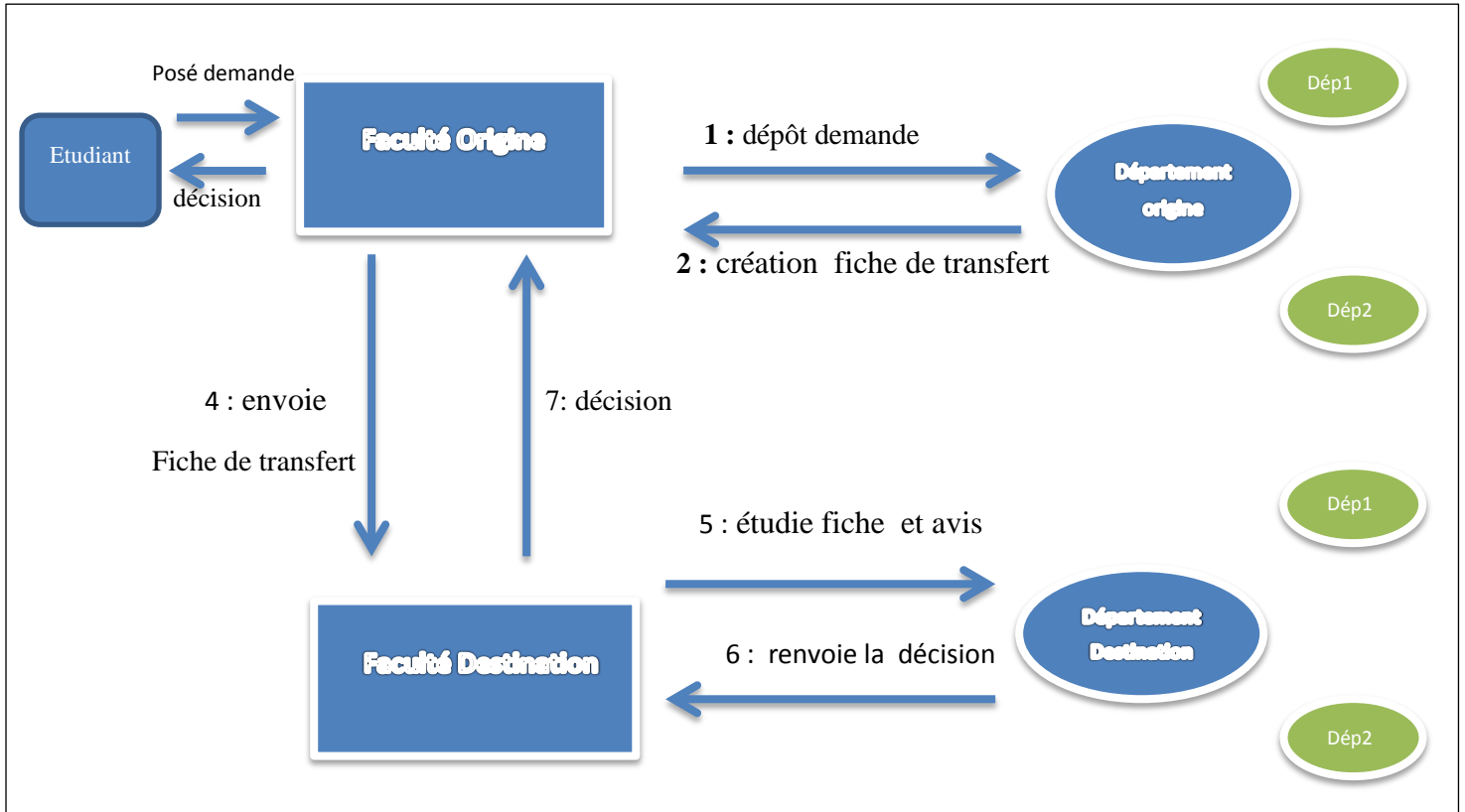


Figure 2.3 : Schéma décrit Scenario 3

Les Avantages :

- ✓ Etudiant Poser la demande directement au niveau de faculté origine
- ✓ moins de répétition de saisir .
- ✓ système centralisé au sein de chaque faculté.
- ✓ La collaboration claire dans cette scénario .

Parmi les scénario précédant nous avons choisi ce dernier selon les efforts et les facilités pour comprendre la méthode de transfert , nous avons Fait La conception de ce scénario .

3. UML

3.1. Présentation du langage UML

« UML (en anglais Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

UML est un langage de modélisation objet, en tant que tel, il facilite l'expression et la communication de modèles en fournissant un ensemble de symboles (la notation) et de règles qui régissent l'assemblage de ces symboles (la syntaxe et la sémantique). [4]

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système :

- Vue des cas d'utilisation : c'est la description du modèle vu par les acteurs du système. Elle correspond aux besoins attendus par chaque acteur (c'est le QUOI et le QUI).
- Vue logique : c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs (c'est le COMMENT).
- Vue d'implémentation : cette vue définit les dépendances entre les modules.
- Vue des processus : c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, etc.
- Vue de déploiement : cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est le OÙ). [12]

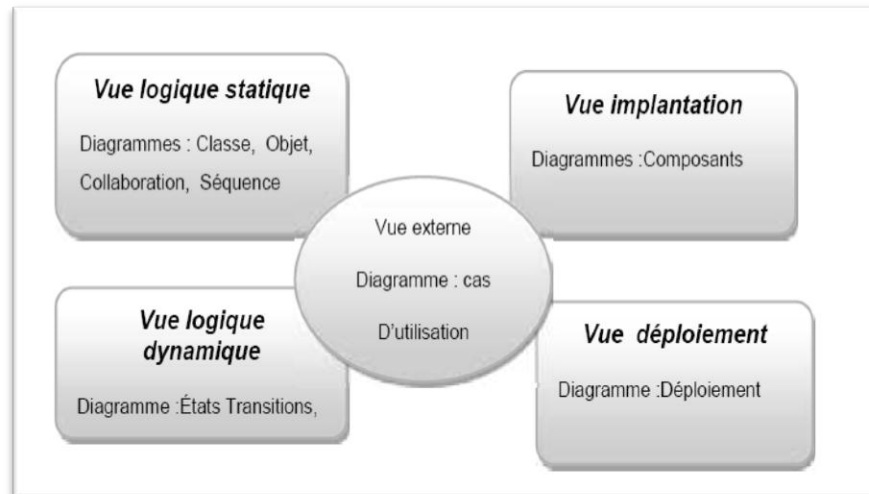


Figure 2.4 :les vues d'UML

Les diagrammes sont des éléments graphiques, ceux-ci décrivent le contenu des vues, qui sont des notations abstraites. Les diagrammes peuvent faire partie de plusieurs vues.

4. Processus de développement suivi

On a utilisé pour la conception de notre système comme les étapes suivantes :

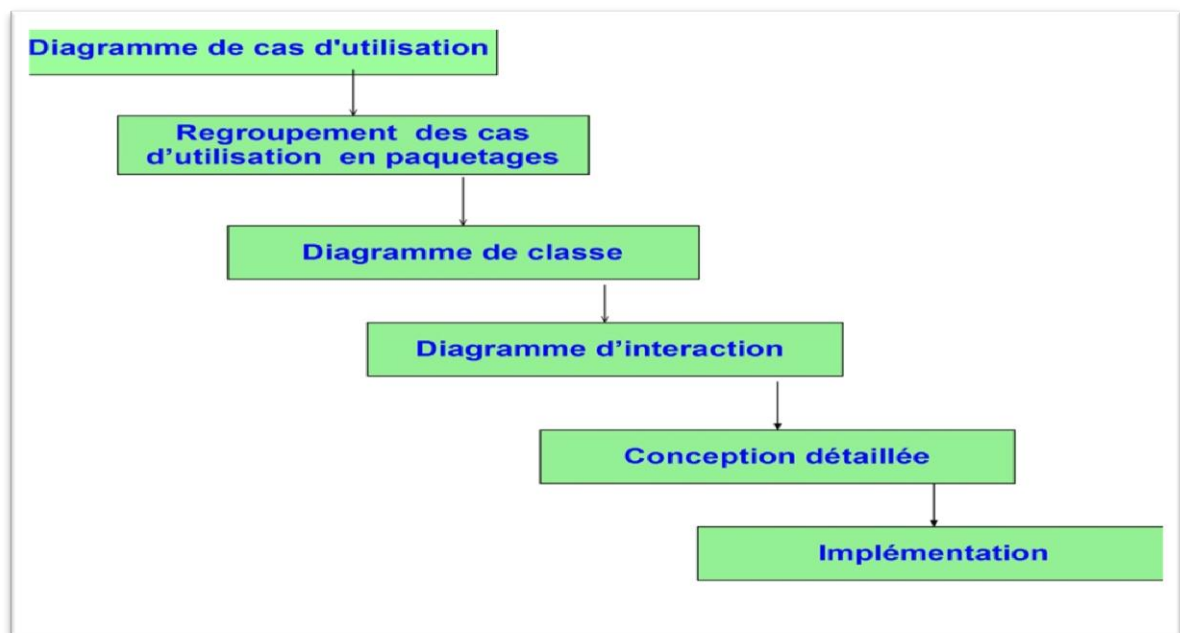


Figure 2.5 :processus de développement

5. Diagrammes des cas d'utilisation

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique. Il ne faut pas négliger cette première étape pour produire un logiciel conforme aux attentes des utilisateurs. Pour élaborer les cas d'utilisation, il faut se fonder sur des entretiens avec les utilisateurs.

- Montre les interactions fonctionnelles entre les acteurs et le système à l'étude :

- **Acteur** : rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation.

- **Cas d'utilisation (use case)** : ensemble de séquences d'actions réalisées par le système produisant un résultat observable intéressant pour un acteur particulier.

Collection de scénarios reliés par un objectif utilisateur commun.

- **Association** : utilisée dans ce type de diagramme pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement « participe à ».

- **Inclusion** : le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire, à un endroit spécifié dans ses enchaînements.

- **Extension** : le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle, à un endroit spécifié indirectement dans celui qui procède à l'extension

- **Généralisation** : les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des relations spécifiques supplémentaires avec d'autres acteurs ou cas d'utilisation. [10]

Les relations entre cas d'utilisation :

- Relation d'inclusion
- Relation d'extension
- Relation de généralisation.

