



Projet Tutoré

---

## Daily Print

---

*Auteurs :*

Saliou BA  
Julien DAURAT  
Ariella LEVY  
Bastien MARAIS

*Encadrant :*

M. Fabrice HOGUIN

Version 0.1 du  
21 février 2018

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Cahier des charges</b>	<b>3</b>
1.1 Énoncé . . . . .	3
1.2 Pré-requis . . . . .	5
1.3 Priorité . . . . .	5
<b>2 Analyse des besoins</b>	<b>6</b>
2.1 Lecture du cahier des charges . . . . .	6
2.1.1 Identification des actions (fonctions), acteurs et objets du problème.	6
2.1.2 Identification des exigences . . . . .	7
2.2 Analyse des interactions logiciel/environnement . . . . .	7
2.2.1 Diagrammes des cas d'utilisations . . . . .	8
2.2.2 Spécification des cas d'utilisation . . . . .	8
2.3 Exigence non fonctionnelle . . . . .	12
2.4 Exigence Organisationnelle et Métiers . . . . .	12
2.5 Glossaire . . . . .	12
<b>3 Dossier concepteur</b>	<b>13</b>
3.1 Spécification détaillé de la structure du système . . . . .	13
3.1.1 Comportement . . . . .	15
3.2 Spécification détaillé des interface utilisateur . . . . .	22
3.2.1 Prototype de l'interface . . . . .	22
3.2.2 Spécification des rapport et formulaires . . . . .	23

<b>Table des matières</b>	<b>1</b>
3.2.3 Politique et stratégie . . . . .	23
3.2.4 Choix des outil a utilisé . . . . .	24
<b>4 Dossier architecture</b>	<b>25</b>
4.1 Vue modulaire . . . . .	25
4.2 Vue composant/Connecteur . . . . .	26
4.3 Vue d'attribution . . . . .	26
<b>Conclusion</b>	<b>28</b>
<b>Remerciement</b>	<b>29</b>

# Introduction

Le projet tutoré représente l'aboutissement de nos savoirs acquis lors de notre cursus par une application ; la réalisation d'une application pour un client spécifique. Nous devons faire une application Web de pour la gestion de la reprographie pour Monsieur Huguin. Ce projet est une application web qui cherche à faciliter la gestion du système de reprographie de l'IUT de Vélizy. Celle-ci doit permettre une meilleure communication entre les clients, ici les professeurs et le service de reprographie. Notre projet sera développé en PHP, HTML, CSS, JavaScript, MySQL, PL/SQL et SQL. Plusieurs domaines toucheront notre projet comme la MPA, la Base de données, la programmation web, la gestion de projet et l'expression communication. Notre projet composé de 4 membres la répartition de leur tâche est la suivante en termes de rédaction des dossiers :

- BA Saliou : Dossier architecture, Analyse des besoins et cahier des charges
- DAURAT Julien : Développement et cahier des charges
- MARAIS Bastien : Chef de projet, développement et cahier des charges
- LEVY Ariella : Dossier concepteur, cahier des charges et rédaction du rapport

Dans ce rapport nous allons vous communiquer nos résultats à travers notre dossier. Nos dossier et notre rapport représente notre projet sur une version minimal ou certaine partie de l'application n'ont pas encore été développés. Dans notre rapport nous vous présenterons l'équipe de travail nous expliquerons les solutions que nous avons trouvées pour présenter le web ainsi que le problème que nous avons eu et comment nous l'avons résolu et enfin nous analyserons la réalisation de ce projet. Cette présentation sera appuyée sur nos différentes annexes.

# Chapitre 1

## Cahier des charges

Ce projet est une application web qui cherche à faciliter la gestion du système de reprographie de l'IUT de Vélizy. Celle-ci doit permettre une meilleure communication entre les clients, ici les professeurs et le service de reprographie. Dans ce document nous commencerons par présenter une description avancée du problème qui se pose avec le système de reprographie actuel, chose qui nous permettra de mettre en avant la solution envisagée en donnant les cas d'utilisation courants pour les clients. Nous procéderons ensuite à une présentation des détails techniques en spécifiant les outils que nous utiliserons au cours de ce projet. Pour finir, il sera important de mettre en évidence les différentes contraintes présentes ainsi que les exigences du client, chose essentielle à la bonne réalisation du projet.

### 1.1 Énoncé

Le système de reprographie actuel n'est pas le plus efficace. Les professeurs ont deux possibilités :

- Faire une demande sous la forme d'un email via un formulaire pdf
- Faire une demande directement à la reprographie avec un formulaire papier à remplir.

Avec le système actuel, les professeurs ne sont pas informés du statut de leur demandes d'impressions ; en effet des déplacements sont nécessaires pour obtenir des informations. Ce n'est donc pas un système extrêmement pratique et efficace. L'objectif sera donc de proposer une alternative à ce projet afin d'améliorer le confort de l'utilisateur ainsi que de faciliter le travail de la reprographie. Le projet serait créé dans le cadre du remplacement du système actuel par une application Web qui permettrait d'optimiser le temps de traitement des demandes et faciliter la création de celles-ci. Le projet devrait permettre de créer un système où les professeurs peuvent demander des photocopies, en spécifiant des informations comme le nombre de copies, la façon dont elles sont photocopiées (recto,

recto/verso...), les finitions (agrafe, livret...). Ils auront la possibilité de suivre l'état des demandes de photocopies (reçues par la reprographie, traitées ou prêtes à être récupérées). Les personnes qui travaillent à la reprographie recevront les demandes dans l'ordre de leur arrivée, et au fur et à mesure que ces demandes seront traitées, ils auront la possibilité d'en modifier le statut, pour informer les professeurs de l'état d'avancement de ces dernières par l'envoi d'un email ou d'une notification. Les différents statuts possibles pour accéder au site sont les suivants : "Professeur" et "Reprographie". Un professeur pourra donc en se connectant faire une demande d'impression qui sera gérée par les personnes qui travaillent à la reprographie. Pour que les demandes soient traitées plus facilement grâce à un système de tri et pour faire des études statistiques, les professeurs seront classés en fonctions de leur principal département d'enseignement (INFO, MMI, GEII, RT, ISTY...). Ces statistiques pourront, à terme, permettre une meilleure gestion et répartition des tâches au sein de la reprographie. De plus, notre projet permettra de centraliser les demandes. Un exemple de cas d'utilisation du point de vue d'un professeur serait :

- *Le professeur se connecte sur le site grâce à son email et son mot de passe.*
- *Il ne voit que ses demandes et a la possibilité de créer autant de requêtes d'impression qu'il le souhaite.*
- *Pour créer une nouvelle demande, il doit accéder à l'onglet "Créer une nouvelle demande", sur cette page il doit entrer différentes informations nécessaires, tel que le nombre de copies, la façon de photocopier, la finition, le fichier à imprimer et également signaler les éventuelles publications protégées. Une fois les informations entrées, il n'a plus qu'à valider en appuyant sur le bouton en bas de la page.*
- *Une fois cette demande envoyée, elle sera transférée au service de reprographie.*
- *Le professeur peut afficher à tout moment ses demandes depuis l'onglet "Mes demandes".*

Un exemple de cas d'utilisation du point de vue d'un membre de la reprographie serait :

- *Le membre de la reprographie se connecte sur le site grâce à son email et son mot de passe.*
- *Il a accès à toutes les demandes et aura la possibilité de filtrer celles-ci en fonction du département auquel appartient le professeur demandeur.*
- *Une fois qu'une demande est sélectionnée, il pourra accéder à ces informations et donc traiter cette dernière tout en mettant à jour progressivement son statut d'avancement.*
- *Lorsque cette demande est traitée, le membre de la reprographie met à jour le statut de celle-ci sur "Terminée", ceci entraînera l'envoi d'un email automatique à l'adresse du professeur pour le prévenir.*

## 1.2 Pré-requis

Pour la réalisation de ce projet nous utiliserons différents outils :

- Langages **WEB : PHP, HTML, CSS et JavaScript**
- Base de données : **MySQL**
- Langages de base de données : **PL/SQL et SQL**
- Support de fonctionnement : **Ordinateurs et mobiles**

L'interface devra être pratique et fonctionnelle.

