



**Institut Supérieur des Etudes Technologiques de Jendouba**  
Département des Technologies de l'Informatique

---

# **Guide de Rédaction du rapport de PFE**

---

**Elaboré par : Riadh BOUSLIMI**

***Décembre 2015***



REPUBLIQUE TUNISIENNE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
**Direction Générale des Études Technologiques**  
**Institut Supérieur des Etudes Technologiques de Jendouba**  
Département des Technologies de l'Informatique

## **PROJET DE FIN D'ÉTUDES**

Présenté en vue de l'obtention du  
**Diplôme de Licence Appliquée**  
**en Technologies de l'Informatique**  
**Spécialité : Développement des Systèmes d'Information**

### **TITRE**

**[Votre sujet de PFE]**

***Réalisé par :***

**Prénom NOM**

**Prénom NOM**

***Sous la direction de:***

**Xx. Prénom NOM**

***Au sein de (Organisme d'accueil) :***



**Année universitaire : 20xx/20xx**

# Signatures

## Encadrant entreprise

[Xx. Prénom NOM |

## Encadrant ISET

[Xx. Prénom NOM |

# *Dédicaces*

Les dédicaces du rapport sont généralement destinées à toutes sources de soutien moral (généralement sont des membres de votre famille, ...).

**La mise en forme de cette page est personnelle à l'étudiant.**

# Remerciements

Les remerciements du rapport sont généralement destinés à l'encadreur de projet et à toutes personnes ayant joué un rôle important pendant la période de réalisation de votre projet. Citez le nom, le poste de chaque personne et la justification de votre remerciement.

**Attention : Les remerciements ne sont pas adressés aux membres du jury.**

Ces remerciements sont exprimés en une dizaine de lignes au maximum, **de la façon la plus simple possible, sans platitude ni exagération.**

La mise en forme de cette page est au gré de l'étudiant.

# Table des matières

La table des matières (sommaire) permet, grâce à la pagination, de retrouver l'endroit où se trouve un élément recherché par le lecteur. La table des matières doit être générée d'une façon automatique. Elle ne doit pas présenter plus que trois niveaux de sous-titres.

Pour générer la table des matières, veuillez suivre la procédure suivante :

- 1) Ouvrir le menu "Références" ensuite "Table des matières"
- 2) Choisir un modèle et le niveau désiré.

**Attention :** Pour pouvoir utiliser la table automatique, il faut utiliser les styles prédéfinis pour les titres et sous-titres.

La table des matières doit être claire, pas besoin de plus de 1 ou 3 niveaux de sous-titres (la table des matières ne doit pas dépasser 1 à 3 pages sinon le lecteur ne voit pas la progression des parties).

On donne ici la table des matières de ce guide :

<b>Introduction générale</b> .....	1
<b>Chapitre 1 : Présentation du cadre de projet</b> .....	2
1. Introduction .....	3
2. Présentation de la [société d'accueil] .....	3
3. Etude de l'existant .....	3
3.1. Description de l'existant.....	3
3.2. Critique de l'existant .....	3
3.3. Solution proposée .....	3
4. Choix de modèle de développement.....	3
5. Planning prévisionnel .....	3
6. Conclusion.....	4
<b>Chapitre 2 : Spécification des besoins</b> .....	5
1. Introduction .....	6
2. Spécification des besoins fonctionnels .....	6
2.1. Besoin fonctionnel [global 1] .....	6
2.1.1. [Sous-besoin 1].....	6
2.1.2. [Sous-besoin 2].....	6
2.2. Besoin fonctionnel [global 2] .....	6
2.2.1. [Sous-besoin 1].....	6
2.2.2. [Sous-besoin 2].....	6
3. Spécification des besoins non fonctionnels .....	6
4. Présentation des cas d'utilisation.....	6

