

# Remerciements

Nous tenons, avant de présenter notre travail, à exprimer notre grande reconnaissance envers les personnes qui nous ont, de près ou de loin, apporter leurs soutiens. Qu'ils trouvent ici collectivement et individuellement l'expression de toute notre gratitude.

Nous tenons à remercier tout particulièrement et à témoigner toute notre reconnaissance à Mr.Chtioui Mounir pour l'expérience enrichissante et pleine d'intérêt qu'il nous a fait vivre durant la période du stage et pour tous les conseils et les informations qu'il nous a prodigué.

Aussi, nous exprimons notre parfaite reconnaissance et nos remerciements à notre encadrant Mr. Kebair Fahem pour le temps qu'il a bien voulu consacrer à l'encadrement et le suivi de ce travail ; les conseils qu'il nous a prodigué après son minutieuse lectures et pour les réunions qui ont rythmées les différentes étapes de la rédaction de ce rapport. Les discussions que nous avons tenus ont permis d'orienter ce travail d'une manière sûre et pertinente. Nous le remercions vivement pour son effort, sa disponibilité et surtout ses conseils qui ont largement contribué à rehausser la valeur de ce travail.

Que les membres de jury trouvent, ici, l'expression de nos remerciements pour l'honneur qu'ils nous font en acceptant de juger ce travail.

# Table des matières

<b>Liste des figures</b>	<b>v</b>
<b>Liste des tableaux</b>	<b>vi</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Cadre du projet</b>	<b>3</b>
Introduction . . . . .	4
I Cadre du projet . . . . .	4
I.1 Présentation de l'organisme . . . . .	4
I.2 Problématique . . . . .	6
I.3 Etude de l'existant . . . . .	6
I.4 Solution proposée . . . . .	6
II Méthodologie de gestion de projet . . . . .	7
II.1 Choix de la méthodologie . . . . .	7
II.2 Présentation de la méthodologie SCRUM . . . . .	8
II.2.1 Principes essentiels de la méthode . . . . .	9
II.2.2 Organisation . . . . .	10
Conclusion . . . . .	10
<b>2 Planification</b>	<b>11</b>
Introduction . . . . .	12
I Planification d'un projet . . . . .	12
I.1 Principe de découpage d'un projet . . . . .	13
I.1.1 Décomposition en produit PBS . . . . .	13
I.1.2 Décomposition en tâches WBS . . . . .	14

I.1.3	Diagramme de Gantt . . . . .	15
I.1.4	Répartition des rôles OBS . . . . .	15
II	Présentation du planning de notre projet : . . . . .	16
II.1	Décomposition en sous-produits . . . . .	16
II.2	Décomposition en tâches . . . . .	17
II.3	Enchaînement des tâches . . . . .	18
	Conclusion . . . . .	21
<b>3</b>	<b>Analyse et spécification des besoins</b>	<b>22</b>
	Introduction . . . . .	23
I	Le Backlog du produit . . . . .	23
II	Identification des acteurs . . . . .	24
III	Spécification des besoins . . . . .	26
III.1	Besoins fonctionnels . . . . .	26
III.2	Besoins non fonctionnels . . . . .	27
IV	Analyse des besoins . . . . .	27
IV.1	Diagramme de cas d'utilisation du module «Gestion des change- ments» . . . . .	27
IV.1.1	Diagramme de cas d'utilisation détaillé du module «Faire le suivi des demandes» . . . . .	29
IV.1.2	Diagramme de cas d'utilisation détaillé du module « Demande d'intervention » . . . . .	31
IV.1.3	Diagramme de cas d'utilisation détaillé du module « Traiter les demandes des clients » . . . . .	34
V	Diagramme d'activité du système gestion des changements . . . . .	37
	Conclusion . . . . .	39
<b>4</b>	<b>Réalisation</b>	<b>40</b>
	Introduction . . . . .	41
I	Architecture logicielle de notre solution . . . . .	41
II	Environnement de développement logiciel . . . . .	43
III	Choix technologiques . . . . .	43
III.1	Plateforme JAVA EE . . . . .	43