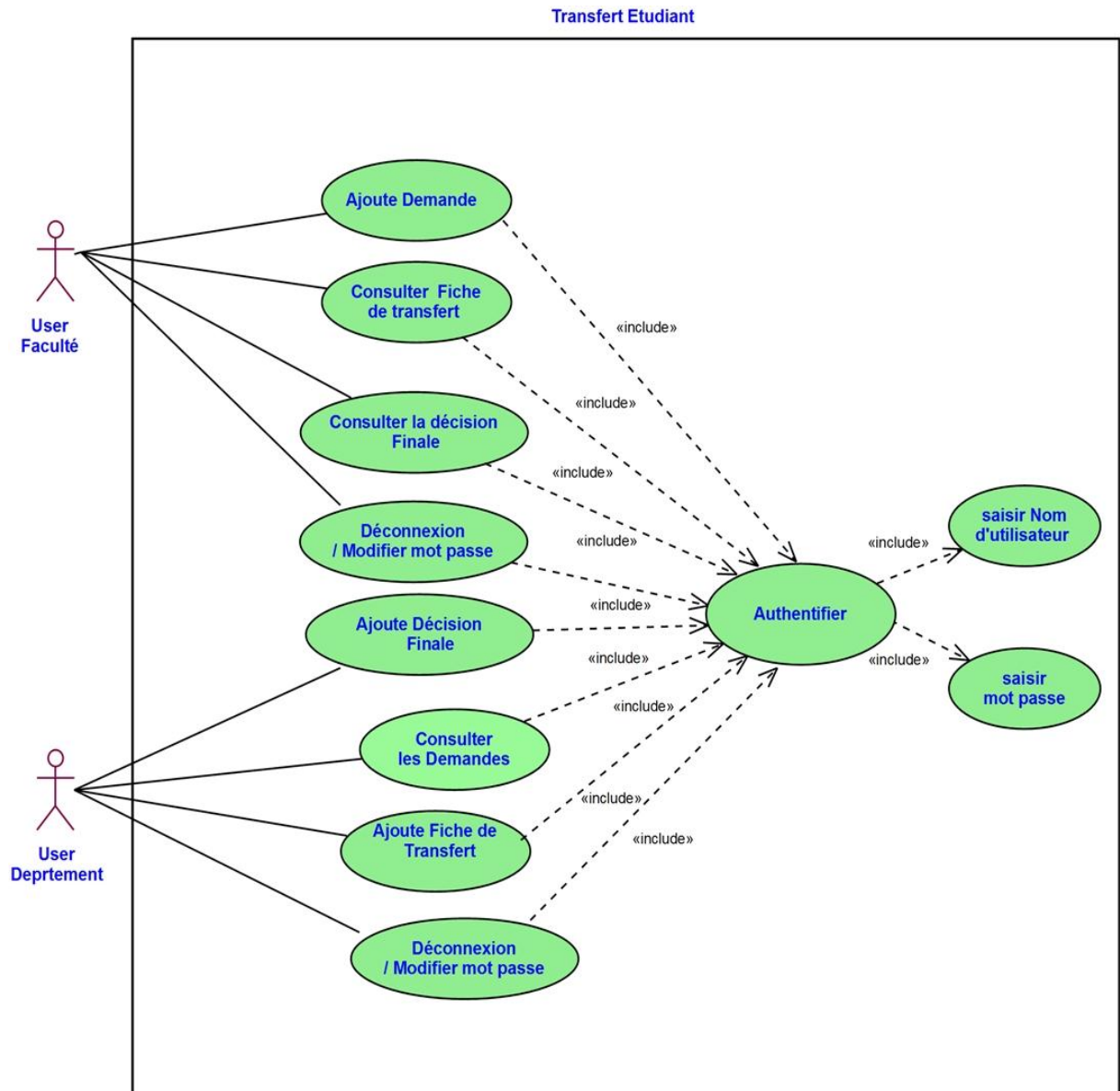


Figure 2.6 : diagramme de cas d'utilisation <User faculté + User département>

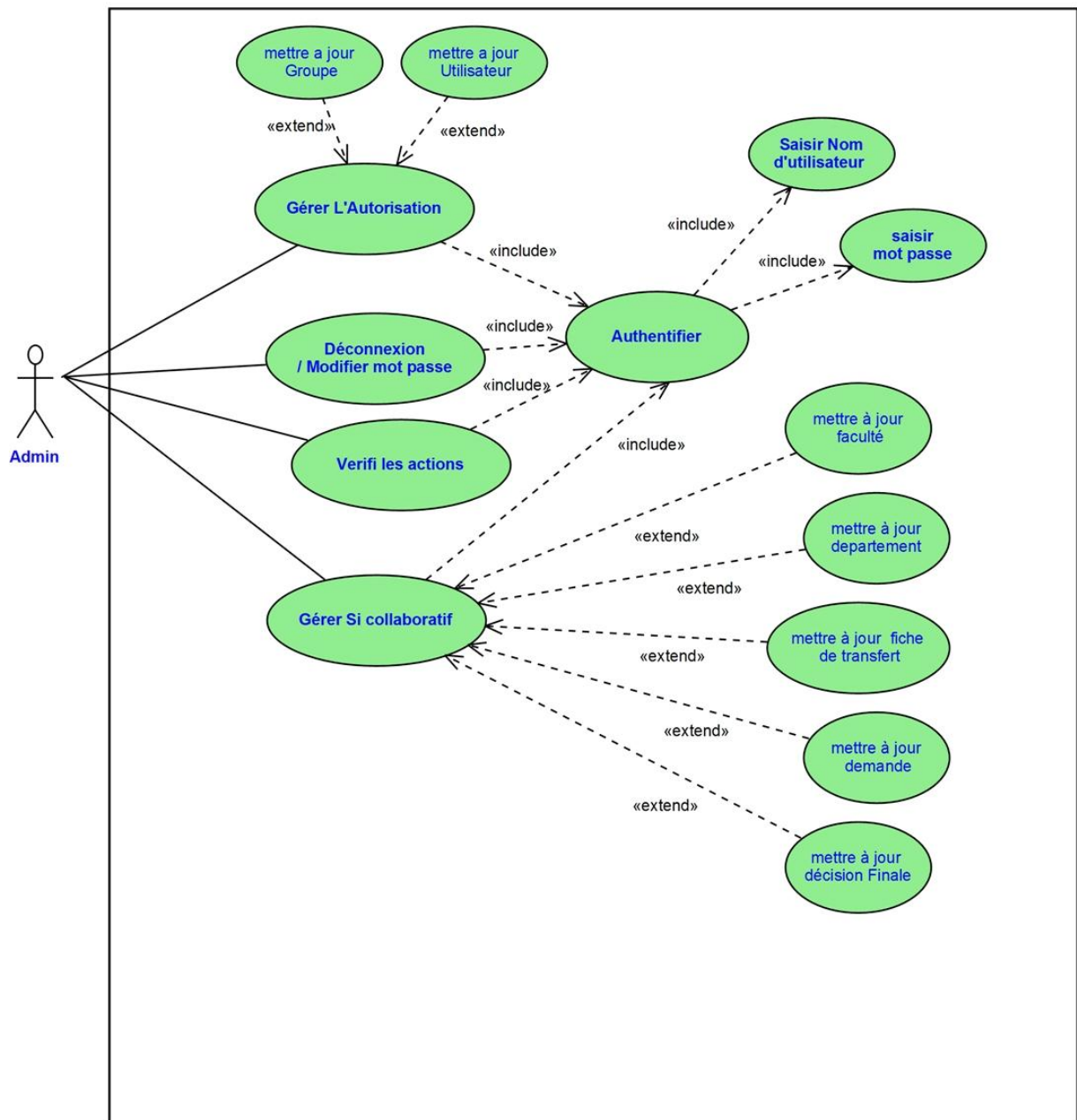


Figure 2.7 :diagramme de cas d'utilisation <Admin>

5.1. Description textuelle des cas d'utilisation

Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Bien que de nombreux diagrammes d'UML permettent de décrire un cas, il est recommandé de rédiger une description textuelle car c'est une forme souple qui convient dans bien des situations.

❖ **Authentification :**

○ **Identifier le cas :**

Nom : authentification.

Objectif : Pour faire les cas d'utilisation de utilisateur administratif (User faculté, User département) dans le Statut officiel

Acteur principaux : User faculté , User département, Admin

Acteur secondaire : //

○ **Description du fonctionnement du cas :**

❖ Les pré-conditions :

Demande d'authentification.

❖ Les scénarios :

✓ Scénarios nominal :

L'utilisateur veut accéder à l'application.

L'utilisateur saisir le Nom d'utilisateur et mot passe pour authentifiée

Le système autorise l'accès ou envoie un message d'erreur.

✓ Scénarios alternatifs :

Le utilisateur elle doit avoir une seule méthode pour l'authentification.

✓ Scénarios d'exception:

Le utilisateur peut être en cas d'erreur si l'utilisateur saisit

les informations fausses ou n'est pas inscrit.

❖ Les post-conditions :

Utilisateur est authentifié.

❖ **Ajoute Demande :**

○ **Identifier le cas :**

Nom : rapporter les renseignements d'étudiant à Son département.

Objectif : Apportez des informations importantes sur un étudiant qui veut Transfert à une autre faculté.

Acteur principaux : User faculté.

Acteur secondaire : //.

○ **Description du fonctionnement du cas :**

❖ Les pré-conditions :

Demande les renseignements importantes (moyenne ,discipline. ...)

❖ Les scénarios :

✓ Scénarios nominal :

User faculté authentifier.

Le système affiche formulaire de demande à remplir.

Ajoute la Demande des renseignement .

Le système vérifie si tous les champs sont remplis.

✓ Scénarios alternatifs :

Quelque champ du formulaire est vide.

Envoie système doit remplir tous les champs.

✓ Scénarios d'exception:

Agent faculté laisse des champs obligatoire est vides....

❖ Les post-conditions :

L'objet demande a été ajouté avec succès.

❖ **Consulter Demande :**

○ **Identifier le cas :**

Nom : Consulter la demande .

Objectif : pour crée fiche de transfert de étudiant.

Acteur principaux : User département.

Acteur secondaire : //.

○ **Description du fonctionnement du cas :**

❖ Les pré-conditions :

Voir les renseignements importantes de la Demande(motif de transfert..) .

❖ Les scénarios :

✓ Scénarios nominal :

User département authentifier.

Accédé a les espaces des demandes.

Filtrer et Voir les demandes qui ajoutée récemment.

Consultée la demande et puis remplir le champs réservée.

❖ Les post-conditions :

L'objet demande a été ajouté avec succès.

❖ *Ajoute Fiche de Transfert :*

○ **Identifier le cas :**

Nom : Ajoute fiche de transfert .

Objectif : Ajoute Fiche de transfert pour faire la transfert vers la faculté destinée.

Acteur principaux : User département.

Acteur secondaire : //

○ **Description du fonctionnement du cas :**

❖ Les pré-conditions :

Fiche de transfert ajoute pour consulter par l'autre faculté .

❖ Les scénarios :

✓ Scénarios nominal :

User département authentifier.

Le système affiche formulaire de renseignement à remplir

Saisir les renseignement .

Le système vérifie si les champs obligatoire sont remplis.

✓ Scénarios alternatifs :

Quelque champ obligatoire du formulaire est vide.

Envoie système doit remplir tous les champs.

✓ Scénarios d'exception:

User faculté laisse des champs ou n'est pas remplir vides....

❖ Les post-conditions :

La consultation est réussite.

❖ *Consulter fiche de transfert :*

○ **Identifier le cas :**

Nom : Consulter la fiche de transfert.

Objectif : pour renvoie la décision par la faculté destinée.

Acteur principaux : User département, .

Acteur secondaire : User Faculté

○ **Description du fonctionnement du cas :**

❖ Les pré-conditions :

Voir les renseignements importantes de la fiche (motif de transfert , moyenne ,discipline.) .

❖ Les scénarios :

✓ Scénarios nominal :

User authentifier.

Accédé a les espaces des fiches de transfert.

Filtrer et Voir les fiches de transfert qui ajoutée récemment.

Consultée fiche de transfert et puis remplir le champs réservée.

❖ Les post-conditions :

La consultation est succès.

❖ *Décision Finale :*

○ **Identifier le cas :**

Nom : insert la décision finale de transfert

Objectif : Envoyer la décision a l'étudiant et son département pour désinscrire

Acteur principaux : User département

Acteur secondaire : //

○ **Description du fonctionnement du cas :**

❖ Les pré-conditions :

Ajoute la décision de transfert pour consulter par faculté d'origine

Les scénarios :

✓ Scénarios nominal :

User département authentifier.

Le système affiche formulaire de renseignement à remplir

Saisir la décision finale.

Le système vérifie si les champs obligatoires sont remplis.

✓ Scénarios alternatifs :

Quelque champ obligatoire du formulaire est vide.

Envoi système doit remplir tous les champs obligatoires.

✓ Scénarios d'exception:

User département laisse des champs obligatoires ou n'est pas remplir vides....

❖ Les post-conditions :

La décision est réussie.

❖ **Consulter décision :**

○ **Identifier le cas :**

Nom : Consulter la décision finale.

Objectif : étudiant accorde ou no.

Acteur principaux : User Faculté .

Acteur secondaire : User département.

○ **Description du fonctionnement du cas :**

❖ Les pré-conditions :

Voir la décision de transfert (accorde , motif décision).

❖ Les scénarios :

✓ Scénarios nominal :

User authentifier.

Accéder à les espaces de décision.

Filtrer et Voir les décisions qui ajoutées récemment.

Consultée la décision.

❖ Les post-conditions :

La décision est succès.

6. Regroupement des cas d'utilisation en paquetages

UML permet de regrouper des cas d'utilisation dans une entité appelée « paquetage ». Le regroupement peut se faire par acteur ou par domaine fonctionnel. Un diagramme de cas d'utilisation peut contenir plusieurs paquetages et des paquetages peuvent être inclus dans d'autres paquetages. [5]

6.1. Diagramme de Paquetage

Un paquetage permet d'organiser des éléments de modélisation en groupe. Un paquetage peut contenir des classes, des cas d'utilisations, des interfaces, etc.

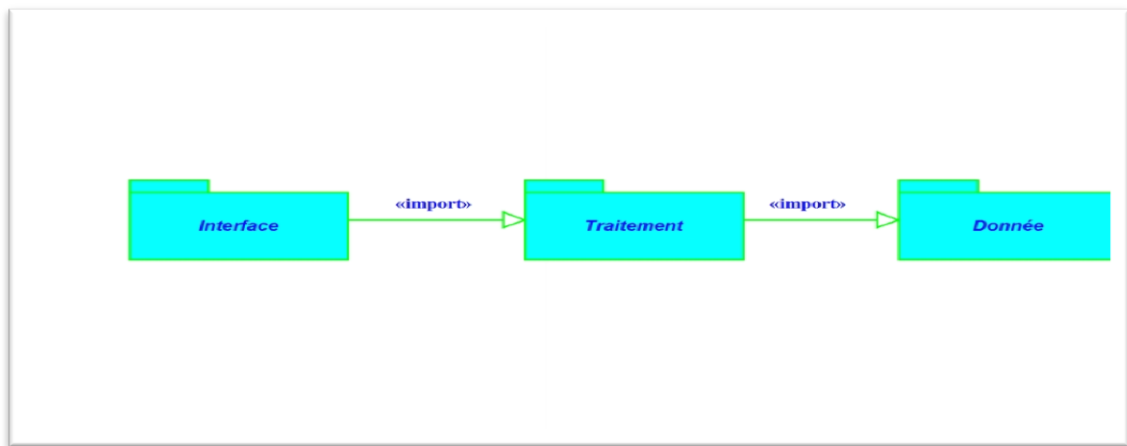


Figure 2.8 :diagramme de paquetage

- **Paquetage d'interface** :Cet package contient les pages d'accueil et les formulaires De demande et d'autre pages ..
- **Paquetage de traitement** :Cet package contient les méthodes des besoins d'utilisateur.
- **Paquetage de donnée** :Cet package contient les données particulières des méthodes De ce système.

