

Projet de LEE

Master, département informatique, 2ième année

**Gestion d’une médiathèque**

***Elaboré et encadré par :***

M. Michael FORTIER

***Réalisé par :***

Florian CHOISELLE

Marie-Eve PICARD GRAVEL

**ANNEE UNIVERSITAIRE 2016/2017**

[Introduction 3](#_Toc467065975)

[Analyse approfondie 4](#_Toc467065976)

[Présentation du projet 4](#_Toc467065977)

[Analyse des besoins 5](#_Toc467065978)

[Spécifications fonctionnelles 7](#_Toc467065979)

[Fonctionnalités du logiciel 7](#_Toc467065980)

[L'environnement matériel et logiciel 8](#_Toc467065981)

[Conception 9](#_Toc467065982)

[Conception architecturale 9](#_Toc467065983)

[Conception de l'interface utilisateur 10](#_Toc467065984)

[Tests 11](#_Toc467065985)

[Tests de cycle complet 11](#_Toc467065986)

[Tests unitaires 12](#_Toc467065987)

[Planning prévisionnel et réel 13](#_Toc467065988)

[Bilan de compétences 13](#_Toc467065989)

[WBS Global 15](#_Toc467065990)

[Matrice RACI Globale 16](#_Toc467065991)

[PERT Global 17](#_Toc467065992)

[GANTT Global 19](#_Toc467065993)

[PERT des spécifications détaillées 20](#_Toc467065994)

[GANTT des spécifications détaillées 21](#_Toc467065995)

[Mesures d’avancement 22](#_Toc467065996)

[Manuel d’installation 23](#_Toc467065997)

[Liste des fichiers nécessaires à l’exécution du logiciel 23](#_Toc467065998)

[Les fichiers constituant le logiciel 23](#_Toc467065999)

[Manœuvre de compilation 23](#_Toc467066000)

[Manuel d’utilisation 24](#_Toc467066001)

[Procédure pour démarrer le logiciel 24](#_Toc467066002)

[Fonctionnalité 1 24](#_Toc467066003)

[Fonctionnalité 2 24](#_Toc467066004)

[Fonctionnalité 3 24](#_Toc467066005)

[Conclusion 25](#_Toc467066006)

# Introduction

Le présent document est un rapport d'analyse concernant le projet «Gestion d’une médiathèque» qui est un projet d’Environnements et Langages Evolués (LEE) pour les étudiants de deuxième année de master informatique.

Le projet « Gestion d’une médiathèque » à pour sujet la réalisation d’une application Java Enterprise d’une médiathèque permettant la gestion de médias qui peuvent être empruntés dans une médiathèque par des membres inscrits.

Alors, c’est en nous appuyant sur les technologies J2EE et UML ainsi que nos connaissances en gestion de projet, que nous devons proposer une solution complète à ce sujet.

Ce document a donc pour but de présenter l'ensemble des étapes préparatoires nécessaires au lancement de notre projet, afin de décrire entièrement le projet que nous saurions réaliser.

Ainsi, dans ce document il sera question de présenter d'abord l'analyse approfondie de notre projet à travers une analyse des besoins. Puis nous présenterons son analyse fonctionnelle à l'aide d'une liste détaillée de ses fonctionnalités. Ensuite, nous exposeront l'architecture logicielle qui devrait nous aider à concrétiser ce projet en vue de son implémentation.



# Analyse approfondie

Cette analyse permet de déterminer le but du logiciel à développer, ceci à travers une analyse du sujet et des besoins de manière à formuler les besoins de l'utilisateur ainsi que le contexte d'utilisation du logiciel. A cette fin, il est important de présenter d'abord le sujet de notre projet ainsi que les problèmes inhérent à celui-ci.

## Présentation du projet

### L'énoncé

« *L’objectif de ce projet est de mettre en place une application Java Enterprise de gestion d’une médiathèque.*

*Il s’agit de gérer des médias (livres, CD audio, DVD vidéos...) qui peuvent être empruntés par des membres inscrits de la bibliothèque.*

*L’application devra permettre, entre autres, de :*

*— de lister la disponibilité des médias*

*— de lister et gérer les membres inscrits à la médiathèque*

*— de permettre à des individus d’emprunter et de rendre des médias*

*— de gérer l’inventaire des médias*

*Votre application doit utiliser les librairies et framework vues en cours :*

*— JPA*

*— EJB*

*— Servlet, JSP*

*— JSTL*

*Remarque : pensez aux commentaires dans votre code, ainsi qu’à vos tests unitaires.* »

### Problèmes inhérent à l'énoncé :

L’énoncé étant assez peu détaillée, il nous laisse une grande liberté concernant ce que l’application peut permettre de faire. Il faut alors faire attention à ne pas se perdre en fioritures ni perdre de vue les principaux objectifs de ce projet.

