

Rapport de Projet de fin d'année

Spécialité : Génie logiciel

Intitulé :

**Application web d'automatisation de
processus de gestion des projets**

Réalisé par : Héra Nemri

Encadré par : M. Belguith Mouhieddine

**Année Universitaire
2017 / 2018**

Dédicaces

À mes parents,

*Pour tout leur soutien, pour tous leurs sacrifices, pour leur amour et pour tout
l'enseignement qu'ils m'ont transmis.*

À mes sœurs,

Avec tous les souhaits d'un grand succès dans leur vie.

À mes chers amis,

*Qui ont contribué à mon épanouissement. Merci d'être toujours près de moi, merci
de m'avoir aidé chaque jour à avancer.*

Remerciement

Ce travail a été rendu possible grâce à de nombreuses personnes qui ont eu la patience de m'accompagner, me guider, me soutenir et m'aider tout au long de cette année d'activité

C'est avec un grand plaisir que je réserve ces quelques lignes de gratitude et reconnaissances à tous ceux qui ont contribué à la réalisation de ce projet.

J'adresse tout d'abord mes vifs remerciements à Monsieur Belguith Mouhieddine pour son encadrement, son assistance, son soutien. Ses précieux conseils ont été d'une grande aide pour la rédaction de ce rapport.

Mes remerciements s'adressent également à mes amis ainsi que tous mes collègues pour la fidélité de leur soutien moral.

Je tiens aussi à exprimer ma profonde gratitude aux membres de jury d'avoir accepté de juger mon travail.

Introduction Générale	1
Chapitre 1 : Etude de l'existant.....	3
1. Introduction	4
2. Critique de l'existant :	6
3. Problématique :	7
4. Solution proposée :	7
5. Méthodologies du travail	8
5.1. Etude méthodologique	8
5.1.1. Les méthodes agiles	8
5.1.2. Le processus unifié	8
5.1.3. Etude comparative des sous méthodes du processus unifié	9
5.2. Critère de choix de la méthodologie 2TUP	10
5.2.1. Principe de la méthodologie 2TUP	10
Conclusion :	11
Chapitre 2 : Analyse et spécification des besoins	12
Introduction	13
1. Branche fonctionnelle	13
a. Les besoins fonctionnels	13
b. Modélisation des besoins	14
i. Les acteurs du système	14
ii. Spécifications fonctionnelles détaillées	14
c. Raffinement des cas d'utilisation	16
i. Raffinement de cas d'utilisation « créer un projet »	16
ii. Raffinement de cas d'utilisation « Générer un planning »	17
iii. Raffinement de cas d'utilisation « Générer un rapport de test »	18
iv. Raffinement de cas d'utilisation « Générer les documents financiers »	19
d. Spécification détaillée des exigences	21
i. Diagramme de séquence système « s'authentifier »	21

ii.	Diagramme de séquence système « Ajouter utilisateur »	21
iii.	Diagramme de séquence système « Créer un projet »	23
iv.	Diagramme de séquence système « générer planning »	24
v.	Diagramme de séquence système « Générer un rapport de test »	25
vi.	Diagramme de séquence système « générer document financier »	26
5.	Branche techniques	27
5.1.	Les besoins techniques	27
5.2.	Choix technologique	28
Chapitre 3 : Etude conceptuelle.....		29
Introduction		30
1.	Conception détaillée	30
1.1.	Les patrons de conception	30
1.2.	Modélisation du Processus métier.....	31
2.	Diagramme de classe	31
3.	Diagramme de séquence détaillé	33
3.1.1.	Diagramme de séquence « Ajouter projet »	33
3.1.2.	Diagramme de séquence « Modifier Customer »	34
4.	Diagramme d'activité.....	34
4.1.1.	Diagramme d'activité du cas d'utilisation « S'authentifier »	35
4.1.2.	Diagramme d'activité du cas d'utilisation « gestion projet »	36
Conclusion.....		36
Chapitre 4 : Réalisation		38
Introduction		39
1.	Environnement de Travail.....	39
1.1.	Environnement matériel	39
1.2.	Environnement logiciel	39
2.	Présentation des interfaces	40
2.1.	Processus final de l'application	40
2.2.	Interfaces de l'application d'automatisation de processus de gestion des projets	40
Conclusion.....		49
Conclusion générale et perspectives		50

Liste des figures

Figure 1 Nuxeo workflow	6
Figure 2 : Cycle de développement en Y (Processus 2TUP).....	10
Figure 3 : Diagramme de cas d'utilisation général.....	15
Figure 4 : raffinement de cas d'utilisation "créer un projet"	16
Figure 5 : Raffinement de cas d'utilisation "Générer un planning"	17
Figure 6 : Raffinement de cas d'utilisation "générer un rapport de test"	18
Figure 7 : Raffinement de cas d'utilisation "générer les documents financiers".....	19
Figure 8 : Diagramme de séquence système "s'authentifier"	21
Figure 9 : Diagramme de séquence système "Ajouter utilisateur"	22
Figure 10 : Diagramme de séquence système "créer un projet"	23
Figure 11 : Diagramme de séquence système « générer planning ».....	24
Figure 12 : Diagramme de séquence système "générer un rapport de test".....	25
Figure 13 : Diagramme de séquence système « générer document financier ».....	26
Figure 14 : Processus métier	31
Figure 15 : Diagramme de classe.....	32
Figure 16 :Diagramme de séquence détaillé "Ajouter Projet"	33
Figure 17 : Diagramme de séquence détaillé "Modifier Customer"	34
Figure 18 : Diagramme d'activité du cas d'utilisation « S'authentifier ».....	35
Figure 19 : Diagramme d'activité du cas d'utilisation « gestion projet ».....	36
Figure 20 : Processus final de l'application	40
Figure 21 : Interface d'authentification	41
Figure 22 : Interface d'ajout d'un nouveau projet	41
Figure 23 : Interface d'ajout des besoins du client.....	42

Figure 24 : Interface d'extraction des taches du projet	43
Figure 25 : Interface de génération d'un document financier	44
Figure 26 : Un document financier (facture)	44
Figure 27: Interface d'ajout d'un contrat de travail	45
Figure 28 : Contrat de travail	46
Figure 29 : Interface du planning de travail	47
Figure 30 : Interface de gestion des taches	48
Figure 31 : Interface d'ajout d'un utilisateur	49

Liste des tableaux

Table 1 : Les solutions du marché	6
Table 2 : Tableau comparatif entre les méthodes agiles et le processus unifié	9
Table 3 : Tableau Comparatif entre les différentes méthodologies	9
Table 4 : Liste des acteurs de notre système.....	14
Table 5 : Description textuelle de cas d'utilisation « Créer un projet »	17
Table 6 : Description textuelle de cas d'utilisation « Générer un planning ».....	18
Table 7 : Description textuelle de cas d'utilisation « Générer un rapport de test ».....	19
Table 8 : Description textuelle des cas d'utilisations « générer les documents financiers » ...	20

Introduction Générale

L'introduction de l'informatique dans la vie professionnelle date de trente ans, celle de la micro-informatique de dix ans. Affaire de spécialistes au départ, orientée principalement vers l'automatisation des tâches de gestion administrative ou de contrôle de processus, elle est devenue plus récemment culture, outil privilégié de la modernisation des services, parfois vecteur du projet d'entreprise. Aujourd'hui, par décantation de l'expérience collective, se dégagent des lignes de force, des leçons générales, une méthodologie du bon usage de l'informatique. A la fois démythification et mise en valeur des acquis, un tel bilan devrait marquer la fin d'une période pionnière, sorte de ruée vers l'or. Maintenant s'ouvre un temps moins romantique, plus exigeant, sans doute encore plus efficace.

Désormais, l'informatique n'est plus seulement l'un des instruments de productivité, elle devient un outil de gestion et de pilotage de l'entreprise, voire un instrument stratégique apportant les moyens d'évolution des métiers de l'entreprise. On ne choisit donc plus les applications logicielles pour leur aptitude à remplir une fonction informatique précise, mais comme élément d'une chaîne de production de valeur.

Depuis maintenant plus d'un siècle, les grandes industries s'intéressent et s'investissent dans la gestion de projet. En effet, alors que l'origine du projet date du 14ème siècle où la phase de conception est séparée de la phase de réalisation lors de la construction d'un édifice, la « gestion » de projet n'apparaît qu'à la fin du 19ème siècle, où les entreprises et leurs affaires deviennent complexes. Elle prendra de l'ampleur au fur et à mesure des années où l'on verra apparaître les diagrammes de Gantt, les organigrammes PERT et la professionnalisation de cette discipline.

C'est dans ce cadre que s'inscrit notre projet de fin d'année afin de mettre en place un workflow de gestion et de suivi des projets informatiques.

Un workflow qui permet d'assurer l'échange des informations entre les différents utilisateurs et d'éliminer les échanges de documents dans des supports éparpillés ce qui facilite le suivi de l'état du processus métier de gestion de projet.

Le présent rapport comprend quatre chapitres dans lesquels nous essayons de synthétiser le travail réalisé tout au long de ce projet :

- Le premier chapitre intitulé « Etude de l'existant » permet d'étudier l'existant, le critiquer et proposer notre solution. Puis nous allons étudier la méthodologie du travail.
- Par la suite, Dans le deuxième chapitre « Analyse et spécification des besoins » nous allons étudier l'approche fonctionnels et non fonctionnels de la solution, ainsi que les choix technique et technologiques de l'application.
- Le troisième chapitre intitulé « Étude conceptuelle » nous allons présenter l'architecture générale. Elle sera réalisée en utilisant le langage de modélisation UML.
- Le quatrième chapitre « Réalisation » nous allons présenter quelques imprimés écrans de notre application.
- Et en fin nous clôturons par une conclusion générale dans laquelle nous décrivons les résultats ainsi que l'ouverture d'une perspective.

.

Chapitre 1 : Etude de l'existant

1. Introduction

Dans ce chapitre, nous Dans cette rubrique, nous allons analyser les solutions du marché dans l'objectif de réaliser une application moderne et plus performante critiquons et nous finissons par proposer notre solution.

- **Dans la plupart des entreprises :**

Le processus actuel de gestion de projet suivit par les entreprises est manuel et se compose des phases suivantes :

- **Phase d'étude** : traiter le cahier des charges puis extraire les fonctionnalités et fixer un plan de réalisation à suivre et l'enregistrer dans un support papier.
- **Phase d'Achat et d'approvisionnement** : dégager les produits nécessaires et les fournisseurs optimaux.
- **Phase de réalisation** : exécuter les tâches du projet et générer un rapport sous format papier critiquant le travail en détaillant chaque tâche.
- **Phase de Facturation et paiements**

Notre étude de l'existant ne se limite pas à étudier seulement le processus suivi par ces entreprises, mais aussi à étudier quelques solutions qui existent sur le marché pour ensuite en dégager ses points faibles et montrer en quoi notre solution est sophistiquée.

Il existe des plateformes open source qui permettent d'organiser et d'exécuter le processus de gestion de projet d'une manière meilleure que celle utilisée par la plupart des entreprises.

- **Sur le marché :**

JIRA workflow

JIRA workflow est un ensemble d'états et de transitions, un problème se déplace d'un état vers un autre dans le cycle de vie, on peut classer le workflow avec JIRA en deux catégories :

- **Default workflow**



C'est un workflow qui ne peut pas être modifié. Toutefois, vous pouvez copier et utiliser ce workflow pour créer votre propre processus.

- **Workflow personnalisé**



C'est l'une des caractéristiques principales de JIRA, la Création d'un workflow personnalisé permet JIRA de reproduire des processus internes spécifiques, À un niveau très élevé, les workflows peuvent être personnalisés pour différents projets et types d'émission. [1]

Nuxeo workflow

Le moteur de workflow Nuxeo fournit les fonctionnalités habituelles que vous attendez d'un moteur de workflow et exploite les principaux modules de la plate-forme : le dépôt, le service d'automatisation, mises en page pour toutes les interactions de l'utilisateur et les mises en œuvre de processus. Il fournit des fonctionnalités de haut niveau en matière de gestion des tâches telles que les listes de tâches filtrables, des rappels, tâche réaffectation, la délégation des tâches, des rappels de tâches. Outre le moteur de workflow fournit des moyens pour mettre en œuvre la plupart des concepts BPMN : Fork, fusion, décision, point exclusivisme... [2]

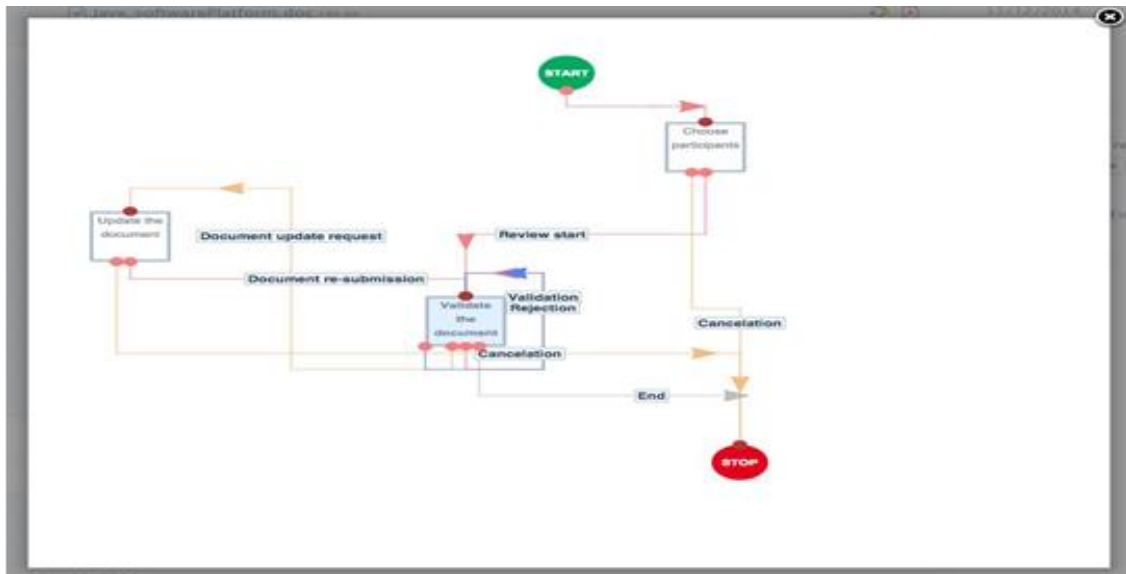


Figure 1 Nuxeo workflow

2. Critique de l'existant :

Le tableau ci-dessous va nous permettre de bien extraire les points forts et les points faibles des solutions du marché :

Solution	Points forts	Points faibles
JIRA workflow	<ul style="list-style-type: none"> -Rapidité d'exécution. -facilité de l'utilisation - multiplateforme. 	<ul style="list-style-type: none"> - fonctionnalités limitées. -pas d'interfaces spécifiques à chaque tâche. - pas de modifications tout au long du processus.
Nuxeo workflow	<ul style="list-style-type: none"> -Rapidité d'exécution - facilité de l'utilisation 	<ul style="list-style-type: none"> -Cout trop cher - fonctionnalités limitées. - pas d'interfaces spécifiques à chaque tâche.

Table 1 : Les solutions du marché

3. Problématique :

Le processus actuel de gestion de projet de la plupart des entreprises se fait principalement à travers des supports semi-papier ou papier, ces moyens empêchent souvent le bon déroulement de tout le processus dans ses différentes phases, ainsi, la communication et la collaboration inter-membres n'est plus organisée de la manière nécessaire à la réussite d'un projet.