7. Diagramme de classes

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets.

Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensembles par un champ sémantique.

Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Elles sont finalement instanciées pour créer des objets (une classe est un moule à objet : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés). [7]

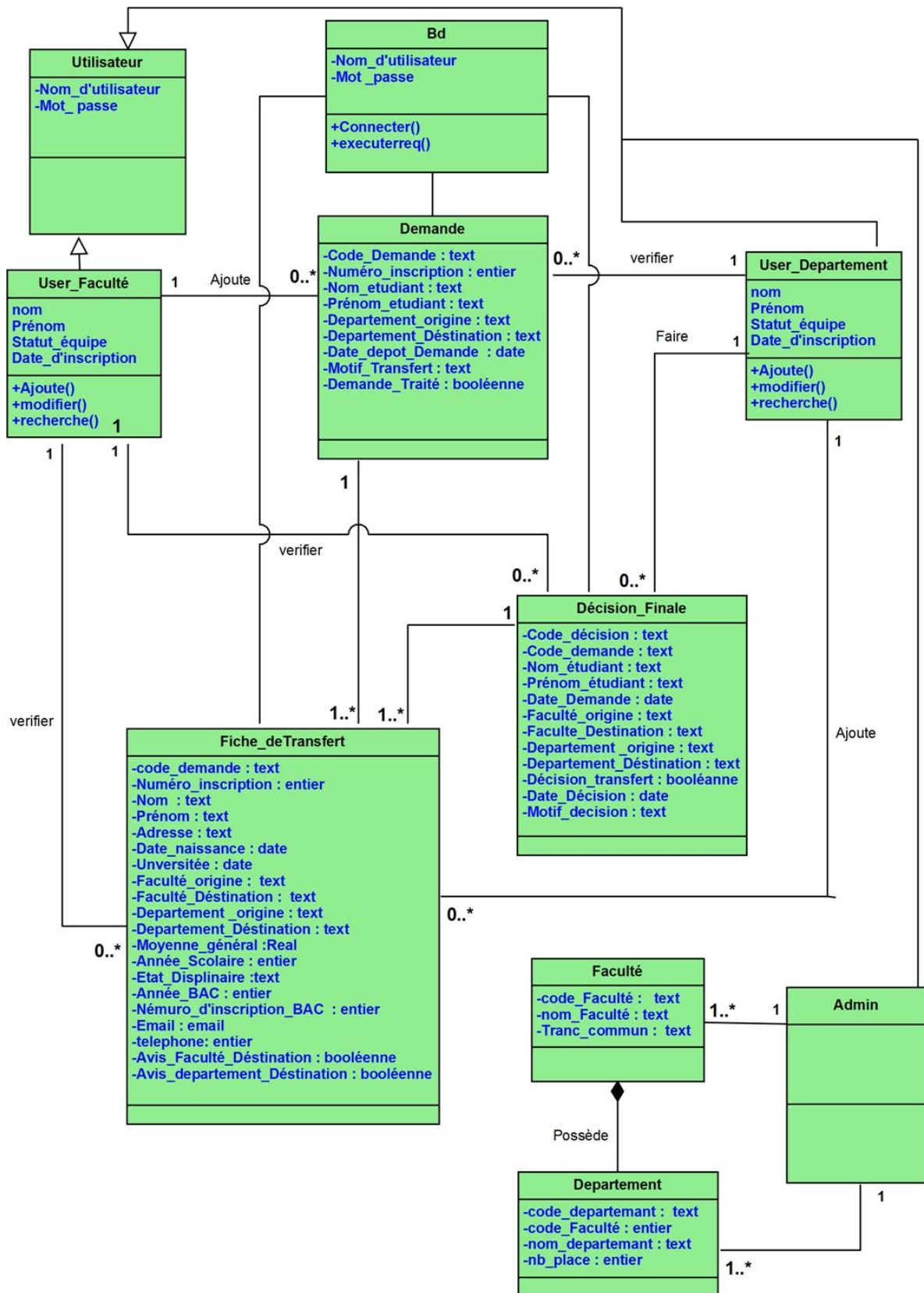


Figure 2.9 : diagramme de classe <paquetage donnée>

8. Les Diagramme d'interaction

Les diagrammes d'interaction permettent d'établir un lien entre les diagrammes de cas d'utilisation et les diagrammes de classes : ils montrent comment des objets (des instances de classes) communiquent pour réaliser une certaine fonctionnalité. Ils apportent un aspect dynamique à la modélisation du système . On trouve dans cette catégorie les diagrammes de séquence, du global d'interaction, de communication et du temps. [5]

8.1. Diagramme de collaboration

Un diagramme de collaboration montre une interaction organisée autour d'un ensemble d'objets et de leurs liens. En revanche, un diagramme de collaboration ne montre pas le temps dans une dimension séparée; ainsi la séquence des messages et les fils concurrents doivent être déterminés en utilisant les numéros de séquence.

C'est une autre représentation des scénarios des cas d'utilisation qui met plus l'accent sur les objets et les messages échangés.

❖ Diagramme de collaboration d'authentification :

Ce diagramme décrit les messages échangés entre les différents objets pour montrer le fonctionnement de l'opération d'authentification : l'utilisateur saisit le mot de passe puis le système vérifie sa validité, ensuite c'est le système qui retourne la page d'accueil de à l'application l'utilisateur.



Figure 2.10 :diagramme de collaboration <Authentification>

❖ **Diagramme de collaboration (demande transfert):**

Ce diagramme décrit les messages échangés entre les différents objets pour montrer le fonctionnement de l'opération de demande transfert : user de faculté saisit la demande puis le système vérifie sa validité, ensuite c'est le système qui enregistre la demande dans la base de donnée et agent de département voir la demande .

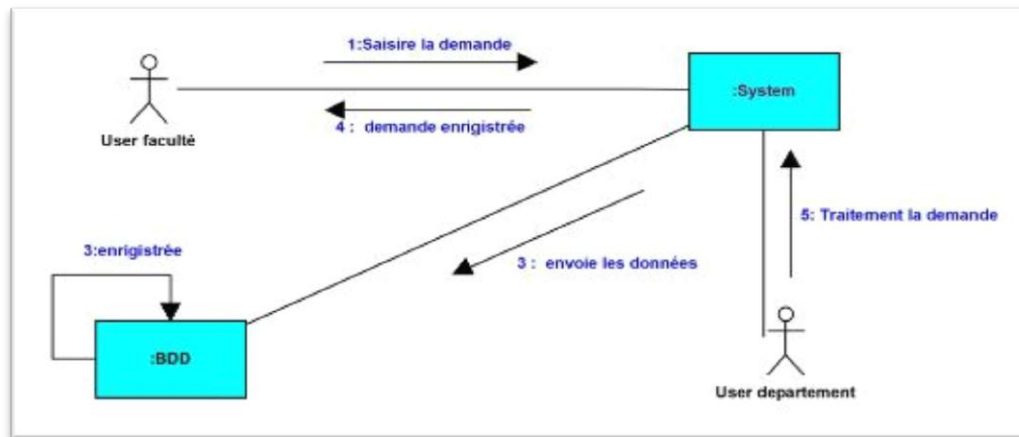


Figure2.11 :diagramme de collaboration <Demande de transfert>

○ **Diagramme de collaboration (décision transfert):**

Ce diagramme décrit les messages échangés entre les différents objets pour montrer le fonctionnement de l'opération de décision transfert : user de département saisit la décision et puis le système vérifie sa validité, ensuite c'est le système qui enregistre la décision dans la base de donnée et user de faculté voir la décision .

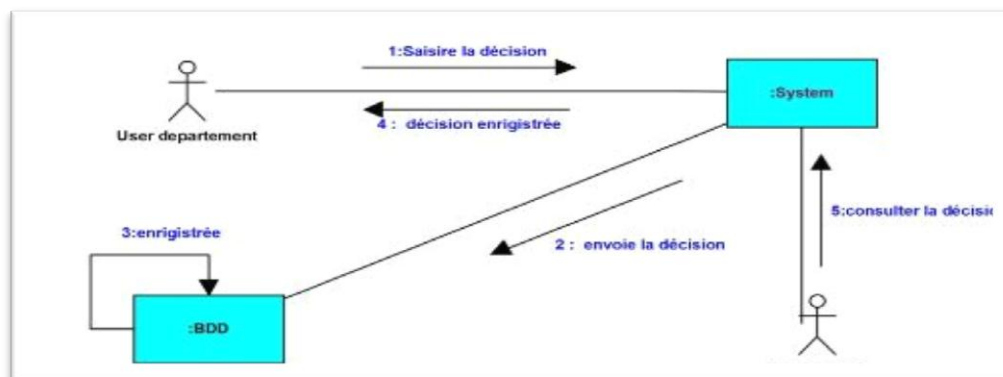


Figure 2.12 :diagramme de collaboration <Décision de transfert>

❖ **Diagramme de collaboration :**

Ce diagramme décrit les messages échangés entre les différents objets pour montrer le fonctionnement de système.

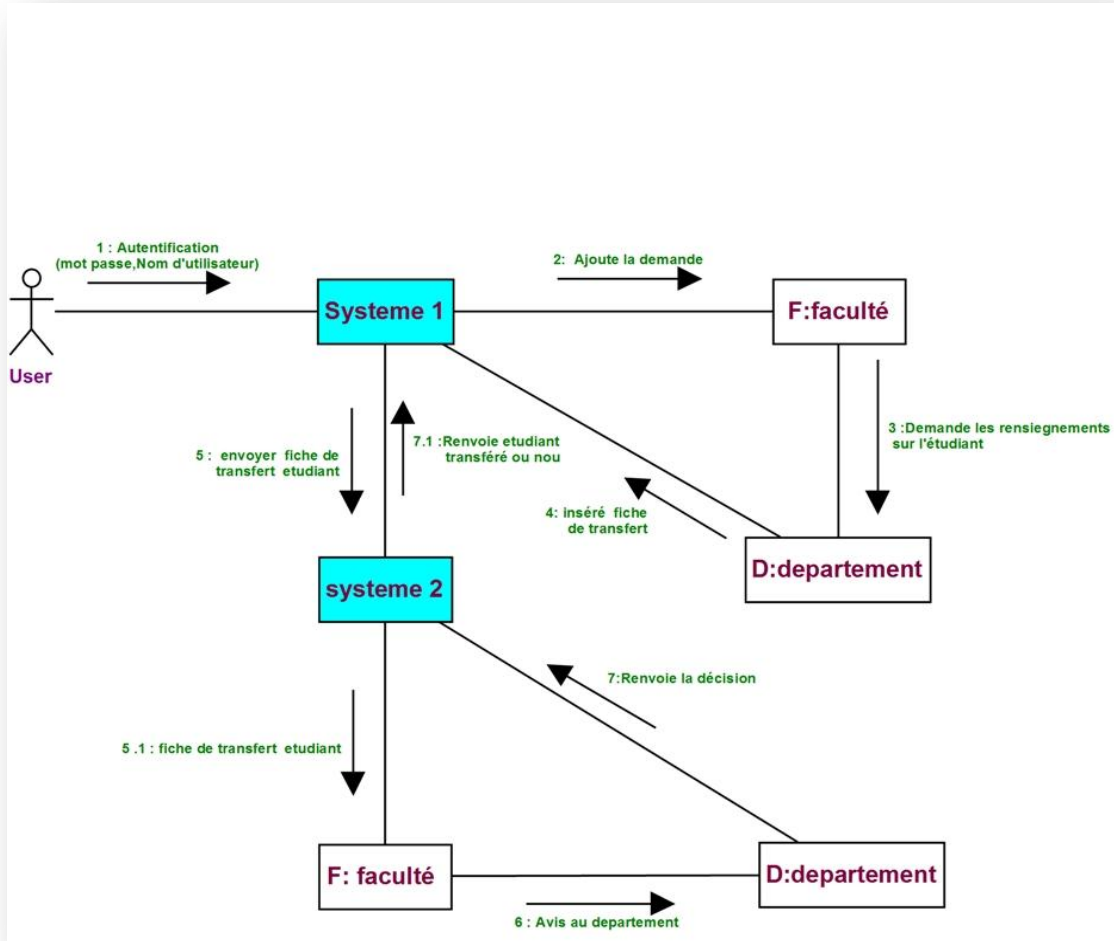


Figure 2.13 :diagramme de collaboration

8.2. Diagramme de séquence

Un diagramme de séquence montre des interactions entre un nombre d'objets en coopération afin d'accomplir une fonction précise. Les principales informations contenues dans ce diagramme sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique. Ainsi, contrairement au diagramme de communication, le temps y est représenté explicitement par une dimension (la dimension verticale).