III.2	Le framework Struts 2 pour l'implémentation de la couche pré- sentation . . . . .	44
III.3	Hibernate pour l'implémentation de la couche d'accès au données	44
III.4	HTML5, CSS3, jQuery et JavaScript pour la construction des interfaces riches . . . . .	45
IV	Diagramme de classe . . . . .	45
V	Déroulement des Sprints . . . . .	48
V.1	Sprint1- « Authentification et demande d'intervention » . . . .	49
V.1.1	Conception détaillée du sprint 1 . . . . .	49
V.1.2	Développement du sprint 1 . . . . .	53
V.1.3	Tests relatifs au sprint 1 . . . . .	58
V.2	Sprint 2 - « Traitement des requêtes des clients» . . . . .	59
V.2.1	Conception détaillée du sprint 2 . . . . .	59
V.2.2	Développement du sprint 2 . . . . .	62
V.2.3	Tests relatifs au sprint 2 . . . . .	64
V.3	Sprint 3 – «Suivi des projets et consultation de leurs historiques»	65
V.3.1	Conception détaillée du sprint 3 . . . . .	65
V.3.2	Développement du sprint 3 . . . . .	67
V.3.3	Tests relatifs au sprint 3 . . . . .	70
V.4	Sprint 4 – «Consultation de la liste des documents disponibles»	71
V.4.1	Développement du sprint 4 . . . . .	71
V.4.2	Tests relatifs au sprint 4 . . . . .	72
VI	Chronogramme . . . . .	73
	Conclusion . . . . .	73
	<b>Conclusion générale</b>	<b>74</b>
	<b>Bibliographie</b>	<b>76</b>
	<b>Webographie</b>	<b>77</b>
	<b>Annexes</b>	<b>78</b>

# Table des figures

1.1	Organigramme d'Amen Bank . . . . .	5
1.2	Cycle de vie de la méthode SCRUM . . . . .	9
2.1	Découpage structurel – PBS . . . . .	14
2.2	Découpage en tâches – WBS . . . . .	15
2.3	Organigramme basé sur le découpage PBS . . . . .	17
2.4	Organigramme de la décomposition en tâches WBS . . . . .	18
2.5	Diagramme de Gantt théorique . . . . .	20
3.1	Backlog de notre produit . . . . .	24
3.2	Diagramme de cas d'utilisation «Gestion des changements » . . . . .	28
3.3	Diagramme de cas d'utilisation du module « Faire le suivi des demandes» . . . . .	29
3.4	Diagramme de cas d'utilisation du module « Demande d'interventions» . . . . .	32
3.5	Diagramme de cas d'utilisation du module «Traiter les demandes» . . . . .	35
3.6	Diagramme d'activité «Gestion des changements» . . . . .	38
4.1	Fonctionnement de Struts2 . . . . .	42
4.2	Diagramme de classe «Gestion de changement» . . . . .	46
4.3	Planning du travail . . . . .	48
4.4	Diagramme de séquence « Authentification » . . . . .	50
4.5	Diagramme de séquence « Demande d'un nouveau projet » . . . . .	51
4.6	Diagramme de séquence « Demande de modification ou de correction » . . . . .	52
4.7	Interface d'authentification . . . . .	53
4.8	Interface d'authentification cas de champs vides . . . . .	54
4.9	Interface d'authentification cas des données erronées . . . . .	55
4.10	Interface d'accueil . . . . .	56

4.11 Interface des demandes clientes . . . . .	56
4.12 Interface des demandes d'un nouveau projet . . . . .	57
4.13 Interface de demande d'amélioration d'une tâche . . . . .	58
4.14 Diagramme de séquence « Téléchargement des documents » . . . . .	60
4.15 Diagramme de séquence « Rattachement des documents » . . . . .	61
4.16 Interface des utilisateurs du département informatique d'AMEN BANK	62
4.17 Interface de consultation des notifications . . . . .	63
4.18 Interface de validation des documents . . . . .	64
4.19 Diagramme de séquence « Suivi de la liste des demandes et de leurs historique» . . . . .	66
4.20 Diagramme de séquence « Ajout d'une réclamation» . . . . .	67
4.21 Interface de suivi des demandes clientes . . . . .	68
4.22 Interface de consultation des demandes en cours . . . . .	69
4.23 Interface des réclamations . . . . .	70
4.24 Interface de suivi des projets réalisés . . . . .	71
4.25 Interface de consultation des différents documents d'une demande de changement . . . . .	72
4.26 Diagramme de Gantt réel . . . . .	73

# Liste des tableaux

3.1	Description textuelles du cas d'utilisation «Faire le suivi des demandes»	30
3.2	Description textuelles du cas d'utilisation «Demande d'interventions» .	33
3.3	Description textuelles du cas d'utilisation «Traiter les demandes» . . .	36

# Introduction générale

L'un des défis auxquels se confronte une entreprise moderne est la nécessité de maintenir une vue d'ensemble sur les nombreux changements dans son système d'information, quelque soit leur nature. Ceci est l'un des piliers garantissant sa compétitivité et sa pérennité.

