



ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET
D'ANALYSE DES SYSTÈMES

GÉNIE LOGICIEL

Plateforme freelance

Réalisé par:

Fatima Zahra HAMDI
Mohamed Abdellahi EL WAGHF
Reda EL MOUADDIN

Encadré par:

M.Mahmoud EL HAMLAOUI

January 30, 2019

Résumé

Le Projet « Plateforme Freelance » représente une interface de Freelance, qui est une interface offrant à un membre « Freelancer » de vendre ses services à un employeur « Entreprise » exprimant des besoins.

Pour répondre aux objectifs que nous nous sommes assignés, il a été nécessaire de procéder tout d'abord par une étude fonctionnelle de notre projet ; cette étape nous a permis d'élaborer un cahier de charges bien détaillé sur lequel nous nous sommes basés pour la conception de l'application.

Durant notre projet, nous avons pour mission dans une première étape de parvenir à traduire nos besoins en termes de conception en une base de données. Ceci dit, nous avons veillé à adopter une base de données la plus cohérente qui soit, et la plus adéquate qui nous a parues.

Notre travail s'est fait en trois phases : une phase d'analyse, de conception et une phase de développement. Nous vous présenterons donc tout au long de ce rapport, les étapes que nous avons suivies ainsi que les outils que nous avons utilisés pour réaliser notre projet.

Mots-clé : JEE , MySQL , Freelance ,Entreprise.

Table des figures

2.1	Diagramme de cas d'utilisation	3
2.2	Diagramme d'activité 1	4
2.3	Diagramme d'activité 2	4
2.4	Diagramme de classe	5
3.1	Design Pattern MVC	7
3.2	Logo HTML	7
3.3	Logo CSS	8
3.4	Logo JSP	8
3.5	Logo MySQL	9

Table des matières

Introduction	1
1 Contexte général du projet	2
1.1 Introduction	2
1.2 Problématique et objectifs	2
1.3 Spécifications des besoins fonctionnelles	2
2 Analyse et conception	3
2.1 Introduction	3
2.2 Diagramme de cas d'utilisation	3
2.3 Diagramme d'activité	4
2.4 Diagramme de classe	5
2.5 Conclusion	5
3 Réalisation	6
3.1 Introduction	6
3.2 Architecture du projet	6
3.3 Outils et technologies utilisées	7
3.3.1 Front end	7
3.3.2 Back end	8
3.4 Conclusion	10
Conclusion	11

Introduction

Durant ce travail, Il est question pour nous, la création d'une plateforme freelance sous forme de site web. On avait plusieurs thèmes pour choisir (un site de vente achat, ...), mais notre choix s'est fixé sur le freelance, de par le réel besoin se faisant sentir ainsi que son importance en étant le futur de toute transaction informatique.

Le but de ce projet est donc de comprendre dans un premier temps ce qu'est le freelance, son utilité, son importance etc... Et dans un second temps, être capable de réaliser le projet (la plateforme freelance) pour utiliser les outils et technologies informatiques acquises pendant notre formation, les difficultés et les solutions qui se présentent lors de la réalisation de ce genre de travail.

Le présent document permettra de synthétiser les résultats de ce projet et afin d'assurer cela, le rapport sera articulé autour de trois grands chapitres : Le premier est consacré à l'analyse et la spécification des besoins où nous allons traiter le cas d'utilisations, les besoins fonctionnelles ainsi que le diagramme d'utilisation, le deuxième chapitre sera consacré à la conception où nous allons décortiquer le contenu des différents diagrammes de séquence et de classe qui concerne notre projet, finalement pour le troisième chapitre nous allons entamer la réalisation, où nous allons parler des outils de travail.

Chapitre 1

Contexte général du projet

1.1 Introduction

Durant ce premier chapitre nous allons donner une vue globale de notre projet, en présentant le sujet d'une manière générale, puis présenter notre objectif et notre démarche.

1.2 Problématique et objectifs

Notre projet consiste donc à développer une application web afin de gérer les offres de freelance. L'objectif est de garantir à l'utilisateur une interface conviviale et facile d'utilisation pour rendre l'interaction aussi instinctive et intuitive que possible. Cela va permettre aux utilisateurs de s'inscrire à notre plateforme et par la suite soit visualiser et postuler pour l'ensemble des projet proposer ,soit proposer des projets.