Les diagrammes des séquences de cet système:

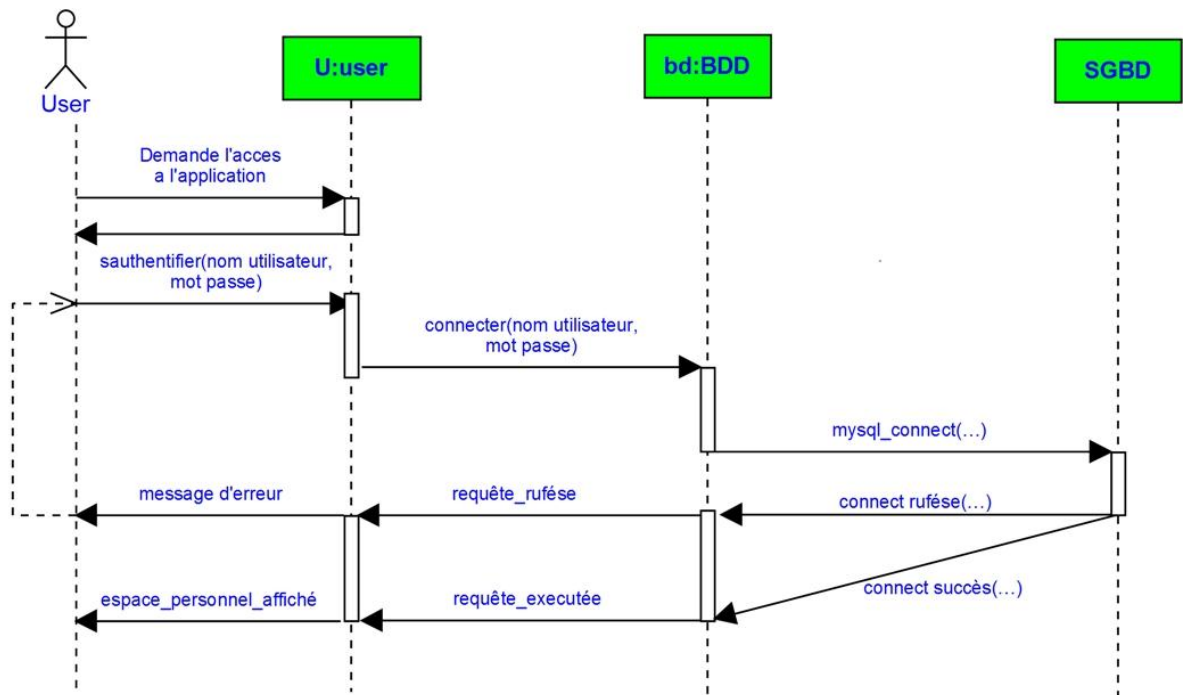


Figure 2.11: diagramme de séquence <Authentication>

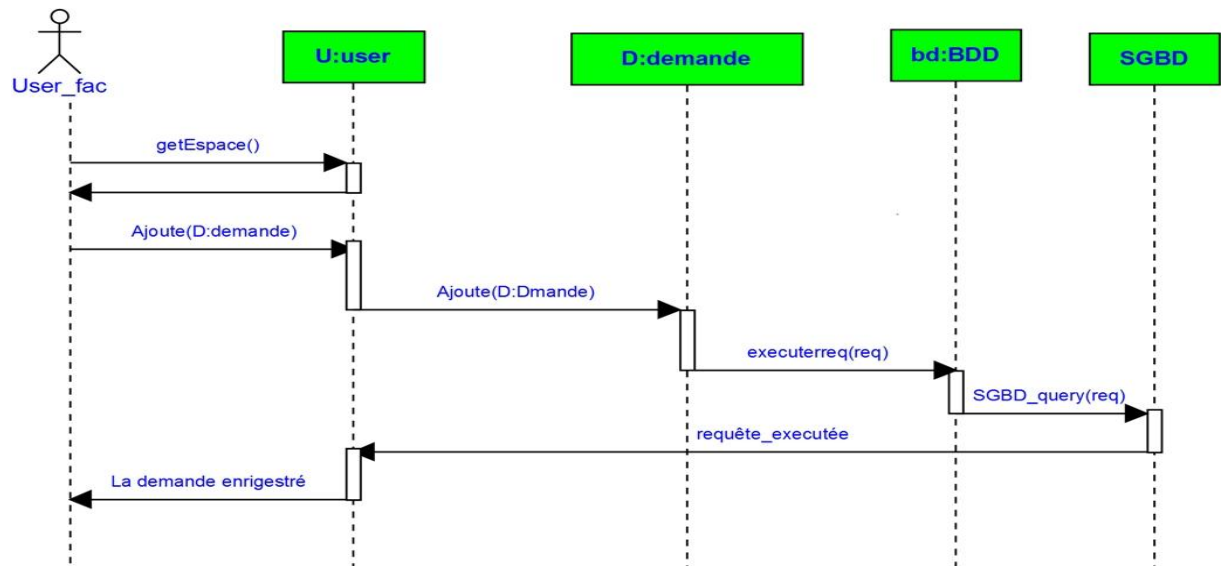


Figure 2.12 :diagramme de séquence <Demande >

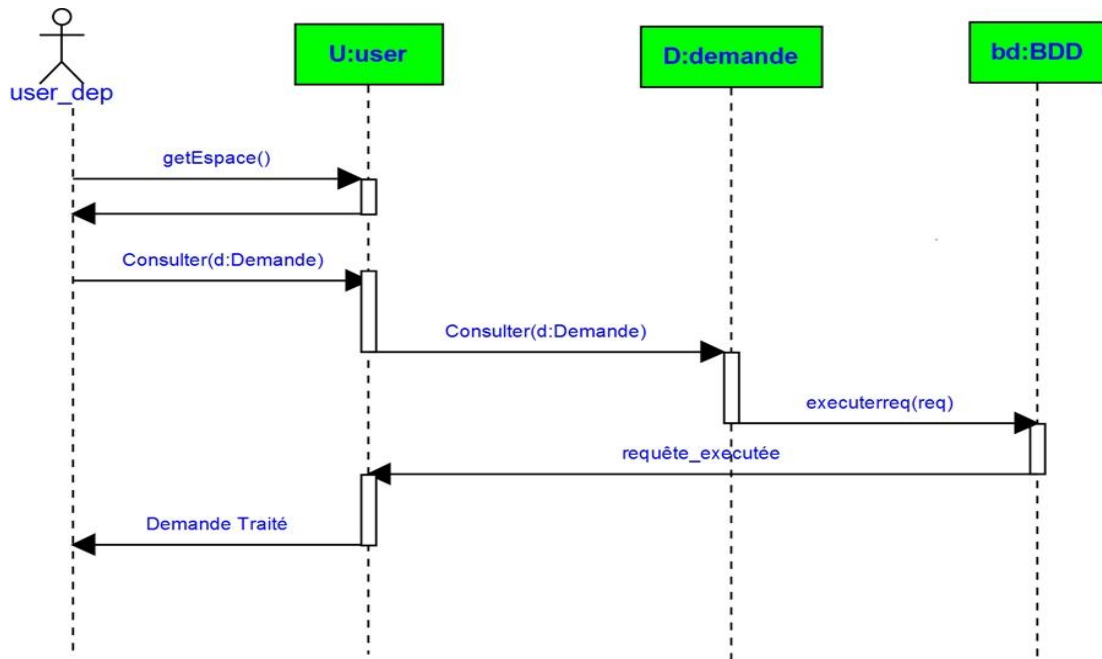


Figure 2.12 :diagramme de séquence <Consulter Demande >

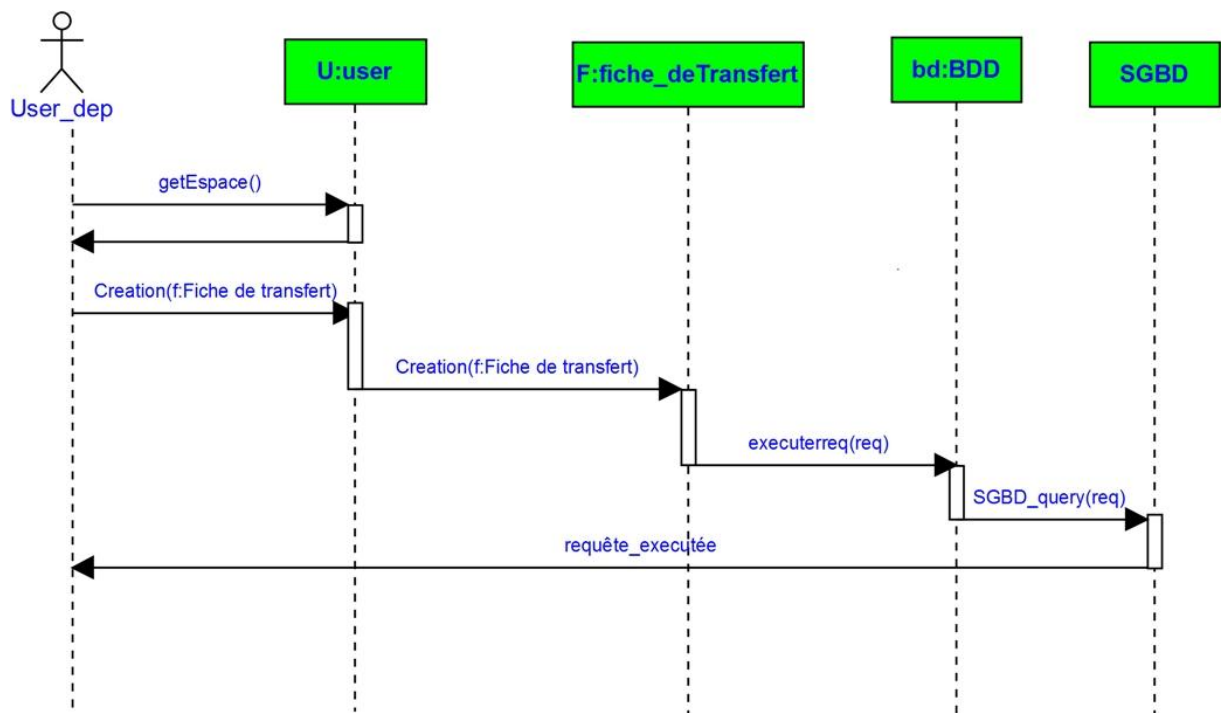


Figure 2.13 :diagramme de séquence < Création Fiche de Transfert >

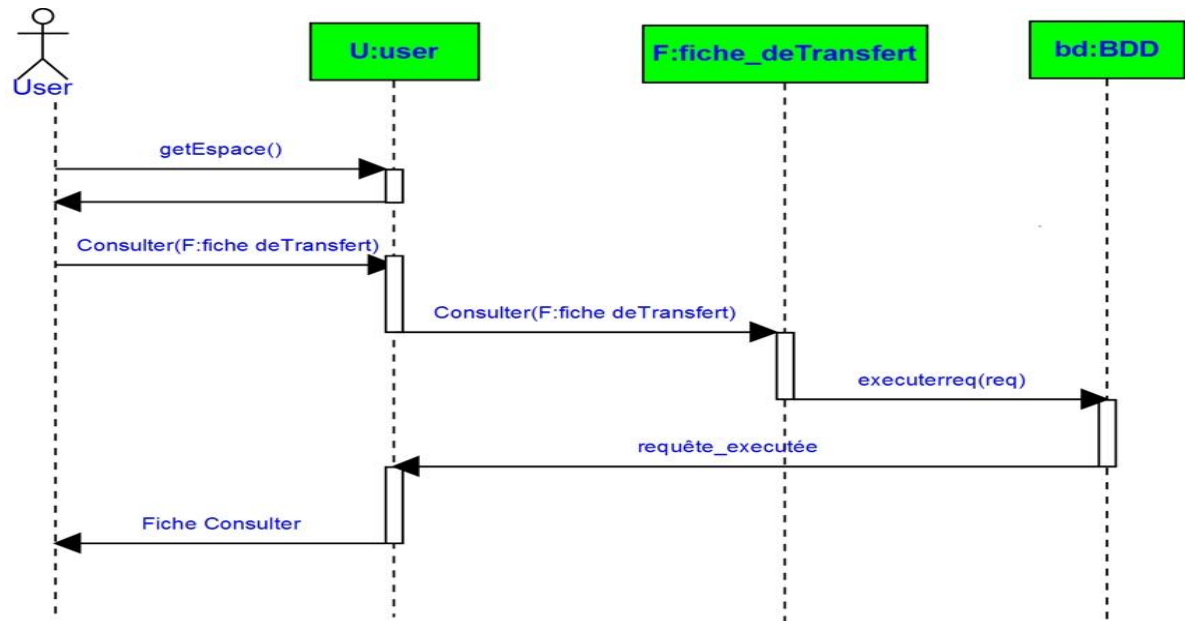


Figure 2.14 :diagramme de séquence < Consulter fiche de transfert>

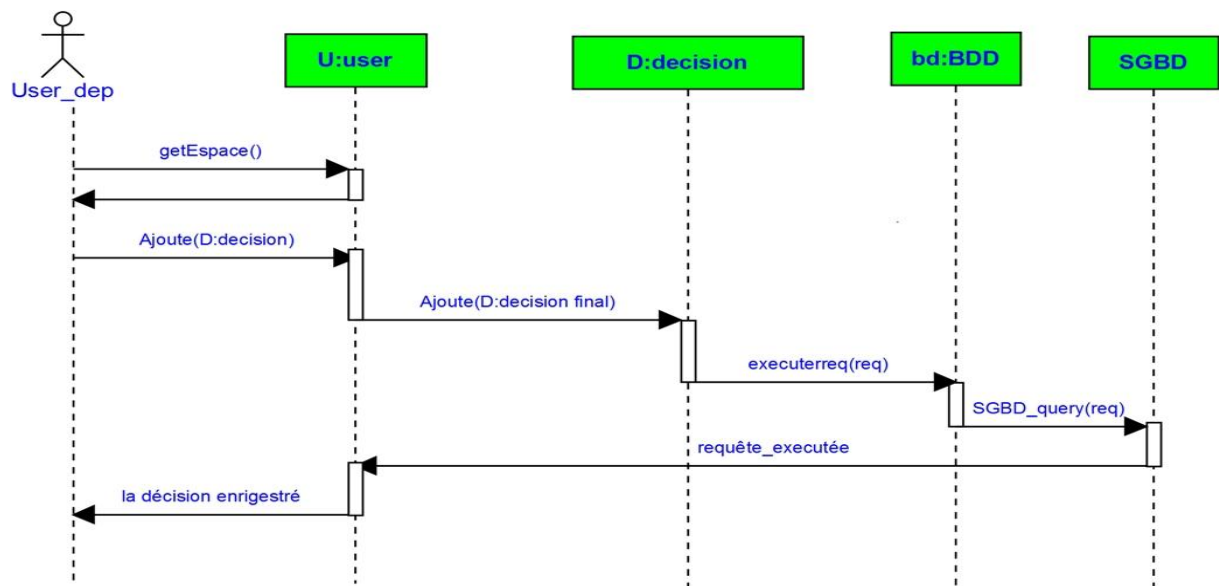


Figure 2.15 :diagramme de séquence < décision finale >

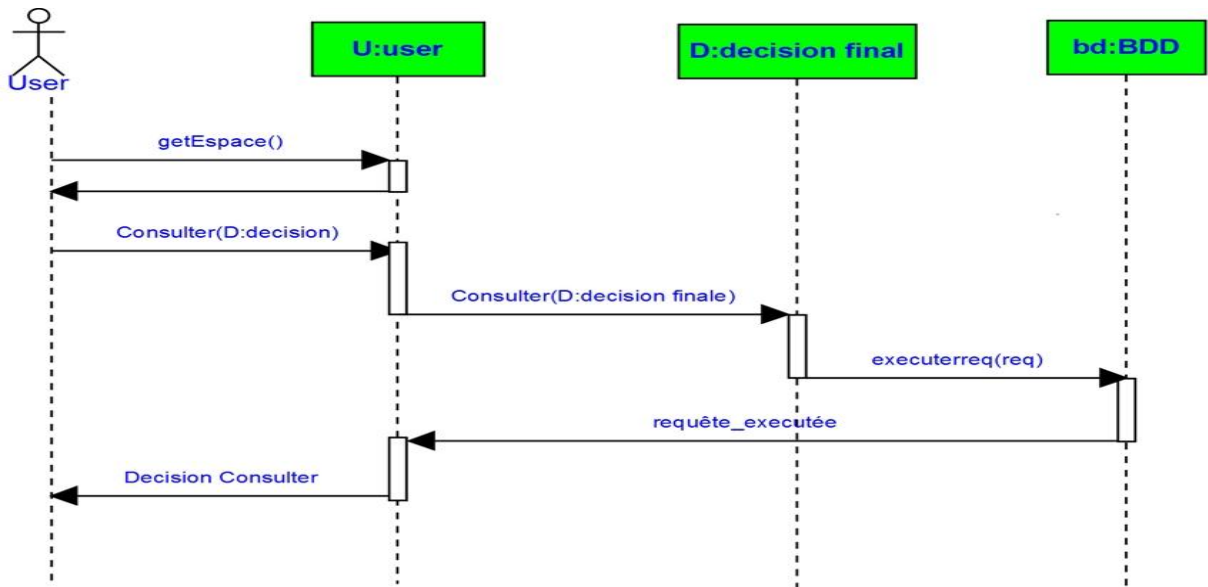


Figure 2.15 :diagramme de séquence <Consulter la décision finale >

9. Conception du base de données

9.1. Introduction sur la base de données

Une base de données (son abréviation est BD, en anglais DB, data base) est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents.

Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de base. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données. [6]

9.2. Importance de clé relationnelle

Pour mettre une base de donnée compatible avec le système il est nécessaire d'identifier le diagramme de classe ces classe sont un attribue comme une clé primaire et les classes qui ont besoin d'une clé primaire sont obliger de définir un **ID** comme une clé primaire pour identifier et concevoir la relation entre les classes.

La clé étrangère de classe est une clé primaire d'une autre classe on sachant qu'il existe une relation entre ces deux classes.

C'est pour cela qu'on utilise le schéma relationnel .

9.3. schéma Relationnelle

- ❖ Utilisateur (**id** , nom_d'utilisateur, mot_passe)
- ❖ User_dep(**id**, nom_d'utilisateur, mot_passe nom, Prénom, Statut_équipe, Date_d'inscription)
- ❖ User_fac(**id**, nom_d'utilisateur, mot_passe nom, Prénom, Statut_équipe, Date_d'inscription)
- ❖ Demande(**id** , Code_Demande , Numéro_insc ,Nom_Etudiant , Prénom_Etudiant ,**#Département_origine** ,**#Département_Destination** , Date_dépôt_Demande ,Motif_Transfert ,Demande_Traité)
- ❖ Fiche de transfert(**id** , Code_demande , Numéro_inscription , Nom ,Prénom ,Adresse ,Date_naissance ,Université , **#Faculté_origine** , **#Faculté_Destination** , **#Département_origine**,**#Département_Destination** ,Moyenne_général , Année_Scolaire, Etat_Displinaire ,Année_BAC, Numéro_d'inscription_BAC ,Email ,Avis_Faculté_Destination , Avis_département_Destination)
- ❖ Décision finale (**id** , Code_décision , Code_demande ,**#Numéro_inscription**, Nom_Etudiant ,Prénom_Etudiant ,Date_Demande , **#Faculté_Destination**, **#Faculté_Destination** ,**#Département_Destination** , Décision_Transfert, Date_Décision ,Motif_décision)
- ❖ Faculté (**id** , code_Faculté, nom_Faculté , Tronc_commun)
- ❖ Département (**id**,code_département, **#code_Faculté** , nom_département , Nb_place)

9.4. Les Tables des bases de données

Dans ce qui suit, nous étudions en détail les différentes tables de la base de données de notre application.

➤ Table utilisateur :

Cette table Utilisée pour sauvegarder les information de utilisateur essentiellement nom d'utilisateur, mot passe .

Le détail des champs est donné par le tableau suivant :

Nom de champ	Type	Dimension	Designation
Id	Int	/	Identification
Nom_d'utilisateur	String	20	Nom d'user
Mot_passe	String	20	Mot passe d'user

Table 01 : Structure de la table utilisateur

➤ Table Demande :

Utilisée pour l'enregistrement les informations de demande , la clé primaire id et le nom Département un clé étrangères.

Le détail des champs est donné par le tableau suivant :

Nom de champ	Type	Dimension	Designation
Id	Int	/	Identification
Code_Demande	String	30	Code de demande
Numéro_inscription	String	30	Numéro d'inscription d'étudiante
Nom_Etudiant	String	20	Nom de étudiant
Prénom_Etudiant	String	20	Prénom de étudiant