## 1.3 Priorité

Nous devons rendre la partie conception, qui comprend le cahier des charges, le diagramme de classe, le diagramme de cas d'utilisation et le diagramme de séquence pour le 23 février 2018.

# Chapitre 2

## Analyse des besoins

Ce document présente les différentes facettes des besoins exprimés par l'utilisateur. On cherche dans ce projet à créer un système de ticketing pour un système de reprographie. Le but de cette annexe est de définir clairement les besoins du client afin d'y répondre au mieux. Dans une première partie, nous étudierons le cahier des charges que nous avons réalisés, en collaboration avec le client et en feront une lecture complète de celui-ci. Puis, nous étudierons les différents cas d'utilisations possibles que nous avons pu identifier au sein du système.

### 2.1 Lecture du cahier des charges

#### 2.1.1 Identification des actions (fonctions), acteurs et objets du problème.

Acteurs	Actions	Objets
Service reprographie	<ul style="list-style-type: none"><li>— Reçoit les demandes</li><li>— Traite les demandes</li><li>— Gestion du système de statistiques</li></ul>	
Professeur	<ul style="list-style-type: none"><li>— Faire une demande</li><li>— Annuler une demande</li></ul>	
Recherche	<ul style="list-style-type: none"><li>— Faire une demande</li><li>— Annuler une demande</li><li>— Choisir les spécifications de la demande</li></ul>	



### 2.1.2 Identification des exigences

Dans le tableau ci-dessous, l'acteur identifié par Utilisateur regroupe tout les utilisateurs de l'application, donc les professeurs, les personnes travaillant à la reprographie et l'administrateur. **Client** : Fabrice Huguin, professeur de l'IUT de Vélizy.

Exigences	Exigences	Acteur	Criticité
1	Pouvoir faire une demande d'impression avec les options déjà existante	Utilisateur	Critique
2	Gestion des demandes entrantes	Reprographie	Critique
3	Récupération du mot de passe	Utilisateurs	Mineur
4	Confirmer les inscriptions	Administrateur	Critique
5	Visualisation des demandes	Utilisateurs	Critique
6	Afficher des statistiques pour aider à la gestion	Service repro-graphie	Mineur
7	Système ergonomique	Utilisateurs	Majeur

## 2.2 Analyse des interactions logiciel/environnement

Le logiciel agira sur les systèmes d'exploitation Windows, Linux. Nous utiliserons comme serveur Apache afin d'héberger le site. Il est important de noter une compatibilité optimale avec les navigateurs **Google Chrome** et **Mozilla Firefox**, et à contrario une compatibilité réduite avec internet Explorer. Le site web Daily'Print sera disponible sur **plateforme mobile**.

### 2.2.1 Diagrammes des cas d'utilisations

FIGURE 2.1 – Diagramme de cas d'utilisation de l'inscription

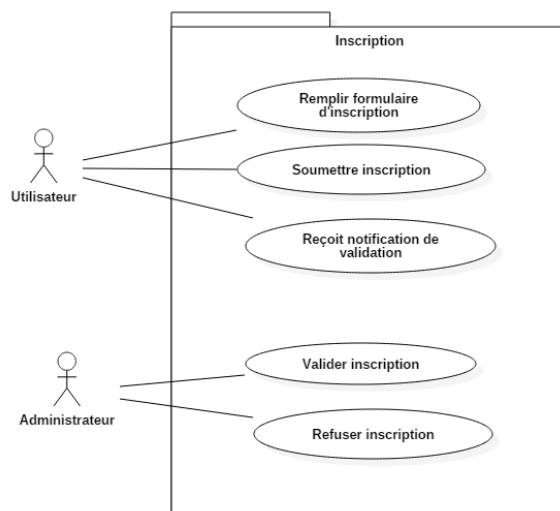
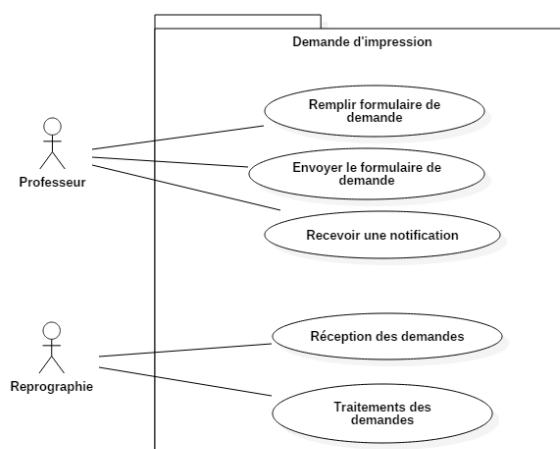


FIGURE 2.2 – Diagramme de cas d'utilisation de la création d'une demande



### 2.2.2 Spécification des cas d'utilisation

#### Cas d'utilisation Administrateur

##### Cas d'utilisation 1 : Gestions des demandes d'inscription

**Contexte d'utilisation :** Cas d'utilisation du système de gestion des demandes d'inscription utilisé par l'administrateur.

**Portée :** Interface de validation.

**Niveau :** utilisateur / stratégique (dépend du temps de réaction de l'administrateur).

**Acteur principal :** Administrateur.

**Intervenants et intérêts :** Administrateur, utilisateur.

**Scénario nominal :**

1. L'administrateur insère son adresse mail.
2. L'administrateur insère son mot de passe.
3. Les données sont bonnes, il est redigé vers son espace de gestion.
4. L'administrateur reçoit les demandes d'inscriptions déroulées entre sa dernière connexion et celle actuelle.
5. L'administrateur valide les demandes d'inscriptions à l'aide du bouton de validation.
6. Notification automatique envoyée à l'utilisateur.
7. L'administrateur se déconnecte.

**Scénario annexe :**

1. L'administrateur insère son adresse mail.
2. L'administrateur insère son mot de passe.
3. Les données ne sont pas bonnes, l'action recommence.

### **Cas d'utilisation 2 : non-validation des inscriptions**

**Contexte d'utilisation :** Cas d'utilisation du système de gestion des demandes d'inscription utilisé par l'administrateur dans le cas où l'inscription n'est pas validée.

**Portée :** Interface de validation.

**Niveau :** utilisateur / stratégique (dépend du temps de réaction de l'administrateur).

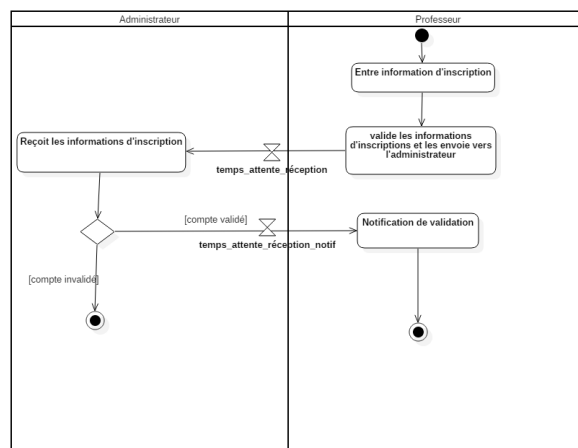
**Acteur principal :** Administrateur.