Cette méthode limitée peut être suffisante pour une société de petite taille mais avec la croissance de la valeur sur le marché, l'augmentation du nombre de ses projets et sa brillance dans le domaine d'informatique, l'excellence et la perfection deviennent des exigences et ne pas des options.

Pour remédier à ces défaillances, et afin de renforcer ses positions dans le marché et d'évoluer ses processus de gestion des projets et rendre ses productivités plus efficace et plus rentable (rendre le processus métier flexible) ainsi qu'améliorer ses qualités de service, nous avons proposé une solution permettant la réalisation d'un système informatique qui intervient dans toutes les phases de réalisation d'un tel projet, depuis l'identification des besoins des clients jusqu'à la livraison.

4. Solution proposée :

Dans le but de remédier aux problèmes énumérés précédemment, la nécessité d'avoir un outil de gestion de projet s'impose, permettant la centralisation des données et facilitant le travail collaboratif en temps réel entre les différents intervenants sur un même projet. C'est dans ce sens que l'idée de la plateforme *PROJECTMANAGER.FLOW* prend naissance afin de répondre à plusieurs besoins et de pallier plusieurs contraintes spatiales (éviter les supports de stockages traditionnels), temporel, disponibilité et les limites des solutions existantes. Les technologies utilisées ont rendu notre solution plus performante et avantageuse par rapport aux autres solutions.

Cette solution doit assurer les objectifs suivants :

- Le développement et l'exécution du processus de gestion des projets qui commence par l'importation du cahier de charge, passant par l'extraction des tâches à faire, génération du document financière et du contrat, détermination des ressources en faveur du projet, arrivant au planning des tâches et estimation du cout et du temps finissant par la réalisation, l'extraction du rapport de test et la gestion de facturation.
- L'intégration au sein du système grâce à un système de notification

5. Méthodologies du travail

Afin d'aboutir à un logiciel à la hauteur répondant aux besoins, le génie logiciel offre des méthodologies et des normes pour remédier à ces problèmes. Bien qu'il existe une multitude de méthodologies de développement logiciel, le choix d'une méthode pour un projet donné devient une décision de plus en plus draconienne.

5.1. Etude méthodologique

Plusieurs études ont montré l'efficacité des méthodes itératives et incrémentales pour produire un logiciel de qualité répondant aux besoins du client, donc nous avons opté à ces méthodes qu'on vient maintenant d'en parler plus en détails.

5.1.1. Les méthodes agiles

Elles reposent sur des cycles de développement itératifs et adaptatifs en fonction des besoins évolutifs du client. Elles permettent d'impliquer le client dans la réalisation de l'application. Ces méthodes consacrés de mieux répondre aux besoins du client en un temps limité (Scrum, XP, RUP...).

5.1.2. Le processus unifié

Le processus unifié est une méthodologie itérative et incrémentale, guidée par les cas d'utilisation, pilotée par l'architecture et orientée risque.

Le tableau ci-dessous montre un récapitulatif des différences entre les deux méthodes :

Agile	Processus unifié
Faible formalisme	Extrêmement formaliste
Concentré sur le code exécutable	Production de nombreux artefacts
Choix idéal pour les petits projets	Pour les projets complexes

Table 2 : Tableau comparatif entre les méthodes agiles et le processus unifié

Vu le caractère conceptuel et technique de ce travail, nous allons utiliser les méthodes descriptives et analytiques basées sur les principes de modélisation UML 2.0 issu d'UP (Unified Process).

5.1.3. Etude comparative des sous méthodes du processus unifié

Ci-dessous un tableau qui résume cette comparaison :

Sous méthode	Description	Avantages	Limites
Cascade	-Les phases du projet se déroulent séquentiellement	-Phases de projet claires.	-Non itératif -Pas de modèles de documents
2TUP	-S'articule autour de l'architecture. -Cycle en V.	-Itératif -La technologie et la gestion des risques sont les plus importants.	-Plutôt superficiel sur les phases -Pas de documents types.
XP (eXtreme Programming)	-Ensemble de « Meilleures pratiques » de développement -Efficace pour les projets de moins de 10 personnes	-Itératif -Mise en œuvre simple. -Fait une grande importance aux aspects techniques Innovante	-Pas de phase en amont et en aval au développement -très floue dans sa mise en œuvre :quels intervenants, quels livrables ?
RUP	-Outil prêt à l'emploi -Destiné pour les grands projets	-Itératif et incrémental. -Piloté par les cas d'utilisation. -Réduction des risques -Qualité acceptable	-Axé processus -Coût élevé.

Table 3 : Tableau Comparatif entre les différentes méthodologies

5.2. Critère de choix de la méthodologie 2TUP

D'après le tableau comparatif, nous avons opté pour le processus 2TUP pour des raisons multiples. D'une part, 2TUP donne une grande importance à la technologie ce qui est important pour notre projet, d'autre part, 2TUP est un processus en Y composé par une branche technique et branche fonctionnelle. [3]

5.2.1. Principe de la méthodologie 2TUP

Ce processus commence par une étude préliminaire qui permet d'identifier les acteurs du système à mettre en œuvre qui est considéré comme une boîte noire tout en présentant les différents messages entre les utilisateurs et ce système et d'élaborer le cahier des charges. La figure 2, montre les différentes étapes de processus 2TUP :

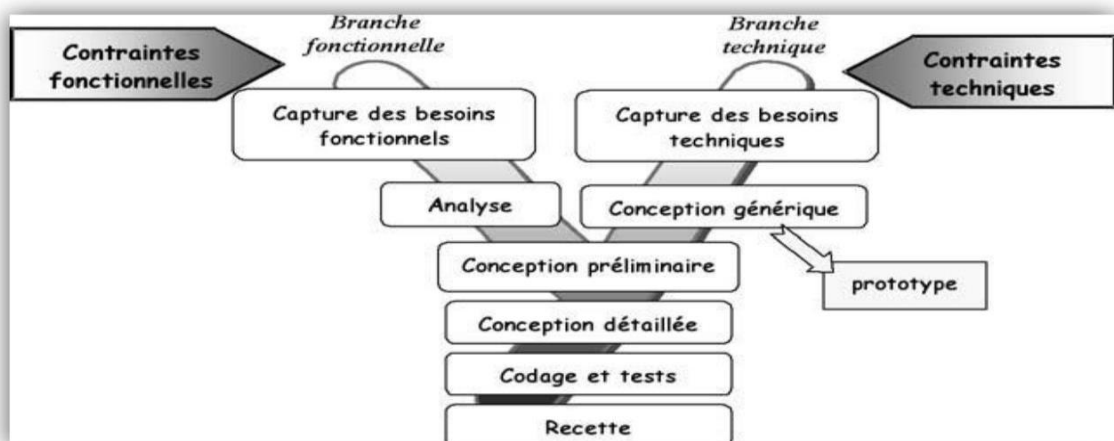


Figure 2 : Cycle de développement en Y (Processus 2TUP)

D'après la figure ci-dessus, on remarque que 2TUP est composé essentiellement de trois branches:

➤ Une branche fonctionnelle (à gauche) :

- Capture des besoins fonctionnels, qui produit les modèles des besoins en se basant sur le métier des utilisateurs. Cette étape élimine le risque d'avoir un système inadapté aux besoins des utilisateurs.
- L'analyse qui consiste à étudier les besoins fonctionnels pour avoir une idée sur le traitement métier du système.

**Une branche technique (à droite) :**

Capture des besoins techniques, cette étape permet de recenser les contraintes de choix de dimensionnement et de conception du système. Les outils sélectionnés ainsi que les contraintes d'intégration (pré requis d'architecture technique).

- L'étape conception générique définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels.

**Une branche de réalisation (au milieu) :**

- La conception préliminaire, elle intègre le modèle d'analyse fonctionnel dans l'architecture technique de comprendre les composants du système à développer.
- La conception détaillée, permettant d'analyser chaque composant.
- L'étape de codage, c'est l'étape de réalisation de ces composants ainsi que les tests unitaires sur chaque composant.
- L'étape de recette, qui consiste enfin à la validation des différentes fonctionnalisées du système.

Conclusion :

L'étude de l'existant nous permet de bien analyser les procédures et la démarche du processus de Gestion de projet à implémenter et nous présente clairement la problématique ainsi que la solution proposée qui sera élaborée conformément aux principes de la méthodologie 2TUP.

Chapitre 2 : Analyse et spécification des besoins

Introduction

Pour extraire l'ensemble des fonctionnalités que doit répondre notre application, ainsi que les différentes contraintes, nous allons consacrer ce chapitre pour l'analyse ainsi que la spécification des besoins

1. Branche fonctionnelle

L'analyse de ce sujet nous a permis d'identifier les différents besoins auxquels doit répondre notre application. Les besoins de la branche fonctionnelle sont seulement les besoins fonctionnels.

1.1. Les besoins fonctionnels

L'analyse fonctionnelle est indispensable pour une bonne réalisation d'un projet, pour cela, nous commençons par la présentation des différents besoins fonctionnels qui doivent répondre aux exigences du futur système. Les besoins recensés sont comme suit :

- La création d'un nouvel utilisateur et ajouter leur rôle et leur groupe.
- La gestion des utilisateurs.
- La création d'un nouveau projet et l'ajout de ses informations et clients.
- L'ajout du cahier de charge et des besoins du client.
- La gestion des tâches du projet, l'estimation du temps ainsi que l'ordre d'exécution.
- La création du devis.
- La création de contrat.
- La planification de la partie réalisation (les tâches mentionnées) par la génération d'un diagramme de Gantt.
- L'affectation des acteurs nécessaires pour le projet (la liste des utilisateurs, la liste des fournisseurs) et la liste des produits qu'on a besoin.
- L'automatisation des tâches du projet (un workflow appart).
- La génération d'un rapport de qualité avant de valider la partie réalisation.
- La création d'une facture et livraison du projet.

1.2. Modélisation des besoins

1.2.1. Les acteurs du système

Un acteur est un élément qui communique directement avec l'application. Après avoir étudié les différentes interactions internes et externes du système, nous avons pu dégager ces acteurs ainsi que leurs interactions:

Acteurs	Besoins fonctionnels
Le gérant	-s'authentifier -créer un nouveau projet -ajouter le cahier de charge et les besoins de client. -ajouter les taches du projet et affecter un contrat de travail. -affecter les acteurs nécessaires pour le nouveau projet. -gérer la liste des utilisateurs.
Le directeur Technique	-s'authentifier -générer un planning (diagramme de Gant). -contrôler l'avancement de réalisation avec un moteur de workflow qui automatise la gestion des taches du projet.
Le responsable de qualité	-s'authentifier -générer un rapport sur la phase de réalisation et validation du travail.
Le responsable financier	-s'authentifier -générer les opérations de financement (devis, facture).

Table 4 : Liste des acteurs de notre système

1.2.2. Spécifications fonctionnelles détaillées

Les cas d'utilisation permettent de comprendre les relations fonctionnelles entre les acteurs et le système. UML propose des concepts issus de bonnes pratiques pour la représentation d'un cas d'utilisation.

Les figures ci-dessous présentent les diagrammes de cas d'utilisation, ils décrivent la majorité des fonctionnalités du système de point de vue des acteurs, mais ne décrivent pas de façon détaillée l'interaction entre les acteurs.

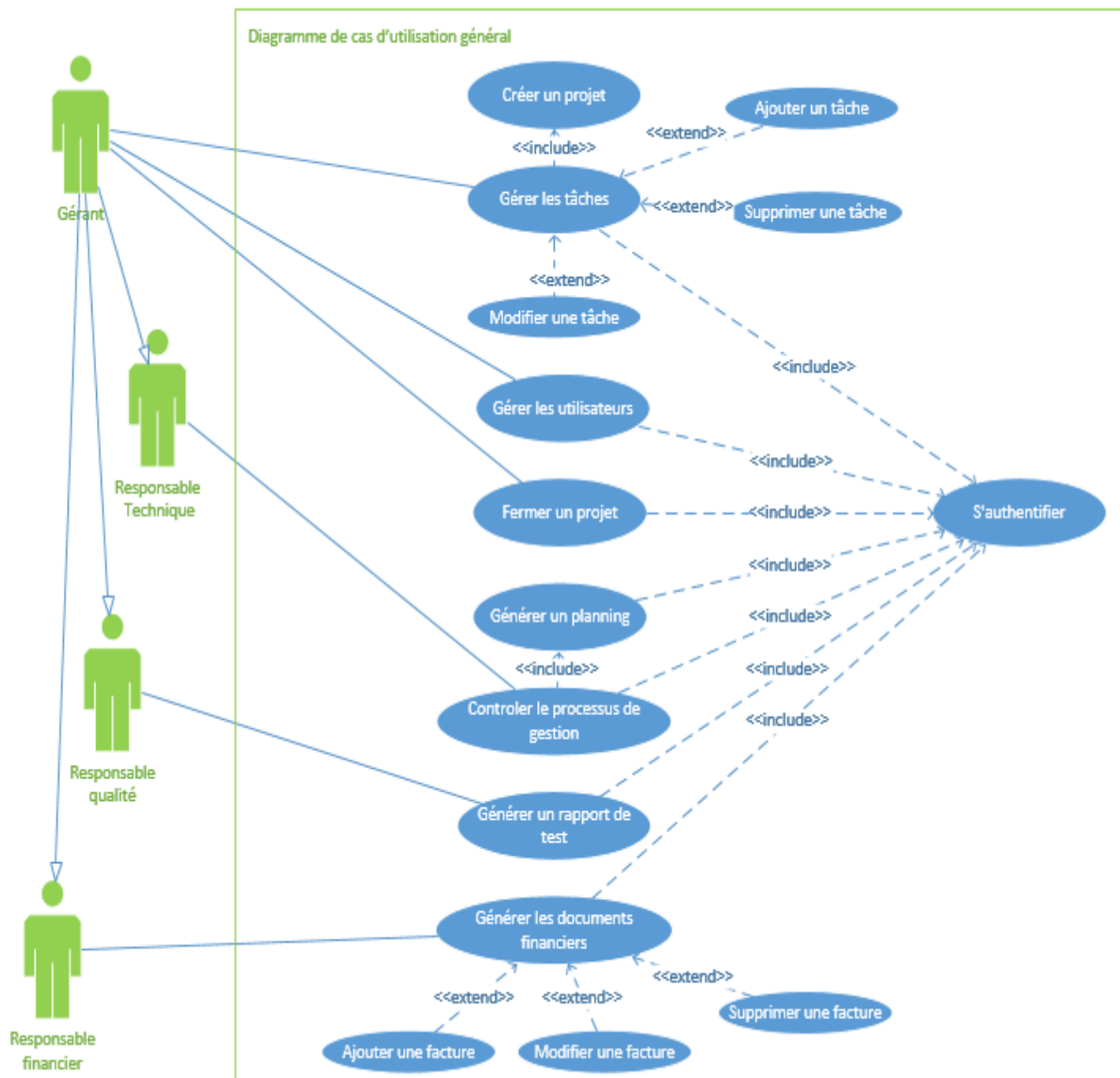


Figure 3 : Diagramme de cas d'utilisation général

1.3. Raffinement des cas d'utilisation

Au cours de cette activité nous allons détailler les cas d'utilisations dégagés grâce aux besoins fonctionnels vus dans le premier chapitre. Pour chaque cas, nous allons parler textuellement des préconditions et des post-conditions liées au cas courant ainsi qu'on va décrire le scénario de base, éventuellement les exceptions.