Département_origine	String	50	Nom de département
Département_Destination	String	50	Nom de département
Date dépôt_Demande	Date	/	Date d'un demande
Motif_Transfert	Text	/	Motif et cause de transfert

Demande_Traité	Booléen	/	Vérification la demande
----------------	---------	---	-------------------------

Table 02 : Structure de la table demande

➤ Table Fiche de transfert :

Utilisée pour l'enregistrement des informations de l'étudiant, la clé primaire id et le nom Département une clé étrangère.

Le détail des champs est donné par le tableau suivant :

Nom de champ	Type	Dimension	Designation
Id	Int	10	Identification
Code_Demande	String	30	Code de demande
Numéro_inscription	String	30	Numéro d'inscription d'étudiante
Nom_etudiant	String	20	Nom de étudiant
Prénom_etudiant	String	20	Prénom de étudiant
Date_naissance	String	20	Date de naissance de l'étudiant
Université	String	20	Le nom de université
Faculté_origine	String	20	Nom de faculté
Faculté_Destination	String	20	Nom de faculté
Département_origine	String	20	Nom de département
Département_Destination	String	20	Nom de département
Moyenne_général	float	/	Moyenne général
Etat_discipline	String	/	Etat discipline l'étudiant
Date_admission	Date	/	Date admission étudiant au faculté
Année_BAC	Int	/	Année de bac
Numéro_d'inscription_BAC	String	30	Numéro d'inscription BAC
Email	email	/	Email de l'étudiant
Numéro_téléphone	int	/	Numéro téléphone de l'étudiant
Avis_Faculté_Destination	Booléen	/	Avis Faculté Destination
Avis_département_Destination	Booléen	/	Avis département Destination

Table 03 : Structure de la table Fiche de transfert

➤ Table décision finale:

Cette table Utilisée pour sauvegarder les information de décision la clé primaire le champs id et le nom Département et Numéro d'inscription comme des clés étrangères.

Le détail des champs est donné par le tableau suivant :

Nom de champ	Type	Dimension	Designation
Id	Int	10	Identification
Code_décision	String	20	Code décision
Code_Demande	String	30	Code de demande
Numéro_inscription	String	30	Numéro d'inscription d'étudiante
Nom_etudiant	String	20	Nom de étudiant
Prénom_etudiant	String	20	Prénom de étudiant
Faculté_origine	String	20	Nom de faculté
Faculté_Destination	String	20	Nom de faculté
Département_origine	String	50	Nom de département
Département_Destination	String	50	Nom de département
Date_Demande	Date	/	Date d'un demande
Décision_Transfert	Booléen	/	Accord ou no
Date_Décision	Date	/	Date Décision
Motif_décision	Text	/	Motif et cause d'admission

Table 04 : Structure de la table décision finale

➤ Table Faculté :

Cette table Utilisée pour sauvegarder les information de faculté essentiellement Id comme clé primaire

Le détail des champs est donné par le tableau suivant :

Nom de champ	Type	Dimension	Designation
Id	Int	/	Identificateur
Code_Faculté	String	20	Code de Faculté
Nom_Faculté	String	20	nom de Faculté
Tronc_commun	String	30	Tronc commun de faculté

Table 05 : Structure de la table Faculté

➤ Table département :

Cette table Utilisée pour sauvegarder les informations de département essentiellement Id comme clé primaire et code faculté comme clé étrangère.

Le détail des champs est donné par le tableau suivant :

Nom de champ	Type	Dimension	Designation
id	Int	/	Identificateur
Code_Faculté	String	20	Code département
Code_département	String	20	Code département
Nom_département	String	20	nom de département
Tronc_ècommun	Int	/	Nombre de place au département

Table 06 : Structure de la table département

10. Conclusion

Dans ce chapitre, on a utilisé le langage de modélisation UML pour la spécification, et la conception en s'appuyant sur les principaux diagrammes structurels, leurs caractéristiques et leurs utilisations tels que les diagrammes de cas utilisation, de classes, de séquences ...

Ces diagrammes donnent à l'utilisateur un moyen de visualiser et de manipuler l' Application avant sa réalisation.