La gestion des changements est le processus qui consiste à gérer de façon systématique et efficace l'établissement, l'exécution et l'administration des changements afin d'optimiser le rendement et de réduire les risques inhérents. La gestion des changements couvre le cycle de vie d'un nouveau projet, la modification et la correction des applications informatique auxquels participent plusieurs intervenants.

Les requêtes, provenant de la part des utilisateurs et des propriétaires métiers concernant les demandes de développement ou de modifications intervenant sur le système, sont le plus souvent gérées manuellement par les différents acteurs. Leurs circuits de validation sont encore informels. Cette manière archaïque de gestion des changements empêche les utilisateurs d'avoir une vision à la fois synthétique et dynamique de leurs demandes.

La prise de conscience des risques liés à l'absence de gestion des changements grandit de plus en plus : des résultats incohérents, la non-conformité aux requêtes des utilisateurs et de leurs exigences, etc. Dès lors, la direction centrale de l'organisation et du système d'information d'AMEN BANK se dit prête à s'investir pour avoir un système de gestion des changements qui résout les diverses demandes des utilisateurs. C'est dans ce cadre que s'inscrit notre projet de fin d'études au sein d'AMEN BANK qui a repéré l'importance de réaliser une application générique de gestion des changements intitulée « Etude, conception et développement d'une application de gestion des



changements : nouveaux développements, modifications et corrections des applications informatiques » pour assurer l'organisation et la sécurité des actions intervenant sur le système d'information de la banque.

L'étude du cycle de vie de la gestion des nouveaux projets, la réalisation du développement des différents modules de la solution sont détaillés dans le rapport de ce présent projet de fin d'études.

Ce rapport est articulé autour de quatre chapitres :

Le premier chapitre comporte une brève présentation de l'organisme d'accueil AMEN BANK et du cadre général de ce projet. Il expose en effet, l'étude de l'existant et met l'accent sur la solution proposée. Et il aborde à la fin la méthodologie appliquée pour assurer le bon déroulement de notre travail.

Le deuxième chapitre se focalise sur la planification de notre travail. Il présente alors le découpage de notre projet nécessaire pour ressortir les différents modules à développer en respectant les délais.

Le troisième chapitre présente tout d'abord le backlog du produit de notre système. Il détaille aussi les besoins fonctionnels et non fonctionnels et aborde l'analyse de ces besoins en se basant sur le langage de modélisation UML.

Dans le quatrième chapitre, il y a une présentation de la phase de réalisation de notre application. Cette étape est basée sur la méthodologie AGILE SCRUM dans le but d'une bonne gestion de projet.

Nous clôturons par une conclusion générale qui présente une analyse du travail réalisé au sein d'AMEN BANK. Cette partie met en évidence non seulement un résumé du travail effectué tout au long de la période du stage de projet de fin d'études mais aussi des propositions des diverses perspectives.

# Chapitre 1

## Cadre du projet

# Introduction

Au cours de ce chapitre, nous allons nous intéresser tout d'abord à la présentation du cadre de notre projet. Il s'agit en effet d'une présentation de la banque pour laquelle ce travail a été réalisé. Nous exposons le sujet du travail qui nous a été confié ainsi que l'environnement qui a servi à son achèvement.

Après l'exposition de la problématique qui a engendré ce travail, nous abordons l'étude de l'existant et nous présentons ensuite la solution proposée. Enfin, nous entamons ce chapitre par une présentation de la méthodologie de gestion de projet adaptée.

## I Cadre du projet

### I.1 Présentation de l'organisme

Amen Bank est la cinquième banque en Tunisie et est la première banque privée, totalement vouée aux services de sa clientèle et au développement de l'économie nationale. Amen Bank se fixe trois objectifs essentiels à savoir : servir, innover, et faire réussir. Amen Bank opère sur l'ensemble du territoire tunisien et intervient essentiellement dans les secteurs des services et de l'industrie. Parmi les douze banques commerciales tunisiennes, Amen Bank se distingue par sa longue histoire qui remonte à plus d'un siècle.[w3]

C'est une institution qui dispose d'atouts majeurs :

- D'actionnaires stables (68% détenue par le groupe Amen, famille Ben Yedder), ce qui garantit la stabilité de la gestion et de la stratégie.
- Des partenaires de taille : Tunisie Leasing, COMAR, HAYAT.
- Un positionnement de banque universelle qui offre un large spectre de produits aux entreprises et aux ménages.
- Une forte productivité : une centaine d'agences pour un effectif réduit de 960 personnes.

