

# RAPPORT DE PROJET

*Conseils méthodologiques  
et aide à la rédaction du  
**rapport de projet tuteuré et  
du rapport de stage.***





# Structuration du rapport

- 1 – Les avant-textes
- 2 – Le texte principal
- 3 – Les post-textes

# Structuration du rapport de projet

## 1 - Les avant-textes

- ❑ **La page de couverture**
- ❑ **Les remerciements** : au minimum le tuteur pour le projet, et le tuteur de stage en entreprise, ainsi que le tuteur à l'IUT qui fait la visite de stage.
- ❑ **Le résumé en français et en anglais** d'environ 100 mots et des mots clés.
- ❑ **Le sommaire** (3 niveaux de hiérarchie maximum)
- ❑ **La table des figures**
- ❑ **Le glossaire** : termes techniques propres au sujet du rapport ou terme spécifique au domaine d'application du projet

# Structuration du rapport de projet

## 2 - Le texte principal

### Introduction

- ❑ Présentation du projet (contexte, objectifs) et du plan du rapport
- ❑ Lecteur : un lecteur quelconque extérieur au projet (rapporteur)

### 1 – Cahier des charges (ou « Analyse »)

- ❑ Analyse du contexte, analyse de l'existant, des besoins fonctionnels et non-fonctionnels
- ❑ Lecteur : le commanditaire (un non-informaticien)

### 2 – Rapport technique

- ❑ Développement du programme : de la conception aux résultats
- ❑ Lecteur : informaticien du même niveau que vous, équipe de développement qui pourrait poursuivre votre projet
- ❑ Style : technique et professionnel ; temps du présent (descriptif)

### 3 – Résultats

- ❑ Test et validation
- ❑ Manuel d'installation
- ❑ Manuel d'utilisation

### 4 – Rapport d'activité

- ❑ Planification et organisation du travail ; recul sur le travail effectué.
- ❑ Lecteurs : enseignants de l'IUT, membres du jury
- ❑ Style : personnel (marque de 1<sup>ère</sup> personne possible), temps du passé (temps du bilan).

**Conclusion** : synthèse et bilan.

# Structuration du rapport de projet

## 3 – *Les post-textes*

- ▣ **Les références bibliographiques et sitographiques**
- ▣ **Les annexes techniques**
  - ▣ Dossier d'analyse.
  - ▣ Dossier de programmation : extraits de code trop long pour être évoqués dans le corps du rapport, diagrammes techniques...
  - ▣ Tout autre document permettant d'assurer la maintenance de l'application ou de comprendre le projet.



# Méthodologie du rapport de projet

*Conseils méthodologiques  
détaillés, partie par partie.*

# La page de couverture

**Logos de l'IUT et de l'Université de Montpellier**

**Prénom et NOM des auteurs du rapport**

→ Nom en majuscule

→ Respecter l'ordre Prénom NOM.

**Prénom et NOM du / des tuteurs du projet**

**Type de rapport** « Rapport de projet » « Rapport de stage ».

**Titre du rapport** : ne pas hésiter à donner un titre long afin qu'il soit bien précis et qu'il rende compte de l'ensemble du projet (projet tuteuré ou projet de stage).

**Illustration facultative**

**Année universitaire – Diplôme**

UNIVERSITÉ DE MONTPELLIER

I.U.T.  
MONTPELLIER - SETE

---

RAPPORT DE PROJET

[Titre du rapport = titre du projet]

<Illustration facultative>

Réalisé par  
Prénom NOM des auteurs

Sous la direction de  
Prénom NOM du / des tuteur(s)

Pour l'obtention du [diplôme]  
Année universitaire 20XX-20XX

# Résumé et mots clés

## □ Résumé :

- Environ 100 mots.
- Présentation très descriptive du projet et des technologies utilisées pour le mettre en œuvre.