**Intervenants et intérêts :** Administrateur, utilisateur.

**Scénario nominal :**

1. L'administrateur insère son adresse mail.
2. L'administrateur insère son mot de passe.
3. Si les données sont bonnes, il est redigé vers son espace de gestion.
4. L'administrateur invalide les demandes d'inscriptions.
5. L'administrateur se déconnecte.

FIGURE 2.3 – Diagramme d’activité représentation du système de gestion de l’administrateur.



## Cas d'utilisation Utilisateurs

### Cas d'utilisation 1 : Envoi d'une demande

**Contexte d'utilisation :** Cas d'utilisation du système de création des demandes utilisé par les professeurs.

**Portée :** Interface de création des demandes

**Niveau :** Utilisateur.

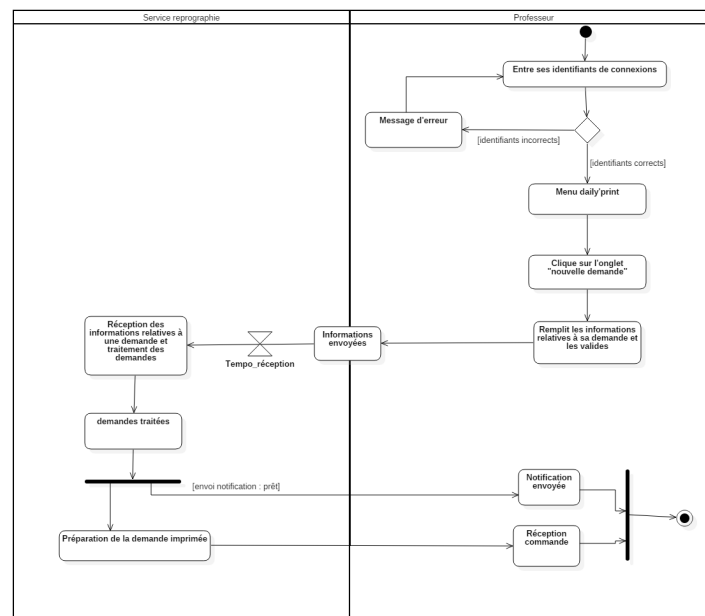
**Acteur principal :** Utilisateur..

**Intervenants et intérêts :** Utilisateur, service reprographie.

**Scénario nominal :**

1. L'utilisateur insère son adresse mail
2. L'utilisateur insère son mot de passe
3. Si les données sont bonnes, il est redirigé vers son espace personnel
4. L'utilisateur se dirige sur le panel de création des demandes
5. L'utilisateur sélectionne les informations pour sa demande
6. L'utilisateur valide sa demande qui est envoyée au service de reprographie
7. temporisation
8. Le service de reprographie reçoit la demande
9. Le service de reprographie traite la demande
10. temporisation
11. L'utilisateur reçoit une notification sur son mail

FIGURE 2.4 – Diagramme d'activité représentation la création et traitement des demandes.



12. temporisation

13. L'utilisateur récupère le résultat de sa demande

### Cas d'utilisation 3 : Inscription validée.

**Contexte d'utilisation :** Cas d'utilisation où l'inscription de l'utilisateur est validée.

**Portée :** Site web

**Niveau :** Utilisateur.

**Acteur principal :** Utilisateur..

**Intervenants et intérêts :** Utilisateur.

**Scénario nominal :**

1. L'utilisateur se dirige sur l'espace d'inscription
2. L'utilisateur insère un nom
3. L'utilisateur insère un prénom
4. L'utilisateur insère une adresse mail valide
5. L'utilisateur insère un mot de passe et le confirme
6. L'utilisateur choisit un statut
7. L'utilisateur spécifie son département d'appartenance
8. L'utilisateur valide son inscription
9. Le système affiche un message lui expliquant que sa demande d'inscription est en cours

10. Temporisation inconnue
11. Si l'administrateur valide la demande d'inscription l'utilisateur reçoit une notification par mail

## 2.3 Exigence non fonctionnelle

Les exigences non fonctionnelles sont par exemples celles qui vont gérer les droits d'accès, en effet l'administrateur sera le seul à pouvoir accéder au panel de gestion des inscriptions. Les professeurs quand-à-eux pourront exclusivement accéder à leurs demandes et ne pourront pas visualiser celles des autres. Pour le service de reprographie les demandes seront accessibles pour chacun des utilisateurs appartenant à ce service, il y aura la présence de filtre pour pouvoir séparer les demandes en fonction du département d'appartenance du professeur.

## 2.4 Exigence Organisationnelle et Métiers

Les exigences sont définies par une organisation : l'IUT de Vélizy et son personnel.

Il est nécessaire de respecter ces exigences, créer un système viable permettant de gérer la reprographie et disponible sur les navigateurs de l'IUT. Un site web pouvant être utilisé sur un serveur de l'IUT (d'où le choix d'Apache)

## 2.5 Glossaire

Reprographie : ensemble des procédés permettant la reproduction directe d'un document

# Chapitre 3

## Dossier concepteur

Le dossier concepteur est destiné aux programmeurs qui veulent comprendre la méthode de conception appliquée, le raisonnement suivi, les choix faits et les raisons de ces choix. Ce dossier montre le modèle de conception, dans lequel on y ajoute des concepts informatiques présents dans les outils, les langages de programmation ou les plate-formes de développement. Ainsi, dans ce rapport, nous étudions la Spécification détaillé de la structure du système, le Comportement, la Spécification détaillé des interface utilisateur, le Modèle de réalisation, la Politique et stratégie et enfin nous aborderons les Choix des outils à utiliser.

### 3.1 Spécification détaillé de la structure du système

Ce schéma représente l'organisation des pages de notre application et les différentes interactions qu'elles possèdent les unes avec les autres :

- la page `index.php` : C'est la page d'accueil de l'application dans laquelle l'utilisateur (quel qu'il soit pourra se connecter). Elle permet un accès à la page `accueil.php` si la connexion est effective. Autrement elle permet un accès `inscription.php` et `oubli.php`.
- la page `inscription.php` : Elle comporte un formulaire d'inscription qui permet à tout individu de faire une demande de création de compte. Étant donné qu'il faut une validation de l'inscription par l'administrateur, cette page renvoi (à l'aide d'un bouton retour) vers `index.php` si l'on se résigne ou alors affiche un message d'information.
- la page `oubli.php` : Cette page permet, en cas d'oubli de mot de passe, de faire une demande de changement de celui-ci. Auquel cas l'utilisateur entrera son adresse e-mail. Elle ne renvoi vers aucune autre page, hormis `index.php` à l'aide du bouton retour.
- la page `accueil.php` : C'est la page d'accueil de l'application qui nous affiche une petite indication sur ce que permet de faire ce site (documentation). Elle renvoi grâce