4.1.	Présentation des acteurs.....	6
4.2.	Description des cas d'utilisation.....	7
4.3.	Diagramme des cas d'utilisation global.....	7
<b>Chapitre 3 : Conception du système</b> .....		<b>8</b>
1.	Introduction .....	9
2.	Modélisation dynamique .....	9
2.1.	Diagrammes de séquences.....	9
2.2.	Diagrammes de collaboration.....	10
2.3.	Diagrammes d'états .....	10
2.4.	Diagrammes d'état-transition .....	10
2.5.	Diagrammes d'activité.....	10
3.	Modélisation statique .....	11
3.1.	Diagramme de classes .....	11
3.2.	Modèle relationnel.....	11
3.3.	Dictionnaire de données .....	11
3.4.	Architecture de l'application .....	11
3.4.1.	Architecture logiciel .....	11
3.4.2.	Architecture matériel .....	11
4.	Conclusion.....	11
<b>Chapitre 4 : Réalisation du système</b> .....		<b>12</b>
1.	Introduction .....	13
2.	Environnement de développement .....	13
2.1.	Environnement matériel .....	13
2.2.	Environnement logiciel .....	13
3.	Principales interfaces graphiques .....	13
4.	Conclusion.....	13
Conclusion générale .....		14
Bibliographie et Nétographie .....		15
ANNEXES .....		16
ANNEXE A : Que placer en annexes ?.....		17
ANNEXE B : Proposition de mise en forme.....		18
ANNEXE C : Diverses recommandations .....		20

# Liste des figures

Insérer ici la liste des figures qui existent dans le rapport en fixant le nom de chaque figure avec le numéro de sa page.

**Attention :** La liste des figures doit être générée automatiquement.

**NB :** le titre de la figure doit être placé en dessous de la figure.

Exemple de la liste des figures de ce guide :

Figure 1 : Différents messages dans un diagramme de séquences .....	9
Figure 2 : Diagramme de séquences "Authentification" .....	10

# Liste des tableaux

Insérer ici la liste des tableaux qui existent dans le rapport en fixant le nom du tableau avec le numéro de sa page.

**Attention :** La liste des tableaux doit être générée de manière automatique.

**NB :** Le titre du tableau doit être placé au-dessus du tableau.

Exemple de liste des tableaux :

Tableau 1 : Planning prévisionnel .....	4
Tableau 2 : Description du cas d'utilisation « xxx » pour l'acteur x.....	7
Tableau 3 : Dictionnaire de données .....	11

## **Introduction générale**

[Tous le long de ce rapport, l'étudiant doit utiliser le pronom « *nous* » à la place de « *je* »]

L'introduction générale comporte, généralement, deux parties.

Dans la première partie, l'étudiant présentera son sujet à travers des informations précises et posera par la suite la problématique à résoudre avec clarté et sans évocation de résultats.

[Il ne faut pas parachuter des introductions « **passe partout** »]

Dans la deuxième partie, l'étudiant présentera le plan de son rapport en évoquant, brièvement, le contenu de chaque chapitre.

L'étudiant doit impérativement suivre ce guide et doit aussi **respecter** la mise en forme recommandée dans l'ANNEXE B.

[Un étudiant du département Technologies de l'informatique doit développer dans son stage une application pas nécessairement avec un langage de programmation vu dans son parcours]

**Attention** : La numérotation du rapport commence par l'introduction, c'est la **page numéro 1**.

# C

## hapitre

# 1

## Présentation du cadre de projet

### Objectifs du chapitre

L'étudiant évoquera implicitement les différents objectifs de ce chapitre.

Ce chapitre comprend, généralement, trois parties ; la présentation de la société où s'est déroulé le stage, une étude de l'existant sur les modalités de travail actuelles, la critique de l'existant et les solutions envisagées par l'étudiant.

## **1. Introduction**

Insérez ici le texte d'introduction du chapitre et l'objectif à atteindre, elle sert de transition avec le reste du document.

## **2. Présentation de la [société d'accueil]**

Cette partie comprend une **brève** description de la société d'accueil : son secteur d'activité, un bref historique, les différents départements ou services et enfin son organigramme. Il faudrait, surtout, insister sur le service ou département auquel l'étudiant a passé son stage et s'accentuer en outre sur les outils informatiques utilisés.

**Attention :** La présentation de la société n'est pas une publicité pour celle-ci ; il ne s'agit pas de vanter ses mérites ou les services qu'elle offre.

## **3. Etude de l'existant**

Cette partie comprend, généralement, trois parties.

### **3.1. Description de l'existant**

Il est question d'expliquer comment le travail s'effectue, actuellement, au sein de la société (en rapport avec l'application qui va être développée par l'étudiant).

### **3.2. Critique de l'existant**

Cette partie permet d'évoquer les insuffisances de la solution actuelle. En effet, les points faibles de la solution actuelle doivent figurer dans la section suivante et aussi dans l'application qui sera développée par l'étudiant.