1.3.1. Raffinement de cas d'utilisation « créer un projet »

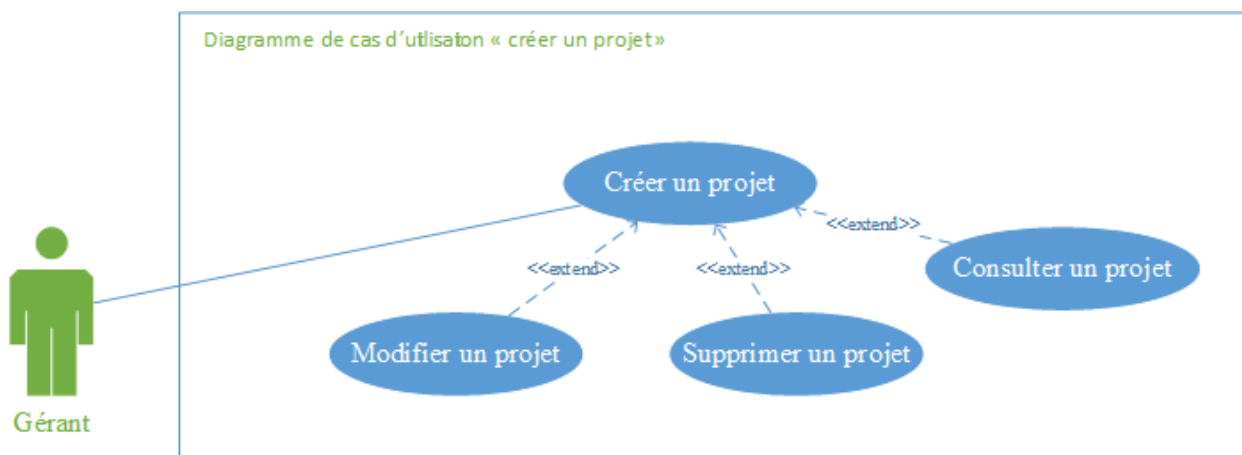


Figure 4 : raffinement de cas d'utilisation "créer un projet"

La figure ci-dessus représente le cas d'utilisation « Créer un projet » raffiné.

En effet, l'accès ne se fait qu'après authentification. Le gérant, qui est dans notre cas le chef de projet, ajoute un nouveau projet, le consulte, le modifie ou le supprime.

- **Description textuelle de cas d'utilisation « Créer un projet » :**

Titre	Créer un projet	
Résumé	Détaille les étapes nécessaires pour ajouter un projet	
Acteur	Gérant	
DESCRIPTION D'ENCHAINEMENT		
Préconditions		Post-conditions
Le gérant s'authentifie		Un nouveau projet ajouté
SCENARIO PRINCIPAL		

<ol style="list-style-type: none"> 1) L'utilisateur demande d'ajouter un nouvel projet 2) L'utilisateur remplit le formulaire concernant le projet 3) Consulter la liste des clients 4) Sélectionner un client de la liste des clients 5) L'utilisateur confirme les informations ajoutées
ENCHAINEMENT ALTERNATIF
<p style="text-align: center;"><u>Enchaînement alternatif A1 :</u></p> <ol style="list-style-type: none"> 4) Choisir d'ajouter un nouveau client. 5) L'utilisateur remplit le formulaire pour ajouter un nouveau client 6) Il confirme l'ajout

Table 5 : Description textuelle de cas d'utilisation « Créer un projet »

1.3.2. Raffinement de cas d'utilisation « Générer un planning »

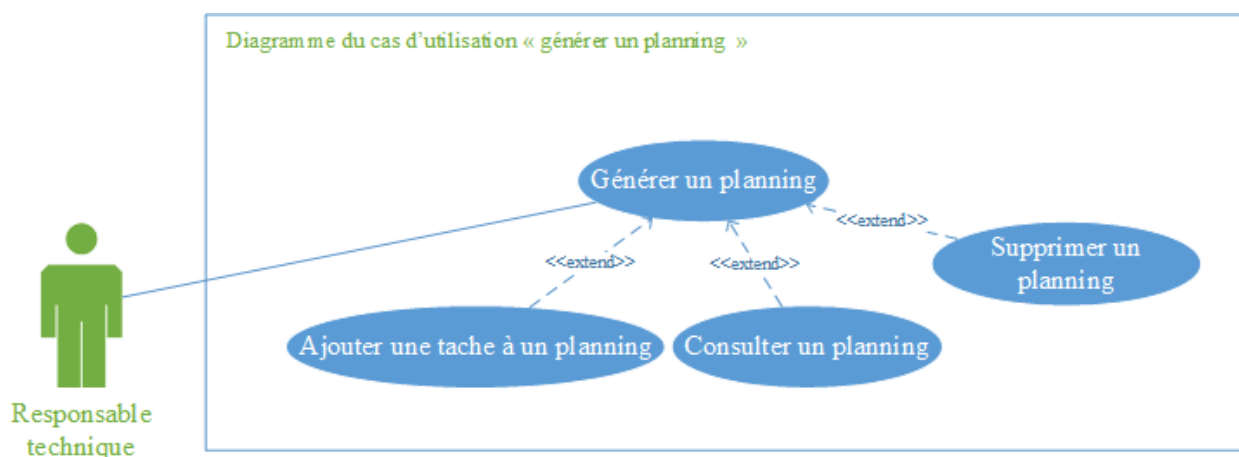


Figure 5 : Raffinement de cas d'utilisation "Générer un planning"

La figure ci-dessus, concerne le cas d'utilisation « générer un planning », maintenant c'est au responsable technique d'orchestrer les tâches et y préparer un planning garantissant le bon déroulement du travail dans les délais prédéfinis, et ce par le fait d'ajouter, consulter ou modifier un planning.

- Description textuelle de cas d'utilisation « Générer un planning » :

Titre	Générer un planning	
Résumé	Détaille les étapes nécessaires pour ajouter un planning (diagramme de Gantt)	
Acteur	Responsable technique	
DESCRIPTION D'ENCHAINEMENT		
Préconditions		Post-conditions
La liste des tâches est ajoutée		Un diagramme de Gantt est généré
SCENARIO PRINCIPAL		
1) Consulter la liste des tâches 2) Ajouter un diagramme de Gant		

Table 6 : Description textuelle de cas d'utilisation « Générer un planning »

1.3.3. Raffinement de cas d'utilisation « Générer un rapport de test »

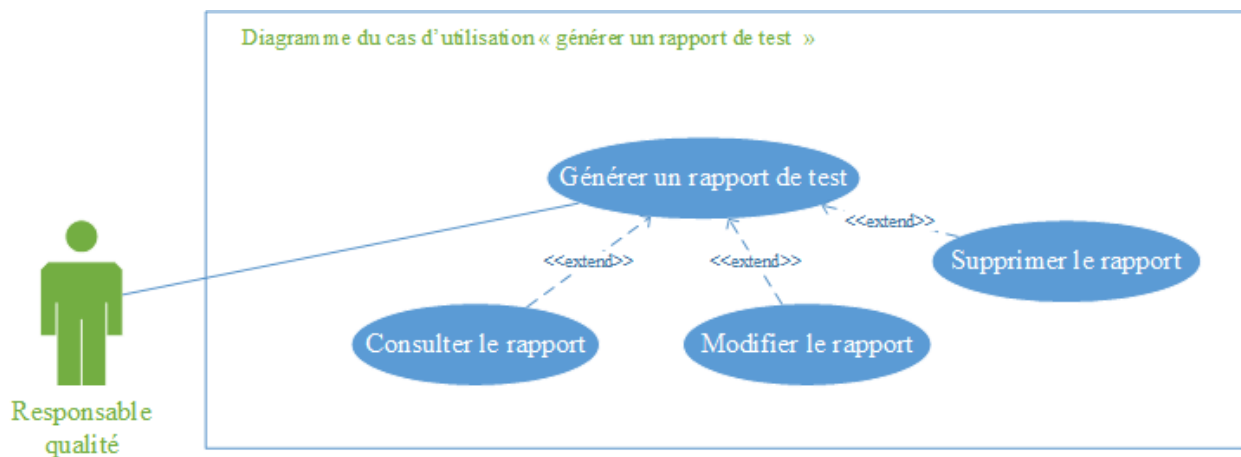


Figure 6 : Raffinement de cas d'utilisation "générer un rapport de test"

La figure ci-dessus, concerne le cas d'utilisation « générer un rapport de test », qui est affecté au responsable de qualité, il s'implique à la fin de chaque projet pour relever un rapport de test au gérant lui permettant d'avoir une vision microscopique sur le pourcentage d'accomplissement des tâches.

• **Description textuelle de cas d'utilisation « Générer un rapport de test »:**

Titre	Générer un rapport de test	
Résumé	Détaille les étapes nécessaires pour générer un rapport de test	
Acteur	Responsable de qualité	
DESCRIPTION D'ENCHAINEMENT		
Préconditions		Post-conditions
La phase de réalisation est terminée		Un rapport de test est ajouté
SCENARIO PRINCIPAL		
1) Consulter la liste des projets. 2) Choisir le projet à tester. 3) Consulter la liste des tâches. 4) Noter chaque tâche. 5) Générer le rapport de test (des charts).		

Table 7 : Description textuelle de cas d'utilisation « Générer un rapport de test »

1.3.4. Raffinement de cas d'utilisation « Générer les documents financiers »

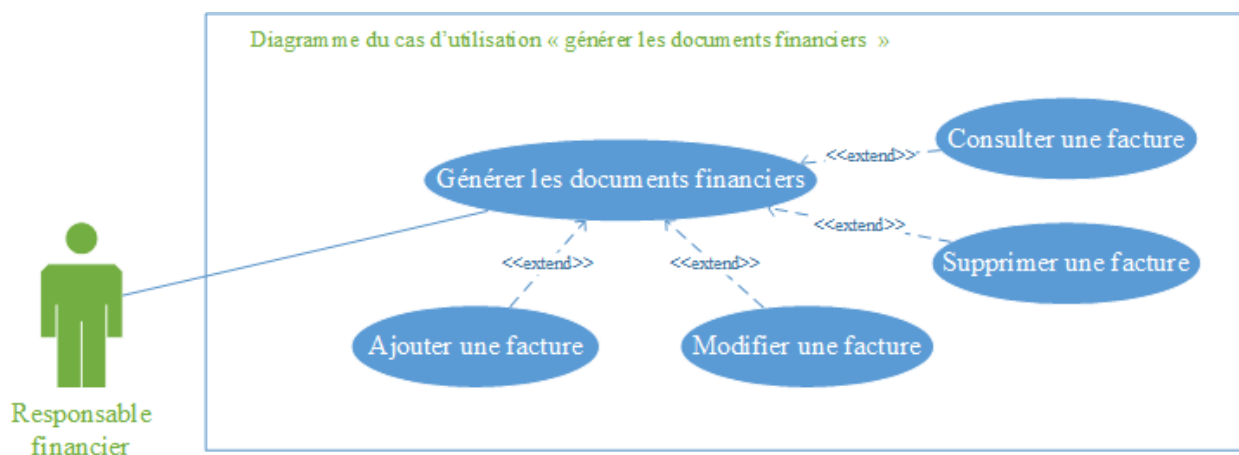


Figure 7 : Raffinement de cas d'utilisation "générer les documents financiers"

La figure ci-dessus, concerne le cas d'utilisation « générer les documents financiers », ce dernier volet est accordé au responsable financier, qui génère un devis puis le passer en facture pour le fournir au client.

- **Description textuelle des cas d'utilisations « générer les documents financiers »**

Titre	Générer les documents financiers	
Résumé	Détaille les étapes nécessaires pour générer les factures	
Acteur	Responsable financier	
DESCRIPTION D'ENCHAINEMENT		
Préconditions		Post-conditions
Un projet crée		Un document financier ajouté
SCENARIO PRINCIPAL		
1) Consulter la liste des projets. 2) Choisir un projet. 3) Choisir le document financier à ajouter. 4) Remplir le formulaire. 5) Confirmer le remplissage et générer le document.		

Table 8 : Description textuelle des cas d'utilisations « générer les documents financiers »

1.4. Spécification détaillée des exigences

1.4.1. Diagramme de séquence système « s'authentifier »

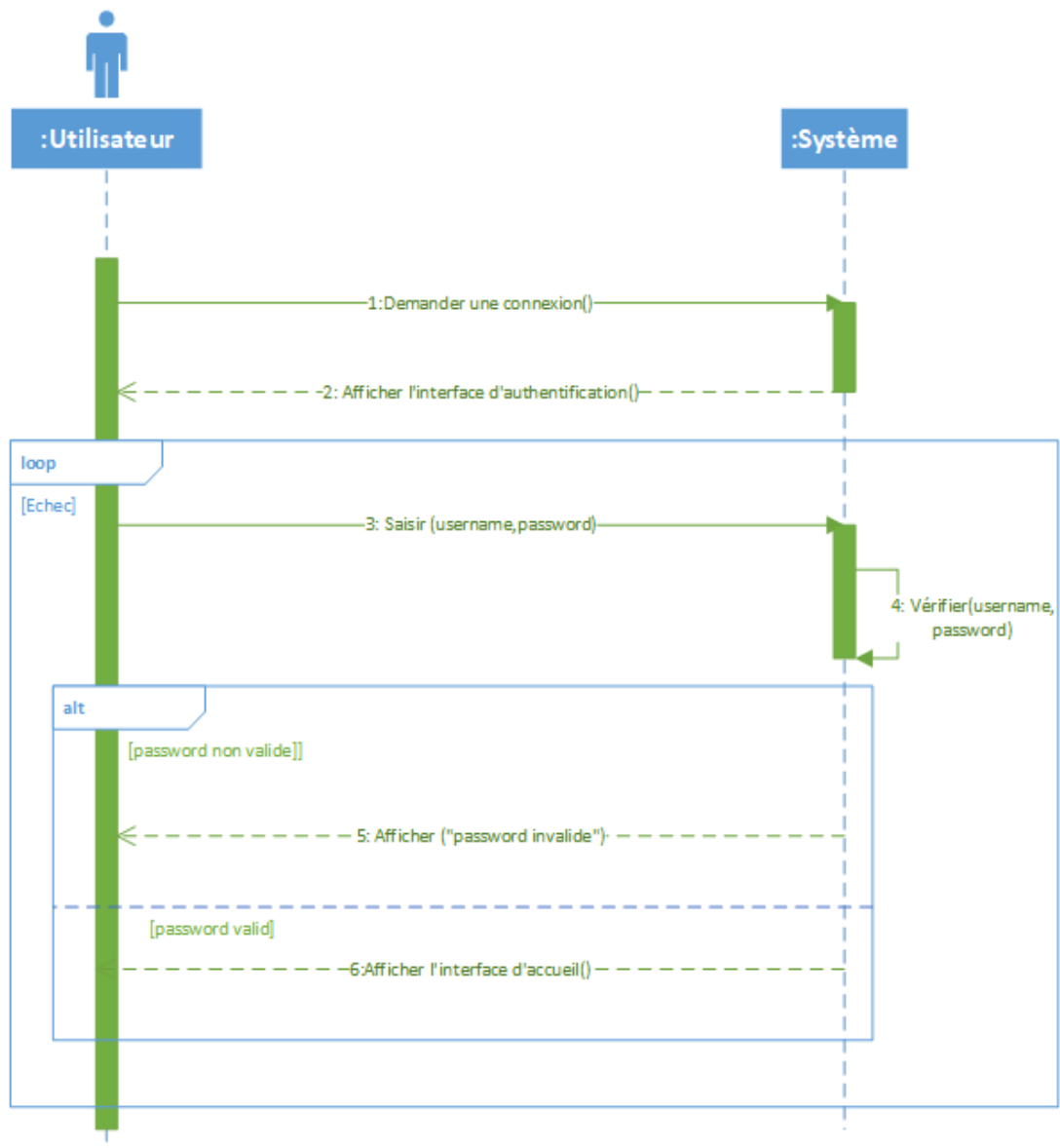


Figure 8 : Diagramme de séquence système "s'authentifier"

Pour passer à exploiter les fonctionnalités du système, il faut tout d'abord se connecter comme étant utilisateur. En cas de mauvais renseignement de paramètres d'authentification, le système va notifier l'utilisateur, et l'interface de connexion s'auto initialise de nouveau.