## □ **Mots clés** : 5-7 mots clés sur le fond du sujet et sur les technologies utilisées.

## □ **Résumé et mots clés en anglais** : ne pas les bâcler.

*Le projet du site des horaires décalés est un site intranet dont le but est de permettre aux membres du personnel de l'Institut de Génomique Fonctionnelle (IGF) de déclarer leur présence quand ils souhaitent travailler en dehors des heures d'ouverture des bâtiments, ceci dans le but d'améliorer la sécurité de l'IGF et des employés. Le site doit permettre un accès et des fonctionnalités adaptés aux quatre types d'utilisateurs identifiés : un membre du personnel, un gestionnaire, un administrateur et un gardien.*

*Le site WEB a été développé dans le framework Zend Framework (PHP). Il utilise la bibliothèque JQuery (JavaScript) et le serveur de base de données MySQL.*

**Mots clés** : IGF, site intranet, horaires décalés, Zend Framework, JQuery, JavaScript.



# Résumé et mots clés

## ❑ Contre-exemple

- ❑ Ne pas préciser le contexte dans lequel s'inscrit ce projet (projet tuteuré ou stage) : aller directement à l'essentiel en présentant le projet.
- ❑ Ne pas annoncer les différentes parties du rapport (cahier, des charges, rapport technique, manuel d'utilisateur...)
- ❑ Ne rien mettre de personnel ou de subjectif : le résumé est entièrement descriptif.

## Résumé

Ce rapport concerne le déroulement d'un stage de dix semaines effectué à Sherpa Technologies en 2012 pour l'Année Spéciale Informatique de l'IUT de Montpellier. Plus précisément, il explique la mise en place du projet Imagygame, de sa base de données MySQL à son développement en HTML, CSS, PHP, JavaScript et SQL, mais aussi le stage en lui-même. Il se compose donc de documents professionnels tels que la cahier des charges, le rapport technique et le manuel utilisateur ainsi que de documents plus pédagogiques comme la présentation de l'entreprise et la rapport d'activité.

Cette expérience est plutôt riche car des aspects essentiels du projet ont été confiés presque exclusivement au stagiaire. On retrouve donc l'application des méthodes enseignées à l'IUT et leur adaptation dans un vrai projet d'entreprise.

# Sommaire

- ❑ **Insérer un sommaire automatique** : Penser à l'actualiser afin que les numéros de pages et les titres correspondent bien au contenu réel.
- ❑ **Numérotation**
  - ❑ **Utiliser une numérotation de type 1.1.1** pour que le lecteur sache immédiatement dans quelles parties ou sous-parties il se situe.
  - ❑ **Faire apparaître la numérotation** dans les titres des parties du sommaire.
- ❑ **Une ou deux pages maximum**
  - ❑ **Trois de niveau hiérarchie maximum** : 1. / 1.1 / 1.1.1
  - ❑ Soigner la mise en page (espace avant les paragraphes) pour que le sommaire donne au lecteur une **vision d'ensemble** du plan du rapport.
- ❑ **Ne pas numéroter l'introduction et la conclusion** (à moins qu'elles ne comportent des sous-parties et comportent plusieurs pages).
- ❑ **Faire débiter le sommaire à l'introduction** : ne pas inclure les avant-textes (remerciements, glossaires...)

# Table des figures

- ❑ **Insérer une table des figures automatiques** : Penser à l'actualiser afin que les noms des figures et les numéros des pages correspondent bien au contenu réel.
- ❑ **Utiliser l'outil d'insertion de légende des traitements de texte** : cf. document sur les traitements de texte.

## Table des figures

Figure 1 - Use case Annonceur et /ou exploitant.....	2
Figure 2 - Diagramme d'activité d'upload de fichier.....	3
Figure 3 - Espace membre.....	4
Figure 4 - Cycle de développement itératif.....	5

# Glossaire

- ❑ Le glossaire doit contenir des **termes techniques d'informatique** qui ne seraient pas compris par n'importe quel informaticien de niveau bac +2.

**Regex** : expression rationnelle, appelée parfois "motif". Décrit un ensemble de caractères possibles à l'aide d'une syntaxe spécifique. L'application d'une expression rationnelle sur une chaîne de caractères initiale permet d'en extraire les caractères qui respectent l'expression.