1.3 Spécifications des besoins fonctionnelles

L'application tient en compte deux acteurs :

- Freelancer :
 - S'inscrire
 - Afficher son profil
 - Chercher les offres postés
 - Postuler pour un projet

- Entreprise :
 - S'inscrire
 - Afficher son profil
 - Afficher les freelancers
 - Poster un projet

Chapitre 2

Analyse et conception

2.1 Introduction

Ce chapitre abordera une analyse du sujet pour présenter une approche originale et une modélisation de l'application.

2.2 Diagramme de cas d'utilisation

Chaque usage que les acteurs font du système est représenté par un cas d'utilisation. Chaque cas d'utilisation représente une fonctionnalité qui leur est offerte afin de produire le résultat attendu. Ainsi, « le diagramme de cas d'utilisation décrit l'interaction entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur ». Ci-dessous les diagramme de cas d'utilisation général de notre système :

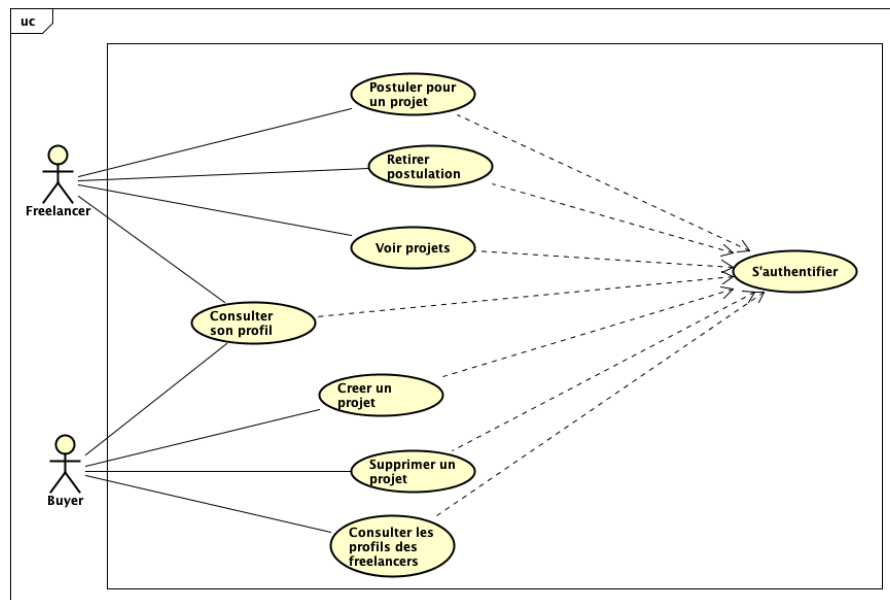


FIGURE 2.1 – Diagramme de cas d'utilisation

2.3 Diagramme d'activité

Ce diagramme permet de décrire toutes sortes de processus : des processus industriels, des processus métier, le déroulement d'un cas d'utilisation ou encore un algorithme.