1.4.2. Diagramme de séquence système « Ajouter utilisateur »

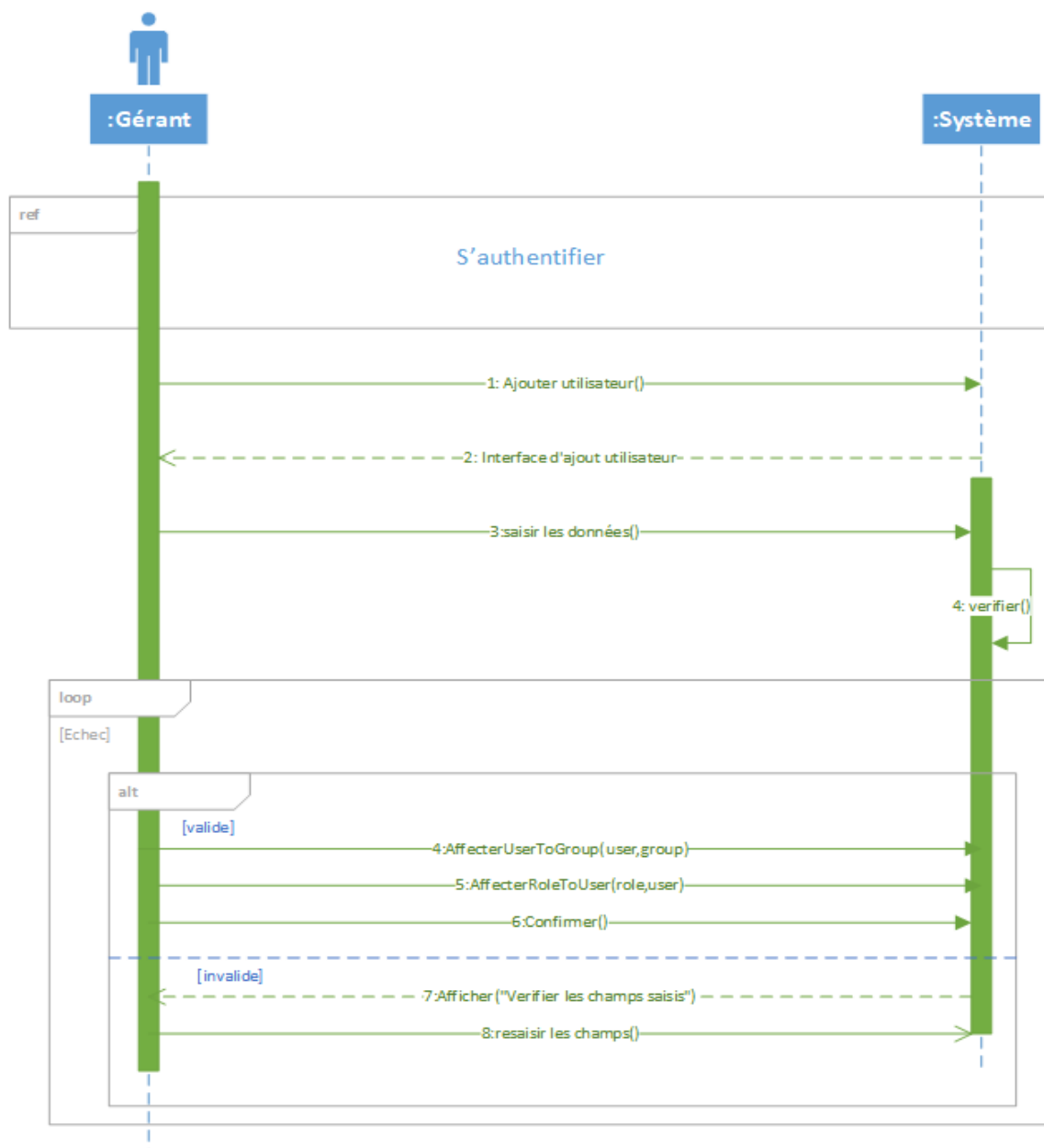


Figure 9 Diagramme de séquence système "Ajouter utilisateur"

Ce diagramme de séquence système présente le scénario à suivre pour ajouter un nouvel utilisateur à la base de données.

1.4.3. Diagramme de séquence système « Créer un projet »

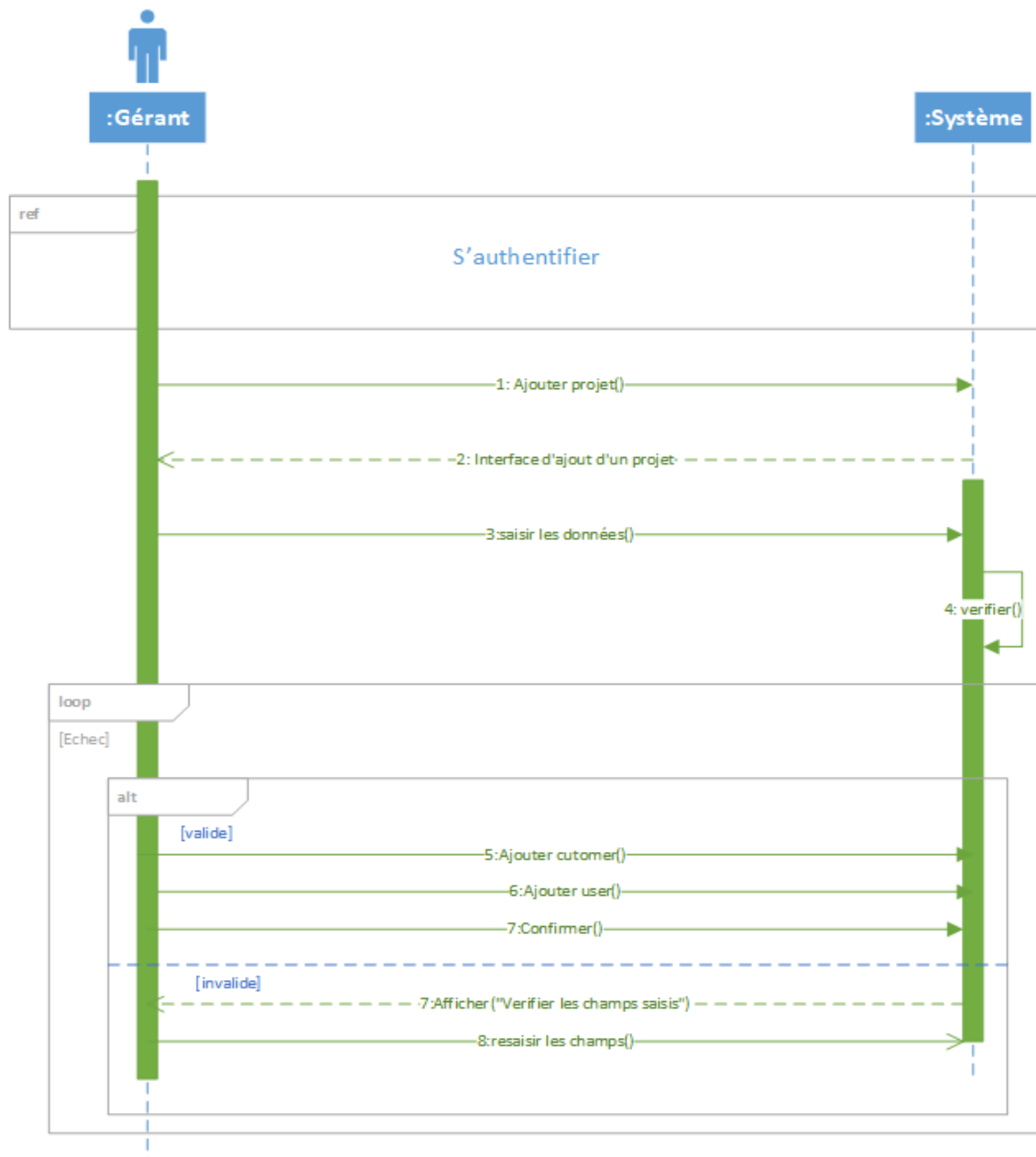


Figure 10 : Diagramme de séquence système "créer un projet"

L'ajout des projets est accordé au gérant, donc le système va réagir aux demandes de création de projet comme suit : Un utilisateur connecté demande d'ajouter un projet, le système vérifie s'il possède les droits de création, si non, il sera bloqué par une alerte, sinon, il sera invité à saisir les données et les ressources nécessaires, en cas de mal remplissage, le système ne cesse de signaler à l'utilisateur.

1.4.4. Diagramme de séquence système « générer planning »

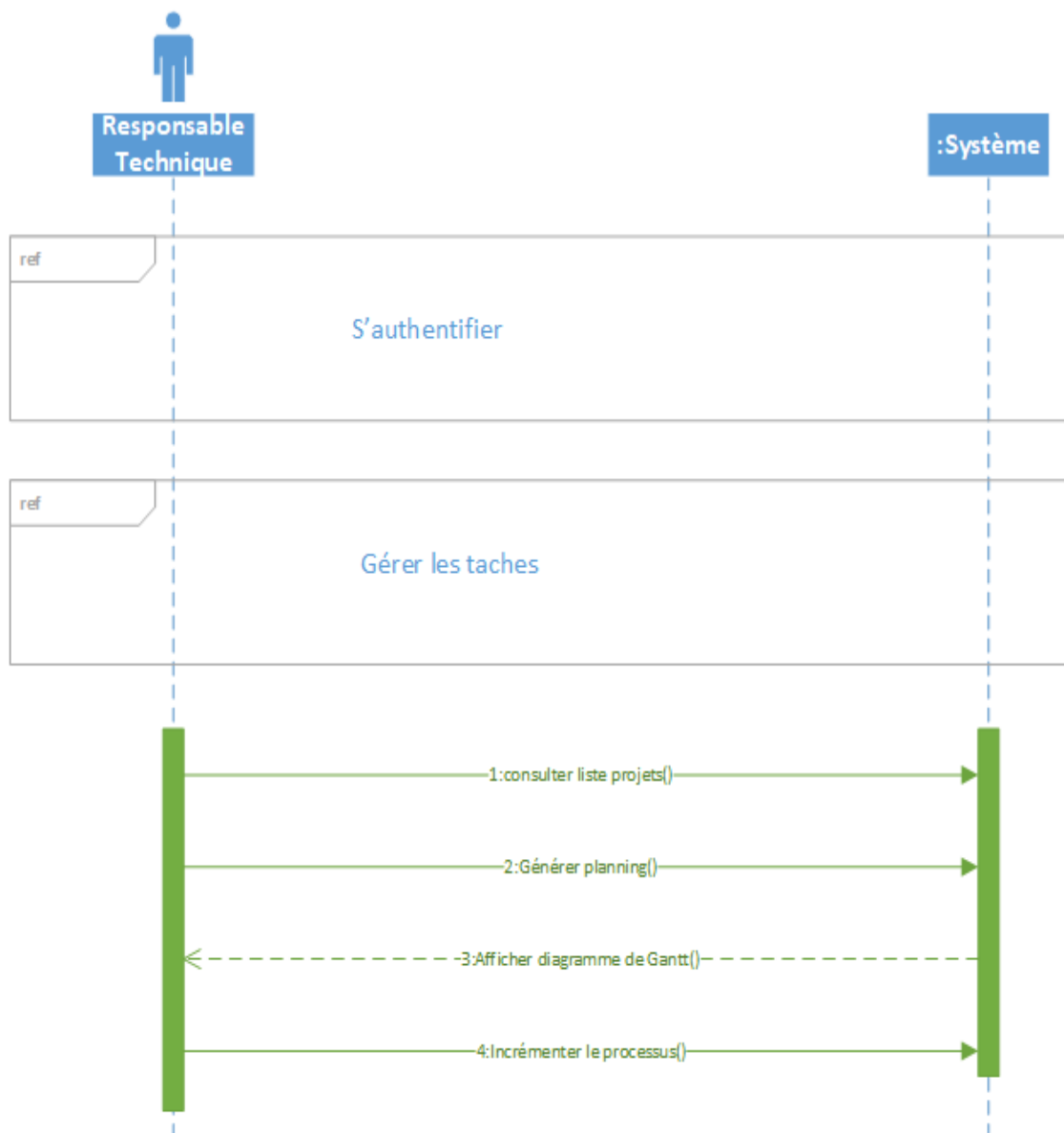


Figure 11 : Diagramme de séquence système « générer planning »

1.4.5. Diagramme de séquence système « Générer un rapport de test »

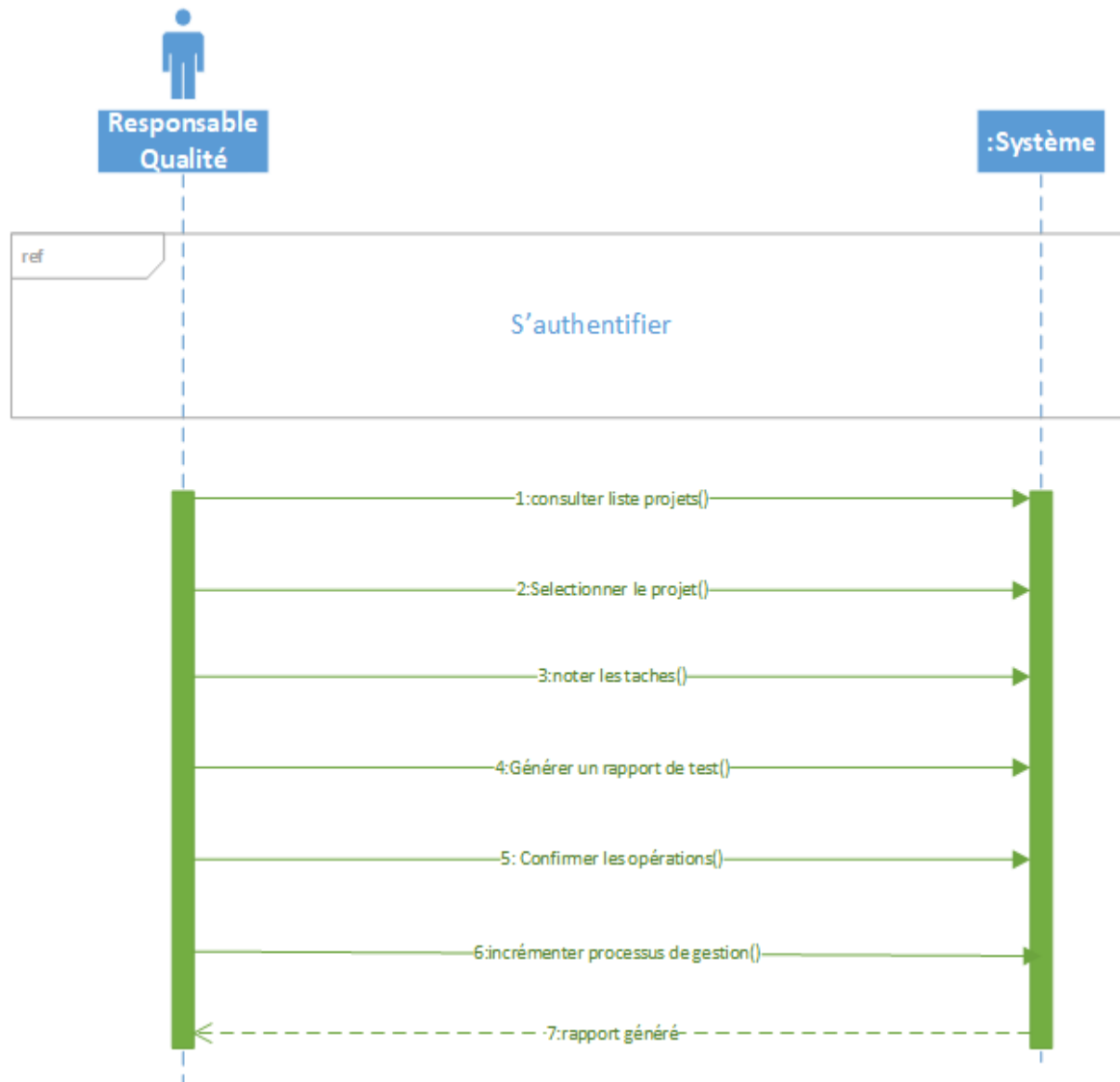


Figure 12 : Diagramme de séquence système "générer un rapport de test"