CHAPITRE III

Implémentation

CHAPITRE 3 : Implémentation

1. Introduction

Dans ce chapitre nous allons présenter nos outils, nous exposons également choix concernant les technologies utilisées dans la réalisation et le développement de notre Application web, telles que les langages de programmation, les bases de données, les serveurs etc. et puis expliquée le technique mvc.

Ensuite nous allons présenter les pages et les fonctionnalités de ce Application .

Pour accomplir notre application nous avons utilisé les outils suivants :

- Python.
- Framework Django.
- MVC.
- Postgrsql.
- Apache.

2. Les outils de développement

2.1. Python

Python est un langage portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) une approche modulaire et orientée objet de la programmation. Python est développé depuis 1989 par Guido van Possum et de nombreux contributeurs bénévoles.

2.1.1. Définition Python

Python est un langage de programmation, comme C, Fortran, BASIC, PHP, etc... Voici quelques caractéristiques du langage :

- un langage interprété et non compilé. Contrairement au C ou au Fortran par exemple, l'utilisateur ne doit pas compiler son code Python avant de l'exécuter. Python est par ailleurs un langage qui permet l' utilisation interactive :
 - ✓ il existe plusieurs interpréteurs de Python qui permettent d'exécuter du code Python en mode interactif.
- un langage distribué sous une licence open-source : on peut l'utiliser gratuitement, même pour construire des logiciels commerciaux payants.

- multi-plateforme : Python tourne sous Windows, Linux/Unix, Mac OS X, OS/2, les Palm et les téléphones Nokia, les machines virtuelles Java et .NET, donc dans beaucoup d'environnements.
- un langage avec de nombreuses bibliothèques devenues des standards dans de nombreux domaines, de la construction d'applications web au calcul scientifique .
- python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur, par un mécanisme de comptage de références (proche, mais différent, d'un garbage collector).
- Enfin, quelques caractéristiques techniques que nous allons illustrer par la suite : Python est un langage orienté-objet, au typage dynamique (le type d'un objet peut changer au cours d'un programme).[11]



Figure 3.1 : Logo python

2.1.2. Domaines d'application

Les domaines d'application naturels de Python incluent entre autres:

- L'apprentissage de la programmation objet.
- Les scripts d'administration système ou d'analyse de fichiers textuels.
- Tous les développement liés à l'Internet et en particulier au Web: scripts CGI, navigateurs Web, moteurs de recherche, agents intelligents, objets distribués...
- L'accès aux bases de données (relationnelles).
- La réalisation d'interfaces graphiques utilisateurs.
- Le calcul scientifique et l'imagerie. Python ne sert alors pas à écrire les algorithmes, mais à combiner et mettre en œuvre rapidement des bibliothèques de calcul écrites en langage compilé (C, C++, Fortran, Ada,...).

- Le prototypage rapide d'applications. L'idée générale est de commencer par écrire une application en Python, de la tester (ou de la faire tester par le client pour d'éventuelles modifications du cahier des charges). Trois cas peuvent alors se présenter:
- Les performances sont satisfaisantes, après optimisation éventuelle du code Python. On livre alors le produit tel quel au client.

❖ Voici une liste de projets représentatifs basés sur Python:

- ✓ Zope, un serveur d'application innovant, et CPS, un Framework de gestion de contenu et de travail collaboratif basé sur Zope.
- ✓ Les programmes d'administration système spécifiques à la distribution Red Hat Linux.
- ✓ Des moteurs de recherche comme Google ou Yahoo!.
- ✓ Chandler, le projet de PIM (Personnel Information Manager) de l'Open Source Applications Fondation.

2.2. Framework Django

2.2.1. Définition Django

Django est un Framework python open-source consacré au développement web 2.0. Le projet a le slogan suivant: "Le Framework web pour les perfectionnistes sous pression". Il est donc clairement orienté pour les développeurs ayant comme besoin de produire un projet solide rapidement et sans surprise : c'est à dire à tous les développeurs web en fait. Django est une belle boîte à outils qui aide et oriente le développeur dans ses projets. Pour la petite histoire Django a vu le jour en 2003 et a été publié sous licence BSD en juillet 2005. [14]



Figure 3.2 : Logo Django

2.2.2. Pourquoi utiliser ce Framework ?

Si vous êtes fan de python c'est déjà une bonne raison de s'intéresser à Django. Beaucoup de développeurs sont beaucoup plus productifs sur python que sur n'importe quel autre langage. La technologie derrière ce Framework peut être une bonne raison, mais Django est une raison à part entière. Tout est pensé de base. Inutile de réinventer la roue à chaque projet web: les bases sont déjà présentes. Django s'inspire du modèle MVC, c'est à dire que la structure du projet fait que les données sont séparées des traitements qui sont eux-mêmes séparés de la vue.

Le moteur de Template de base est le plus simple que j'ai rencontré, efficace souple et facile à prendre en main. Un contrôleur de base permet de rediriger les actions en fonctions des URL et une API permet de fournir des informations sur votre projet sans passé par la case SQL. D'ailleurs l'ORM inclu vous éloignera de toute requête SQL.

Une des forces de Django c'est de pouvoir lancer son propre serveur web et d'y faire à peu près tout ce que l'on veut dans un environnement de test.

Une des choses que j'apprécie le plus dans ce Framework c'est qu'une fois que vous avez créé vos modèles (c'est à dire que vous avez défini la structure du contenu de votre projet), un espace utilisateur est automatiquement accessible et vous pouvez ajouter modifier et supprimer des données via une interface web. De la pure magie, il travaille à votre place!

Un nouveau développeur qui intègre un projet Django est opérationnel en deux secondes. D'une part par l'installation et la configuration de son environnement de développement mais également par le fait que tous les projets Django ont la même structure, celui-ci ne perdra jamais de temps à trouver une fonction ou comprendre un mécanisme de programmation. C'est d'ailleurs très intéressant cette homogénéité de structure pour des échanges de données entre plusieurs projets Django.

On aime aussi dans Django sa grande popularité, ce qui fait qu'il existe de nombreux tutoriels et la documentation officielle est d'ailleurs très bien faite.[15]

❖ En résumé, pourquoi choisir Django:

- simplicité d'apprentissage
- efficacité de votre développement
- solidité de vos projets
- sécurité finale
- facilité de maintenance
- facilité d'intégration de nouveaux développeurs.

2.3. Présentation de l'architecture MVC

L'architecture MVC (modèle, vue et contrôleur) est un concept très puissant qui intervient dans la réalisation d'une application. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur).

Ce concept peut très bien être appliqué à un site internet. De nombreux sites internet disposent d'une plateforme d'administration pour modifier facilement les contenus. Il est bien évident que l'utilisateur qui va utiliser ce système pour créer des pages ne doit pas modifier la structure complète du site, ni les actions ni l'affichage. Pour qu'il ne modifie que les données, celles-ci doivent alors être isolées.

C'est sur ce constat que se base le système MVC. Chaque partie est bien distincte. Les données peuvent provenir d'une source quelconque et la vue peut être conçue par des webdesigners n'ayant pas de connaissance en programmation. .[15]

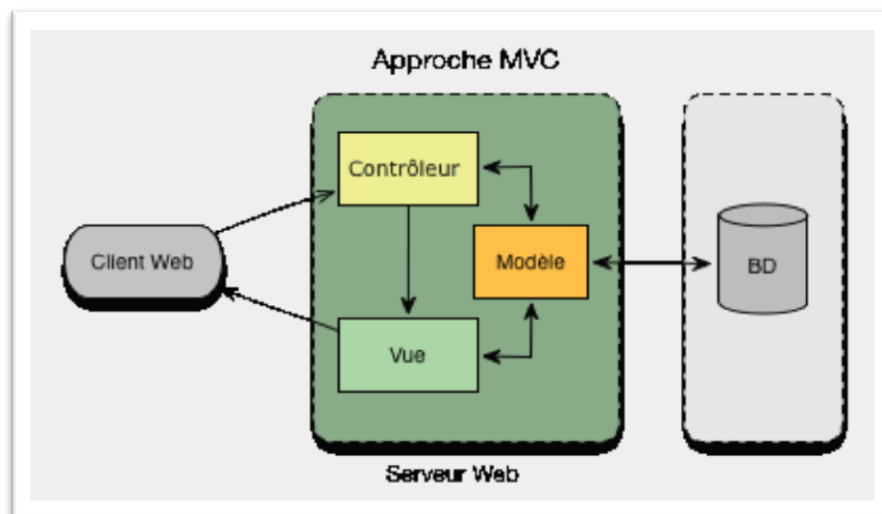


Figure 3.3: Approche MVC

2.3.1. Avantages du MVC

L'approche MVC apporte de réels avantages:

- Une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur
- Un gain de temps de maintenance et d'évolution du site.

- Une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données, de l’affichage (webdesign) et des actions)

2.3.2. Inconvénients du MVC

L’inconvénient majeur du modèle MVC n’est visible que dans la réalisation de petits projets, de sites internet de faible envergure.

En effet, la séparation des différentes couches nécessite la création de plus de fichiers (3 fois plus exactement):

1. Un fichier pour le modèle.
2. Un fichier pour le contrôleur.
3. Un fichier pour la vue.

Il n’est donc pas très intéressant de recourir à ce système dans ce cas-là.

2.3.3. MVC

L’architecture MVC est donc décomposée en trois étapes :

➤ Le Modèle

Le modèle correspond aux données, la plupart du temps stockées dans une base de données. Mais celles-ci peuvent également être contenues dans un fichier XML ou dans des fichiers texte. Les données peuvent être exploitées sous forme de classes, dans un langage de programmation orientée objet (comme PHP, python).

➤ Le Contrôleur

Le contrôleur est l’élément qui va utiliser les données pour les envoyer à la vue. Son rôle est donc de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue (par l’utilisateur, exemple: afficher les derniers articles), puis de renvoyer à la vue les données afin d’être affichées.

- Le contrôleur peut donc instancier différents objets (classe User, classe Articles, ...) qui enverront des requêtes vers la base de données ou récupéreront des données XML.
- La transmission à la vue des données récupérées se fait généralement à l’aide d’un Template : insertion des données dans un fichier HTML qui va être envoyé au navigateur.

➤ La vue

La vue correspond à un fichier HTML la plupart du temps (peut être un fichier XML avec une feuille de style XSL). Elle se contente d'afficher le contenu qu'elle reçoit sans avoir connaissance des données.

Elle ne contient que les informations liées à l'affichage. Dans le cas d'une page HTML, elle contient les éléments `<h1>`, `<p>`, `<div>` ... dans lesquels vont venir s'insérer les contenus.

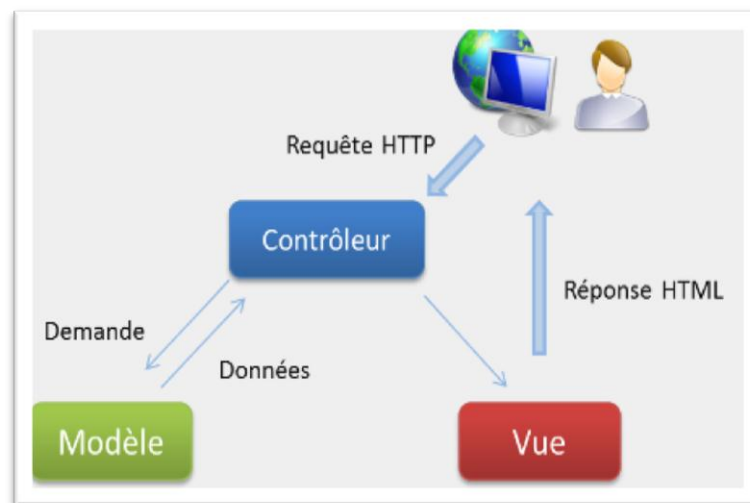


Figure 3.4 :Schéma de l'architecture MVC

2.3.4. La spécificité de Django

- Le modèle MVT :

L'architecture utilisée par Django diffère légèrement de l'architecture MVC classique. En effet, la « magie » de Django réside dans le fait qu'il *gère lui-même la partie contrôleur* (gestion des requêtes du client, des droits sur les actions...). Ainsi, nous parlons plutôt de Framework utilisant l'architecture MVT : **Modèle-Vue-Template**.