**Servlet** : classe Java qui permet de créer dynamiquement des données au sein d'un serveur web.

**SHA1** : Secure Hash Algorithm, fonction de hachage cryptographique. À partir d'une chaîne de caractères, elle produit un résultat numérique ayant une forte probabilité d'être unique. Son résultat est souvent appelé "hash".

~~**Classe** : Représente une catégorie d'objets, en définissant des attributs et des fonctions communs à ces objets.~~

~~**Objet** : Structure de données chargée de stocker, classer, et traiter des données. C'est l'instanciation d'une classe.~~

- ❑ Indiquez dans le glossaire les **termes techniques propres au domaine ou à l'activité** sur lesquels porte votre projet ou votre stage.

**Cultivar** : un cultivar est une variété de plantes (arbres compris) obtenue en culture, généralement par sélection, pour ses caractéristiques « réputées uniques ». Il peut s'agir de qualités esthétiques, techniques, de vitesse de croissance (pour les arbres par exemple), d'adaptation à un biotope ou de résistance à certaines maladies.

**Curateur** : gestionnaire d'une collection. Il prend soin de l'ensemble des plantes de sa collection, l'organise et la documente de façon cohérente.

- ❑ Les mots contenus dans le glossaire doivent être suivis d'un **astérisque\*** dans le corps du rapport pour inviter le lecteur à consulter le glossaire.

# Introduction

- ❑ **Présentation générale du contexte**
- ❑ **Problématique** : poser un problème que le projet devra résoudre. Montrer que le projet répond à un enjeu, un besoin.
- ❑ **Présentation du projet** en lui-même : objectifs, fonctionnalités principales.
- ❑ **Annonce du plan** :
  - ❑ Utiliser des connecteurs logiques qui ponctuent la phrase.
  - ❑ Essayer d'ajouter des précisions pour l'annonce de chacune des parties afin que l'annonce du plan ne soit pas interchangeable d'un rapport à l'autre.
  - ❑ Inutile d'annoncer la conclusion dès l'introduction (permet d'alléger l'annonce du plan).

***Un paragraphe pour chacune des parties de l'introduction.***  
*Mais la problématique peut faire l'objet d'un paragraphe à part ou bien être rattachée à la présentation générale du contexte ou à la présentation du projet.*

# Introduction - exemple

## Présentation générale du contexte

Le département Informatique de l'IUT de Montpellier possède un réseau interne (Intranet) permettant à l'administration de communiquer des informations et de mettre à disposition des fichiers à destination des étudiants. Au fil des années, cet Intranet s'est développé en accueillant de nouvelles fonctionnalités. Il permet aujourd'hui la gestion complète des projets tuteurés des étudiants, allant de la création des binômes jusqu'à l'affectation des sujets et à la réalisation du planning des soutenances.

## Problématique

Ce système, bien que fonctionnel, ne permet toutefois pas au responsable des projets d'obtenir une vue de l'ensemble des sujets disponibles à un instant donné, ni même de partager ces informations ou de les récupérer afin de pouvoir travailler dessus sans être connecté.

## Présentation du projet

Le projet qui nous a été confié consiste donc à proposer et implémenter des solutions permettant de répondre à ce besoin.

## Annonce de plan

Après avoir mis au point dans le cahier des charges l'ensemble des besoins fonctionnels et des spécifications techniques, nous présenterons dans le rapport technique les choix de conception que nous avons faits pour développer une nouvelle version du site, dont nous exposerons les résultats de manière critique. Vous trouverez également dans ce document un manuel d'utilisation à destination de l'administrateur du site Intranet de l'IUT, ainsi qu'un rapport d'activité rendant compte des méthodes de travail que nous avons utilisées pour conduire ce projet.

# « Cahier des charges » ou « Analyse »

## ▣ Analyse du sujet et de son contexte

- ▣ Analyse de l'existant (quels autres logiciels similaires , algorithmes, etc.. existent déjà ? )
- ▣ Analyse de l'environnement dans lequel le logiciel va être utilisé (système d'information de l'entreprise par exemple).