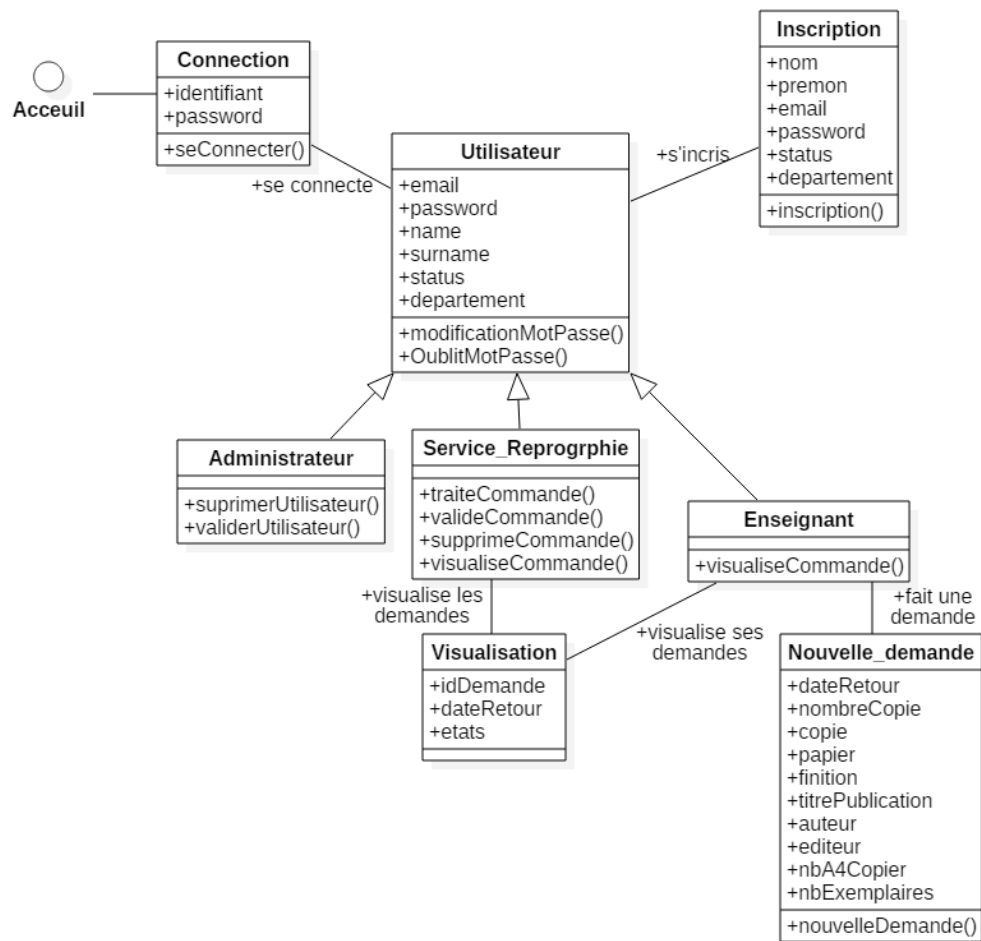
à une barre de navigation aux pages Visual.php, new\_demande.php et aide.php (en terme de pages fonctionnelles), et aussi vers param.php et index.php en cas de déconnexion.

- la page visual.php : C'est une des pages centrale de l'application, elle permet à l'utilisateur de visualiser ses demandes ainsi que leur statut d'avancement. Elle renvoi grâce à une barre de navigation aux pages accueil.php , new\_demande.php et aide.php (en terme de pages fonctionnelles), et aussi vers param.php et index.php en cas de déconnexion.
- la page new\_demande.php : c'est une des pages centrales de l'application puisqu'elle permet à l'utilisateur de faire ses demandes d'impressions et de spécifier leur caractères. Elle renvoi grâce à une barre de navigation aux pages accueil.php, Visual.php et aide.php (en terme de pages fonctionnelles), et aussi vers param.php et index.php en cas de déconnexion.
- la page aide.php : Elle contient les informations relatives à l'application, une sorte de page de documentation permettant à l'utilisateur de comprendre le fonctionnement de l'application. Elle renvoi grâce à une barre de navigation aux pages accueil.php, new\_demande.php, Visual.php(en terme de pages fonctionnelles), et aussi vers index.php en cas de déconnexion.
- la page param.php : Elle contient un formulaire permettant à l'utilisateur de modifier certains de ses paramètres comme son mot de passe et d'afficher les informations de l'utilisateur.

L'interface accueil nous dirige vers la première page du site c'est a dire la page de connexion



FIGURE 3.1 – Représentation organisationnelle de l'application Web.



### 3.1.1 Comportement

Plusieurs méthode comporte un comportement dynamique qui sont :

- La méthode SeConnecter de la page connexion
- La méthode Inscription de la page inscription
- La méthode ModificationMotDePasse de la page modificationMotDePasse
- La méthode OubliMotDePasse de la page OubliMotDePasse
- La méthode SupprimerUtilisateur de la page administrateur
- La méthode ValideUtilisateur de la page administrateur
- La méthode TraiteCommande de la page administrateur
- La méthode ValideCommande de la page administrateur
- La méthode SupprimeCommande de la page administrateur
- La méthode NouvelleComande de la page nouvelle demande

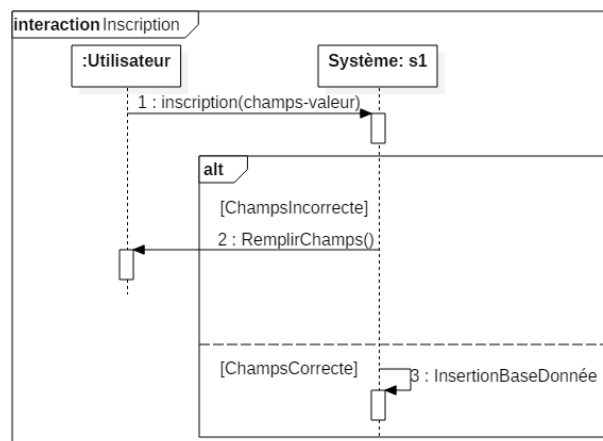
Pour chacune de ses méthode nous allons vous montrer les diagramme d'état transition et

de comportement. Certaines méthodes du diagramme de classe n'apparaissent pas dans le comportement car elles n'ont pas un comportement dynamique.

### Méthode d'inscription

La classe inscription est une classe composée de plusieurs champs permettant de créer un utilisateur dans la base de données lorsque que l'utilisateur instance anonyme va interagir avec la méthode inscription l'utilisateur va être entré dans la base de données et son compte sera en attente de validation (voir classe administrateur).

FIGURE 3.2 – Diagramme de séquence représentant la méthode Inscription.

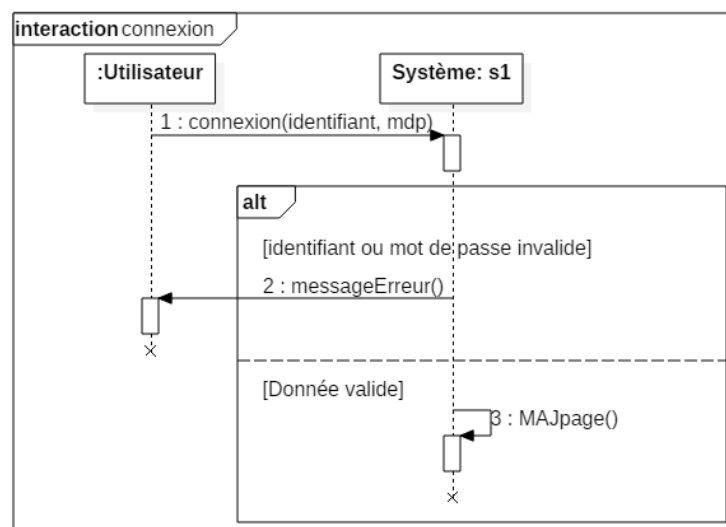


Dans le diagramme ci dessous on peut voir que une instance anonyme de utilisateur envoie un message a une instance s1 qui est la système. Le système va renvoyer un message d'erreur si le champs est incorrecte et si le champs est correct il va s'envoyer un message a lui même pour ajouter l'utilisateur a la base de donnée.