1.4.6. Diagramme de séquence système « générer document financier »

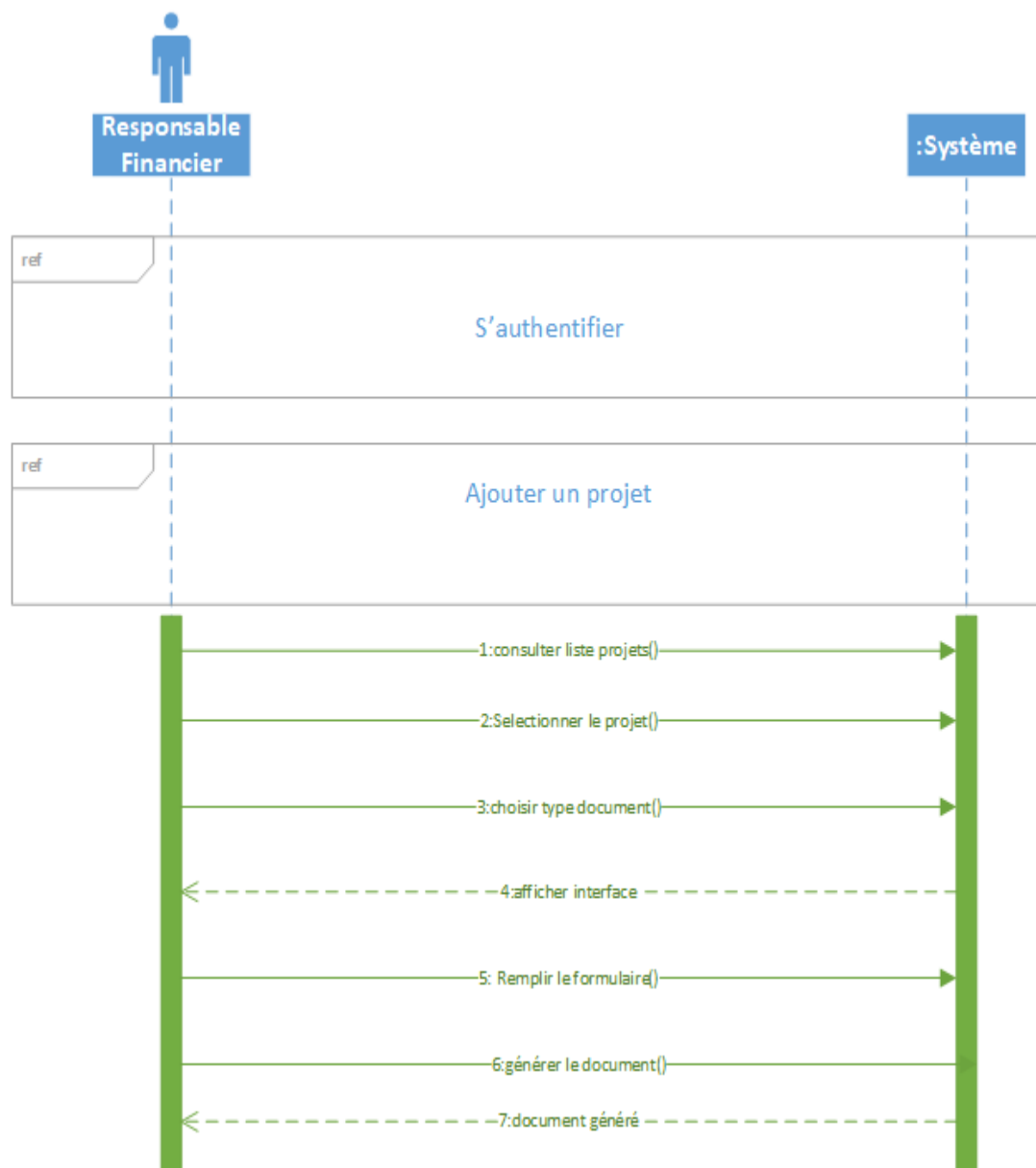


Figure 13 : Diagramme de séquence système « générer document financier »

L'utilisateur consulte la liste des projets, sélectionne un projet, puis crée un devis pour finalement le générer.

5. Branche techniques

Pour réaliser notre projet nous avons appliqué des ressources techniques que nous allons les présenter par la suite.

5.1. Les besoins techniques

Etant donné qu'une application fonctionnelle et opérationnelle ne garantit pas la satisfaction et la fidélité des utilisateurs, nous devons prendre en considération des critères non fonctionnels lors de la conception et de l'implémentation de notre solution. Parmi ces exigences, certains sont basiques, dont tout produit informatique doit doter. Nous citons :

- **Architecture en couches :**
Nous allons classer l'application en des couches, nous utilisons la technologie EJB (Entreprise Java Bean) pour assurer la communication avec la base de données, qui est composée de deux modules les entités qui représentent les tables de la base de données, le parton DAO qui assure l'accès aux données, ainsi nous avons utilisé pour les traitements métiers le Framework JSF avec Spring, pour la couche présentation nous avons utilisé le framework interfaces utilisateur de JSF PrimeFaces
- **La Sécurité :** nous avons utilisé le Spring Security comme un outil de protection contre les accès non autorisés.
- **La Compréhension :** les interfaces graphiques se sont organisés d'une façon plus lisible et clair pour faciliter la compréhension et l'utilisation et pour nous avons utilisé la Framework Primefaces 6 de JSF.
- **Apprentissage :** La facilité pour un utilisateur à utiliser le produit (la tâche de gestion des projets (métier) est clair et ses instructions sont mentionnées ainsi ne complique pas les interfaces d'interactions avec l'utilisateur de l'application).

5.2. Choix technologique

- **JEE (Java Enterprise Edition)** : un ensemble de spécifications pour plate-forme java d'oracle destinée au application d'entreprise
- **Maven** : conçu pour supprimer les tâches difficiles du processus de build par l'utilisation d'une approche déclarative
- **Java Server Faces:** Ou JSF est un Framework java utilisé pour développer des applications web. JSF est un Framework MVC basé sur la notion des composants.
- **Hibernate** : Framework open-source Java ayant pour gérer la persistance des données de base ainsi que la notion d'ORM.

Conclusion

Durant ce chapitre, nous avons analysé les besoins non fonctionnels et fonctionnels de notre application. Ces analyses sont devenues formelles par la suite à l'aide des diagrammes des cas d'utilisation. Les interactions utilisateurs système sont décrites dans des tableaux spécifiques Le chapitre qui suit va être consacré pour l'étude conceptuelle de notre application.

Chapitre 3 : Etude conceptuelle

Introduction

La conception permet de décrire le fonctionnement désiré du système afin d'en faciliter la réalisation et la maintenance. Pour ce faire, la première partie dédiée à l'exposition du choix de méthode de conception ainsi que l'architecture du système à développer. La deuxième partie présente la conception détaillée de notre système et la base de données utilisée.

1. Conception détaillée

1.1. Les patrons de conception

En génie logiciel, un patron de conception vise à résoudre les problèmes selon l'approche orienté objet. Un modèle de conception peut être vu comme un document contenant la description de la structure d'un ensemble de classes permettant de répondre à une situation particulière. Le but principal des patrons de conception est de minimiser le temps de développement d'un logiciel tout en augmentant la qualité de résultat.



MVC (Model-View-Controller) :

Le modèle vue contrôleur est un modèle de conception responsable de l'organisation de l'interface homme machine (IHM) d'un logiciel. En résumé, lorsqu'un utilisateur envoie une requête, elle va être envoyée par la « vue » et analysé par le « contrôleur ». En suite le contrôleur demande au model approprié d'effectuer l'ensemble des traitements nécessaires et notifie la vue que la requête est effectuée. Enfin une notification est effectuée vers le modèle pour se mettre à jour.

1.2. Modélisation du Processus métier

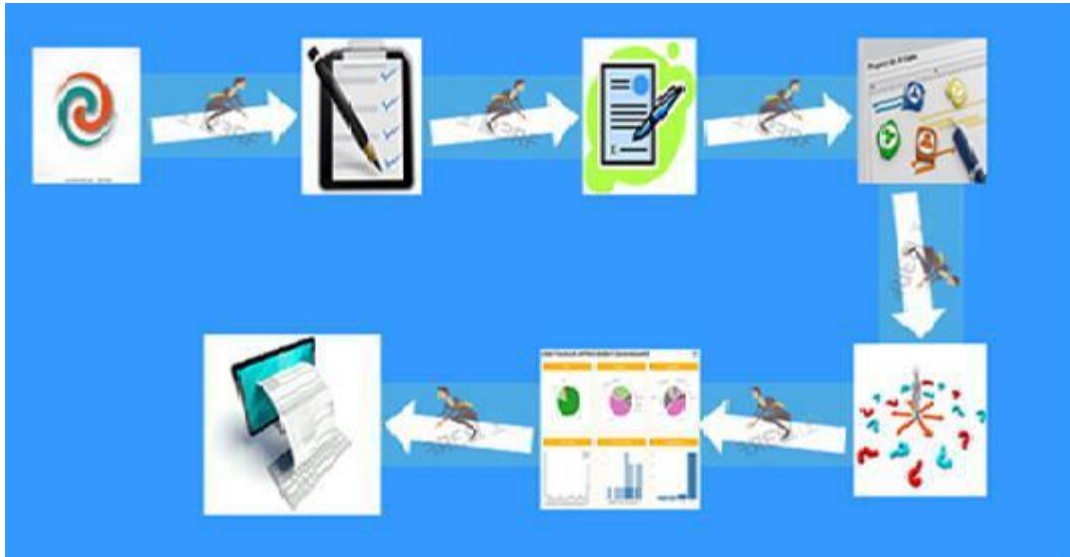


Figure 14 Processus métier

Ce processus représente les différentes tâches pour gérer les phases du projet. Ce processus métier va permettre à l'utilisateur de contrôler l'avancement de ses projets, en commençant par la création du projet et il va ajouter les différents documents nécessaires à la réalisation du projet et aussi il assure la gestion des tâches. Et Enfin, il fournit un rapport de test à l'utilisateur.

2. Diagramme de classe

Le diagramme de classe exprime l'aspect statique du système en termes de classes et de relations entre ces classes. Il nous permet de présenter la structure interne du système et d'organiser le travail de développeur. Nous avons utilisé dans notre diagramme de classe les concepts suivant Classe : une classe est une description abstraite (condensée) d'un ensemble d'objets similaires du point de vue de leur structure (attributs), leur comportement (opérations) et leurs relations à l'extérieur (association). L'association : une association exprime une connexion sémantique bidirectionnelle entre classes. C'est une abstraction des liens qui existent entre les objets instances des classes associées.