## ▣ Analyse des besoins fonctionnels (besoins utilisateurs)

- ▣ Spécifications fonctionnelles avec des maquettes si nécessaire. La formalisation de votre problème dans un langage mathématique fait partie de l'analyse.
- ▣ Diagrammes d'analyse utiles à la définition des besoins (UML, Merise...)
  - ▣ Cas d'utilisation (Use case)
  - ▣ Diagramme de séquences
  - ▣ Diagramme d'activités

## ▣ Analyse des besoins non-fonctionnels

- ▣ Spécifications techniques
  - ▣ Toutes les contraintes techniques imposées par le commanditaire (ex : langage de programmation).
  - ▣ Licence, normes, fiabilité, performance
- ▣ Autres contraintes : ergonomiques, juridiques



# Rapport technique

## Le contenu

### ▣ Conception

L'objectif de cette section est de donner une vue logique du logiciel à produire. Vous devez identifier toutes les entités du domaine, les processus qui vont agir sur ces entités et les règles d'interaction

- ▣ Présentation et justification des choix technologiques :
  - ▣ Lister les technologies utilisées pour le programme en soulignant la pertinence de leur utilisation (que ce soit une contrainte du cahier des charges ou un choix de votre part)
  - ▣ Présenter de manière synthétique les outils qui ne sont pas connus de tout informaticien de niveau bac +2.
- ▣ Diagrammes UML de programmation (de classes, d'objets, de séquence, de collaboration, d'état-transition, de composants)
- ▣ Description des algorithmes

### ▣ Réalisation

L'objectif de section est de présenter l'architecture interne du logiciel réalisé en termes de composants ainsi l'architecture de déploiement.

- ▣ Description de l'architecture du programme (ex : MVC ,MAVEN).



# Rapport technique

## Le contenu

- ▣ Diagramme de composant et de déploiement
- ▣ Découpage en fonctions
- ▣ Arborescence des fichiers du programme

### ▣ Validation , résultats et perspectives

L'objectif de cette section est de montrer comment le logiciel réalisé a été validé et testé dans son environnement de fonctionnement.

- ▣ Examen des résultats obtenus par rapport aux objectifs initiaux (diagramme de déploiement).
- ▣ Suites possibles de développement.

**Attention à bien identifier et distinguer ce qui relève des résultats du rapport technique** (rédigé de manière impersonnelle, sous la forme d'un bilan très descriptif de ce qui fonctionne et ce qui n'a pas été développé, ou ce qui pourrait être développé, mais qui ne figurait pas dans le cahier des charges), **ce qui relève de la gestion de projet** (prise de recul sur l'organisation et ce qui a permis ou non de tenir les délais et les objectifs du cahier des charges initial) **et enfin ce qui relève du manuel d'utilisateur.**

# Rapport technique

## Conseils de rédaction

### □ Adopter un style très professionnel et technique

- **Bannir les marques de première personne.**
- **Bannir les temps du passé**, car il ne faut pas raconter la manière dont vous avez développé, mais expliquer comment fonctionne et comment a été conçu votre programme.

### □ Analyser en détail les diagrammes et les extraits de codes choisis.

- **Choix des extraits de code** : toute partie du code qui mérite d'être explicité par son importance ou sa difficulté.
- **Numéroter et donner un nom aux figures**, aux diagrammes et aux extraits de code. Indiquer dans le cours de la lecture à quel moment il faut regarder le diagramme / la figure : (cf. figure 1) ou « Comme l'indique la figure 1... »
- **Placer les diagrammes ou extraits de codes d'une page ou plus en annexe** ; préférer des extraits de code courts et bien commentés.
- **Analyser et commenter en détail les diagrammes et les extraits de code** : tout diagramme et toute figure doivent être accompagnés d'explications textuelles.