## Analyse des besoins

Après une discussion animée en salle de TP, il était possible de définir les besoins de notre logiciel.

### Objectif du logiciel

|  |  |
| --- | --- |
| Objectif du logiciel | L'objectif du logiciel est de permettre la gestion de médias qui peuvent être empruntés dans une médiathèque par des membres inscrits. |
| Utilisateurs | - Personne  - Adhérent  - Bibliothécaire |
| Contexte d'utilisation | Il y a des interactions entre le système et ses utilisateurs lorsque :  - Une personne souhaite devenir adhérent  - Une personne souhaite consulter la liste des médias disponibles  - Un adhérent souhaite gérer son compte  \* Il peut modifier des informations le concernant  \* Il peut décider de se désinscrire  - Un adhérent souhaite emprunter un média  \* Il peut emprunter seulement si sa limite d’emprunt n’est pas dépassée  \* Il peut emprunter seulement s’il n’a pas de médias en retard  - Un adhérent souhaite rendre un média  \* Il peut rendre un média à tout moment  \* S’il rend un média après un retard, il peut à nouveau emprunter  - Un bibliothécaire souhaite accéder à la gestion des inscriptions  \* Il peut obtenir la liste des adhérents  \* Il peut ajouter un adhérent en validant une demande  \* Il peut faire une demande en tant que personne pour une personne  \* Il peut supprimer un adhérent  \* Il peut modifier les informations d’un adhérent  - Un bibliothécaire souhaite accéder à la gestion des prêts  \* Il peut enregistrer un emprunt  \* Il peut enregistrer une restitution  - Un bibliothécaire souhaite accéder à la gestion de l’inventaire  \* Il peut obtenir la liste des médias enregistrés dans la base de données  \* Il peut ajouter un média dans la base de données  \* Il peut supprimer un média de la base de données |

### Diagramme de cas d'utilisation

Pour exprimer les besoins d'un client de manière simplifiée, nous avons eu envie de modéliser un diagramme UML intitulé *diagramme de cas d'utilisation*.

<<primary>>

<<primary>>

<<Include >>

<<Extend >>

<<Extend >>

<<Include >>

<<secondary>>

<<secondary>>

<<primary>>

<<Extend >>

<<Extend >>

<<Extend >>

<<Extend >>

<<Extend >>

<<Extend >>

<<Extend >>

<<Include >>

<<primary>>

<<primary>>

<<Extend >>

<<primary>>

<<primary>>

Médiathèque

Adhérent



<<Include >>

Bibliothécaire



Personne



<<Extend >>

Dans ce diagramme il est possible d'observer ce que permet de faire le logiciel de la médiathèque pour ses utilisateurs. Une personne ne peut qu’adhérer ou consulter la liste des médias disponibles. Un bibliothécaire ou un adhérent sont des personnes spéciales. Elles ont les droits d’une personne plus d’autres droits. Les liens « extends » montrent les possibilités d’un cas d’utilisation, tandis que les liens « includent » montrent une dépendance entre des cas d’utilisation. Ainsi, par exemple, *Emprunter un média* entraine toujours le cas d’utilisation *Gestion des prêts*. La notion de retard ne figure pas dans le schéma mais un message peut être envoyé automatiquement par le système lorsqu’un média n’a pas été rendu à la date prévue.

# Spécifications fonctionnelles

## Fonctionnalités du logiciel

En nous basant sur l'analyse des besoins, il nous a été possible de spécifier les fonctionnalités de notre logiciel, c'est-à-dire, de déterminer précisément ce que le logiciel permet de faire.