### Méthode connexion

La classe connexion est composer de deux champs un champs identifiant et un champs mot de passe ainsi que une méthode connexion qui va chercher dans la dans la base de donnée si l'identifiant existe et si le mots de passe entré correspond a celui enregistré.

FIGURE 3.3 – Diagramme de séquence représentant la méthode Connexion.

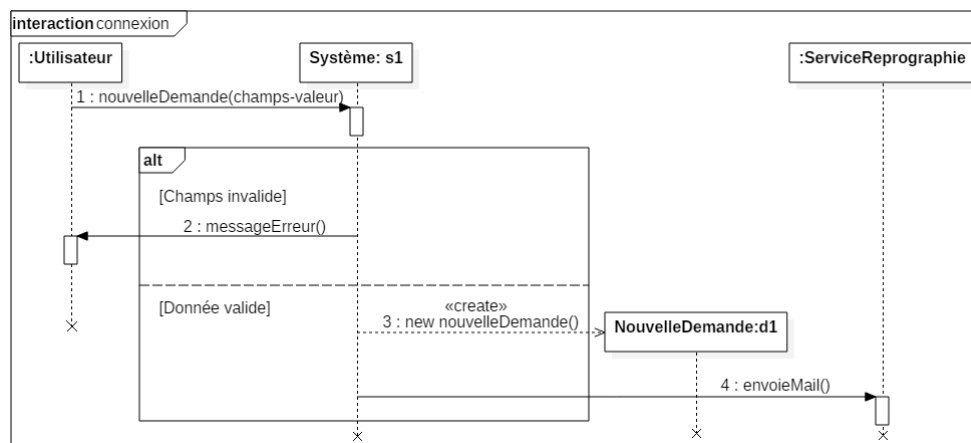


Dans le diagramme ci dessous on peut voir que une instance anonyme de utilisateur envoie un message a une instance s1 qui est la système. Le système va renvoyer un message d'erreur si le donnée envoyer sont invalide et si les donnée envoyer sont correct il va s'envoyer un message a lui même pour ajouter l'utilisateur a la base de donnée.

### Méthode NouvelleDemande

La méthode nouvelle demande permet a un utilisateur de faire une nouvelle demande ainsi après que celui ci ai rempli tout les champs de demander lors de l'interaction final avec la page la demande est crée et inséré dans la base de donnée un mail automatique est envoyer au service de reprographie pour leur informer qu'une nouvelle demande a été transmise et qu'elle est en attente de traitement

FIGURE 3.4 – Diagramme de séquence représentant la méthode NouvelleDemande.

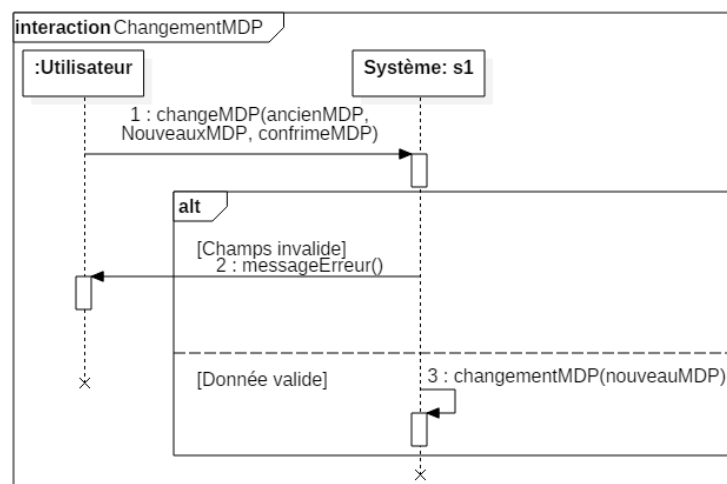


Dans le diagramme suivant une instance anonyme utilisateur envoie un message qui est une demande de nouvelle demande le système reçoit le message si la demande est invalide un message d'erreur est envoyé à l'utilisateur sinon le système crée un objet « d1 » de type nouvelleDemande puis il crée un message envoiMail qui va envoyer un mail au service de reprographie

### Méthode ModificationMotDePasse

Cette méthode se déclenche quand un utilisateur a interagit avec la page modification du mot de passe lorsque que l'utilisateur fait une modification de mot de passe le mot de passe chiffré dans la base de donnée est changer.

FIGURE 3.5 – Diagramme de séquence représentant la méthode ChangementMDP.

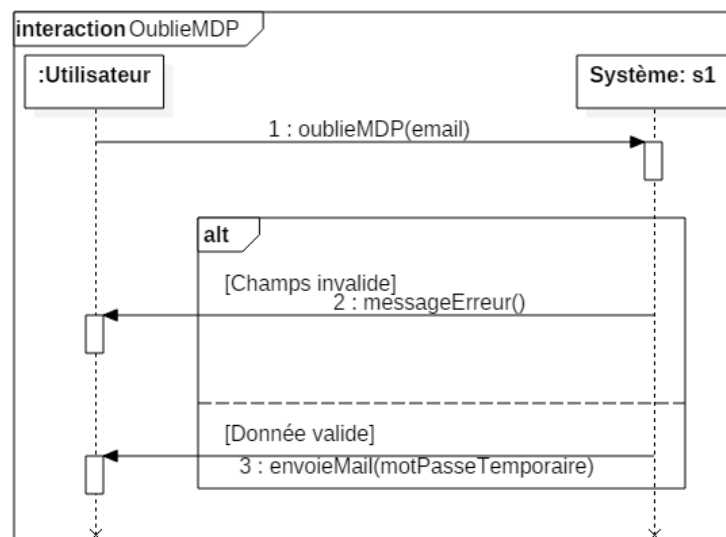


Dans le diagramme ci dessous on peut voir que une instance anonyme de utilisateur envoie un message a une instance s1 qui est la système. Le système va renvoyer un message d'erreur si le donnée envoyer sont invalide et si les donnée envoyer sont correct il va s'envoyer un message a lui même pour ajouter l'utilisateur a la base de donnée.

### Méthode OubliMotDePasse

Cette méthode se déclenche quand un utilisateur a interagit avec la page oubli du mot de passe lorsque que l'utilisateur clique sur le bouton « valider » un mail est envoyer a l'utilisateur avec un mot de passe temporaire le mot de passe est changer par celui qui est temporaire l'utilisateur est invite a se connecter avec le mot de passe temporaire puis il est inviter a changer son mot de passe

FIGURE 3.6 – Diagramme de séquence représentant la méthode OublieMDP.