# Exemple de code bien commenté

## ► Configuration de la web application WebCgrdAdmin

Que nous utilisions Spring ou pas, notre module est une application web qui a vocation à être publiée au travers d'Internet par un conteneur de Servlet, ce qui implique donc que nous ayons besoin de configurer ce conteneur à l'aide d'un fichier /WEB-INF/web.xml. L'intégralité de ce fichier de configuration se trouve en annexe 3 ; nous allons ici nous consacrer à une configuration qui utilise Spring :

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        /WEB-INF/applicationContext.xml ①
        /WEB-INF/applicationContext-security.xml ①
    </param-value>
</context-param>

<servlet> ②
    <servlet-name>SpringMvcDispatchingServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping> ③
    <servlet-name>SpringMvcDispatchingServlet</servlet-name>
    <url-pattern>*.do</url-pattern>
</servlet-mapping>

<servlet-mapping> ③
    <servlet-name>SpringMvcDispatchingServlet</servlet-name>
    <url-pattern>*.json</url-pattern>
</servlet-mapping>

<servlet-mapping> ③
    <servlet-name>SpringMvcDispatchingServlet</servlet-name>
    <url-pattern>*.text</url-pattern>
</servlet-mapping>
```

① : Ces deux instructions permettent de délocaliser la configuration de Spring dans d'autres fichiers.

Le premier fichier, applicationContext.xml, définit notamment les objets dont on a besoin à l'intérieur de notre application et quels sont les paramètres pour créer ces objets. Par le principe d'inversion de contrôle que nous avons vu plus tôt, Spring donnera accès à ces objets. Le second fichier, applicationContext-security.xml, permet de délimiter quelles sont les règles de sécurité qu'il y a au sein de l'application, comment les utilisateurs sont authentifiés, que faire lorsqu'un visiteur n'est pas authentifié. Je reviendrai sur ces deux fichiers plus tard.

② : Ici est déclaré le Servlet principal de Spring. Pour rappel, Spring a un rôle d'aiguillage entre les pages et les classes appelées par le navigateur client. C'est au travers de ce Servlet que le travail d'aiguillage qui est celui du conteneur de Servlet est délégué à Spring.

③ : Comme nous l'avons vu dans le chapitre sur JavaEE, l'élément Servlet-mapping permet de définir un masque d'URL qui, lorsqu'il est vérifié, redirige vers le Servlet déclaré pour l'élément Servlet-name. Donc ici, chaque fois que le navigateur client essaiera d'accéder à une URL se terminant par .do, .json ou .text, la requête du client sera transmise à la classe org.springframework.web.servlet.DispatcherServlet, déclarée plus tôt en Servlet.

# Exemple de schéma bien conçu et bien commenté

Le schéma ici aide à la présentation d'un choix technologique, celle d'une bibliothèque de fonction.

L'explication en détail de son fonctionnement est pertinente, car ce n'est pas un outil étudié au cours de la formation.

## C.3.2. Technologie utilisée : Google Web Toolkit

Google Web Toolkit est un outil puissant qui permet aux développeurs de concevoir et réaliser des applications Web qui fonctionnent sur la grande majorité des navigateurs. Ces applications étant en JavaScript sur la partie cliente, elles sont également adaptées aux terminaux mobiles. La possibilité d'utiliser une architecture client/serveur permet aux développeurs de créer des applications complexes et reliées à des systèmes de stockage tout en gardant une interface riche et légère.