**Adhérer**

La fonctionnalité qui permet de … nécessite que …

**Consulter la liste des médias disponibles**

La fonctionnalité qui permet de … nécessite que …

**Gérer son compte**

La fonctionnalité qui permet de … nécessite que …

**Emprunter un média**

La fonctionnalité qui permet de … nécessite que …

**Rendre un média**

La fonctionnalité qui permet de … nécessite que …

**Gestion des inscriptions**

La fonctionnalité qui permet de … nécessite que …

**Gestion des prêts**

La fonctionnalité qui permet de … nécessite que …

**Gestion de l’inventaire**

La fonctionnalité qui permet de … nécessite que …

## L'environnement matériel et logiciel

Ce logiciel doit être réalisé en J2EE. Les librairies et Framework tels que JPA, EJB, Servlet, JSP, JSTL que nous avons vu en cours et TPs sont à utiliser.

La réalisation de ce logiciel aura lieu à travers l'utilisation de l’IDE (interface de développement intégrée) Netbeans car cela facilite l'organisation d'un projet au niveau de l'implémentation. Nous utiliserons également Framagit pour le partage des fichiers de manière à pouvoir avancer simultanément sur différentes tâches du projet et voir où en est l’autre puis pouvoir mieux contrôler le versionning de nos fichiers.

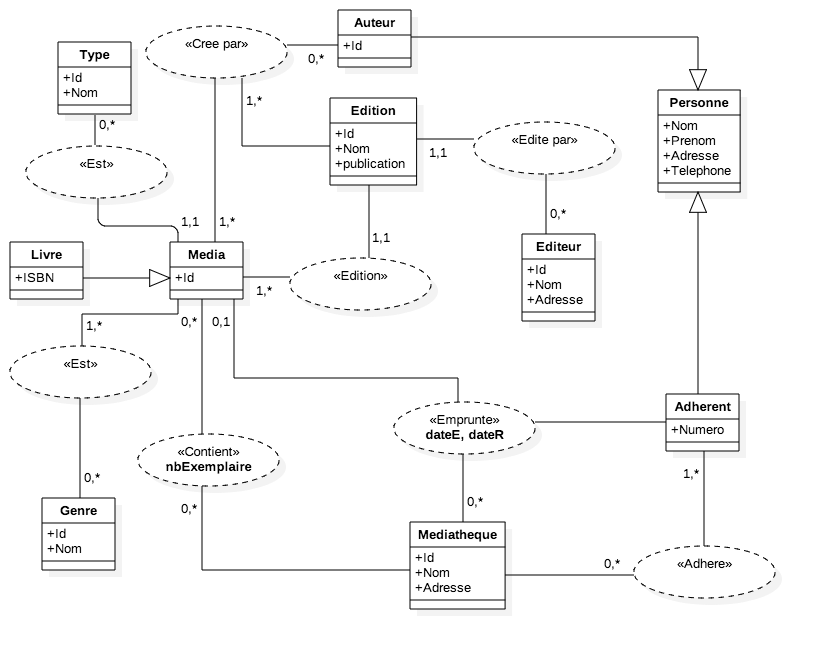
Quant au logiciel de gestion d’une médiathèque que nous implémenteront, il contiendra au moins […] classes, dont une pour […], une pour […[…]].

# Conception

## Conception architecturale

### Diagramme de classe

[…].



## Conception de l'interface utilisateur

### Version préliminaire de l'interface utilisateur

[…].

Note : L’écran de modification d’un compte adhérant permet la visualisation des informations à propos de ce compte.

# Tests

## Tests de cycle complet

Nous avons effectué nos tests de cycles complets en ayant à l’esprit que le but était de vérifier qu’il serait possible pour un utilisateur d’effectuer une action correspondant à un cas d’utilisation souhaité du début à la fin. C'est-à-dire que le but de ces tests est, dans notre cas, de vérifier qu’un utilisateur peut, par exemple, demander à adhérer à la médiathèque. Il fallait alors tester notre logiciel, fonctionnalité par fonctionnalité.

Si le résultat obtenu est « OK » alors l’élément à tester réalise bien le rôle pour lequel il a été défini, sinon, le résultat obtenu est « KO ».