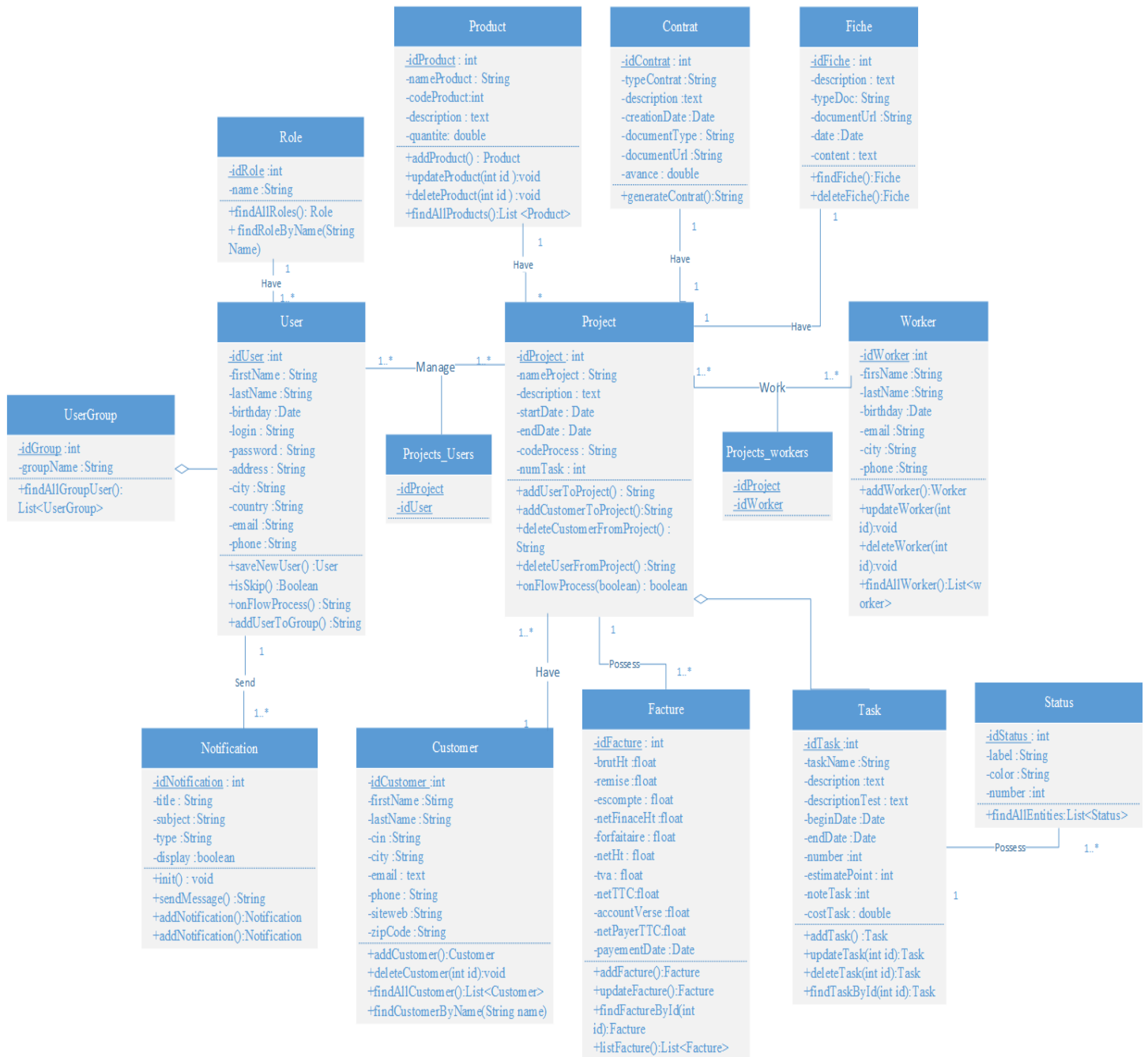


Figure 15 : Diagramme de classe

3. Diagramme de séquence détaillé

3.1.1. Diagramme de séquence « Ajouter projet »

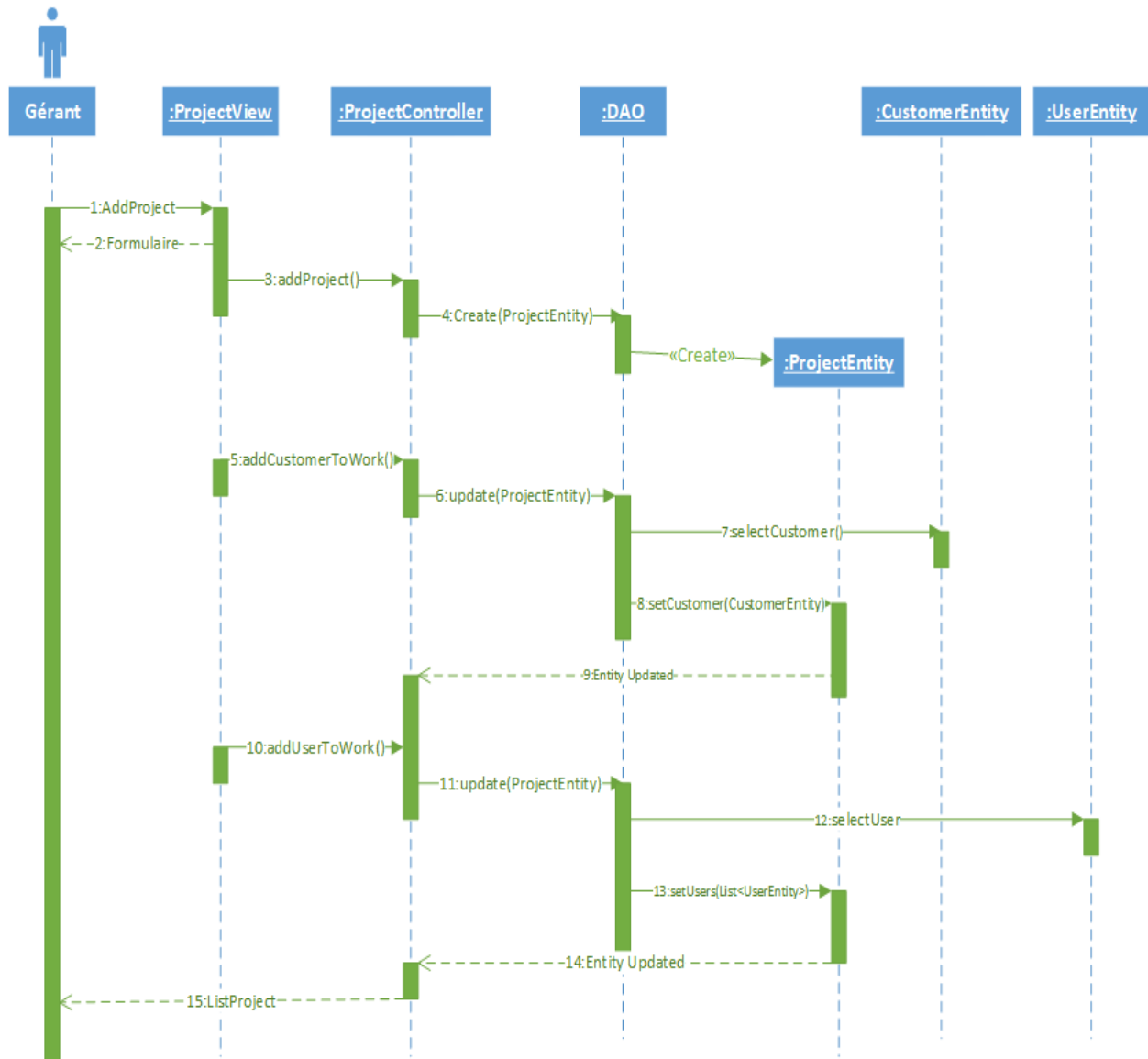


Figure 16 :Diagramme de séquence détaillé "Ajouter Projet"

3.1.2. Diagramme de séquence « Modifier Customer »

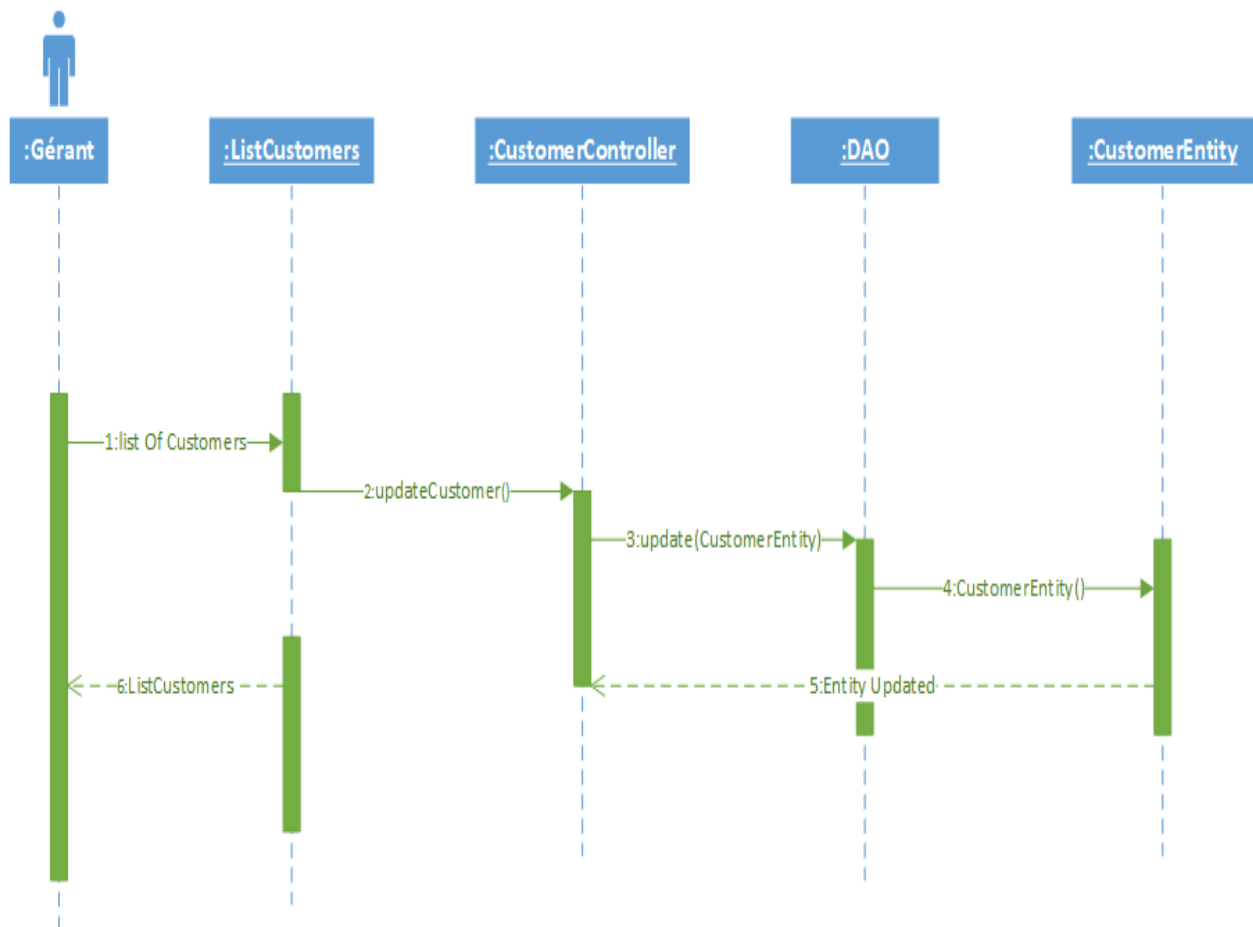


Figure 17 : Diagramme de séquence détaillé "Modifier Customer"

4. Diagramme d'activité

A ce niveau, Il est intéressant de comprendre la logique comportementale de la solution à proposer. Le diagramme d'activité est un procédé qui permet de d'écrire cet aspect.

4.1. Diagramme d'activité du cas d'utilisation « S'authentifier »

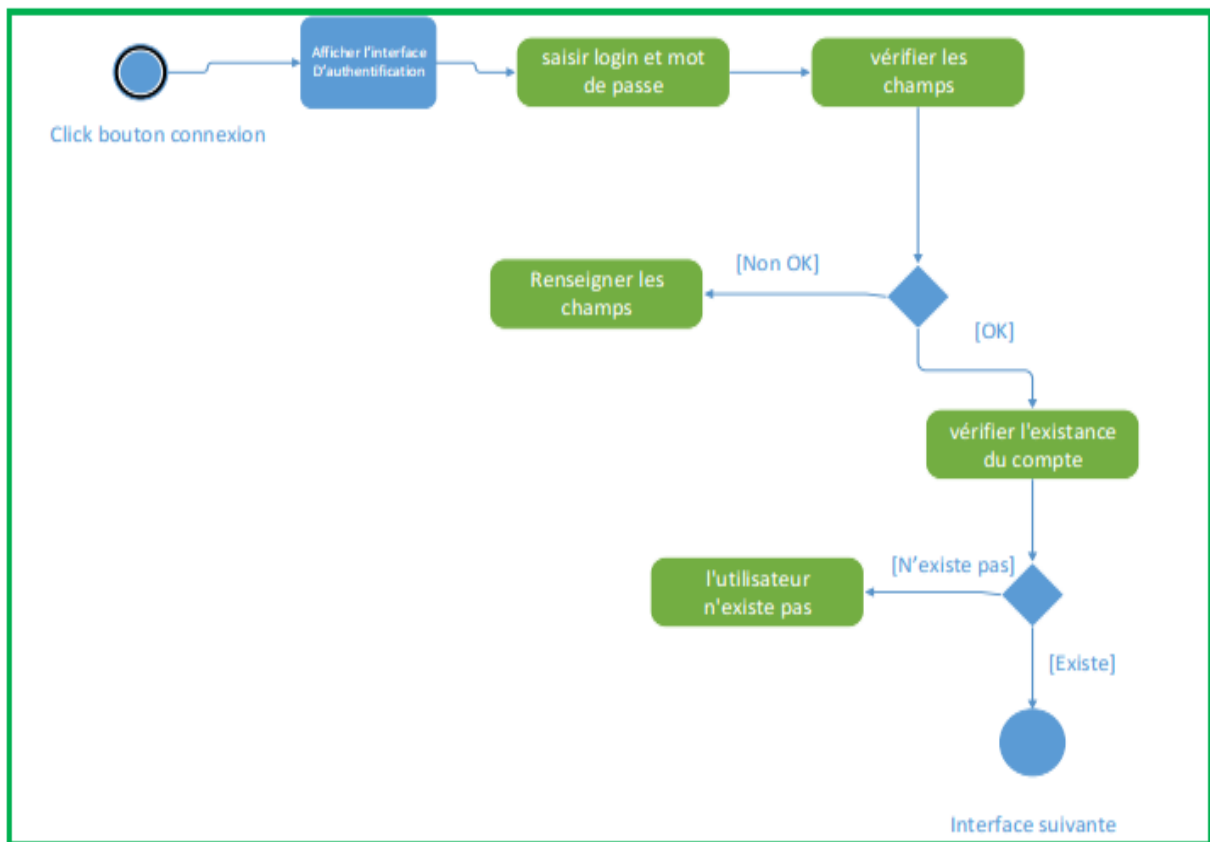


Figure 18 Diagramme d'activité du cas d'utilisation « S'authentifier »

4.2. Diagramme d'activité du cas d'utilisation « gestion projet »

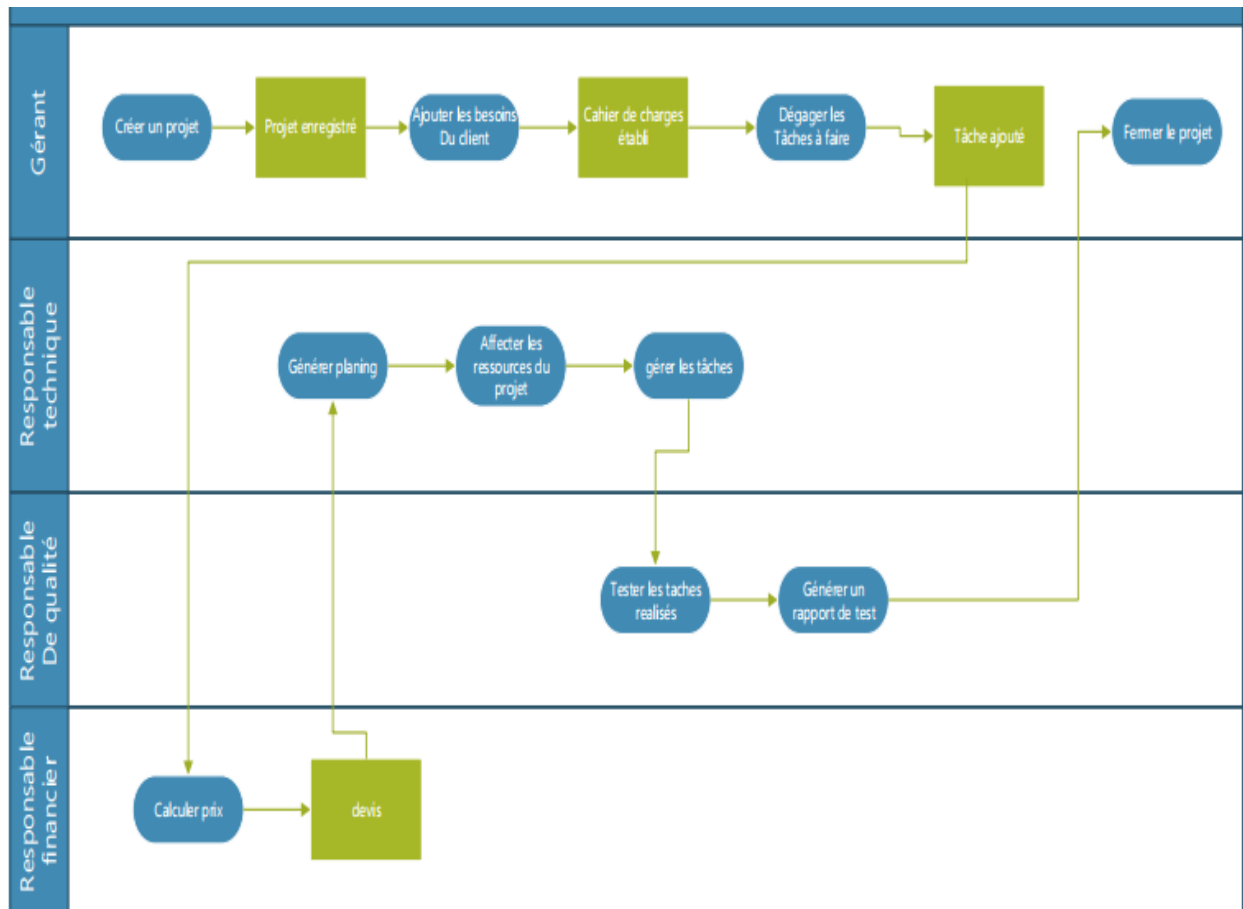


Figure 19 Diagramme d'activité du cas d'utilisation « gestion projet »

Le diagramme d'activités du système montre l'enchaînement et le processus de gestion des projets il comporte quatre acteurs chacun a ses activités. Ceci est entamé et clôturé par le gérant avec ses deux activités (créer un projet, fermer un projet).