### **3.3. Solution proposée**

Deux cas se présentent : soit il y a une application existante qui présente certaines lacunes et, donc, la proposition consiste à apporter des améliorations, soit tout est géré manuellement et il faudrait, donc, informatiser le processus de travail actuel. Dans les deux situations, il faudrait en quelques lignes présenter la ou les propositions possibles (en indiquant ses avantages et ses inconvénients) et justifier le choix de la solution adoptée.

**[L'étudiant insistera ici sur l'originalité de son sujet par rapport aux travaux existants]**

**Remarque :** Il est possible que l'étude de l'existant fasse l'objet d'un chapitre indépendant si l'étudiant n'a pas fait un stage au sein d'une société.

## **4. Choix de modèle de développement**

Ici, l'étudiant doit donner la démarche qu'il va adopter pour développer son application. En outre, le choix de modèle doit être justifié.

Les modèles de cycle de vie de développement d'un logiciel sont vus dans le cours de « Génie logiciel et conduite de projet ».

## **5. Planning prévisionnel**

Ici l'étudiant doit mettre le plan de son travail pendant la période de son stage.

Planning prévisionnel devra être représenté comme suit :

*Tableau 1 : Planning prévisionnel*

Semaine Etape	Février				Mars				Avril				Mai			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
<i>Etude préalable</i>																
<i>Conception</i>																
<i>Réalisation</i>																
<i>Test et Validation</i>																

## 6. Conclusion

Insérez ici le texte de la conclusion du chapitre présentera les différents objectifs déjà cités précédemment.

**Attention :** Chaque chapitre doit comporter une brève introduction et conclusion. La mention des termes « **Introduction** » et « **Conclusion** » n'est pas indispensable.

# C

## h

### ap

#### itre

# 2

# Spécification des besoins

### Objectifs du chapitre

Ce chapitre comprend, généralement, deux parties : les besoins fonctionnels et les besoins non fonctionnels.

## **1. Introduction**

Dans ce chapitre l'étudiant donnera une vision préalable sur le contenu de ce chapitre.

[Si l'étudiant ne fera pas un stage au sein d'une société, il devra contacter son encadrant pour lui aider afin de définir le cahier de charges.]

Lors de l'analyse des besoins, l'étudiant doit faire la distinction entre besoins fonctionnels et besoins non-fonctionnels. Les besoins fonctionnels répondent aux points précis du cahier des charges, et sont donc requis par la société. Ils ne sont pas négociables en général, c'est le "besoin primaire" de la société.

Les besoins non-fonctionnels sont des besoins/contraintes liés à l'implémentation et à l'interopérabilité générale. Ils peuvent être fixés par la société, ou par l'étudiant.

## **2. Spécification des besoins fonctionnels**

Ce sont les besoins indispensables auxquels doit répondre l'application.

Pour plus de clarification, il est recommandé de présenter les besoins en indiquant les besoins globaux puis les détailler.

Pour cela, il est possible d'utiliser les puces ou les numérotations comme suit :

### **2.1. Besoin fonctionnel [global 1]**

[Exemple d'un besoin global : « Gestion des étudiants »]

#### **2.1.1. [Sous-besoin 1]**

[Exemple d'un sous-besoin relatif à la « Gestion des étudiants » on peut mettre : « Ajouter un étudiant »]

#### **2.1.2. [Sous-besoin 2]**

### **2.2. Besoin fonctionnel [global 2]**

#### **2.2.1. [Sous-besoin 1]**

#### **2.2.2. [Sous-besoin 2]**

## **3. Spécification des besoins non fonctionnels**

Ce sont les besoins qui permettraient d'améliorer la qualité des services de l'application comme la convivialité et l'ergonomie des interfaces, l'amélioration du temps de réponse,...

Il est, également, possible de les présenter sous forme de puces.

**Remarque importante** : Les étudiants doivent adopter le langage de modélisation **UML**, le chapitre de spécification des besoins devrait comprendre les diagrammes de cas d'utilisation dont nous détaillons dans la suite la structuration.

## **4. Présentation des cas d'utilisation**

### **4.1. Présentation des acteurs**

Dans ce paragraphe, l'étudiant présentera en bref les différents acteurs de l'application.