### Adhérer

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Consulter la liste des médias disponibles

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Gérer son compte

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Emprunter un média

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Rendre un média

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Gestion des inscriptions

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Résultat attendu | Résultat obtenu |
| … |  | OK | … |
| … |  | OK | … |

### Gestion des prêts

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Gestion de l’inventaire

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

## Tests unitaires

Nous avons effectué nos tests unitaires à l'aide des tableaux de validation suivant, qui sont classés par fonctionnalité de notre logiciel. Ces tableaux présentent chaque élément à tester, par fonctionnalité et selon l’affichage souhaité, en comparant les résultats attendus à ceux obtenus.

Si le résultat obtenu est « OK » alors l’élément à tester réalise bien le rôle pour lequel il a été défini, sinon, le résultat obtenu est « KO ».

### Adhérer

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Consulter la liste des médias disponibles

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Gérer son compte

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Emprunter un média

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Rendre un média

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Gestion des inscriptions

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Résultat attendu | Résultat obtenu |
| … |  | OK | … |
| … |  | OK | … |

### Gestion des prêts

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

### Gestion de l’inventaire

|  |  |  |
| --- | --- | --- |
|  | Résultat attendu | Résultat obtenu |
| … | OK | … |
| … | OK | … |

# Planning prévisionnel et réel

Nous avons choisi d'organiser notre planning prévisionnel et réel en fonction des qualités de chacun. Pour ce faire, nous avons d'abord effectué un bilan des compétences, puis nous avons organisé les tâches et distribué celles-ci selon nos qualités respectives tout en veillant à ce que chacun soit impliqué en tout temps dans le travail des autres.

## Bilan de compétences

Ainsi, nous avons tout d'abord procédé à une analyse des compétences de chacun afin de préparer au mieux l'organisation et le partage des différentes tâches à effectuer relatives à notre projet, tant au niveau de l'implémentation de notre logiciel que de la rédaction des divers documents. Cette étape nous a permis d'évaluer les forces et les faiblesses de chacun de manière à être capable de profiter des premières tout en tentant de combler les secondes. Nous pouvions ainsi nous entraider de manière réfléchie.

Ce bilan de compétence a alors été pensé et rédigé en équipe, en prenant soin de mettre en avant des compétences techniques et organisationnelles telles que présentées dans le tableau suivant :

|  |  |  |
| --- | --- | --- |
| Numéros | Compétences | Sous compétences |
| 1 | Communication & Rédaction | Capacités de communication orale et écrite  Maîtrise de la langue française  Maîtrise des outils de traitement de texte |
| 2 | Gestion de Projet & Organisation | Expérience en tant que chef de projet  Sens de l'organisation, minutie, rigueur, etc. |
| 3 | Conception & Modélisation | Capacités d'abstraction, d'imagination, de conception  Connaissances en modélisation |
| 4 | Implémentation | Facilités à programmer des algorithmes solides  Compétences en Java & J2EE |
| 5 | Compétences d'Analyse | Analyse des besoins ou des problèmes, réflexion  Analyse documentaire |
| 6 | Design d'interface WEB | Sens de l'esthétique et compétences techniques |
| 7 | Conception de Tests | Capacités d'appréhensions fortes |

Nous avons ensuite pu classer ces compétences selon deux catégories, d'après le niveau d'aptitude de chacun tel que l'on peut l'observer dans le tableau ci-dessous :

|  |  |  |
| --- | --- | --- |
| Noms | Compétences Normales | Compétences Fortes |
| Florian CHOISELLE | 1, 2, 7 | 3, 4, 5, 6 |
| Marie-Eve PICARD GRAVEL | 4, 5, 6 | 1, 2, 3, 7 |

Le bilan de compétences terminé, il a ensuite été possible de déterminer le rôle de chacun, car les tâches devaient être partagées de manière équitable.

Etant donné les compétences, points forts et points faibles de chacun, voici les rôles qui ont été donnés, respectivement :

* Florian CHOISELLE

**Concepteur développeur logiciel**

Activités :

- Analyse des besoins

- Responsable de la conception architecturale

- Programmation des diverses fonctionnalités du logiciel

- Réalisation des tests de cycle complet

- Conception des tests unitaires

- Conception du manuel d’installation

- Analyser les problèmes techniques et proposer des solutions

* Marie-Eve PICARD GRAVEL

**Analyste concepteur**

Activités :

- Analyse des besoins

- Rédaction du cahier des charges en précisant les solutions techniques

- Modélisation des traitements et des données

- Conception des tests de cycle complet

- Réalisation des tests unitaires

- Conception du manuel utilisateur

- Analyser les problèmes techniques et proposer des solutions

## WBS Global

Dans cet organigramme de type « WBS » (*Work Breakdown Structure*), il est possible d’observer que le projet est découpé en deux grandes parties, selon un cycle de vie itératif. En effet, il y a d'un côté les tâches liées à la création du logiciel, et de l'autre celles liées à la rédaction des rapports.