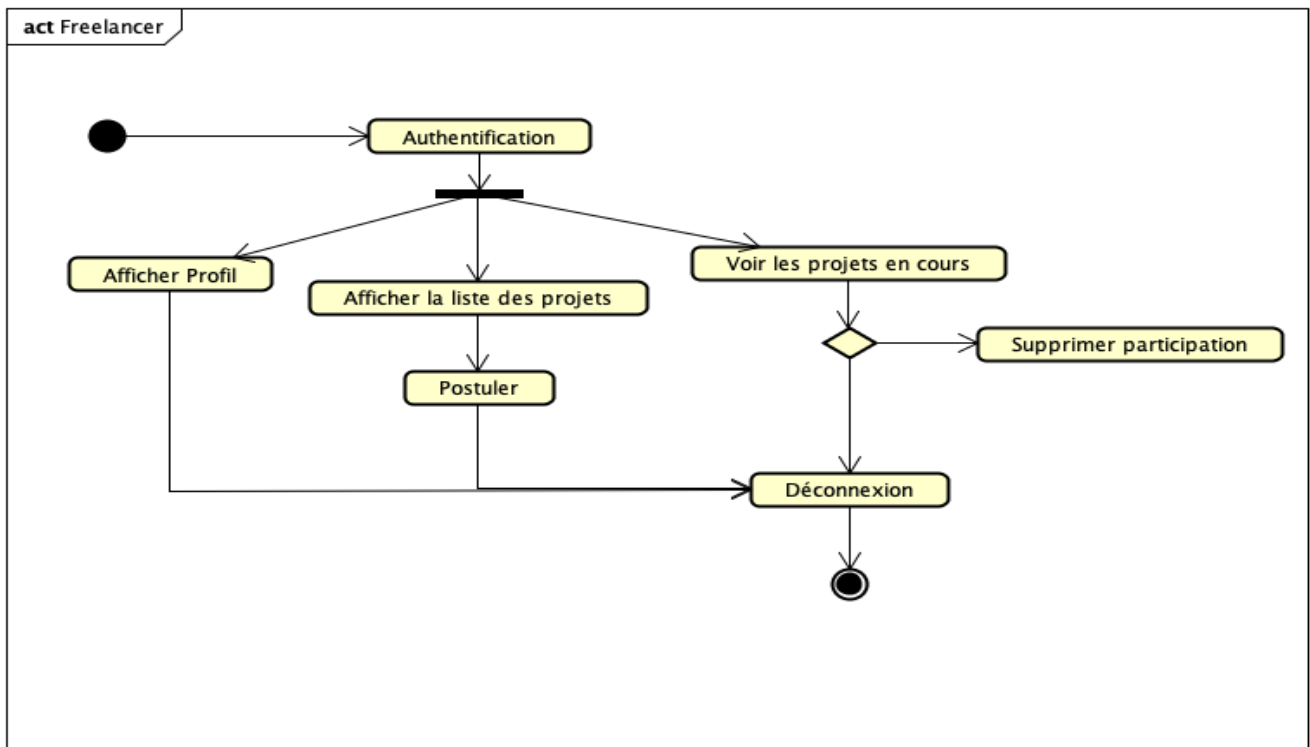


FIGURE 2.2 – Diagramme d'activité du freelancer

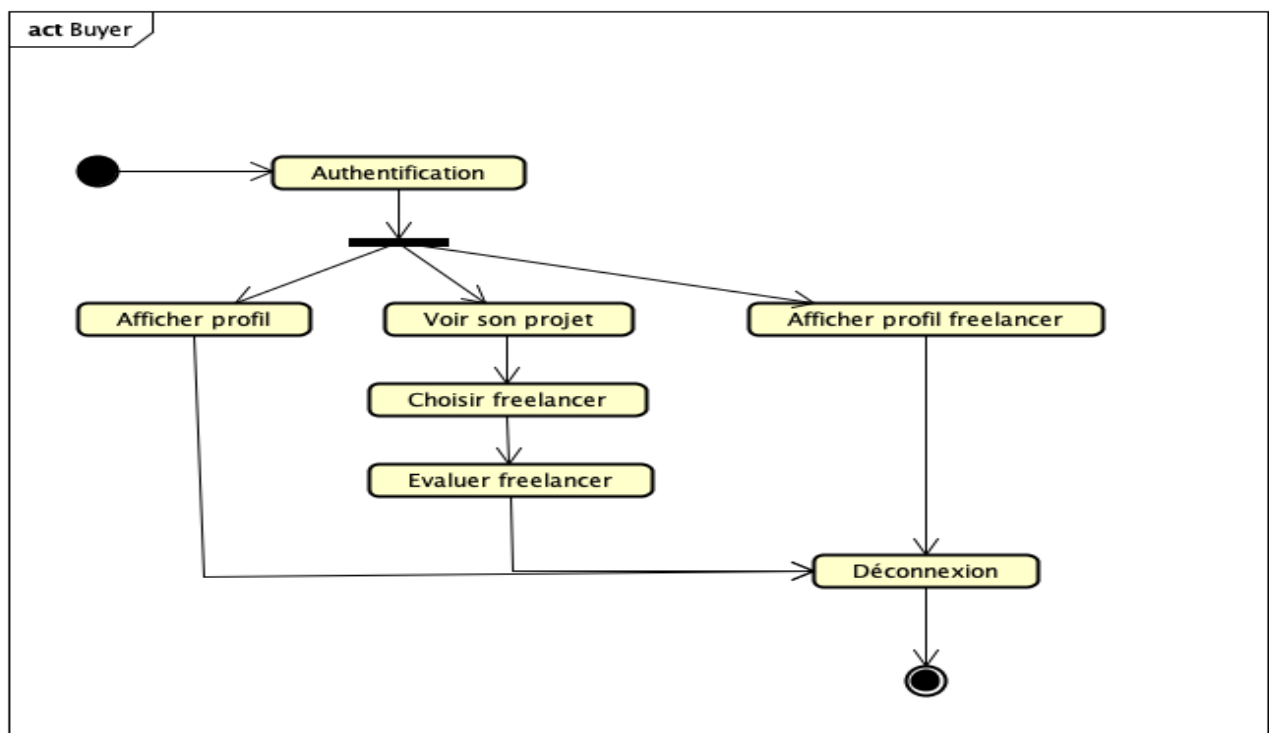


FIGURE 2.3 – Diagramme d'activité du buyer

2.4 Diagramme de classe

Ce diagramme consiste à consolider et valider toute l'analyse des cas d'utilisation. Il s'agit d'un regroupement de l'ensemble des classes en un seul diagramme (diagramme de classe récapitulatif).