#### 4.2. Description des cas d'utilisation

A ce niveau, les cas d'utilisation peuvent être présentés par deux façons que ce soit par acteur, soit par fonctionnalité. Généralement, si les fonctions des acteurs sont complètement indépendantes, c'est la première solution qui est adoptée. Si en revanche, une fonctionnalité du système fait intervenir plusieurs acteurs, c'est la deuxième possibilité qui est adoptée.

Les cas d'utilisation présentant certaines ambiguïtés doivent être complétés par une description textuelle en présentant les points suivants :

*Tableau 2 : Description du cas d'utilisation « xxx » pour l'acteur x*

<b>Cas n°</b>	<i>numéro du cas d'utilisation</i>
<b>Acteur(s) :</b>	<i>Nom(s) de(s) acteur(s)</i>
<b>Objectif :</b>	<i>C'est le but du cas d'utilisation.</i>
<b>Pré-condition(s) :</b>	<i>Condition(s) devant être remplie(s) pour exécuter le cas d'utilisation.</i>
<b>Post-condition(s) :</b>	<i>Condition(s) nécessaire(s) pour que le cas d'utilisation soit considéré comme achevé.</i>
<b>Scénario nominal :</b>	<i>C'est le scénario indiquant les étapes pour réaliser le cas d'utilisation (il ne comprend pas d'alternatives) : il peut être, également, remplacé par un diagramme de séquence.</i>
<b>Scénario alternatif :</b>	<i>C'est le scénario indiquant les éventuelles étapes différentes liées aux choix de l'utilisateur, par exemple.</i>

#### 4.3. Diagramme des cas d'utilisation global

A ce niveau, l'étudiant devra mettre le diagramme de cas d'utilisation global en le dessinant avec le logiciel approprié pour la modélisation conceptuel tel que : *Visual Paradigm, PowerAMC, Rational Rose, StarUML*, etc.



**C** hapitre  
**3**

# Conception du système

## Objectifs du chapitre

Ce chapitre a pour objectif de présenter la solution conceptuelle proposée par l'étudiant.  
En d'autres termes, ce chapitre devrait répondre à la question **COMMENT FAIRE**.

## 1. Introduction

Dans ce chapitre, l'étudiant doit modéliser son application d'un point de vue statique et dynamique. Pour la modélisation dynamique, les diagrammes de séquences, les diagrammes de collaboration et les diagrammes d'états doivent être figurés. Pour modéliser l'aspect statique le diagramme de classes et le diagramme de déploiement doivent être présentés à la fin de ce chapitre.

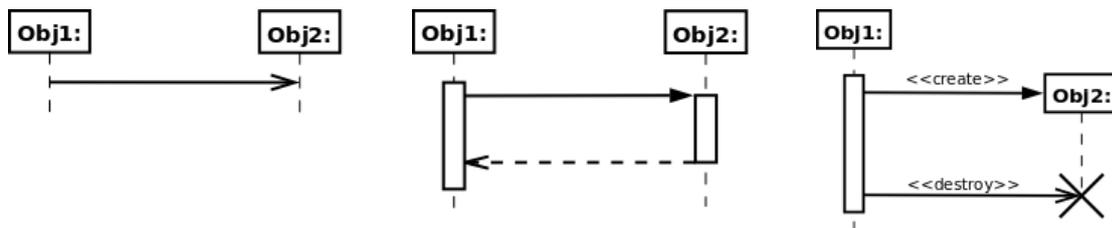
## 2. Modélisation dynamique

### 2.1. Diagrammes de séquences

Un diagramme de séquence est un diagramme d'interaction dont le but est de décrire comment les objets collaborent au cours du temps et quelles responsabilités ils assument. Il décrit un scénario d'un cas d'utilisation [1].

Avec les intéressants ajouts au diagramme de séquences apportés par UML 2, en particulier les cadres d'interactions (avec les opérateurs *loop*, *opt* et *alt* par exemple), ainsi que la possibilité de référencer une interaction décrite par ailleurs, le diagramme de séquence système nous semble constituer une excellente solution [2].

Les différents messages représentés par un diagramme de séquence sont :



**Message asynchrone :**  
Il n'attend pas de réponse et ne bloque pas l'émetteur qui ne sait pas si le message arrivera à destination.