Cette architecture reprend les définitions de modèle et de vue que nous avons vues, et en introduit une nouvelle : le Template (voir figure suivante). Un Template est un fichier HTML, aussi appelé en français « gabarit ». Il sera récupéré par la vue et envoyé au visiteur ; cependant, avant d'être envoyé, il sera analysé et exécuté par le Framework, comme s'il s'agissait d'un fichier avec du code. Django fournit un moteur de Template très utile qui

permet, dans le code HTML, d'afficher des variables, d'utiliser des structures conditionnelles (`if/else`) ou encore des boucles (`for`), etc.[15]

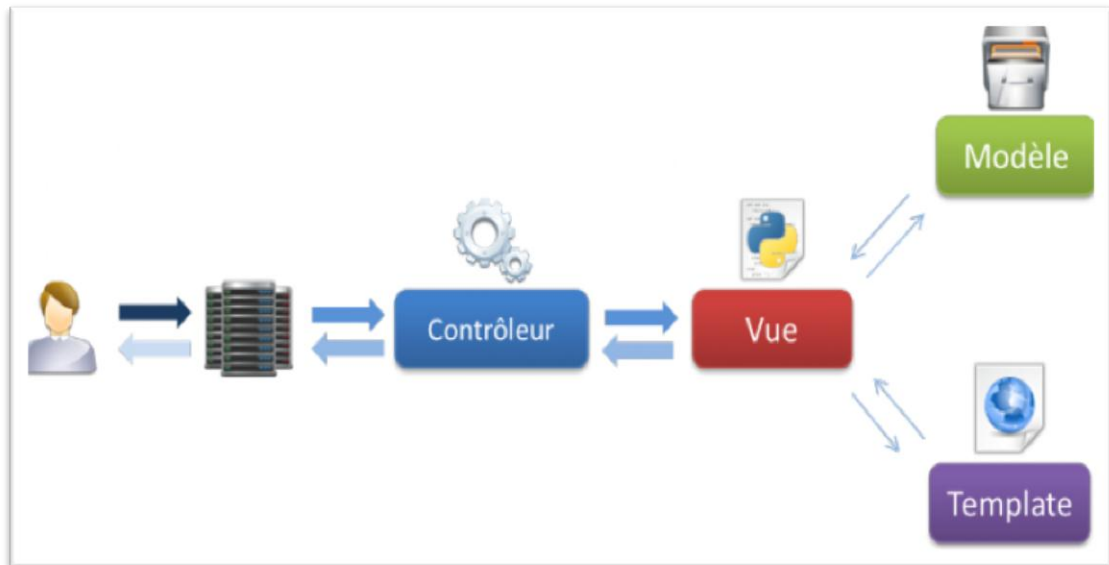


Figure 3.5 : Schéma d'exécution d'une requête (MVT)

2.4. Présentation de PostgreSQL

PostgreSQL est un SGBDR (système de gestion de base de données relationnelles) fonctionnant sur des systèmes de type UNIX (par exemple Linux, FreeBSD, AIX, HP-UX, IRIX, Solaris, ...).

L'une des principales qualités de PostgreSQL est d'être un logiciel libre, c'est-à-dire gratuit et dont les sources sont disponibles.



Figure 3.6 : logo postgresQL

2.4.1. Caractéristiques de PostgreSQL

PostgreSQL possède de nombreuses caractéristiques en faisant un SGBDR robuste et puissant digne des SGBD commerciaux :

- des interfaces graphiques (X-Window est donc nécessaire) pour gérer les tables.
- des bibliothèques pour de nombreux langages (appelés *frontaux*) afin d'accéder aux enregistrements à partir de programmes écrits en :
 - Java (JDBC)
 - langage C/C++
 - Perl
 - Tcl/Tk
- une API ODBC permettant à n'importe quelle application supportant ce type d'interface d'accéder à des bases de données de type PostgreSQL. PostgreSQL fonctionne selon une architecture client/serveur, il est ainsi constitué :
 - d'une partie serveur, c'est-à-dire une application fonctionnant sur la machine hébergeant la base de données (le serveur de bases de données) capable de traiter les requêtes des clients. Il s'agit dans le cas de PostgreSQL d'un programme résident en mémoire appelé postmaster
 - d'une partie client devant être installée sur toutes les machines nécessitant d'accéder au serveur de base de données (un client peut éventuellement fonctionner sur le serveur lui-même).[13]

2.5. serveur web

Un serveur web est un logiciel permettant à des clients d'accéder à des pages web, c'est-à-dire en réalité des fichiers au format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur distant.

Un serveur web est donc un « simple » logiciel capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (par défaut le port 80), et de fournir une réponse avec ce même protocole.

Les principaux serveurs web sur le marché sont entre autres :

- Apache
- Microsoft IIS (Internet Information Server)
- Microsoft PWS (Personal Web Server)

2.5.1. Présentation Apache

Apache est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes.

Apache possède désormais de nombreuses fonctionnalités dont la possibilité de définir une configuration spécifique à chaque fichier ou répertoire partagé. [12]

3. Les Différents tables que nous avons créent avec postgresql

3.1. Présentation de la base de données

Table 1 :

Table	Propriétaire	Commentaires
STE_decision_finale	postgres	
STE_demande	postgres	
STE_departement	postgres	
STE_faculte	postgres	
STE_fiche_transfert_etudiant	postgres	

Figure 3.7 :la structure de la base de données(postgresql)

Table 2 :

Nom	Type de données	Non NULL ?	Clé primaire ?	Valeur par défaut	Commentaires
id	integer	Oui	Oui	nextval('auth_user_id_seq'::regclass)	
password	character varying(128)	Oui	Non		
last_login	timestamp with time zone	Oui	Non		
is_superuser	boolean	Oui	Non		
username	character varying(30)	Oui	Non		
first_name	character varying(30)	Oui	Non		
last_name	character varying(30)	Oui	Non		
email	character varying(75)	Oui	Non		
is_staff	boolean	Oui	Non		
is_active	boolean	Oui	Non		
date_joined	timestamp with time zone	Oui	Non		

Figure 3.8 :la table user

Table 3 :

Nom	Type de données	Non NULL ?	Clé primaire ?	Valeur par défaut	Commentaires
id	integer	Oui	Oui	nextval('"STE_demande_id_seq"::regclass)	
code_demande	character varying(30)	Oui	Non		
num_insc	character varying(30)	Oui	Non		
nom_etud	character varying(20)	Oui	Non		
prenom_etud	character varying(20)	Oui	Non		
nom_faculte_id	integer	Oui	Non		
departement_origine	character varying(50)	Oui	Non		
nom_dep_id	integer	Oui	Non		
date_depot_demande	date	Oui	Non		
motif_trnsfert	text	Oui	Non		
demande_traite	boolean	Oui	Non		

Figure 3.9 :la table demande

Table 4:

Nom	Type de données	Non NULL ?	Clé primaire ?	Valeur par défaut	Commentaires
id	integer	Oui	Oui	nextval('"STE_fiche_transfert_etudiant_id_seq"::regclass)	
code_demande_id	integer	Oui	Non		
date_demande	date	Oui	Non		
num_insc	character varying(30)	Oui	Non		
nom	character varying(30)	Oui	Non		
prenom	character varying(30)	Oui	Non		
adresse	character varying(50)	Oui	Non		
date_naissance	date	Oui	Non		
universite	character varying(50)	Oui	Non		
nom_faculte_dorigine	character varying(20)	Oui	Non		
faculte_destination_id	integer	Oui	Non		
departement_origine	character varying(50)	Oui	Non		
nom_dep_id	integer	Oui	Non		
moyenne_general	double precision	Oui	Non		
annee_scolaire	character varying(20)	Oui	Non		
etat_displinaire	character varying(20)	Oui	Non		
date_admission_faculte	date	Oui	Non		
annee_bac	character varying(20)	Oui	Non		
num_ins_bac	integer	Oui	Non		
email	character varying(100)	Oui	Non		
avis_FD	boolean	Oui	Non		
avis_DD	boolean	Oui	Non		

Figure 3.10 : la table fiche de Transfert

Table 5 :

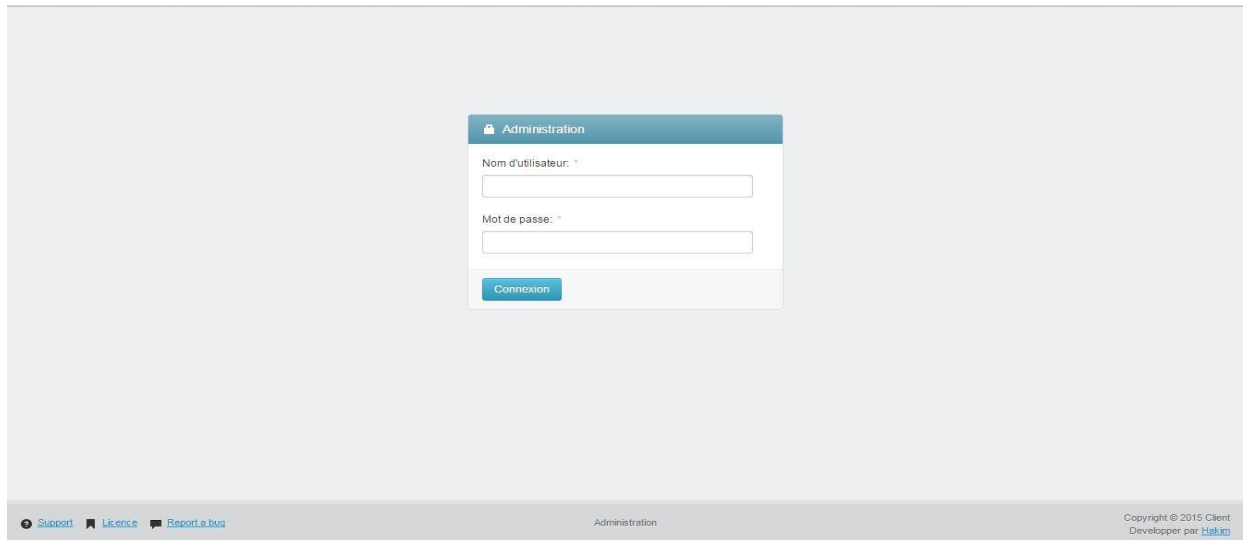
Nom	Type de données	Non NULL ?	Clé primaire ?	Valeur par défaut	Commentaires
id	integer	Oui	Oui	nextval('"STE_decision_finale_id_seq"::regclass)	
num_insc	character varying(30)	Oui	Non		
date_decision	date	Oui	Non		
nom	character varying(20)	Oui	Non		
prenom	character varying(20)	Oui	Non		
nom_faculte_id	integer	Oui	Non		
nom_dep_id	integer	Oui	Non		
decision_trnsfert	boolean	Oui	Non		
motif_decision	text	Oui	Non		

Figure 3.11 : la table décision Final

4. Présentation de l'application web

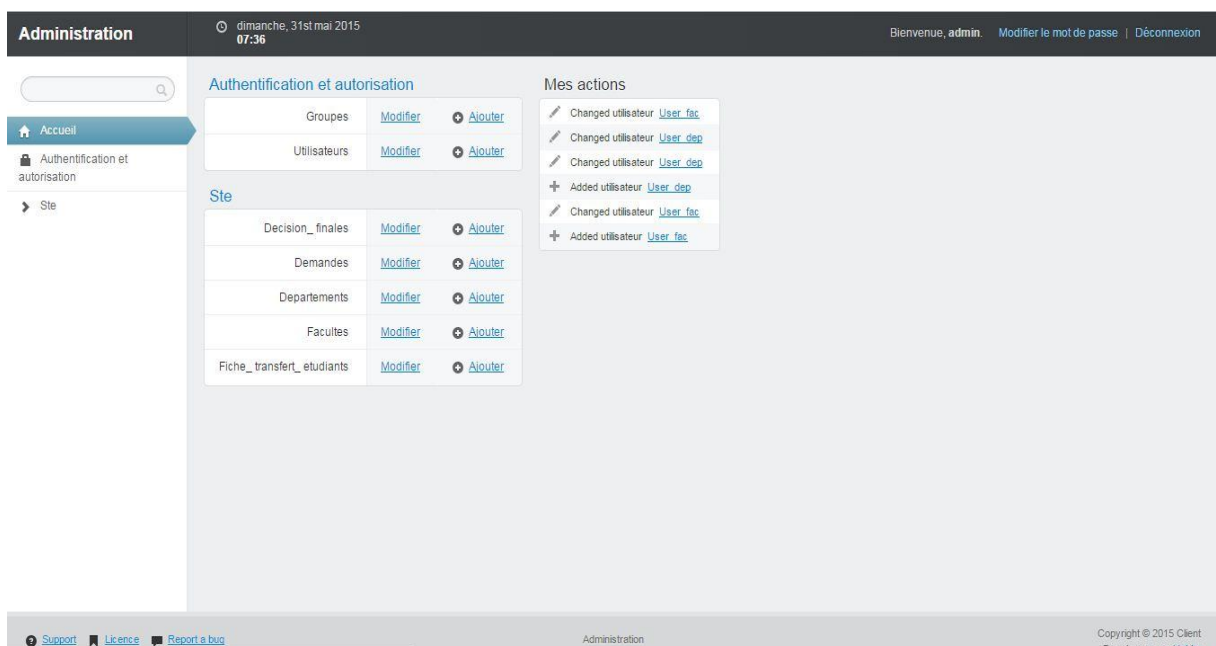
4.1. Authentification

La figure ci-dessous permet à l'utilisateur de s'authentifier et de se connecter au serveur de la base de données. L'utilisateur doit entrer son login et son mot de passe pour accéder à l'application. En cas d'erreur un message d'alerte s'affiche .



