



*Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Direction Générale des Etudes Technologiques
Institut Supérieur des Etudes Technologiques du Kef
Département Technologies de l'Informatique*

Guide pour la rédaction du rapport de stage de perfectionnement



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Direction Générale des Etudes Technologiques
Institut Supérieur des Etudes Technologiques du Kef
Département Technologies de l'Informatique

Sigle
Société

Code stage :

Soutenu le :

Jury:

RAPPORT DE STAGE DE PERFECTIONNEMENT

Sujet :

**Sujet
du stage**

Organisme d'accueil :

Période : De..... à.....

Elaboré par : prénom NOM

Année Universitaire : 2017/2018

Remerciements

Cette page est personnelle et est consacrée, généralement, à remercier l'encadreur de la société ainsi que les personnes (membres de la société, enseignants, personnel technique ou administratif et non pas les membres du jury) qui auraient aidé l'étudiant à mener à terme son stage de perfectionnement en le conseillant ou en lui fournissant de la documentation. 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.]

Introduction générale	1
Chapitre 1 : Présentation du cadre du stage	2
1. Présentation de la société	2
2. Etude de l'existant	2
2.1. Description de l'existant.....	2
2.2. Critique de l'existant	2
2.3. Solution proposée.....	2
Chapitre 2 : Spécification des besoins	4
1. Besoins fonctionnels	4
2. Besoins non fonctionnels	4
3. Diagrammes de cas d'utilisation	4
3.1. Présentation des acteurs	4
3.2. Description des cas d'utilisation	4
Chapitre 3 : Conception	6
1. Diagramme de classes.....	6
2. Conception de la base de données	6
2.1. Modèle relationnel.....	6
2.2. Schéma relationnel de données	6
Chapitre 4: Réalisation	7
1. Environnement de développement.....	7
2.3. Environnement matériel	7
2.4. Environnement logiciel	7
2. Principales interfaces graphiques.....	7
Conclusion générale	8
Bibliographie et Nétographie	9

ANNEXES.....	10
ANNEXE A : Que placer en annexes ?.....	10
ANNEXE A : Que placer en annexes ?.....	11
ANNEXE B : Proposition de mise en forme.....	12
ANNEXE C : Diverses recommandations.....	14

Liste des figures

Notez que le titre de la figure doit être placé en dessous de la figure.

Liste des tableaux

Notez que le titre du tableau doit être placé au-dessus du tableau.

Introduction générale

L'introduction générale comporte, globalement, deux parties.

La première partie présente le sujet à travers des renseignements précis et pose le problème à résoudre avec clarté sans évocation de résultats.

[Il faut **éviter** impérativement les introductions « **passe partout** »]

La seconde partie énonce le plan du rapport en évoquant, brièvement, le contenu de chaque chapitre.

La suite de ce guide illustre un exemple type de structure de rapport pouvant être adoptée par un étudiant du département Technologies de l'informatique dont le stage a pour objectif de développer une application.

Attention !!

La numérotation du rapport commence par l'introduction, c'est la **pagenuméro 1**.

Chapitre 1 : Présentation du cadre du stage

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

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

1. Présentation de la société

Cette partie comprend une brève description de la société d'accueil : son domaine d'activité, un bref historique (si ça apporte une plus-value au travail), son organisation. Il faudrait, surtout, insister sur l'aspect informatique : ses activités dans ce domaine ; la présentation de son parc informatique est, particulièrement, appréciée.

Il est, également, important d'indiquer le département au sein duquel le stage s'est effectué en précisant sa vocation (développement, maintenance,...)

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.

2. Etude de l'existant

Cette partie comprend, généralement, trois parties.

2.1. 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).

2.2. Critique de l'existant

Cette partie soulève les points forts et faibles de la solution actuelle (en exploitation) en insistant sur les lacunes et les insuffisances de celle-ci.

2.3. 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.

Chapitre 2 : Spécification des besoins

Au niveau de ce chapitre, il faut expliquer en détail ce que l'application est censée faire (QUOI FAIRE) à travers l'inventaire des fonctionnalités de l'application.

Notons qu'il est important de rappeler, au niveau de l'introduction de ce chapitre, l'objectif du projet.

Nous présentons, pour la suite, un exemple type d'une structure possible de ce chapitre.