**Message synchrone :**  
L'émetteur reste alors bloqué le temps que dure l'invocation de l'opération.

**Messages de création et destruction d'instance :**  
La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie et la destruction d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet

Figure 1 : Différents messages dans un diagramme de séquences

L'étudiant devra modéliser tous les diagrammes de séquences avec UML2 pour avoir un aperçu comme une « **boite blanche** » et éviter le diagramme de séquence « acteur & système » vu qu'il donne un aperçu « **boite noire** » dans lequel on n'observera pas les interactions entre les objets.

**Exemple d'un diagramme de séquences d'authentification :**

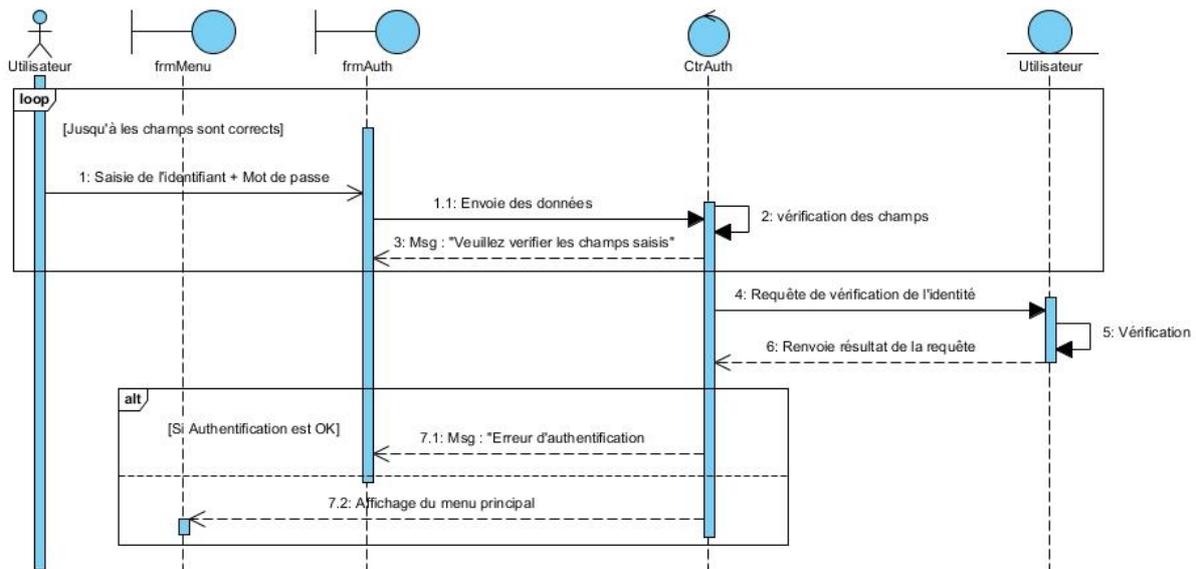


Figure 2 : Diagramme de séquences "Authentification"

**Attention :** l'étudiant doit mettre un seul diagramme de séquences parmi ceux qui se ressemblent d'un point de vu scénario.

**Remarque :** L'étudiant mettre au maximum 10 diagrammes de séquences.

**2.2. Diagrammes de collaboration**

Les diagrammes de collaboration montrent des interactions entre objets (instances de classes et acteurs). En outre, ils permettent de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent [3].

**Remarque :** le diagramme de séquence privilégie l'aspect temporel (cycle de vie des objets) par contre le diagramme de collaboration privilégie l'aspect spatial (objets et liens entre eux).

**2.3. Diagrammes d'états**

**2.4. Diagrammes d'état-transition**

Les diagrammes d'états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets (de type signaux, invocations de méthode) [4].

**2.5. Diagrammes d'activité**

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. [5]

**Attention :** Les diagrammes d'activités sont relativement proches des diagrammes d'états-transitions dans leur présentation, mais leur interprétation est sensiblement différente. En effet

la différence principale entre les deux diagrammes est que les premiers mettent l'accent sur le flot de contrôle d'un objet à l'autre, tandis que les seconds insistent sur le flot de contrôle d'une activité à l'autre.

### 3. Modélisation statique

#### 3.1. Diagramme de classes

Ici sera dressé le diagramme de classes de la future base de données.

**Attention :** les classes doivent comporter les attributs (nom + type de données) et les méthodes.

#### 3.2. Modèle relationnel

Ici sera dressé le modèle relationnel de la base de données en expliquant préalablement les règles de passage.