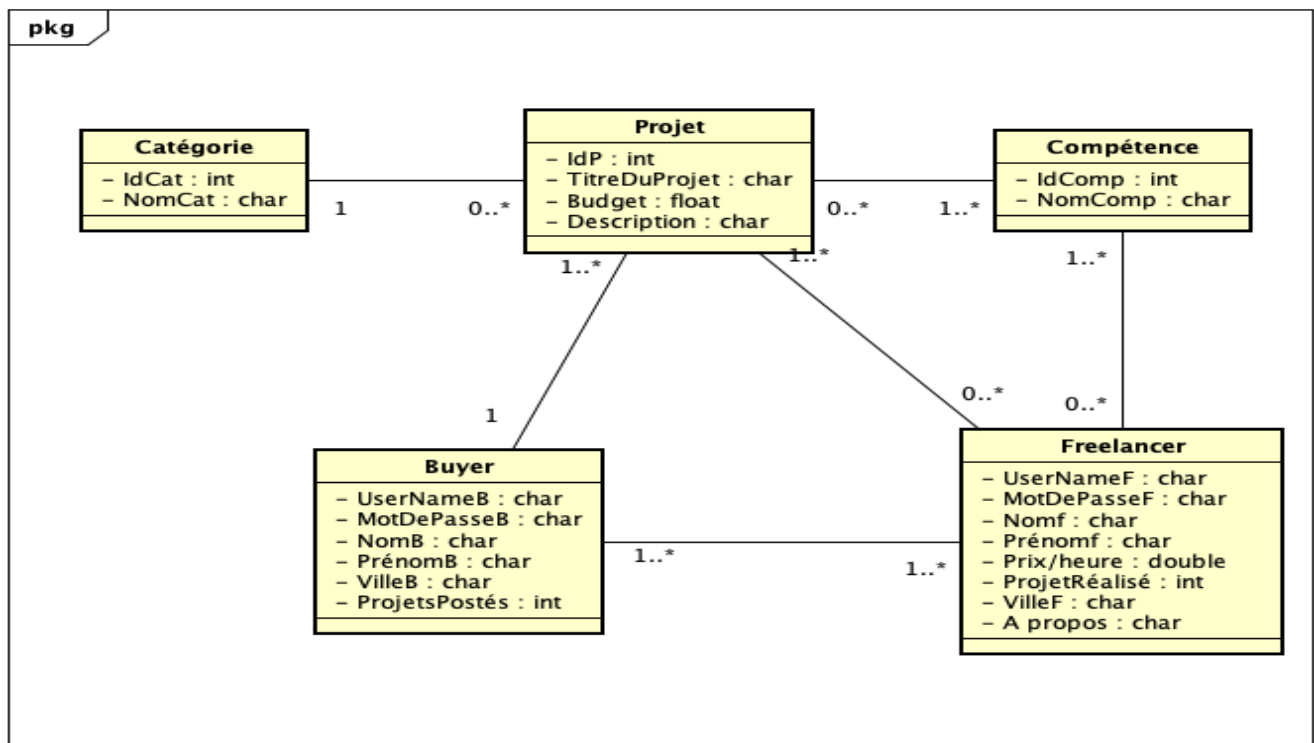


FIGURE 2.4 – Diagramme de classe

2.5 Conclusion

Comme nous pouvons le constater, l'activité de la conception a facilité la compréhension de notre système, qui ébauche vers l'activité d'implémentation.

Chapitre 3

Réalisation

3.1 Introduction

Ce chapitre abordera les outils que nous avons choisis d'utiliser, et présentera le résultat final de l'application.

3.2 Architecture du projet

Pour la réalisation de notre projet nous avons opté pour une architecture en modèle MVC (Modèle-vue-contrôleur).

Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

Dans le modèle, on trouve à la fois les données et les traitements à appliquer à ces données. Ce bloc contient donc des objets Java d'une part, qui peuvent contenir des attributs (données) et des méthodes (traitements) qui leur sont propres, et un système capable de stocker des données d'autre part.