Amen Bank se spécialise dans ses activités, grâce à :

- Un noyau dur.

- Un savoir d'une équipe d'experts.
- Un contact direct avec l'agence.
- Des locaux dédiés à l'activité internationale, tels que dealing room et une plateforme internationale : open space.

Amen Bank dispose d'un réseau de 105 agences, 105 distributeurs automatiques de billets (DAB) dont 15 hors site, 708 TPE et de 518 zones connectés au réseau international VISA et MASTER CARD.[w3]

- **Organigramme d'Amen Bank**

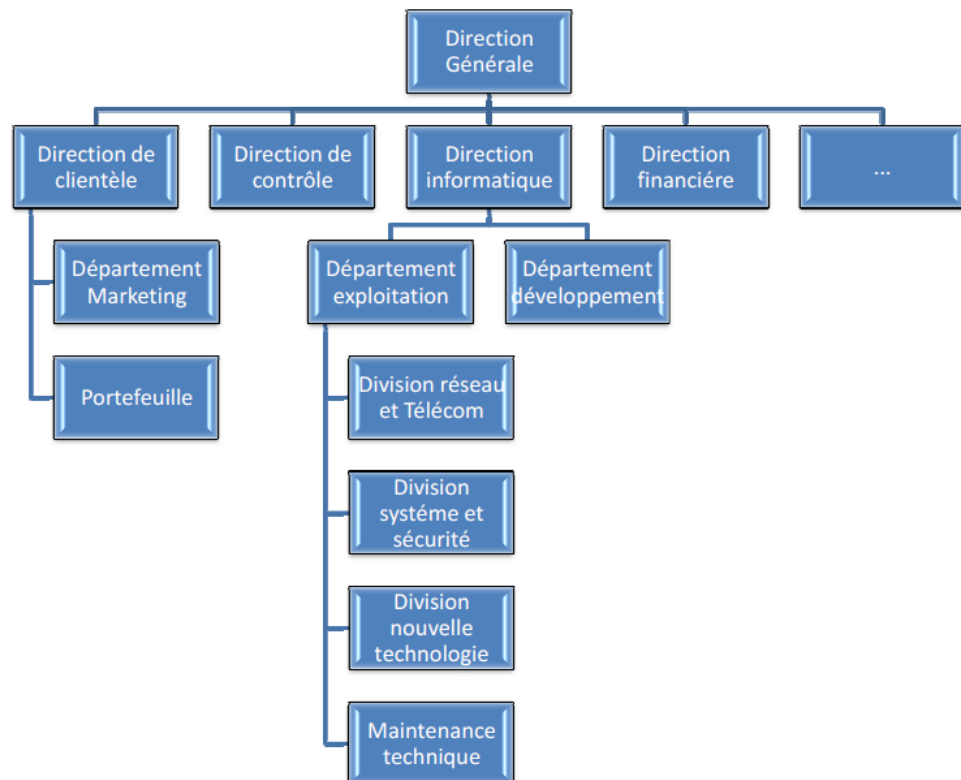


FIGURE 1.1: Organigramme d'Amen Bank

### I.2 Problématique

A l'heure actuelle, les entreprises ont besoin de suivre de près les changements sur leurs systèmes d'information pour pouvoir répondre aux exigences des différents utilisateurs. Ce besoin est apparu dès qu'il y a eu une prise de conscience de la situation des entreprises à cause de la mauvaise administration de la gestion des changements.

Cela a engendré de graves conséquences, entre autres, la non-conformité aux termes et aux conditions prédéfinies ainsi que le non-respect des délais. C'est dans ce cadre qu'AMEN BANK a constaté qu'il est urgent de développer une application de gestion des changements pour répondre à son propre besoin de la gestion de nouveaux projets ou de la modification des applications déjà réalisées.

Pour cette raison, nous avons intégré le département informatique au sein d'AMEN BANK pour accomplir cette mission. Notre projet consiste alors à réaliser une application permettant à la banque en question de contrôler, de communiquer et de répondre plus efficacement aux différents demandes métier dans toutes les phases du cycle de vie du développement applicatif, depuis la réception de la demande jusqu'à sa mise en place.

### I.3 Etude de l'existant

Danc cette étude de l'existant, nous nous sommes basées sur des entretiens avec le personnel d'AMEN BANK pour connaître les problèmes dans la gestion des nouveaux projets, la correction des bugs ou l'amélioration des tâches.