Observons le principe de fonctionnement de GWT :

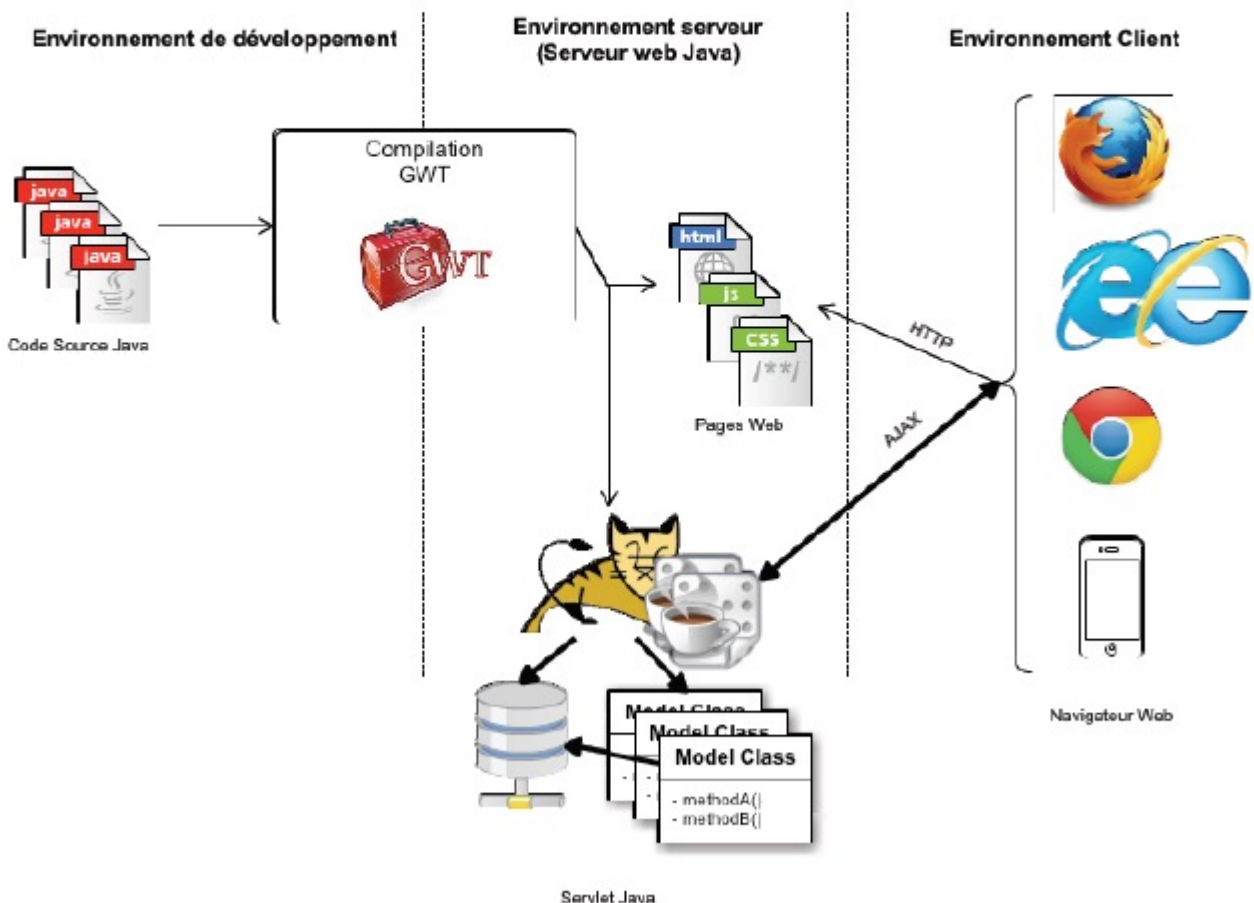


Figure 18 : Cycle de développement d'une application avec GWT



Le schéma ne se substitue pas à l'explication qui est détaillée et structurée.

GWT est d'abord un ensemble de bibliothèques qui permettent aux développeurs de réaliser une interface graphique (éléments graphiques, interactions utilisateurs et événement métier) d'une façon très proche de ce qu'il faudrait faire avec Swing dans une application Java locale. Sans jamais se soucier de connaître du code HTML, JavaScript ou CSS.

Ensuite, un compilateur GWT transforme le code source Java en application web composée de fichiers HTML, CSS, JavaScript, ainsi que des classes Java compilées. Cette application est ensuite prête à être déployée dans un serveur web tel que Tomcat.

Les utilisateurs qui accèdent à l'application peuvent utiliser n'importe quel navigateur web dont l'exécution du code JavaScript est activée.

Lors de l'accès initial, le navigateur charge les fichiers HTML, JavaScript et CSS. Puis lors de l'initialisation des scripts, le chargement des données se fait à l'aide d'une communication asynchrone qui fait appel aux Remote Procedure Call (RPC), hébergés et exécutés sur le serveur. Du point de vue du serveur, ce sont les Servlets dont j'ai parlé plus tôt. On peut donc considérer que le navigateur charge l'interface graphique dans un premier temps, puis ensuite affiche les données reçues dans un second temps.

Lorsque l'utilisateur manipule l'interface, les nouvelles informations lui sont transmises par des échanges asynchrones entre le navigateur et le serveur, qui se font en AJAX. Cette façon de procéder évite d'avoir à transférer toute la page à chaque interaction. Les quantités de données transférées sont alors limitées et les utilisateurs ont une plus grande impression de fluidité de travail.

Pour résumer, GWT est donc un framework qui permet de créer l'interface graphique d'une application web et de gérer les problématiques de communication inhérentes à cette architecture client-serveur qui n'existeraient pas dans une application entièrement Java et exécutée dans la JVM de l'utilisateur final.

# Présentation des choix technologiques : exemples

Dans le schéma précédent, la présentation d'un choix technologique est bien réalisée, car :

- **L'explication en détail de la technologie utilisée est pertinente** : ne pas présenter des choses élémentaires : Java, PHP, programmation orientée objet...
- **L'explication est très claire grâce au schéma** réalisé pour illustrer le propos.
- **L'explication de la technologie est personnelle et adaptée au sujet**: ne pas se contenter de faire des copier / coller d'explications très générales sur une technologie, car il faut justifier l'utilisation de cet outil pour le projet en particulier ou cibler ses explications.