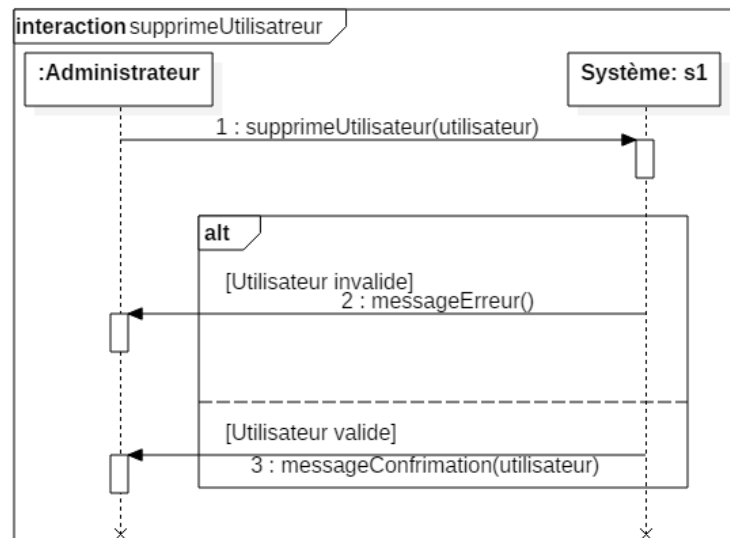


Dans le diagramme ci dessous on peut voir que une instance anonyme de utilisateur envoie un message a une instance s1 qui est la système. Le système va renvoyer un message d'erreur si le donnée envoyer sont invalide et si les donnée envoyer sont correct il va s'envoyer un message a lui même pour ajouter l'utilisateur a la base de donnée.

### Méthode SupprimerUtilisateur

La méthode SupprimeUtilisateur permet a l'utilisateur instance de la classe administrateur de supprimer un autre utilisateur en interagissant avec un bouton de suppression ainsi l'utilisateur est supprimer de la base de donnée

FIGURE 3.7 – Diagramme de séquence représentant la méthode SupprimeUtilisateur.

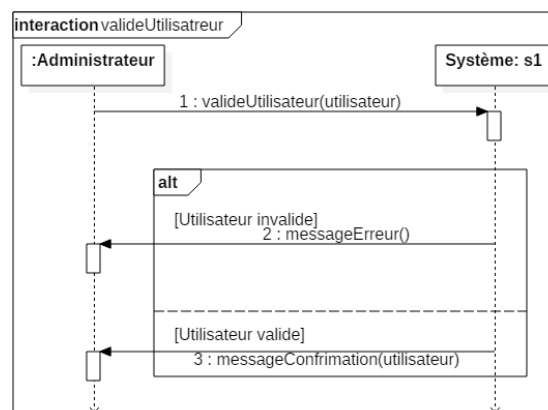


Dans le diagramme ci dessous on peut voir que une instance anonyme de utilisateur envoie un message a une instance s1 qui est la système. Le système va renvoyer un message d'erreur si le donnée envoyer sont invalide et si les donnée envoyer sont correct il va s'envoyer un message a lui même pour ajouter l'utilisateur a la base de donnée.

### Méthode ValideUtilisateur

La méthode `ValideUtilisateur` permet a l'utilisateur instance de la classe administrateur de valider un autre utilisateur en interagissant avec un bouton de validation ainsi l'utilisateur qui était une table de temporaire de en attente de valider est inséré dans la table d'utilisateur non temporaire

FIGURE 3.8 – Diagramme de séquence représentant la méthode ValideUtilisateur.



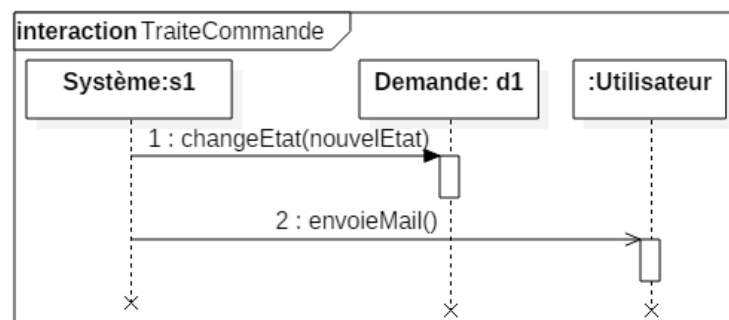
Dans le diagramme ci dessous on peut voir que une instance anonyme de utilisateur envoie un message a une instance s1 qui est la système. Le système va renvoyer un message d'erreur si le donnée envoyer sont invalide et si les donnée envoyer sont correct il va s'envoyer un message a lui même pour ajouter l'utilisateur a la base de donnée.

### Méthode TraiteCommande

#### — Traite commande

La méthode TraiteCommande permet a un utilisateur de la classe service de reprographie de changer l'état de la commande de l'état « en cour » a l'état « à récupéré » ainsi lorsque le changement entre ses deux état a lieu la commande est changer au niveau de le base de donnée. Un mail automatique est envoyer au a l'émetteur de la commande qui lui informe le nouvelle état de la demande.

FIGURE 3.9 – Diagramme de séquence représentant la méthode TraiteCommande.



Dans le diagramme ci dessous on peut voir que une instance de s1 de système envoie un message a une instance d1 de demande. Le système va renvoyer un message a l'instance anonyme utilisateur.

#### — valide commande

La méthode valideCommande permet a un utilisateur de la classe service de reprographie de changer l'état de la commande de l'état « a récupéré » a l'état « fini » ainsi lorsque le changement entre ses deux état a lieu la commande est changer au niveau de le base de donnée.

#### — supprime commande

La méthode SupprimeCommande permet a un utilisateur de la classe service de reprographie de changer l'état de la commande de l'état n'importe quel état a l'état « supprimer » ainsi lorsque le changement entre ses deux état a lieu la commande est supprimer au niveau de le base de donnée. Un mail automatique est envoyer au a l'émetteur de la commande qui lui informe sa demande a été supprimer

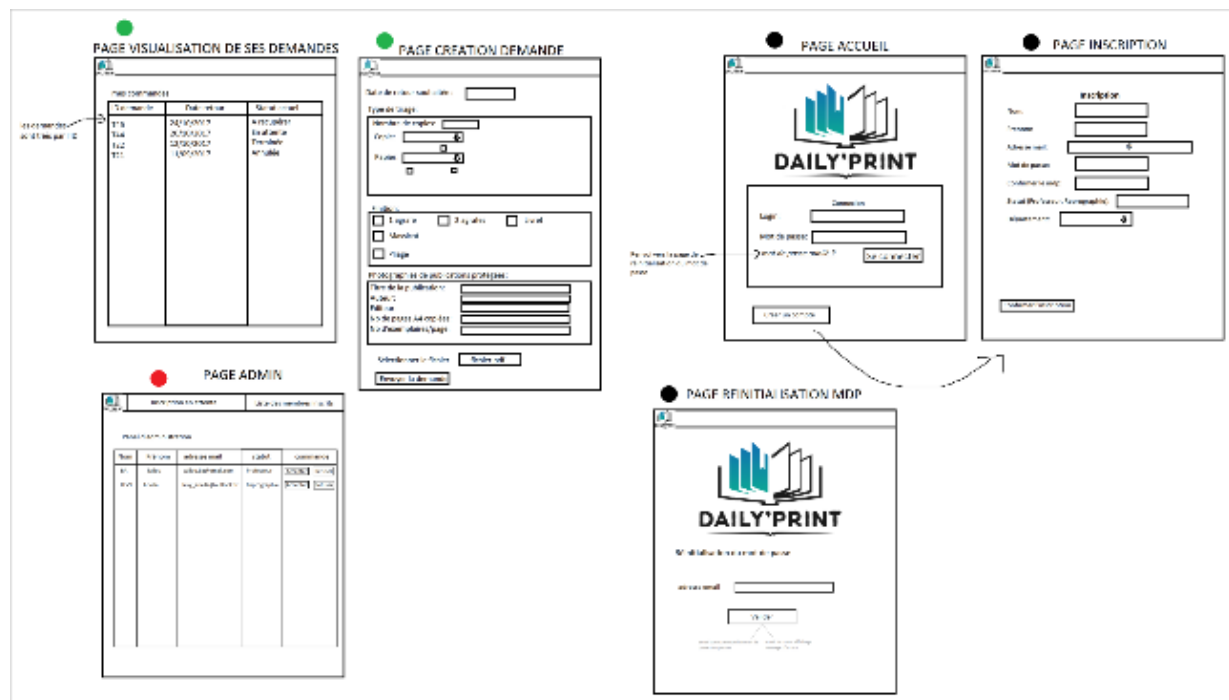
Les méthode valideCommande et Supprime commande fonctionne de la même manière que traiteCommande mais avec des spécifié différentes notamment au niveau de la base de

donnée.