4.2. Espace Administrateur

La figure ci-dessous nous donne un aperçu de l'espace <<Administration>> cette espace offre plusieurs fonctionnalités y compris ajoutée les utilisateurs ,les groupes et gérée le system collaboratif , contrôler les action et les tracs de utilisateur , en plus offrir les permissions de chaque utilisateur et modifier les information de compte d'administrateur) .



4.3. Espace d'ajoute demande

Cette espace permet au user de faculté d'ajouter demande, de remplir les Informations de l'étudiant et les enregistrée .

The screenshot shows the 'Ajouter demande' form in the Administration interface. The form includes the following fields:

- Code Demande : *
- Numéro inscription : *
- Nom Etudiant : *
- Prénom Etudiant : *
- Faculté destination : *
- Département Origine : *
- Département destination : *
- Date depot Demande : *
- Motif Transfert : *

Buttons on the right side of the form include 'Enregistrer' and 'Enregistrer et ajouter un nouveau'. A 'Champ Réserve (Afficher)' button is located at the bottom left of the form area.

4.4. Consulter les demandes

The screenshot shows the 'Demandes' table in the Administration interface. The table has the following columns:

- Code Demande
- Numéro inscription
- Nom Etudiant
- Prénom Etudiant
- Date depot Demande
- Demande Traité
- Ajouté récemment ?

The table contains three rows of data:

Code Demande	Numéro inscription	Nom Etudiant	Prénom Etudiant	Date depot Demande	Demande Traité	Ajouté récemment ?
03	08/10	karim	karim	1 juin 2015	✓	✗
02	06/10	ahmed	karim	30 mai 2015	✗	✗
01	05/10	ahmed	ahmed	31 mai 2015	✗	✗

Below the table, there is a search bar with a 'Rechercher' button and a 'Ajouter demande' button. The table also shows a pagination bar at the bottom: '1 - 3 / 3 demandes'.

Cette espace représente un ensemble des demandes et offre plusieurs Fonction par exemple filteries par la date dépôt demande ou par les demande traite, les notifications de chaque demande, L'User département peut accéder à l'espace propre pour consulter les demandes.

4.5.Création fiche de Transfert

Après la Consultation de demande par User département , Il peut créer un fiche de transfert et l'enregistrées pour le consulter par faculté destinée.

Cette espace inclut tous les information de l'étudiant.

Administration

dimanche, 31st mai 2015
10:39

Bienvenue, User_dep.
[Modifier le mot de passe](#)
[Déconnexion](#)

Accueil

> Site

Decision_finales

Demandes

Fiche_transfert_etudiants

Accueil > Site > Fiche_transfert_etudiants > Ajouter fiche_transfert_etudiant

Code demande : *

Date Demande : *

Numéro inscription : *

Nom : *

Prénom : *

Adresse : *

Date naissance : *

Université : *

Faculté D'origine : *

Faculté destination : *

Département Origine : *

Département destination : *

Moyenne général : *

Année Scolaire : *

Etat Disciplinaire : *

Date Admission au Faculté : *

Année BAC : *

Némuro d'inscription BAC : *

Email : *

Enregistrer

Enregistrer et ajouter un nouveau

Champ Réserve (Afficher)

[Support](#)
[Licence](#)
[Report a bug](#)

Administration

Copyright © 2015 Client.
Developper par [Hakim](#)

4.6. Consulter fiche de Transfert

User faculté et user département accéder à l'espace propre pour consulter fiche de transfert ,cette espace Peut faire plusieurs Fonction par exemple filteries par la date demande ou par les Faculté destinations et qui offre les notifications de chaque fiche de transfert .

Administration | dimanche, 31st mai 2015 11:27 | Bienvenue, User_fac. | Modifier le mot de passe | Déconnexion

Accueil » Ste » **Fiche_transfert_etudiants**

Keyword Date Demande Faculté destination Avis departement Destination Avis Faculté Destination Rechercher

« 2015 | 31 mai

	Code demande	Nom	Date Demande	Avis Faculté Destination	Avis departement Destination	Ajouté récemment ?
<input type="checkbox"/>	02	ahmed	31 mai 2015	✓	✗	✓
<input type="checkbox"/>	01	ahmed	31 mai 2015	✗	✗	✓

Envoyer 0 sur 2 sélectionné

1 - 2 / 2 fiche_transfert_etudiants

Support | Licence | Report a bug | Administration | Copyright © 2015 Client. Developper par [Hakim](#)

4.7. Décision final :

Après la Consultation et l'études de fiche de transfert par Administration.
user département renvoie la décision Finale ,cette page inclut plusieurs champs parmi ces dernier l'accord , refus et motif de décision .

The screenshot shows a web application interface for 'Administration'. The top header bar displays the date and time 'dimanche, 31st mai 2015 11:02' and the user 'Bienvenue, User_dep.' with links for 'Modifier le mot de passe' and 'Déconnexion'. The left sidebar contains a search bar and a menu with 'Accueil', 'Ste' (selected), 'Decision_finales', 'Demandes', and 'Fiche_transfert_etudiants'. The main content area is titled 'Ajouter decision_finale' and contains a form with the following fields: 'Numéro inscription : *' (text input), 'Date Décision : *' (date picker with 'Aujourd'hui' button), 'Nom étudiant : *' (text input), 'Prénom étudiant : *' (text input), 'Faculté Destination : *' (dropdown menu), 'Département Destination : *' (dropdown menu), 'Accord Transfert' (checkbox), and 'Motif decision : *' (large text area). On the right side of the form, there are two buttons: 'Enregistrer' and 'Enregistrer et ajouter un nouveau'. The footer of the page includes links for 'Support', 'Licence', and 'Report a bug', the text 'Administration', and copyright information 'Copyright © 2015 Client' and 'Développeur par Hakim'.

4.8.Consulter décision finale :

The screenshot shows the 'Administration' web application interface. The top header displays the date 'dimanche, 31st mai 2015' and time '11:28', along with user information 'Bienvenue, User_fac.' and links for 'Modifier le mot de passe' and 'Déconnexion'. The sidebar on the left contains navigation links: 'Accueil', 'Ste', 'Decision_finales', 'Demandes', and 'Fiche_transfert_etudiants'. The main content area is titled 'Decision_finales' and features a search bar with a 'Rechercher' button. Below the search bar, there are filters for 'Date Décision' and 'Numéro inscription'. A table displays the following data:

	Numéro inscription	Date Décision	Nom étudiant	Accord Transfert	Faculté Destination	Ajouté récemment ?
<input type="checkbox"/>	06/10	31 mai 2015	ahmed	-	Mathématique et informatique	✓
<input type="checkbox"/>	05/10	31 mai 2015	ahmed	✓	Mathématique et informatique	✓

Below the table, there is a dropdown menu and an 'Envoyer' button, with the text '0 sur 2 sélectionné'. The footer of the page includes links for 'Support', 'Licence', and 'Report a bug', along with the text 'Administration' and 'Copyright © 2015 Client. Développeur par Hakim'.

User faculté accéder à l'espace propre pour consulter la décision finale et puis renvoie la décision à l'étudiant. Cette page offre plusieurs Fonction par exemple filteries les décision par la date ou par numéro d'inscription et les notifications de chaque décision

5. Conclusion

Dans ce chapitre nous avons mis en évidence toutes les techniques nécessaires à la réalisation de notre Application.

Nous avons tout d'abord commence par indiquer la présentation de l'environnement de développement et les outils utiles et nécessaire à la création du Application web, puis nous avons Présenté la structure des table sur la base de donnée puis donné des exemples réels de nos pages web réalisées.

CONCLUSION GENERALE

Ce projet de fin d'étude consiste à réaliser une Application web pour les transferts inter-faculté.

Au cours de ce mémoire, nous avons présenté les différentes étapes de la conception et l'implémentation de notre application , Afin de satisfaire les besoins des utilisateurs nous avons commencé par décrit quelques scénario de transfert étudiant et puis pour la conception on utilisant le formalisme UML et le modelé MVC(modelé-vue-contrôler) pour structurer l'application et facilite aussi le dialogue entre les concepteurs, utilisée le gestionnaire de bases de données postgresql pour la manipulation des données et enfin la concrétisation de l'application sous l'environnement de programmation python .

Il a fait l'objet d'une expérience intéressante, qui nous a permet d'améliorer nos connaissances et nos compétences dans le domaine de la programmation. Nous avons appris à mieux manipuler les langages python, postgresql et MVC (Model-vue-contrôle) Ainsi à travers cette Application web nous pourrons gérer ces activités et offrir à cette structure une chance de s'imposer et d'entrer de plein pied dans le system d'information collaboratif et essayer de Project la collaboration des système d'information dans .le cas de transfert inter-faculté.

En effet, ce travail étant une œuvre humaine, n'est pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à toutes les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer d'avantage cette étude. Etant donné que tout travail informatique a été toujours l'œuvre d'une équipe.

Bibliographie

- [1] : Jihed TOUZI « Aide à la conception de Système d'Information Collaboratif support de l'interopérabilité des entreprises » Directeur de thèse : Pr. Hervé PINGAUD
Encadrant : Frédérick BENABEN Le : 09/11/2007.
- [2] : amen souissi «MODÉLISATION CENTRÉE SUR LES PROCESSUS MÉTIE POUR LA GÉNÉRATION COMPLÈTE DE PORTAILS COLLABORATIFS» Encadrant : Cédric Dumoulin'
Maître de Conférence Université de Lille1, Le :20/12/2013
- [3] : Travail Collaboratif dans les Parlements : Bamako,13-15 Juin 2011.
- [4] : Laurent Debrauwer," UML 2. Modélisation des objets", 2ème édition, ENI édite ,2010
- [5] : Benoît Charroux, Aomar Osmani, Yann Thierry-Mieg- UML2, 3e édition –© 2010 Pearson Education France
- [6] : Lotfi kelili," conceptions et réalisation d'un site web dynamique pour l'INSFP ",
diplôme Ingénieur d'état, session 2006, université Yahia Fares Médéa .
- [7] :UML, Le langage de modélisation objet unifié par Laurent Piechocki.
- [8] : C. Morley, La modélisation des processus : typologie et proposition utilisant UML. Processus et Systèmes d'information – Journées ADELI, Paris, France, 2002.

Webographie

- [9] :http://liris.cnrs.fr/amille/enseignements/IGC_M2_2008/ consulté le : 08/04/2015
- [10] :<http://www.developpez.com/> consulté le : 05/04/2015
- [11] : <http://www.linux-center.org/> consulté le : 20/04/2015
- [12] : <http://www.commentcamarche.net/> consulté le : 10/03/2015
- [13] :<http://www.jenny.bourdiol.org/> / consulté le : 25/03/2015
- [14] : <http://www.webzeo.org/> consulté le : 04/05/2015
- [15] : <http://www.guillaumevoisin.fr/> consulté le : 10/05/2015

الملخص:

الهدف الأول من وراء هذا العمل هو الاطلاع على أنظمة المعلومات المتعاونة وتجسيد تطبيق ديناميكي يعمل على مساعدة الطالب و الإدارة في عملية تحويل الطالب بين الكليات على مستوى الجامعة. مما يسمح بحل عدة مشاكل يعاني منها الطالب و الإدارة معا. ومن اجل تصميم هذا التطبيق اخترنا لغة النمذجة الموحدة UML التي لبت كل متطلباتنا وللتطوير والانجاز قمنا باختيار محيط البرمجة Django وكلغة برمجة اخترنا python باعتماد الخادم Apache .

الكلمات المفتاحية: التعاونية، تطبيق ويب, Inter-faculté , تحويل الطالب

Abstract :

The main objective behind this work is the design and implementation of a collaborative information system, using a dynamic Web application to help the student and administration for inter-faculty transfer at the university. In order to help solving several problems that face student and the administration.

We used UML for designing this system, Django development environment, Python language programming and Apache server are the main tools used to implement our system.

Keywords: Collaborative, Web application, Inter-faculty, transfer student.

Résumé :

L'objectif essentiel de notre travail, c'est de voir les système d'information collaboratif et essayer de contribuer à la réalisation d'une Application Web dynamique qui facilitera a l'étudiant et a l'administration le transfert étudiant inter-faculté au niveau de l'université.

Ce qui permettra de résoudre plusieurs problèmes dont souffrent les étudiants et l'administration ...

Pour la conception et la réalisation de notre application, nous avons choisi le langage de modélisation UML à l'environnement de développement Django avec le langage de programmation python et le serveur Apache .

Mots clés : Collaboratif , Application web , Inter-faculté ,transfert étudiant.