Conclusion:

Ce chapitre nous a permis de détailler la conception statique de notre futur système, une conception globale à travers un diagramme des composants et une conception détaillée à l'aide des diagrammes des classes, ainsi que la conception dynamique, par le biais des diagrammes des séquences et d'activités.

Cette décomposition conceptuelle va être très utile lors de la réalisation de notre système qu'on va l'aborder dans le chapitre suivant.

Chapitre 4 : Réalisation

Introduction

Nous entamons cette partie par la présentation de l'environnement de travail logiciel et matériel. Puis, nous exposons la démarche suivie pour la réalisation.

1. Environnement de Travail

En premier lieu, nous allons présenter l'environnement de développement (que ce soit matériel ou logiciel) avec lequel nous avons travaillé.

1.1. Environnement matériel

Les différentes phases de notre projet à savoir la documentation, la spécification des besoins, la conception et le développement, sont réalisés à l'aide d'un ordinateur disposant des caractéristiques suivantes:

- Modèle: ASUS X541U
- Processeur: intel core i7
- Mémoire vive : 8 GO
- Disque dur : 100 GO.
- Système d'exploitation: Windows 10 pro.

1.2. Environnement logiciel

Kit de développement JDK 1.8 : JDK est l'environnement dans lequel le code Java est compilé.

Netbeans 8.2 : Netbeans est un environnement de développement (IDE) destiné aux plusieurs langages y compris le langage java.

GlassFish Server 4.1.1 est serveur d'application Java EE libre écrit en Java, c'est un support complet de Java EE 6 qui est proposé : EJB3, JPA2, JSF2...

Microsoft Visio 13 :Microsoft Visio (officiellement Microsoft Office Visio) est un logiciel de diagrammes et de synoptiques pour Windows qui fait partie de la suite bureautique Microsoft Office mais se vend séparément.

2. Présentation des interfaces

2.1. Processus final de l'application

Cette figure représente le cycle de vie du processus de gestion des projets.

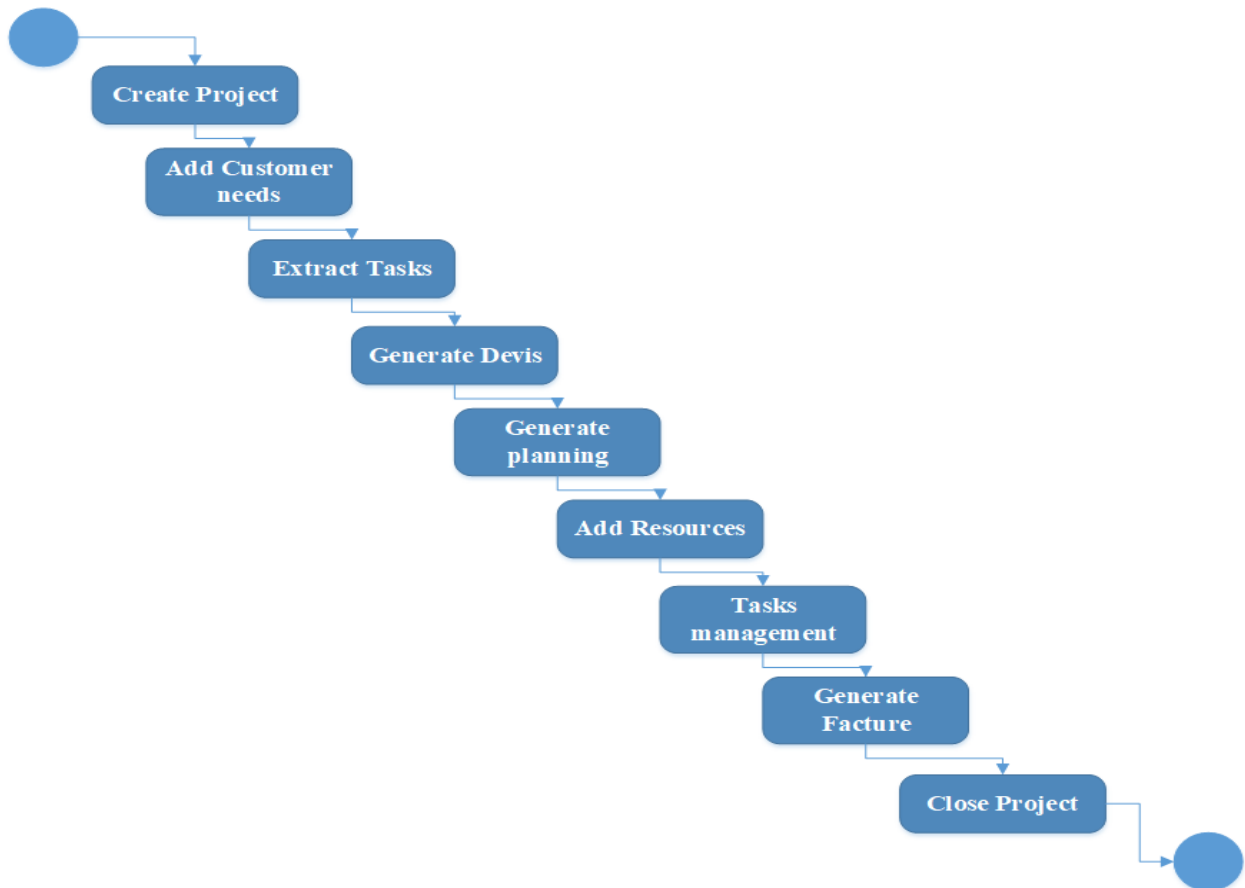


Figure 20 : Processus final de l'application

2.2. Interfaces de l'application d'automatisation de processus de gestion des projets

➤ Interface d'authentification

L'interface d'authentification présentée par la figure ci-dessus est le point d'entrée de notre application.

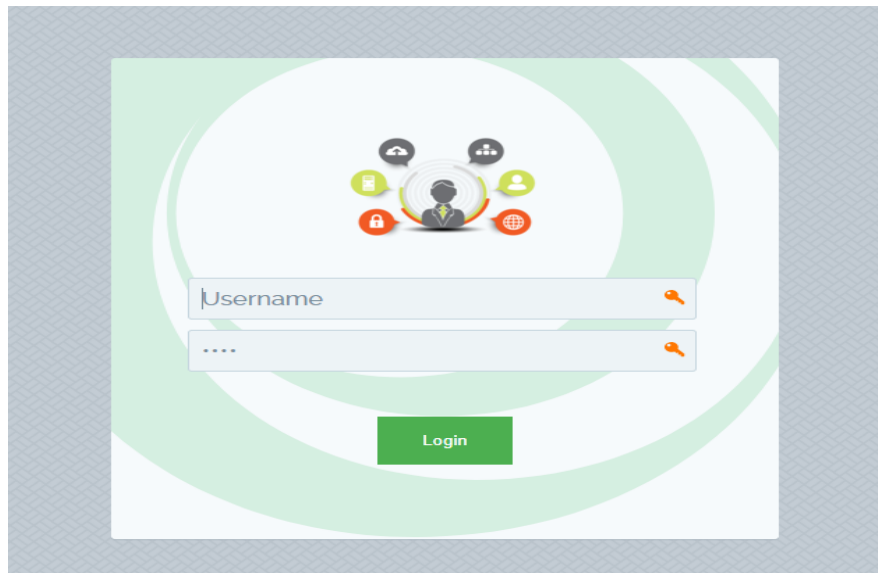


Figure 21 : Interface d'authentification

➤ Création d'un nouveau projet

Le gérant va lancer un projet, il remplit un formulaire et lui affecte les utilisateurs ainsi que le maitre d'ouvrage du projet.

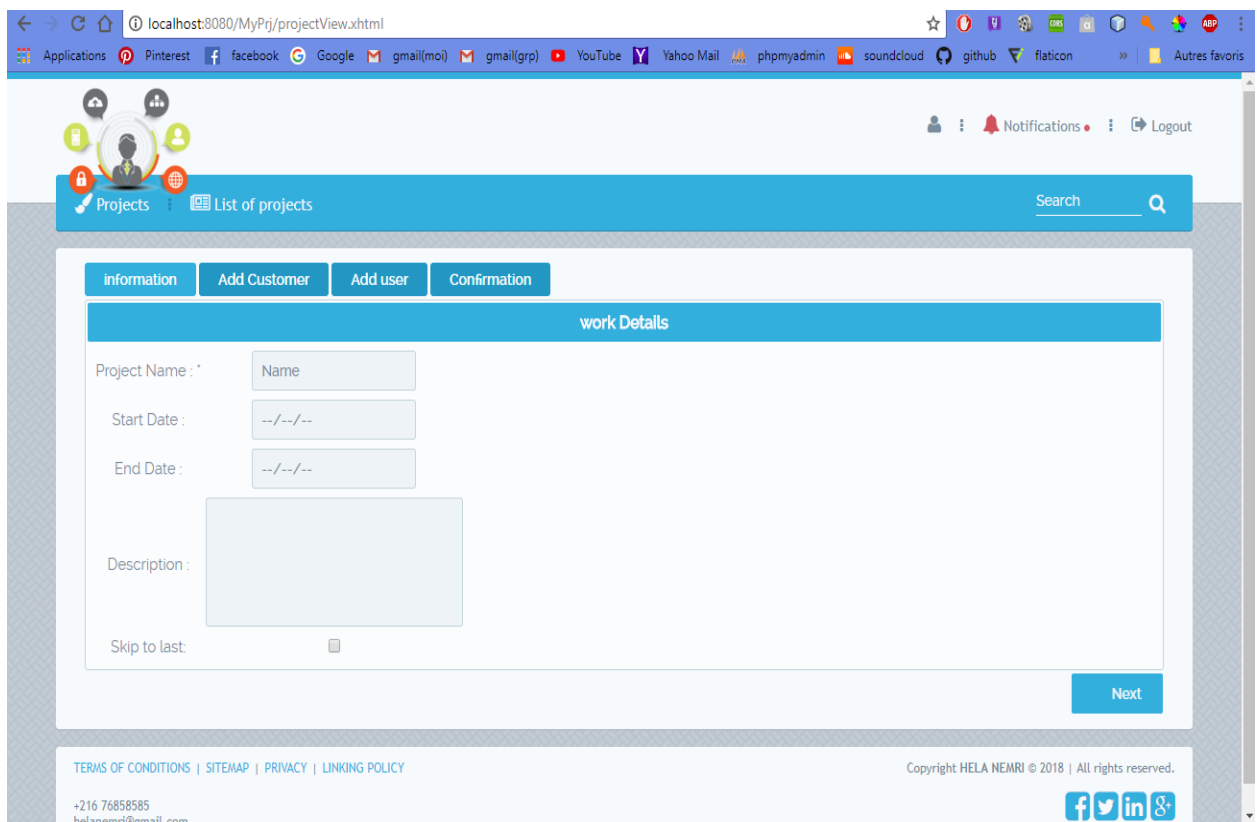


Figure 22 : Interface d'ajout d'un nouveau projet

➤ **Ajouter les besoins du client (cahier des charges)**

Avec cette interface, le gérant remplit le formulaire ci-dessous et ajoute le cahier des charges du client (ce cahier des charges est sous format PDF peut être envoyé par internet ou avec un support de stockage externe).

The screenshot shows a web browser window with the URL `localhost:8080/MyProj/ficheUpload.xhtml`. The page has a blue header with a search bar and a 'Logout' link. The main content area is titled 'Specification details' and contains a form with the following elements:

- Date :** A text input field with a calendar dropdown showing May 2018. The date 10 is selected.
- Document :** A text input field with a 'Choose' button.
- Document type :** A text input field with a 'Choose' button.
- Buttons:** 'Submit', 'Clear', 'Save', and 'Reset Values'.

Figure 23 : Interface d'ajout des besoins du client

➤ Extraire les tâches du projet

The screenshot displays a web application interface for managing project tasks. The browser's address bar indicates the URL is `localhost:8080/MyPrj/tasksProject.xhtml`. The page features a blue navigation bar at the top with links for 'Projects' and 'List of projects', along with a search bar. The main content area is titled 'Task details' and contains a form with the following fields:

- Task Name:
- Number:
- Estimation:
- Description:

Below the form, there are two buttons: 'Save' and 'Reset Values'. The footer of the page includes links for 'TERMS OF CONDITIONS', 'SITEMAP', 'PRIVACY', and 'LINKING POLICY', contact information '+216 76858585 helanemri@gmail.com', and social media icons for Facebook, Twitter, LinkedIn, and Google+.

Figure 24 : Interface d'extraction des taches du projet

➤ Générer un document financier (devis ou facture) :

Cette interface contient l'ensemble des tâches du projet et les éléments liés à la comptabilité, pour générer un document financier, le responsable financier va remplir les champs, ajouter ou modifier une tâche puis pour confirmer les champs et générer un nouveau document et lorsqu' il clique sur le bouton « Save » un document financier sous format PDF est généré.