Grâce à ces réunions, nous avons conclu que la direction informatique fait face à des difficultés surtout lors de l'administration des nouveaux projets. Pire encore, certains acteurs n'ont pas accès ou ne sont pas au courant de certains changements qui les concernent directement ou indirectement. Cela est dû à l'absence d'une traçabilité qui s'avère importante pour la réussite d'une organisation. Les résultats de notre enquête sont illustré dans l'annexe B.

### I.4 Solution proposée

Face aux anomalies de la procédure actuelle et grâce à la technique d'interview que nous avons menée en nous basant sur un guide d'entretiens, nous proposons alors

de concevoir une nouvelle application permettant l'automatisation de la gestion des changements. Nous avons pu alors non seulement dégager les besoins d'AMEN BANK mais aussi de connaître les exigences de ses employés. Les objectifs de ce travail sont :

- Permettre aux employés d'AMEN BANK de faire des demandes de changements :  
soit demande d'un nouveau projet, soit amélioration d'une tâche ou correction des bugs,
- Permettre aux employés de la direction centrale du système de l'information et de l'organisation d'AMEN BANK de traiter les requêtes de demande de changement chacun selon leur rôle,
- Permettre au différents utilisateurs de notre système de télécharger et rattacher les documents nécessaires d'une demande de changement selon leurs privilèges,
- Notifier les différents utilisateurs de notre application s'il y a un nouveau document à télécharger,
- Notifier le département informatique d'AMEN BANK s'il y a des demandes de changements à traiter,
- Permettre aux acteurs de notre système de faire des réclamations concernant une tâche bien particulière,
- Suivre déroulement de la demande de près,
- Mettre en place un système permettant à un employé d'AMEN BANK, utilisateur de notre application, de consulter l'historique de ses demandes,
- Permettre aux employés de la direction informatique de valider les différentes phases d'une demande chacun selon son rôle.

## II Méthodologie de gestion de projet

### II.1 Choix de la méthodologie

Le choix entre une méthode et une autre, dépend de la nature du projet et de sa taille. Pour des projets de petite taille et dont le domaine est maîtrisé, par exemple, un cycle de vie en cascade s'avère largement suffisant. Lorsqu'il s'agit d'un projet où les données ne sont pas réunies dès le départ, où les besoins sont incomplets voire flous, il faut s'orienter vers une méthode itérative ou orientées prototypes.

Parmi les méthodes itératives, nous pouvons distinguer les méthodes AGILE largement utilisées de nos jours à travers le monde. Une méthode AGILE est menée dans un esprit collaboratif et s'adapte aux approches incrémentales. Elle engendre des produits de haute qualité tout en tenant compte de l'évolution des besoins du client.

Une méthode AGILE assure une meilleure communication avec le client et une meilleure visibilité du produit livrable. Elle permet aussi de gérer la qualité en continu et de détecter des problèmes le plus tôt au fur et à mesure, permettant ainsi d'entreprendre des actions correctrices sans trop de pénalités dans les coûts et les délais.

Il y a moult méthodes AGILE et il ne s'agit pas de choisir la meilleure méthode parmi celles existantes. Il s'agit plutôt de sélectionner la méthode la plus adaptée à notre projet.

La nature de projet qui doit être évolutif et dont tous les besoins n'ont pas encore été totalement identifiés, nous a orientées vers une méthode de type AGILE et plus particulièrement SCRUM.

## II.2 Présentation de la méthodologie SCRUM

Le principe de la méthodologie SCRUM est de développer un logiciel de manière incrémentale en maintenant une liste totalement transparente des demandes d'évolutions ou de corrections à implémenter.

Avec des livraisons très fréquentes, toutes les 4 semaines en moyenne, le client reçoit un logiciel fonctionnel à chaque itération. Plus nous avançons dans le projet, plus le logiciel est complet et possède toujours de plus en plus de fonctionnalités.

Pour cela, la méthode s'appuie sur des développements itératifs à un rythme constant d'une durée de 2 à 4 semaines. La figure 1.2 illustre le cycle de vie de la méthode SCRUM :[3]



FIGURE 1.2: Cycle de vie de la méthode SCRUM

Comme nous pouvons le remarquer dans cette figure, pour mettre en place la méthode SCRUM, il faut tout d'abord définir les différentes fonctionnalités de notre application qui forment le backlog du produit. Ensuite, nous procédons à la planification du sprint pour définir le plan détaillé d'une itération.