Gestion d’une médiathèque

1000

Logiciel

1100

Mémoire

1200

**0**

**1**

**2**

**3**

1110

Interface WEB

1120

Tests

Module Central

1130

1210

Tests

1220

Guide utilisateur

Rapport d’analyse

1230

Conception

Programmation

Tâches GL

Conception

Programmation

Tâches GL

Conception

Programmation

Tâches GL

Conception

Rédaction

Tâches GL

Conception

Rédaction

Tâches GL

Conception

Rédaction

Tâches GL

1111

1112

1113

1121

1122

1123

1131

1132

1133

1211

1212

1213

1221

1222

1223

1231

1232

1233

## Matrice RACI Globale

La matrice RACI globale de notre projet n’entre pas dans le détail des itérations mais présente le rôle majeur d’un membre de l’équipe sur une tâche, même si d’autres rôles ont pu être joués.

Ainsi, celui qui a le rôle de «Réalisateur» sur une tâche réalise l'action, celui qui a le rôle «Autorité» sur une tâche peut aussi réaliser l'action mais il est surtout responsable de l'avancement de la tâche. Celui qui a le rôle «Consulté» sur une tâche doit être consulté pendant l'avancement de la tâche et finalement, celui dont le rôle sur une tâche est «Informé» est surtout informé de l'avancement de la tâche et des résultats obtenus qui peuvent lui servir dans l'avancement des tâches sur lesquelles il joue un rôle différent puisqu'il existe des dépendances entre diverses tâches.

|  |  |  |
| --- | --- | --- |
|  | Ressources | |
| Tâches | Florian CHOISELLE | Marie-Eve |
| Logiciel | | |
| Module Central | | |
| Conception | A | R |
| Programmation | A | C |
| Tâches de Génie Logiciel | C | A |
| Module Interface WEB | | |
| Conception | A | C |
| Programmation | A | C |
| Tâches de Génie Logiciel | C | A |
| Tests | | |
| Conception | A | R |
| Programmation | A | C |
| Tâches de Génie Logiciel | C | A |
| Mémoire | | |
| Rapport d'Analyse | | |
| Conception | R | A |
| Rédaction | C | A |
| Tâches de Génie Logiciel | C | A |
| Tests | | |
| Conception | A | R |
| Rédaction | A | R |
| Tâches de Génie Logiciel | C | A |
| Guide Utilisateur | | |
| Conception | R | A |
| Rédaction | C | A |
| Tâches de Génie Logiciel | C | A |

## PERT Global

Avant de pouvoir modéliser un diagramme PERT pour l'organisation des différentes tâches selon les délais de ces tâches, il faut avoir estimé ces délais. Nous avons donc tenté d’effectuer ces estimations selon nos expériences antérieures dans ce type de projet.

Voilà donc ces tâches et leurs estimations en heures-hommes :

|  |  |  |  |
| --- | --- | --- | --- |
| Tâche | Nom | Antériorités | Durée  (heures-hommes) |
| A | Analyse du sujet | - | … |
| B | Ecriture des spécifications générales | A | … |
| C | Ecriture des spécifications détaillées | B | … |
| D | Gestion et organisation du projet | - | … |
| E | Implémentation des spécifications détaillées | M | … |
| F | Ecriture des tests unitaires | E | … |
| G | Ecriture des tests de cycles complets | C | … |
| H | Lancement des tests unitaires | F | … |
| I | Lancement des tests de cycles complets | G | … |
| J | Rédaction du mémoire du projet | - | … |
| K | Rédaction du guide utilisateur | E | … |
| L | Rédaction du cahier de tests | H, I | … |
| M | Rédaction du rapport d’analyse | A, B, C | … |

L’analyse du sujet, l’écriture des spécifications générales et détaillées qui dépendent respectivement les unes des autres donnent lieu à la rédaction du rapport d’analyse.