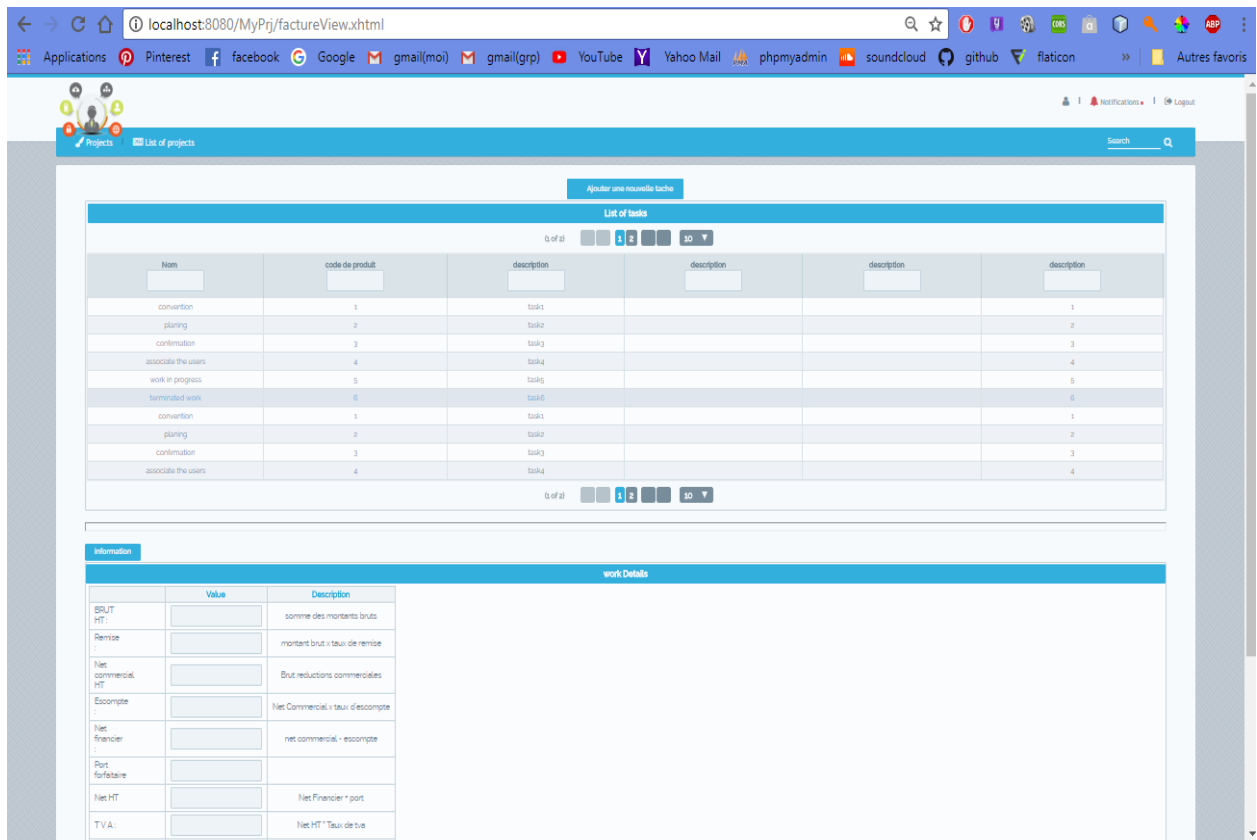


Figure 25 : Interface de génération d'un document financier

Company Name
Address Line 1
Address Line 2
City, State - ZipCode
Country

Account No.	ABC0001
Invoice No.	123456
Invoice Date	05/25/2018

Qty	Item Number	Item Description	Price	Ext Price
1	convention	task1		
2	planing	task2		
3	confirmation	task3		
4	associate the users	task4		
5	work in progress	task5		
6	terminated work	task6		
1	convention	task1		
2	planing	task2		
3	confirmation	task3		
4	associate the users	task4		
5	work in progress	task5		
6	terminated work	task6		
1	convention	task1		
2	planing	task2		
3	confirmation	task3		
4	associate the users	task4		
5	work in progress	task5		
6	terminated work	task6		
	BRUT HT	somme des montants bruts		

Figure 26 : Un document financier (facture)

➤ Générer un contrat de travail

Le gérant génère un contrat de travail avec le client par le remplissage du formulaire qui contient la date de création, une avance de paiement, une description ainsi que les tâches du projet.

The screenshot shows a web browser window with the URL `localhost:8080/MyPj/contratUpload.xhtml`. The browser's address bar and tabs are visible. The main content area displays a table with 4 columns and 8 rows of data. Below the table is a pagination control showing '(1 of 4)' and buttons for pages 1, 2, 3, 4, and 10. Below the table is a form titled 'work Details' with a blue header. The form contains a date field with the value '5/5/18', a text field with the value 'acompte', and a rich text editor with the title 'Description'. The rich text editor has a toolbar with various icons for text formatting and editing. At the bottom right of the form is a blue button labeled 'generate contract'.

convention	1	task1		1
planing	2	task2		2
confirmation	3	task3		3
associate the users	4	task4		4
work in progress	5	task5		5
terminated work	6	task6		6
convention	1	task1		1
planing	2	task2		2
confirmation	3	task3		3
associate the users	4	task4		4

(1 of 4) 1 2 3 4 10

Information

work Details

5/5/18

acompte

Description

generate contract

Figure 27: Interface d'ajout d'un contrat de travail

Thu May 10 00:00:00 WAT 2018

Contrat de travail

Entre name client

Les taches a faire

- 1-convention : task1
- 2-planing : task2
- 3-confirmation : task3
- 4-associate the users : task4
- 5-work in progress : task5
- 6-terminated work : task6

Figure 28 : Contrat de travail

➤ **Générer un planning de travail :**

Cette interface permet au responsable technique d'ajouter un planning de travail pour estimer les durées des tâches avec leurs ordres, le planning est généré avec un diagramme de GANTT, la représentation de ce dernier est en fonction de deux attributs le point d'estimation qui représente le nombre des jours et le rang de la tache dans le projet.

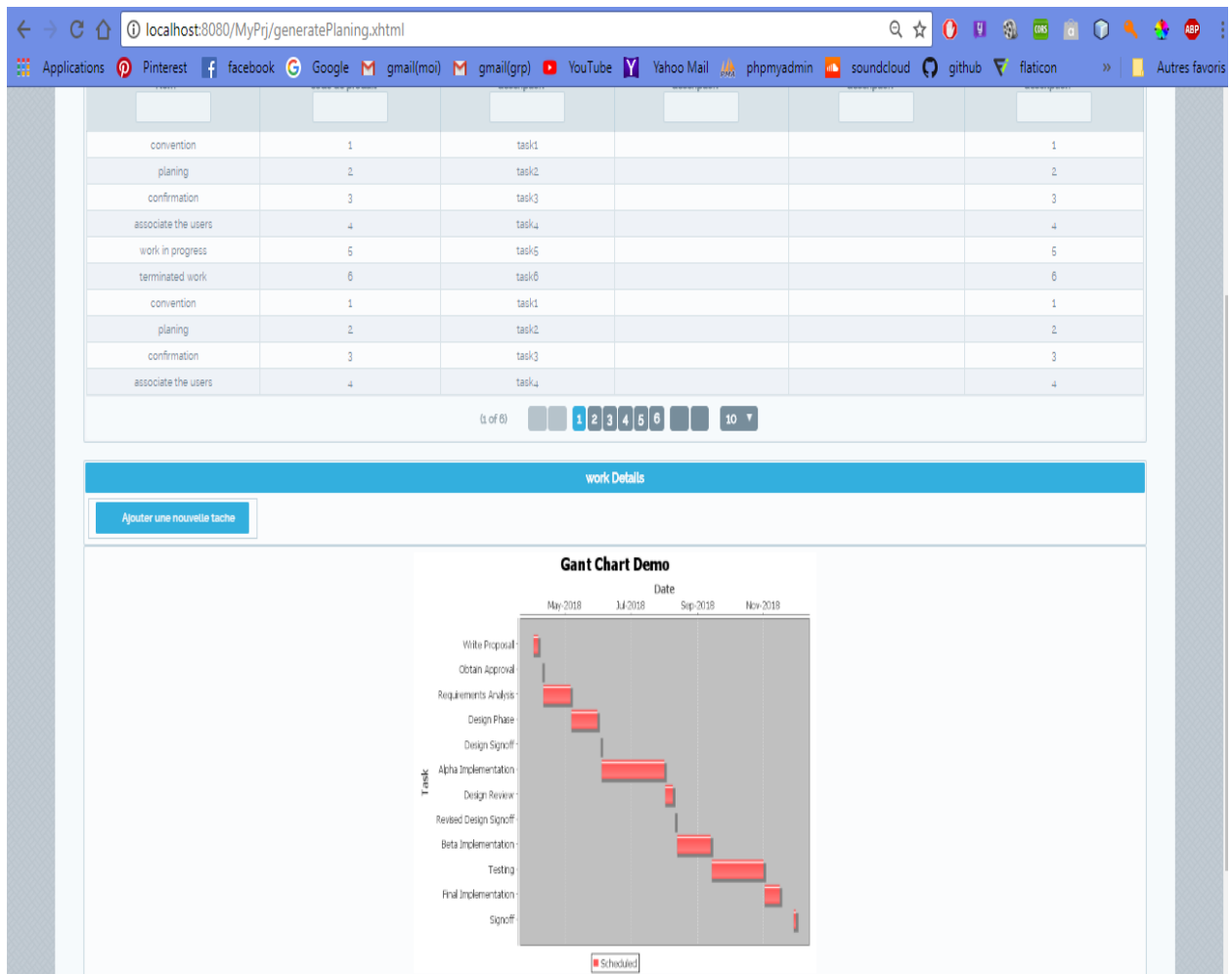


Figure 29 : Interface du planning de travail

➤ Processus de gestion des tâches du projet :

Avec cette interface, le responsable technique a le droit de contrôler l'avancement des tâches et de confirmer ou pas leurs réalisations.

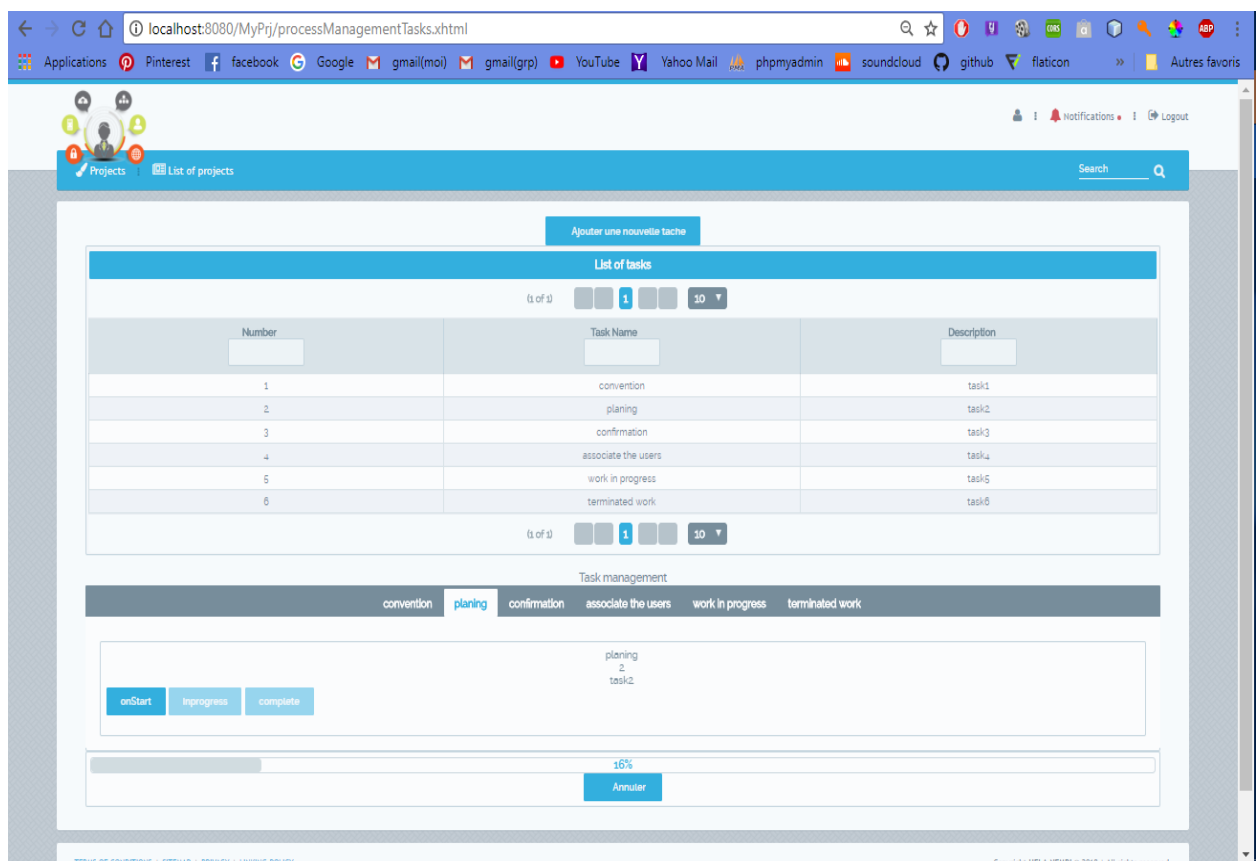


Figure 30 : Interface de gestion des taches

➤ Gestion des utilisateurs :

Cette interface permet l'ajout d'un nouvel utilisateur et l'affecter à un groupe.

Figure 31 : Interface d'ajout d'un utilisateur

Conclusion

Au cours de ce dernier chapitre, nous avons décrit l'environnement matériel, l'environnement logiciel, les différentes interfaces de notre application ainsi que les différentes étapes du processus que nous avons suivi pour la gestion des projets.

Conclusion générale et perspectives

L'objectif de notre projet était d'élaborer une application permettant d'automatiser le processus de gestion des projets, en essayant d'améliorer le système actuel le plus possible avec d'autres fonctionnalités comme la génération des notifications système et la gestion des documents électroniques.

Pour accomplir ce travail, nous avons utilisé la démarche de la méthode « processus 2TUP » qui utilise les notations d'UML, ce qui nous a permis de relever les avantages d'un travail itératif dans le monde réel et qui s'adapte à l'évolution des besoins pour un futur système stable et évolutif.

La réalisation de ce projet nous a permis de mettre en pratique plusieurs connaissances théoriques que nous avons acquis tous au long de notre formation à l'EPI. Mais ce projet nous a également offert une véritable opportunité pour découvrir de nouvelles technologies que les entreprises aujourd'hui utilisent plus précisément d'apprendre à travailler avec la technologie Java EE et de se familiariser d'avantage avec l'architecture MVC.

Par rapport aux objectifs que nous nous sommes fixés, nous avons réussi à réaliser nos attentes. Nous avons rencontré quelques problèmes techniques.

Nous pouvons ajouter quelques perspectives à notre application comme l'ajout d'un espace client pour informer ceci sur les différentes progressions et changement du travail, et aussi l'ajout d'un forum de discussion entre les membres de la société pour faciliter le partage des connaissances et des conseils entre eux.

Références

- [1] « JIRA Workflow » [En ligne]. Available: <https://confluence.atlassian.com/jira064/configuring-workflow-720412524.html>.

- [2] Polymer, « Concevoir une application basée sur des workflows » [En ligne]. Available: <https://www.nuxeo.com/fr/blog/build-a-workflow-based-application-with-polymer/>.

- [3] A. S. Kamla, « Présentation des methodes d'analyse, » 2013. [En ligne]. Available: http://www.memoireonline.com/01/17/9565/m_Application-de-gestion-commerciale-des-produits-alimentaires14.html.

Glossaire

- **API:** Application Programming Interface
- **Http:** HyperText Transfer Protocol
- **Html:** HyperText Markup Language
- **IHM:** Interface Homme Machine
- **ORM:** Objet-Relationnel Mapping
- **SQL:** Structured Query Language
- **UML:** Unified Modeling Language
- **URL:** Uniform Resource Locator
- **XP:** eXtreme Programming
- **2TUP :** 2 Tracks Unified Process

ABSTRACT

The traditional project management process may be sufficient for a small company but with the growth of the market value and the increase in the number of its projects, excellence and perfection become a requirements and not options.

It is in this context that our project is located. He is about making a web application that allows improving and automating the process of management in a company.

RÉSUMÉ

Le processus de la gestion des projets traditionnel peut être suffisant pour une société de petite taille mais avec la croissance de la valeur sur le marché et l'augmentation du nombre de ses projets l'excellence et la perfection deviennent des exigences et ne pas des options.

C'est dans ce cadre que se situe notre projet. Il consiste à réaliser une application web qui permet d'améliorer et d'automatiser le processus de la gestion des projets de l'entreprise.

Mot clé : Processus, 2TUP, Workflow, Projet, Entreprise, Automatisation, JEE, ORM, MVC, Gantt.