Les sprints durent généralement deux à quatre semaines. Durant un sprint, il y a toujours des réunions quotidiennes entre les différents collaborateurs du projet afin de présenter l'état d'avancement des différentes tâches en cours, les difficultés rencontrées ainsi que les tâches restantes à réaliser.

Une fois le produit partiel est prêt, nous vérifions la conformité de ce qui a été fait durant le sprint et nous pouvons alors l'améliorer en procédant à l'étape de rétrospective.

### II.2.1 Principes essentiels de la méthode

Nous pouvons remarquer quatre valeurs principales dans les méthodes agiles :

- L'équipe : nous nous concentrons sur les personnes et leurs interactions plutôt que sur les processus et les outils,
- L'application : le plus important c'est d'avoir une application fonctionnelle plutôt que d'avoir une documentation complète,
- La collaboration : cette méthode se base sur la collaboration avec le client,
- L'acceptation du changement : nous ne suivons pas un plan fixe mais nous réagissons à chaque nouveau changement.



### II.2.2 Organisation

La méthodologie SCRUM fait intervenir 3 rôles principaux qui sont :

- Product owner : dans la majorité des projets, le responsable produit (Product owner) est le responsable de l'équipe projet client. C'est lui qui va définir et prioriser la liste des fonctionnalités du produit et choisir la date et le contenu de chaque sprint sur la base des valeurs (charges) qui lui sont communiquées par l'équipe,
- Scrum Master : véritable facilitateur sur le projet, il veille à ce que chacun puisse travailler au maximum de ses capacités en éliminant les obstacles et en protégeant l'équipe des perturbations extérieures. Il porte également une attention particulière au respect des différentes phases de SCRUM,
- Equipe : l'équipe s'organise elle-même et elle reste inchangée pendant toute la durée d'un sprint. Elle doit tout faire pour délivrer le produit,
- Intervenants : ils observent et donnent des conseils à l'équipe.

Dans notre projet, nous pouvons distinguer les rôles suivants :

- Product owner : M. Mounir Chtioui,
- Scrum Master : M. Mounir Chtioui
- Testeurs : Mlle. Abir Omri,
- Développeurs : Donia Hammami et Amira Azzouz.

## Conclusion

Tout au long de ce chapitre, nous avons évoqué le cadre général du projet. Nous avons commencé tout d'abord par une présentation de l'organisme d'accueil qui a été suivie par une étude de l'existant. Ceci nous a permis de comprendre les besoins et d'envisager la solution la plus adéquate aux attentes du client. Dans le prochain chapitre, nous détaillons la planification de notre projet en se basant sur la démarche de découpage d'un projet.

## Chapitre 2

### Planification

# Introduction

Avant de démarrer un projet, il est nécessaire de le définir et de le planifier afin de bien le piloter et d'atteindre les objectifs fixés par le cahier de charges. La planification d'un projet consiste à identifier les tâches, estimer leurs charges, les ordonnancer afin d'organiser des ressources humaines et matérielles qui permettront d'atteindre avec succès les buts et les objectifs prédéfinis dans les délais.

Dans ce chapitre, nous présentons tout d'abord la notion de planification de projet, en particulier le principe de découpage structurel et temporel. Nous détaillons, par la suite, le plan de notre travail, le résultat d'un découpage selon les normes en vigueur de la gestion de projets.

## I Planification d'un projet

La condition pour réussir toute entreprise réside dans une planification systématique orientée vers l'objectif, établie sur la base du plan d'exécution de projet. La planification doit être réalisée soigneusement, progressivement et conformément aux possibilités, si on veut atteindre les objectifs du plan d'exécution de projet.

La planification n'est pas un processus statique mais dynamique. Dans la phase de planification, l'apparition de nouveaux éléments tout au long des étapes successives nous oblige souvent à modifier les données fixées au début.

Les objectifs de la planification d'un projet sont les suivants :

- être capable, à l'aide d'exemples, de décrire les rapports existants entre les principaux instruments de planification,
- être apte de diviser de manière judicieuse un projet simple en sous-parties,
- être capable de concevoir un plan de structure pour un projet,
- savoir si une planification du déroulement et des détails a été effectuée dans sa totalité,
- planifier les ressources étape par étape,
- concevoir la planification des coûts pour un projet,
- proposer une organisation de projet répondant aux exigences.