## 3.2 Spécification détaillé des interface utilisateur

### 3.2.1 Prototypé de l'interface

FIGURE 3.10 – Prototypé d'interface.



### Définition de la charte graphique

Logo :

images/logoBis.png

Typographie :

- Normal
- Calibri
- Italique
- gras
- A-B-C-D-E-F-G-H-I-J-K-L-M-N-O-P-Q-R-S-T-U-V-W-X-Y-Z



- a-b-c-d-e-f-g-h-i-j-k-l-m-n-o-p-q-r-s-t-u-v-w-x-y-z
- 0-1-2-3-4-5-6-7-8-9 et les caractère spéciaux

**Couleur :**

- Blanc(code couleur : RGB :255,255,255)
- Vert(code couleur : RGB : 0,238,0)
- Gris/Blanc(code couleur : RGB : 207,207,207)
- Noir(code couleur : RGB : 0,0,0)

**Image issu d'internet :** pas d'image issus d'internet

### 3.2.2 Spécification des rapport et formulaires

Pour spécifier les rapport et formulaire nous avons décider de faire le backend qui correspond a l'arrière plan de notre application c'est a dire la parti qui va géré l'application et les interaction que l'utilisateur va avoir. Le backend sera donc en php et javascript et le front-end qui correspond a la parti frontal du programme c'est a dire le l'interface physique de l'application.

### 3.2.3 Politique et stratégie

1. Mécanismes de stockage pour les données persistantes

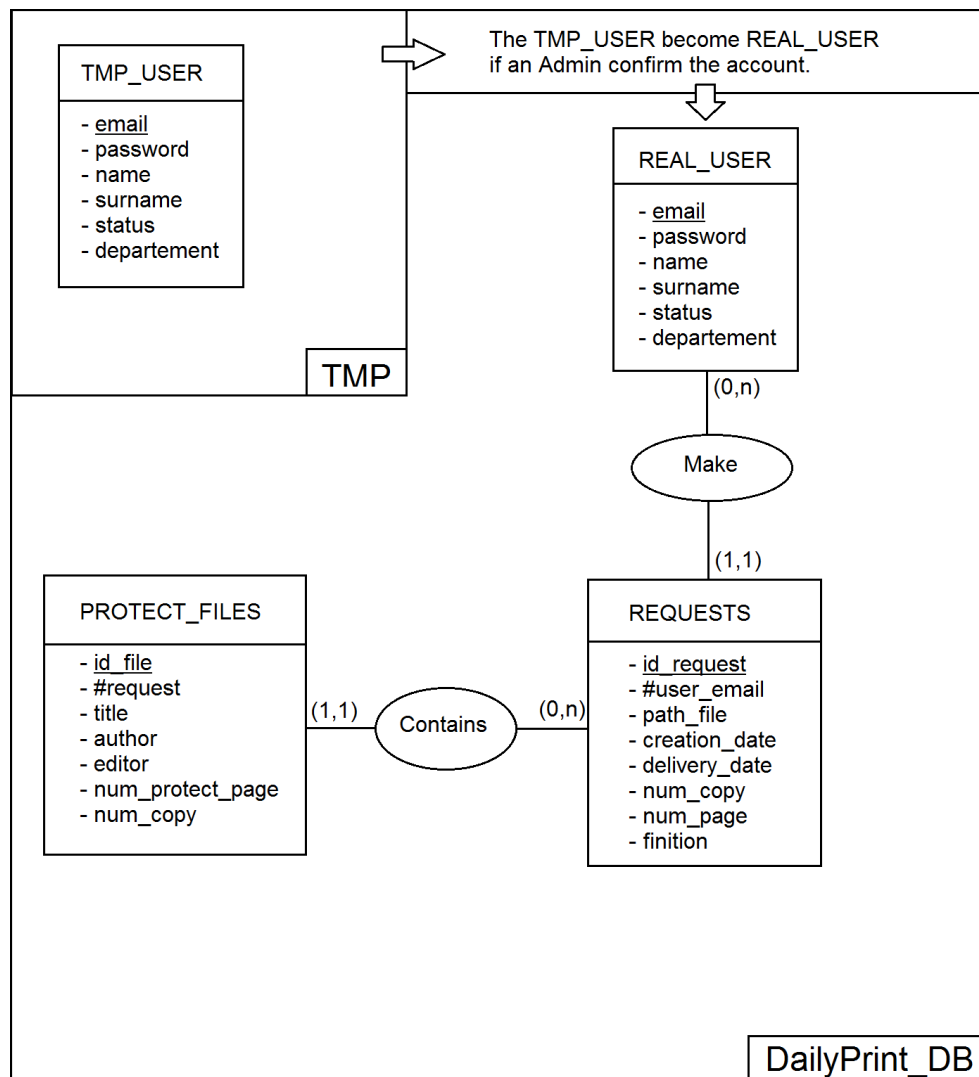
La quasi totalité des applications requièrent une base de données pour le stockage de l'information métier ou comme simple source d'information. Pour cela nous allons vous présenter le schéma de la base de données de notre application pour sa version simplifiée.

voici le schéma de la base de données, nous avons ici 4 tables qui sont :

- TMP\_USER(email, name, surname, password, departement) : ce sont tout les utilisateurs n'ayant pas été validés par l'administrateur.
- REAL\_USER(email, name, surname, password, departement) : ce sont les utilisateurs qui on été validés et peuvent effectuer des commande
- REQUESTS(id\_request, creation\_date, delivery\_date, user\_date, num\_copy, num\_page, finition) : les commandes effectuées par les différents utilisateurs.
- PROTECT\_FILES(title, autor, editor, num\_protect\_page, num\_copy, requests) : ce sont les informations sur les fichiers qui vont passer en reprographie.

Les utilisateurs temporaires deviennent réels quand il sont validés par les administrateurs. Un utilisateur validé peut faire une ou plusieurs commandes, ses fichiers sont protégés.

FIGURE 3.11 – Schéma de la base de donnée.