## Autre exemple pertinent :

### ► Java Enterprise Edition & Servlets

Ce module d'importation de données est développé sous la forme d'une application web, qui utilise Java Enterprise Edition.

JavaEE est une spécification de la plate-forme Java destinée aux applications d'entreprise. À cet effet, elle met à disposition des composants qui permettent de répondre aux besoins spécifiques des entreprises, et plus particulièrement pour traiter les aspects de communication en réseau.

Parmi les composants qui sont proposés par JavaEE, le plus important dans notre contexte est le composant Servlet. Un Servlet est un composant qui s'exécute sur un serveur HTTP et permet de créer dynamiquement des pages qui sont ensuite envoyées aux navigateurs clients. Lors de l'exécution de ce Servlet, il est alors possible de communiquer avec une base de données, de faire appel à d'autres bibliothèques ou modules Java. Ensuite, le résultat de cette exécution est renvoyé au client.

- Le JavaEE n'est pas systématiquement étudié à niveau Bac+2 (contrairement au Java ou au PHP).
- L'explication est courte et adaptée au propos : « le plus important dans notre contexte est le composant Servlet. »

# Manuel d'utilisation

- ❑ **Manuel d'installation** : documentation de type « Lisez-moi » permettant d'installer et de configurer le logiciel.
- ❑ **Manuel d'utilisation** :
  - ❑ Destiné à un utilisateur qui découvre le programme et qui doit pouvoir trouver dans la documentation toutes les réponses à ses questions.
  - ❑ Utiliser des captures d'écran pour guider l'utilisateur.
- ❑ **Test de validation** : comment avez-vous débogué votre programme

## 6.1 Connexion/Déconnexion

- **Le site** : l'accès au back-end Imagygame passe par l'espace membre de son site auquel vous pouvez accéder grâce à l'adresse : <http://www.imagygame.com>. En effet, il vous faudra cliquer sur le bloc vert à droite indiqué par une flèche rouge sur la capture d'écran suivante :



Capture 2: Page de connexion

Cela va ouvrir une pop-up comme on peut voir ci-dessus permettant l'accès au back-end grâce à vos identifiants.

# Gestion de projet

## ▣ Principe

- ▣ Partie dédiée à l'**aspect méthodologique et universitaire du projet**.
- ▣ **Retour critique sur votre gestion du projet** : travail de groupe, pertinence des choix d'outils de communication et de gestion...Quels sont éléments positifs et ceux à revoir ?
- ▣ Seule partie du rapport où l'on peut utiliser les **marques de première personne et les temps du passé** pour faire un bilan critique sur la gestion du projet.

## ▣ Contenu possible

- ▣ **Évaluation de la méthode de développement choisie et du travail de groupe**
- ▣ **Planification** : présentation commentée et critique des plannings prévisionnels et du planning réel.
- ▣ **Pertinence des méthode et outils de travail** : gestionnaire de version de code (Git), communication ...



# Conclusion

## ▣ Phase de résumé

- ▣ Rappel des objectifs du cahier des charges
- ▣ Synthèse des résultats obtenus
- ▣ Perspectives d'évolution du projet (éventuellement)

## ▣ Phase de bilan

- ▣ Apport en termes de connaissances techniques.
- ▣ Apport en termes de méthode de travail.