## I.1 Principe de découpage d'un projet

Le découpage d'un projet consiste à éclater le projet en différents lots d'activités, afin d'obtenir des sous-ensembles dont la complexité est plus facilement maîtrisable et de suivre efficacement le projet et l'atteinte de la qualité. Pour assurer le découpage d'un projet, il existe trois principales normes internationales :[7]

- découpage en sous-projets ou sous-systèmes ou PBS,
- découpage des travaux en tâches ou WBS : il s'agit de découper les travaux en tâches, phases ou parties. En général, cette décomposition est représentée sous la forme d'une arborescence,
- répartition des rôles et responsabilités ou OBS. Cet organigramme permet de définir :
  - qui fait quoi ?
  - qui est responsable de quoi ?
  - qui est responsable de qui ?

### I.1.1 Décomposition en produit PBS

C'est la première étape dans la planification basée sur le produit. Décomposer un produit en sous-produits permet de clarifier et d'identifier tous les travaux nécessaires à sa création. Les objectifs de cette étape sont les suivants :

- Identifier les produits demandés par les clients,
- Identifier des produits supplémentaires nécessaires pour construire et soutenir les produits à la clientèle,
- Construire un consensus sur les meilleures catégories de produits qui doivent être utilisés pour générer des idées sur ce que les produits doivent être créés ou obtenus.

Donc, la décomposition en sous-projets est un moyen d'établir les exigences des clients sous forme d'arborescence. La figure 2.1 montre un exemple de PBS.[w5]

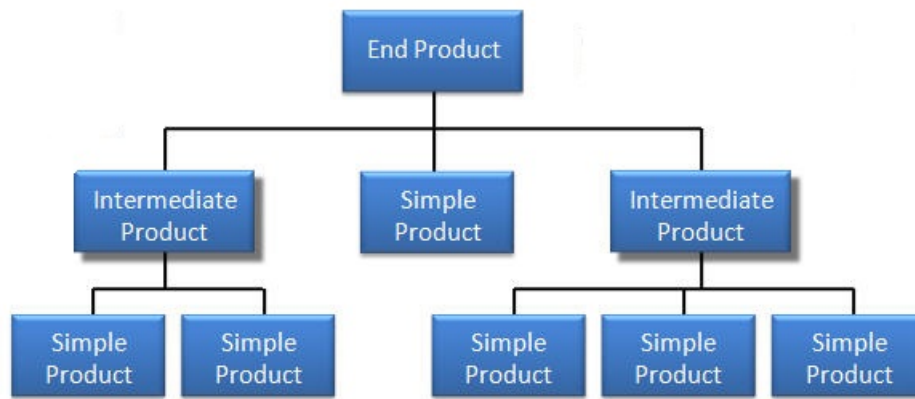


FIGURE 2.1: Découpage structurel – PBS

### I.1.2 Décomposition en tâches WBS

Il s'agit d'un processus de décomposition dans lequel un projet complexe est divisé en petites unités de travail qui peuvent être facilement gérées. Une tâche ou une activité a les caractéristiques suivantes :

- elle a généralement un but unique et peut être gérée comme une entité distincte,
- elle a une date de début et une date de fin spécifique,
- elle nécessite des ressources clairement définies.

Le WBS est alors un organigramme de tâches ou de lots de travaux permettant d'identifier de manière arborescente et exhaustive l'ensemble des activités d'un projet afin d'en maîtriser les coûts, les délais et les performances. Dans la figure 2.2 nous pouvons voir comment peut être représenté le WBS.[4]