#### 3.3. Dictionnaire de données

Les colonnes récupérées depuis le model relationnel doivent être représentées comme suit :

*Tableau 3 : Dictionnaire de données*

Nom de la colonne	Type de données	Taille	Obligatoire (Oui/Non)	Valeur par défaut	Valeurs autorisées	Clé		Nom de la Table
						Primaire	Etrangère	

#### 3.4. Architecture de l'application

##### 3.4.1. Architecture logiciel

Ici sera dressé le diagramme de composant qui donnera un aperçu sur l'architecture logiciel de l'application qui a été développée.

##### 3.4.2. Architecture matériel

Ici sera dressé le diagramme de déploiement qui donnera un aperçu sur l'architecture matériel de l'application qui a été développée.

#### Consigne :

- Les ressources matérielles sont représentées sous forme de nœuds.
- Les nœuds sont connectés entre eux, à l'aide d'un support de communication.

### 4. Conclusion

La conclusion donnera un résumé sur ce qui a été présenté dans ce chapitre.

# C

## hapitre

# 4

## Réalisation du système

### Objectifs du chapitre

Ce chapitre a pour objectif de présenter la solution logicielle et l'environnement de développement qui sont utilisés afin d'aboutir à développer l'application.

## **1. Introduction**

Dans ce chapitre, l'étudiant présentera l'environnement matériel et logiciel adopté pour avoir un produit fini et présentera aussi les principales interfaces graphiques. En effet, ce chapitre sera composé par deux parties. La première partie détaillera l'environnement de développement et la seconde partie concernera la mise en œuvre de la solution proposée.

## **2. Environnement de développement**

### **2.1. Environnement matériel**

C'est l'environnement sous lequel l'étudiant a développé son application : les caractéristiques de l'ordinateur telles que la fréquence du processeur, la taille de la mémoire centrale ou s'il s'agit d'une application réseau, les routeurs ou hubs, serveurs,...

### **2.2. Environnement logiciel**

Ce sont les outils logiciels utilisés pour le développement de l'application ou de la base de données, la modélisation des différents diagrammes de conception,...

## **3. Principales interfaces graphiques**

Au niveau de cette rubrique, il faut placer les principales interfaces graphiques développées qui devraient être toutes commentées par un paragraphe de 2 à 3 lignes expliquant son contenu.

A noter qu'il ne faut pas placer toutes les interfaces de l'application, mais uniquement les plus importantes et celles qui seraient différentes. Les autres interfaces sont placées en annexes.

## **4. Conclusion**

[Il ne faut pas l'oublier]

## **Conclusion générale**

La conclusion du rapport doit comprendre, impérativement, un rappel de l'objectif du stage de perfectionnement et une récapitulation du travail fait en présentant les résultats (en d'autres termes, les réponses aux problèmes posés au début).

Il est, également, recommandé de porter un œil critique sur le travail fait en soulevant certaines insuffisances ou améliorations possibles.

**Remarque :** La conclusion devrait être rédigée en une page sous forme d'un paragraphe et non pas de tirets.

## **Bibliographie et Nétographie**

Cette partie comprend les différents livres, articles, revues et sites internet qui ont servi à la documentation.

### **Bibliographie [Obligatoire]**

L'ordre de ces références peut se faire soit par ordre alphabétique du nom de l'auteur soit par ordre d'apparition dans le rapport.

[i] NOM\_AUTEUR, Prénom. « Titre de l'ouvrage », lieu de publication, nom de l'éditeur, année de publication, nombre de tomes, nombre de pages.

S'il s'agit d'un rapport de PFE, par exemple, on peut ajouter le numéro d'ordre (référence) associé. (i= 1, 2, ...,n).

Exemple :

[1] REEVES, Hubert. « Bases de données relationnelles », Paris, Editions du seuil, 1988, p88.

### **Nétographie**

Sites Web visités lors de l'élaboration du projet, avec une brève description du thème consulté (une ou deux lignes au maximum).

Exemple :

[2] <http://www.asp.net/> : Fondements du langage ASP.NET.

### **A ne pas mentionner :**

- Les moteurs de recherche tels que [www.google.fr](http://www.google.fr) ou [www.yahoo.fr](http://www.yahoo.fr)
- Les cours étudiés au niveau de l'ISET ; ils sont considérés comme faisant partie des connaissances acquises et assimilées par les étudiants.