1. Besoins fonctionnels

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

Par mesure de clarté, il est recommandé de présenter les besoins sous forme WBS (Work Breakdown Structure) ; en d'autres termes, indiquer les besoins globaux puis les détailler.

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

- 1. Besoin global 1
 - 1.1. Sous-besoin 1
 - 1.2. Sous-besoin 2
- 2. Besoin global 2
 - 2.1. Sous-besoin 1
 - 2.2. Sous-besoin 2

2. 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 : Pour les étudiants qui adoptent 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.

3. Diagrammes de cas d'utilisation

3.1. Présentation des acteurs

Au niveau de ce paragraphe, les différents acteurs de l'application sont présentés en bref.

3.2. Description des cas d'utilisation

Il existe globalement deux façons de présenter les cas d'utilisation, soit par acteur, soit par fonctionnalité, les deux sont possibles. 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 (présentée au choix sous forme d'un paragraphe cohérent ou non). Celle-ci comprend, essentiellement, les points suivants :

Objectif : c'est le but du cas d'utilisation.

Pré-condition(s) : Condition(s) devant être remplie(s) pour exécuter le cas d'utilisation.

Enchaînement nominal : 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 deséquence.

Post-condition(s) : Condition(s) nécessaire(s) pour que le cas d'utilisation soit considéré comme achevé.

Il est, également, possible de spécifier d'autres informations telles que les acteurs primaires et secondaires ; tout dépend de la particularité du cas.

Chapitre 3 : Conception

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.

La conception est décrite par un ensemble de diagrammes relevant soit de langageUML.

Notons que pour les sujets de configuration, de paramétrage ou d'intégration, ce chapitre peut être complètement omis.

La structure de ce chapitre dépend de la nature du sujet ; il est, vivement, recommandé de s'adresser au corps enseignant pour tout conseil éventuel.

Nous illustrons, dans la suite, une structuration de ce chapitre dans le cas d'un stage ayant pour objectif de développer une application qui gère une base de données.

1. Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classes est placé au niveau de cette partie. Il est possible d'intégrer la description des classes, à ce niveau.

2. Conception de la base de données

La description de la conception de la base de données s'effectue en plusieurs étapes. L'étudiant adopte la structuration appropriée pour ce chapitre, selon le langage UML.

2.1.Modèle relationnel

Il s'agit de traduire le diagramme de classes en modèle relationnel afin de montrer que l'étudiant traduit, correctement, les classes et associations en tables.

2.2.Schéma relationnel de données

Le schéma relationnel de données se présente sous la forme de liste de relations déduites à partir des entités et des relations du modèle. En outre, il met en exergue les clés primaires et étrangères.

Selon, la spécificité du sujet, la conception peut différer. Il est recommandé à l'étudiant de s'adresser au corps enseignant pour lui porter conseil.

Chapitre 4: Réalisation

Ce chapitre a pour objectif majeur de présenter le « produit fini », c'est-à-dire ce que l'étudiant a développé.

Pour cela, ce chapitre est, généralement, composé de deux parties. La première partie détaille l'environnement de développement. La seconde partie concerne la mise en œuvre de la solution proposée par l'étudiant en présentant les principales interfaces graphiques.

Pour les sujets de stage de configuration ou d'intégration, le déploiement et l'implémentation peuvent être détaillés.

1. Environnement de développement

2.3. 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,...

2.4. 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,...

2. 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.

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, 288p.

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 ou 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.

Il est impératif de **référencer** la bibliographie et nétographie au niveau du rapport !!

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

I. 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.

II. Corps du texte

- Justifié,
- Interligne : 1.5,
- Police : Times New Roman, 12 pts.

III. 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 « . ».

IV. 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.

V. Marges

2.5 cm (haut, bas, droite, gauche)

VI. Couleurs

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

VII. 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 temps à employer au niveau du rapport est impérativement le **présent**,
- 3) Il est préférable d'utiliser l'impersonnel, sinon, le pronom personnel **Nous** même si le stage est effectué par un seul étudiant.
- 4) Tous les chapitres doivent être équilibrés dans la mesure où le nombre de pages devrait être, approximativement, le même.
- 5) Le nombre de pages d'un rapport de stage de perfectionnement (de l'introduction à la conclusion) ne devrait **pas excéder 30 pages** (entre 15 et 30 pages généralement).
L'objectif visé est la qualité et non la quantité.