La gestion et l’organisation du projet donne lieu à des schémas de génie logiciel qui seront utilisés dans les divers documents mais comme tout sera fait au fur et à mesure de l’avancement du projet, cette tâche ne dépend d’aucune autre et aucune tache précise ne dépend de celle-ci.

L’implémentation des spécifications détaillées dépend de l’écriture de celles-ci et plus globalement, de la rédaction du rapport d’analyse. Nous détaillerons les sous-tâches inhérentes à cette tâche-ci dans les pages qui suivent.

L’écriture des tests unitaires (tests qui portent sur le fonctionnement des petites choses comme vérifier qu’un bouton joue bien son rôle) dépend de l’implémentation des spécifications détaillées alors que l’écriture des tests de cycles complets (tests qui portent sur le bon fonctionnement du programme selon un cas d’utilisation donné, du début à sa fin) dépend uniquement de l’écriture des spécifications détaillées. Ces écritures donnent lieu à la rédaction du cahier de tests qui sera complété au fur et à mesure lors du lancement des tests qui dépendent respectivement de l’écriture de ceux-ci.

Voici le diagramme PERT qu'il est possible d'élaborer à partir de l’estimation des tâches précédemment présentées :

L(00)

M(00)

H(00)

F(00)

G(0)

E(000)

A(0)

B(00)

C(00)

0 0

0 0

00 00

00 00

000 000

I(0)

00 000

00 000

00 000

000 000

000 000

D(00)

0 000

00 000

J(00)

0 000

00 000

60 60

K(16)

000 000

000 000

000 000

Le diagramme PERT Global précédent représente les itérations de notre projet, qu'il faut effectuer l'une après l'autre. Mais en réalité, la première itération consiste surtout en une analyse du projet dans sa globalité et c'est cette analyse qu'il faut terminer pour pouvoir procéder au découpage des tâches pour ensuite procéder à l'exécution de l'itération suivante. Le projet ne pouvant être terminé qu’après l’exécution de toutes ces itérations.

## GANTT Global

Une fois ce PERT Global terminé, il a été possible de créer un diagramme GANTT global (par rapport à nos dates au plus tôt), voici donc ce diagramme :

DIAGRAMME

En observant ce diagramme Gantt il est possible de remarquer que les seules tâches qui ne font pas partie du chemin critique et peuvent ainsi profiter d'une marge de temps supplémentaires sont les tâches […] qui correspondent respectivement à […]. Sur toutes les autres tâches, il faudra donc veiller à ne pas prendre de retard.

## PERT des spécifications détaillées

Voici les sous-tâches pour l’implémentation des spécifications détaillées, les dépendances qui existent entre elles ainsi que leurs estimations en heures-hommes :

|  |  |  |  |
| --- | --- | --- | --- |
| Tâche | Nom | Antériorités | Durée  (heures-hommes) |
| A | … | … | … |
| B | … | … | … |
| C | … | … | … |
| D | … | … | … |
| E | … | … | … |
| F | … | … | … |
| G | … | …. | … |
| H | … | … | … |
| I | … | … | … |

Voilà le diagramme PERT qu'il est possible d'élaborer à partir de l’estimation des tâches précédemment présentées :

F(00)

G(00)

I(00)

H(00)

E(00)

A(0)

B(0)

C(00)

0 0

0 0

00 00

00 00

00 00

00 00

00 00

D(0)

00 00

00 00

00 00

0 0

00 00

## GANTT des spécifications détaillées

Voici le diagramme de GANTT pour l’implémentation des spécifications détaillées :

DIAGRAMME

Il est possible de faire des observations similaires sur ce diagramme de GANTT à celles faites pour le diagramme de GANTT global. Ici, les tâches sur lesquelles il sera possible de prendre de la marge sont les tâches […] qui correspondent respectivement à […].

Sur toutes les autres tâches, il faudra donc veiller à ne pas prendre de retard. Pour ce faire, nous ferons régulièrement des tableaux avec mesures d’avancement.

## Mesures d’avancement

Les diagrammes précédents sont globaux et se basent sur des données plus détaillés (récoltées dans des tables Excel) dont nous ne ferons pas figurer les diagrammes ici. Toutefois, ce qu'il est important de savoir c'est que nos calculs pour les mesures d'avancement qui permettent de comparer les temps effectifs aux temps estimés sur chaque tâche sont issus de documents plus précis et dont nous avons veillé à ce qu'ils soient bien élaborés pour permettre un vrai suivi pour l'ensemble du projet.