**Attention à ne pas être trop général dans vos remarques sur ce que le projet ou le stage vous a apporté : bannir les formules creuses et passe-partout.**

# Les sources : notes de bas de page et bibliographie

- **Citation dans le texte** : Indiquer dans le cours de la lecture à quel moment précis vous faites références à une ressource bibliographique avec **le numéro de la référence entre crochets** (style IEEE)

Ex: un Framework est « *un ensemble cohérent de composants logiciels structurels* » [1]

- **Bibliographie :**

- Indiquez l'ensemble des ressources que vous avez consulté au cours de la réalisation de votre projet :
  - ouvrages de référence sur le sujet (livres, périodique et éventuellement un cours d'un enseignant à l'IUT)
  - Site internet
  - Bibliothèque de code.
- **Utilisez Zotero pour l'éditer avec le style IEEE :**

[1]

P. Nom, *Titre*. Lieu: Editeur, date de publication.

[2]

P. Nom, « Titre de la page », *Titre du site web*, Date de publication. [En ligne]. Disponible sur: adresse URL. [Consulté le: 01-nov-2017].

[3]

P. Nom, « Titre du message », *Titre du forum*. Date. Disponible sur: adresse URL. [Consulté le: 01-nov-2017].

## □ Bibliographie

[[1]

F. Beck, « Des outils pour évaluer la sécurité des objets connectés », oct. 01, 2019. <https://interstices.info/des-outils-pour-evaluer-la-securite-des-objets-connectes/> (consulté le oct. 15, 2019).

[2]

N. Boujemma, « Algorithmes : l'éthique n'est pas toute la question », *Le Monde.fr*, sept. 20, 2018.

[3]

S. Cacaly, Y.-F. L. Coadic, P.-D. Pomart, et E. Sutter, Éd., *Dictionnaire de l'information*, 3e ed. Paris, France: Armand Colin, 2008.

[4]

A. Collon, « Approfondissez Swift avec la Programmation Orientée Objet », *OpenClassrooms*, oct. 30, 2107. <https://openclassrooms.com/courses/approfondissez-swift-avec-la-programmation-orientee-objet> (consulté le oct. 04, 2017).

[5]

H. Guillaud, « Rétro-design de l'attention : c'est compliqué ! », *InternetActu.net*. <http://www.internetactu.net/2019/01/14/retro-design-de-lattention-cest-complique/> (consulté le oct. 18, 2019).

[6]

« Choose a License », *Creative Commons*, 2020. <https://creativecommons.org/choose/> (consulté le nov. 09, 2020).

# Annexes

- ❑ **Titre et numérotation** : Il faut numéroter les annexes et leur donner un titre.
- ❑ **Renvoi aux annexes** : dans le corps du texte les renvois aux annexes sont indispensables) pour indiquer au lecteur quand consulter l'annexe (cf. annexe 1).
- ❑ **Table des annexes** : Faire une table des annexes s'il y a un nombre conséquent d'annexes et de pages (avec l'outil « table des illustrations »).

## Table des annexes

ANNEXE 1 : Base de données initiale.....	68
ANNEXE 2 : Base de données finale.....	69
ANNEXE 3 : Champs et règles de la base de données finale .....	71
ANNEXE 4 : Exemple de règles de gestion.....	84
ANNEXE 5 : Page de connexion.....	89
ANNEXE 6 : Exemple de page de détails (détails d'un annonceur).....	90
ANNEXE 7 : Exemple de page de liste (liste des configurations).....	92
ANNEXE 8 : Exemple de page d'ajout (ajout d'une campagne).....	93
ANNEXE 9 : Exemple de vue (Imagygame.html).....	99
ANNEXE 10 : Modele.php.....	103
ANNEXE 11 : Change.js.....	110
ANNEXE 12 : Portail joueur.....	111
ANNEXE 13 : Portail annonceur.....	111