Une page JSP est destinée à la vue. Elle est exécutée côté serveur et permet l'écriture de gabarits (pages en langage "client" comme HTML, CSS, Javascript, XML, etc.). Elle permet au concepteur de la page d'appeler de manière transparente des portions de code Java, via des balises et expressions ressemblant fortement aux balises de présentation HTML. Une servlet est un objet qui permet d'intercepter les requêtes faites par un client, et qui peut personnaliser une réponse en conséquence. Il fournit pour cela des méthodes permettant de scruter les requêtes HTTP. Cet objet n'agit jamais directement sur les données, il faut le voir comme un simple aiguilleur : il intercepte une requête issue d'un client, appelle éventuellement des traitements effectués par le modèle, et ordonne en retour à la vue d'afficher le résultat au client.

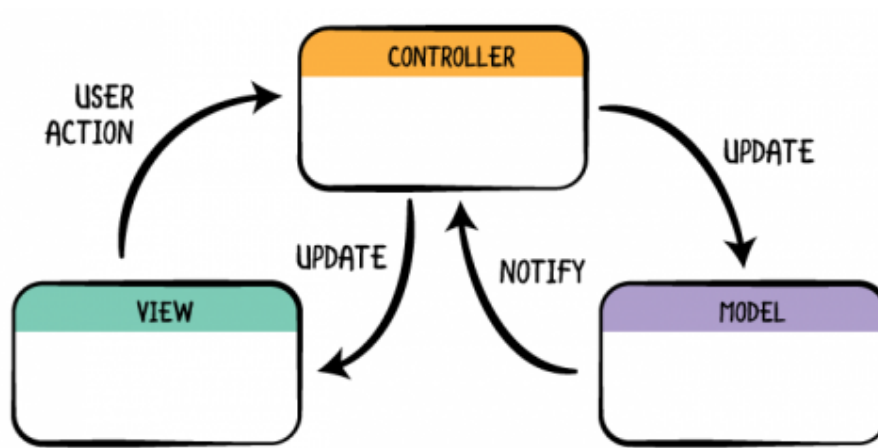


FIGURE 3.1 – Design Pattern MVC

3.3 Outils et technologies utilisées

3.3.1 Front end

- HTML :

HTML a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu. C'est donc en HTML qu'on écrira ce qui doit être affiché sur la page : du texte, des liens, des images. . .



FIGURE 3.2 – Logo HTML

- CSS :

Le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte. . .). Ce langage est venu compléter le HTML en 1996.



FIGURE 3.3 – Logo CSS

- JSP :

Le JSP est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique. Cette technique permet la création de bibliothèques de balises JSP (taglib) qui agissent comme des extensions au HTML ou au XML. Les bibliothèques de balises offrent une méthode indépendante de la plate-forme pour étendre les fonctionnalités d'un serveur HTTP. Il existe aussi un langage de script particulier, appelé Expression Language (EL) destiné à réduire l'injection de code java au sein des pages JSP ainsi qu'à étendre les possibilités des taglibs, tel que la JSTL.



FIGURE 3.4 – Logo JSP

3.3.2 Back end

- Servlet :

Servlet est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus généralement présentées au format HTML, mais elles peuvent également l'être au

format XML ou tout autre format destiné aux navigateurs web. Les servlets utilisent l'API Java Servlet (package `javax.servlet`).

- MySQL :

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde³, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.



FIGURE 3.5 – Logo MySQL

- JDBC :

JDBC est une interface de programmation créée par Sun Microsystems — depuis racheté par Oracle Corporation — pour les programmes utilisant la plateforme Java. Elle permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC. Normalement, il s'agit d'une base de données relationnelle, et des pilotes JDBC sont disponibles pour tous les systèmes connus de bases de données relationnelles.

3.4 Conclusion

Dans ce chapitre nous avons présenté les environnements du travail pour notre projet puis nous avons décrit les différentes interfaces offertes par notre application.

Conclusion

Notre projet a pour objectif la gestion des offres de freelance. Notre travail résume la conception et la réalisation d'une application web permettant la gestion et le suivi des projets de freelance. Dans ce contexte, nous avons cherché à développer une application flexible et évolutive permettant son amélioration par la suite afin d'anticiper les changements continus des besoins des entreprises. Le projet s'est déroulé selon trois axes principaux afin de passer par les étapes essentielles de notre projet : l'analyse, la conception et la réalisation. Pour la réalisation, nous avons utilisé JEE comme langage de programmation et MySQL comme système de gestion de base de données. En outre, ce projet était une opportunité pour bien connaître le développement et apprendre le langage JEE ainsi que les langages liés tel que CSS, JavaScript, HTML. Finalement, cette expérience nous a permis de mettre en pratique ce que nous avons appris en modélisation, et plus particulièrement, la modélisation avec le langage UML.