**Remarque :** Il est impératif de **référencer** la bibliographie et nétographie au niveau du rapport.

On donne ici la Bibliographie et la Nétographie de ce guide :

### **Bibliographie**

[2] Pascal Roques, UML 2 par la pratique Etudes de cas et exercices corrigés, 5ème Edition, 2006, p54.

### **Nétographie**

[1] <http://tvaira.free.fr/dev/uml/UML-DiagrammesDeSequence.pdf>

[3] <http://uml.free.fr/cours/p18.html>

[4] <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-etats-transitions>

[5] <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites>

# **ANNEXES**

**ANNEXE A : Que placer en annexes ?**

**ANNEXE B : Proposition de mise en forme**

**ANNEXE C : Diverses recommandations**

[Les annexes sont facultatives et ne suivent pas de règles particulières]

## **ANNEXE A : Que placer en annexes ?**

L'annexe présente un complément de documents qui ne sont pas indispensables à la compréhension du projet, mais qui présentent un certain intérêt. Ces documents peuvent être :

- Des explications plus détaillées liées au thème du projet, à l'environnement de développement,...
- Des documents qui ont servi de base pour le développement de l'application comme des fiches et formulaires remis par la société d'accueil,
- Des interfaces de l'application qui ne figurent pas au niveau de la réalisation,
- Des diagrammes non présentés précédemment,
- Des bouts de code illustrant soit la difficulté de l'implémentation soit l'originalité liée au codage ou au langage de développement,
- ...

## **ANNEXE B : Proposition de mise en forme**

Cette annexe présente différentes recommandations relatives à la mise en forme du rapport.

### **1. Titres et sous-titres**

- Il est recommandé de précéder le titre du chapitre par son numéro (Chapitre 1 : ...),
- Les titres et sous-titres doivent être sur le même niveau vertical,
- On peut distinguer les niveaux de titres et sous-titres par la taille de police,
- A ne pas utiliser « : » à la fin d'un titre ou d'un sous-titre,
- Les titres et sous titres ne sont ni soulignés ni écrits en italique,
- Un titre ou sous-titre ne doit jamais figurer en fin de page.

**Remarque :** Le titre d'un chapitre peut être placé sur une page indépendante ; dans ce cas, la page en question devrait être comptabilisée mais non numérotée et ne devrait comporter ni entête ni pied de page. La page d'après (contenant le corps du chapitre) ne doit porter aucun titre. En d'autres termes, le titre d'un chapitre doit être mentionné une seule fois.

### **2. Corps du texte**

- Justifié,
- 1er ligne : 0.8 cm,
- Interligne : 1.5 ligne,
- Espacement avant et après : 6pts,
- Police : Times New Roman, 12 pts.

### **3. Puces**

- Il faut adopter le même type de puces pour tout le rapport et conserver le même retrait,
- Chaque puce finit par une virgule« , » à l'exception de la dernière qui finit par un point « . ».

### **4. Entête et pied de page**

1. L'entête peut contenir :

- 1.1. Le titre du chapitre courant
- 1.2. Une ligne le séparant du texte de la page

2. Le pied de page peut contenir :

- 2.1. Le numéro de page
- 2.2. Le titre du projet de fin d'études
- 2.3. Une ligne le séparant du texte de la page

**Remarque :** Il n'est pas apprécié de mentionner le nom de l'étudiant ou de la société en entête ou pied de page dans la mesure où elle ne présente aucune plus-value.

## **5. Marges**

2.5 cm (haut, bas, droite, gauche)

## **6. Couleurs**

A éviter sauf en cas de besoin (Interfaces de l'application, ...)

## **7. Numérotation des pages**

- La pagination débute au niveau de l'introduction.
- Les annexes peuvent avoir une numérotation différente du reste du rapport.

## ANNEXE C : Diverses recommandations

1) Les annexes sont facultatives ; elles pourraient, éventuellement, comprendre un complément d'interfaces graphiques qui n'ont pas été mentionnées au niveau du rapport.

Si une partie de la programmation est jugée intéressante ou innovante, il est possible de placer le code source en annexes. De même, certaines notions théoriques pourraient être détaillées au niveau des annexes.

2) Le nombre de pages d'un rapport de PFE (de l'introduction à la conclusion) ne devrait **pas excéder 60 pages** (entre 50 et 60 pages généralement).

**[L'objectif visé est la qualité et non la quantité]**