### 3.2.4 Choix des outil a utilisé

les outils utiliser sont principalement l'ordinateur cependant nous utilisons plusieurs langage de programmation qui sont :

- Langages WEB : **PHP, HTML, CSS et JavaScript**
- Base de données : **MySQL**
- Langages de base de données : **PL/SQL et SQL**
- Support de fonctionnement : **Ordinateurs et mobiles**

De plus, notre site sera héberger sur Hostinger a l'adresse <http://www.dailyprint.xyz/>. Nous utilisons aussi une système de gestion de version qui est GIT(GITHUB)

# Chapitre 4

## Dossier architecture

Ce dossier a pour objectif de faire paraître la structure du système en utilisant plusieurs vues ainsi qu'en décomposant le système en sous-système. Le logiciel doit assurer le rôle d'une caisse enregistreuse d'un supermarché. Il permet d'enregistrer les articles d'un client, établir une procédure de vente, produire un ticket de caisse (manipuler les données) ainsi que de stocker des données relatives aux ventes, la caisse peut également se réinitialiser. Nous allons voir de quoi ce système se constitue en utilisant les vues modulaires et composants connecteurs.

### 4.1 Vue modulaire

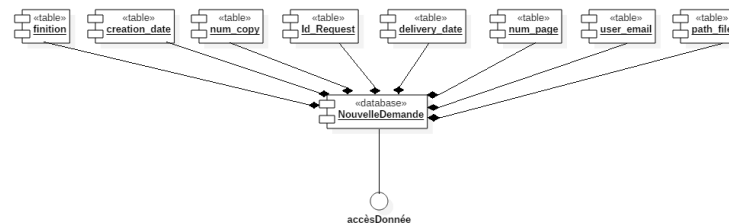
Ce chapitre décrit l'ensemble d'éléments qui composent le système d'un point de vue statique. Il montre également la description des interconnexions entre les composants ainsi que les interfaces offertes par eux. La structure modulaire peut être spécifiée en UML par le biais de diagrammes de paquetages, diagrammes de sous-systèmes et diagrammes de classes. Ce chapitre doit aussi préciser le choix effectué par rapport au style d'architecture (en couches, etc.) Quelques remarques :

- Dans le cas d'un style en couches, le rôle de chaque couche est explicité et la présence de chaque couche est justifiée.
- Pour le diagramme de paquetages ou de sous-systèmes présenter le rôle de chaque sous-système ou paquetage, avec une courte description. Indiquer clairement si le composant est réutilisé tel quel, adapté ou développé. Présenter les classes importantes du paquetage.
- Pour les interfaces, description avec ses opérations

## 4.2 Vue composant/Connecteur

D'un point de vue dynamique, le système utilise une base de données pour recenser toutes les données essentielles à l'application WEB. L'application WEB accède à cette base pour mettre en place le processus de gestion de la reprographie. Elle est constituée des pages et méthodes indiquées dans le schéma de représentation organisationnel, et forme notre exécutable.

FIGURE 4.1 – Diagramme de package représentant la vue de composant/connecteur



## 4.3 Vue d'attribution

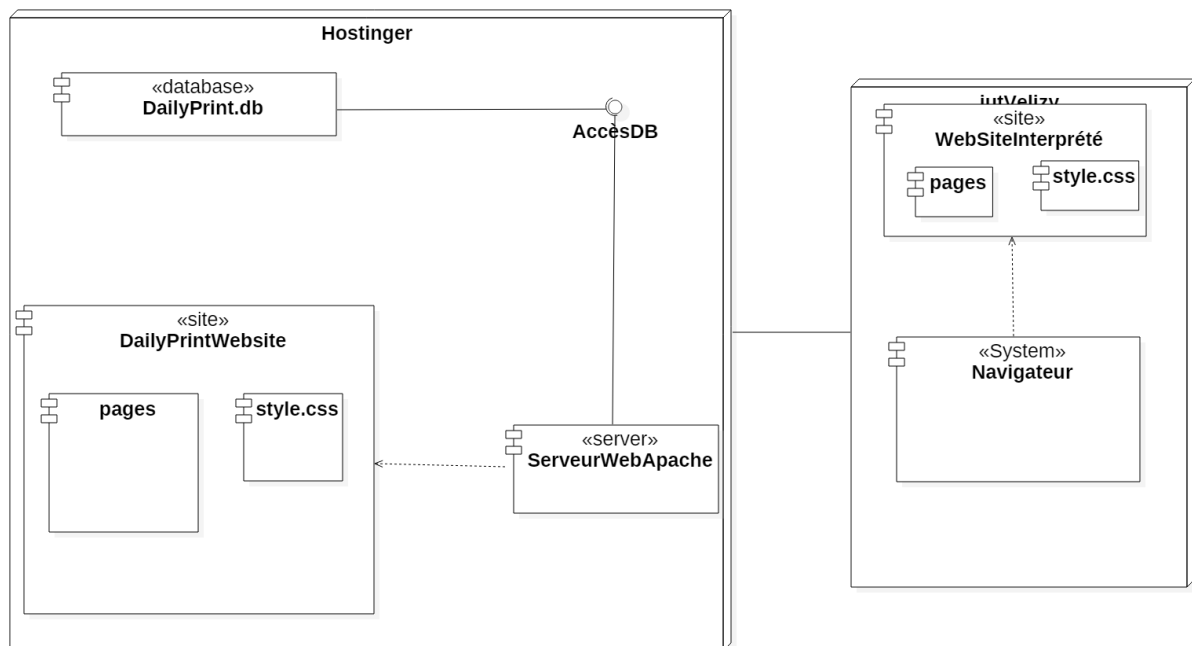
Cette partie contient la description des associations entre les composants logiques et les composants physiques de notre site web. Les structures d'attributions vont permettre de spécifier les composants dans l'infrastructure matérielle. Les associations entre les composants et les nœuds d'exécution sont spécifiées. Chaque nœud et leurs interconnexions sont décrits en UML et leur modélisation peut-être faite via un diagramme de déploiement : Les associations entre les composants et les éléments logiciels physiques (fichiers, tables, bibliothèques, etc..) La spécification en UML peut-être représentée à l'aide d'un diagramme de composant :

Dans notre cas, les éléments seront réarrangés en fonction de leur nœuds d'appartenance :

- La base de données sera hébergée chez Hostinger qui nous fournit un serveur apache
- Le site web non interprété quand-à-lui appartient au nœud des services DailyPrint.
- A contrario le site web interprété appartient au nœud des services de l'IUT de Vélizy car c'est en son sein que sera exécuté le site web.

D'où le diagramme de déploiement suivant :

FIGURE 4.2 – Diagramme de package représentant la vue d'attribution



# Conclusion

Lors de notre projet et la manière dont nous avons abordé nous avons pu voir que pour notre rendu de mi-parcours nous ne pourrions rendre qu'une version minimal. Notre projet comporte donc une évolution possible que ce soit au niveau de ses fonctionnalités. Lors de ce Projet nous avons pu mettre en pratique nos connaissances mais nous avons aussi pu apprendre de nouvelles connaissances. En effet, nous avons pu voir la difficulté d'un projet de cette forme, nous avons aussi appris comment mieux organiser notre travail. Ce projet nous a apporté de la rigueur et nous a permis de mieux aborder la suite des événements qui auront lieu pour la fin du projet. Le plus grand Handicap ou difficulté que nous avons eu est la gestion de temps en effet nous avons du mal à gérer les temps impartis.

# Remerciement

*Nous tenons à remercier le corps enseignant de nous permettre de réaliser ce projet. Nous remercions aussi nos professeurs de nous avoir enseigné les différents enseignements qui nous ont permis de réaliser ce projet. Nous remercions notre tuteur, Monsieur Hoguin pour le projet et l'aide qu'il nous a fourni ainsi que l'IUT de Vélizy-Villacoublay pour nous avoir permis de travailler sur leurs différents outils.*