Voilà donc les mesures d'avancement pour notre projet global, tirées des diagrammes précédents, de nos estimations globales et des dates que nous avons estimées en partant de la date de rendu de projet :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tâche | Lettres | Travail Estimé | Travail Réel | Fin Estimée |
| Analyse du sujet | A |  |  |  |
| Ecriture des spécifications générales | B |  |  |  |
| Ecriture des spécifications détaillées | C |  |  |  |
| Gestion et organisation du projet | D |  |  |  |
| Implémentation des spécifications détaillées | E |  |  |  |
| Ecriture des tests unitaires | F |  |  |  |
| Ecriture des tests de cycles complets | G |  |  |  |
| Lancement des tests unitaires | H |  |  |  |
| Lancement des tests de cycles complets | I |  |  |  |
| Rédaction du mémoire du projet | J |  |  |  |
| Rédaction du guide utilisateur | K |  |  |  |
| Rédaction du cahier de tests | L |  |  |  |
| Rédaction du rapport d’analyse | M |  |  |  |
| Total | |  |  |  |

Comme il est possible de le constater dans le tableau ci-dessus, nous avons dû accuser un retard qui s'est répercuté sur l'ensemble des tâches de notre projet et avons ainsi eu à terminer un bon nombre d'entre-elles le même jour. Ce jour étant la date limite de remise des différents éléments qui constituent l'ensemble de notre projet.

Il est également possible de constater que nous avons eu à travailler plus que prévu. Mais cela n'est pas étonnant puisque nous sommes encore novices et que les estimations concernant les coûts en heures-hommes des diverses tâches se font en général en fonction de l'expérience de ceux qui estiment ces coûts. Du moins, pour un petit projet c'est la méthode préconisée.

Dans tous les cas, cela nous aura servi d'expérience pour mieux estimer les coûts pour des tâches de même nature à l'avenir.

# Manuel d’installation

## Liste des fichiers nécessaires à l’exécution du logiciel

-

-

-

## Les fichiers constituant le logiciel

-

-

-

## Manœuvre de compilation

[…].

# Manuel d’utilisation

## Procédure pour démarrer le logiciel

[…].

## Fonctionnalité 1

[…].

## Fonctionnalité 2

[…].

## Fonctionnalité 3

[…].

# Conclusion

En conclusion, notre projet de LEE nous a permis de développer notre esprit d'équipe tout en nous donnant l'opportunité de tester nos compétences en J2EE ainsi que pour l'analyse préalable à la réalisation d'un projet, notamment nos compétences en UML et en génie logiciel. En effet, nous avons eu à travailler l'un après l'autre sur certaines parties de notre projet, et à améliorer le travail de l'autre avec nos connaissances respectives. Ainsi, nous avons eu à nous investir en tout temps dans le travail de l'autre pour mener à bien notre projet. Nous avons également pris connaissances de difficultés comme celles de […], car nous n'avions jamais […]. Nous nous sommes donc rendu compte que nous avions encore beaucoup à apprendre. Néanmoins, nous avons compris qu'il est important de commencer par analyser un projet de façon globale, si l'on souhaite réfléchir à toutes les éventualités qui s'offrent à nous mais aussi aux problèmes possibles à rencontrer lors de l'implémentation.

Les connaissances que nous avons pu acquérir au cours de notre cursus en informatique sur le campus de l’université Paris13-Nord (IUT, Licence3, Master) nous ont alors été fort utiles, et nous avons pu mettre en application de nombreux aspects de notre formation, aussi bien au niveau méthodologique de technique.

Bien que nous ayons déjà eu le plaisir d'effectuer des projets de ce type, c'est-à-dire avec un thème et un planning à respecter, dont le but est la mise en pratique de nos connaissances mais aussi le dépassement de celles-ci pour apprendre quelque chose à nouveau, c'est toujours un plaisir de tenter de relever le défi et de se rendre compte que l'on apprends un peu plus à chaque fois. Cela est très satisfaisant.

Nous espérons avoir fait les choses comme il le fallait, mais pensons néanmoins pouvoir affirmer que ce projet s’est assez bien déroulé et a été très enrichissant pour chacun de nous.