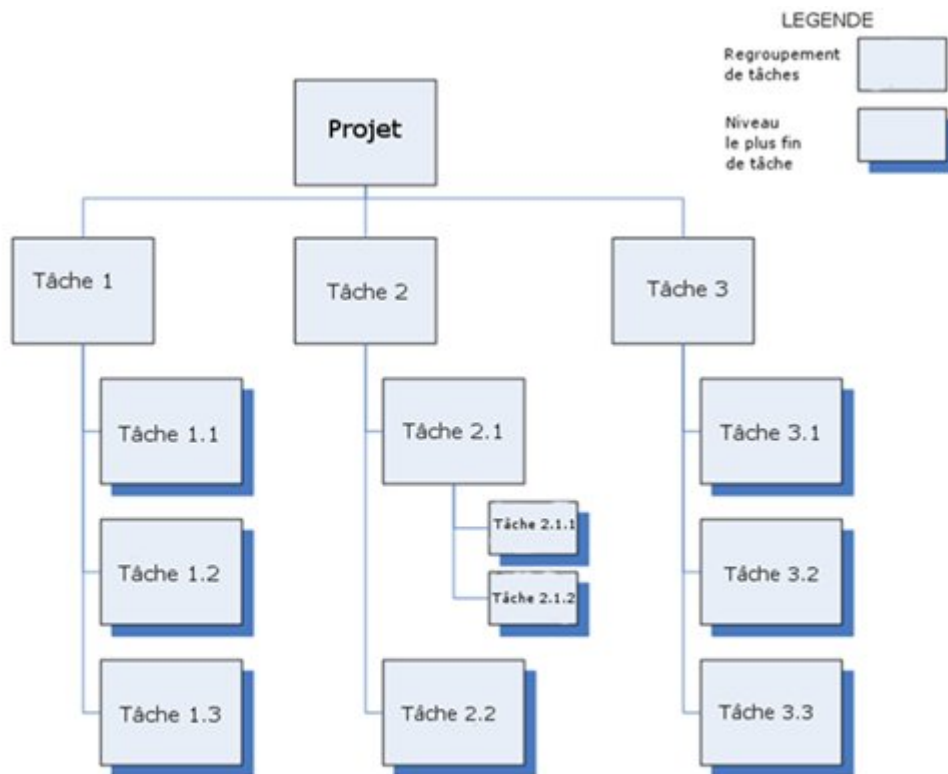


FIGURE 2.2: Découpage en tâches – WBS

Pour présenter l'enchaînement des tâches, nous nous sommes basées sur le diagramme de Gantt.

### I.1.3 Diagramme de Gantt

Le diagramme de GANTT ou diagramme à barres est l'un des outils de planification le plus utilisé. Cet outil permet de situer les tâches à accomplir dans le temps. Il permet restreindre les délais, grâce à la possibilité de chevauchement des tâches qu'offre son tracé. La construction du GANTT retient en abscisse, le temps nécessaire à l'exécution des opérations. A l'ordonnée, on représente soit les tâches soit les ressources affectées aux différentes opérations.

### I.1.4 Répartition des rôles OBS

OBS est une structure permettant l'identification des rôles et des missions des différentes entités d'organisation (départements, services, partenaires, etc.) d'un projet. Elle regroupe donc, le découpage en tâches WBS et les noms des responsables de ces tâches.[7]

## II Présentation du planning de notre projet :

Nous abordons dans cette partie, la planification de notre projet en nous basant sur les différentes normes de découpage. Les différents organigrammes ont été réalisés avec l'outil MatchWare MindView. Nous détaillons tout d'abord le découpage en sous-produits et ensuite la décomposition en tâches. L'organigramme de la répartition des rôles (OBS) n'est pas présenté vu que nous avons effectué toutes les tâches sans l'intervention d'autres personnes.[7]

Nous terminons avec le planning de notre travail tout au long du développement de notre application de gestion des changements. La planification est sous la forme d'un diagramme de GANTT qui a été élaborée avec le logiciel MatchWare MindView.

### II.1 Décomposition en sous-produits

Notre projet s'intéresse à la description de déroulement de toute demande d'intervention émanant de l'utilisateur et validée par le propriétaire métier, il offre une meilleure organisation et sécurité pour les actions intervenantes sur le système d'information de la banque.

La figure 2.3 montre le découpage de notre application en sous-projets par ordre :

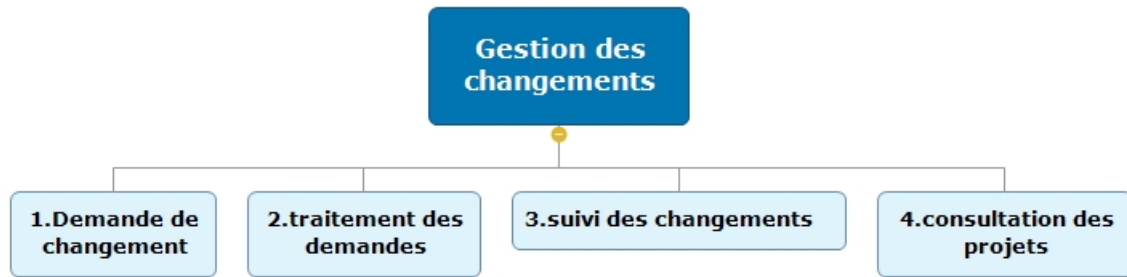


FIGURE 2.3: Organigramme basé sur le découpage PBS

## II.2 Décomposition en tâches

Grâce au découpage en tâches, nous avons pu décomposer les différents sous-produits en éléments pour avoir une vue plus détaillée sur chacun de ces modules et pour faciliter notre travail lors de la phase de réalisation. La figure 2.4 montre les tâches élémentaires des différents modules à développer en tenant compte de leurs enchaînements.



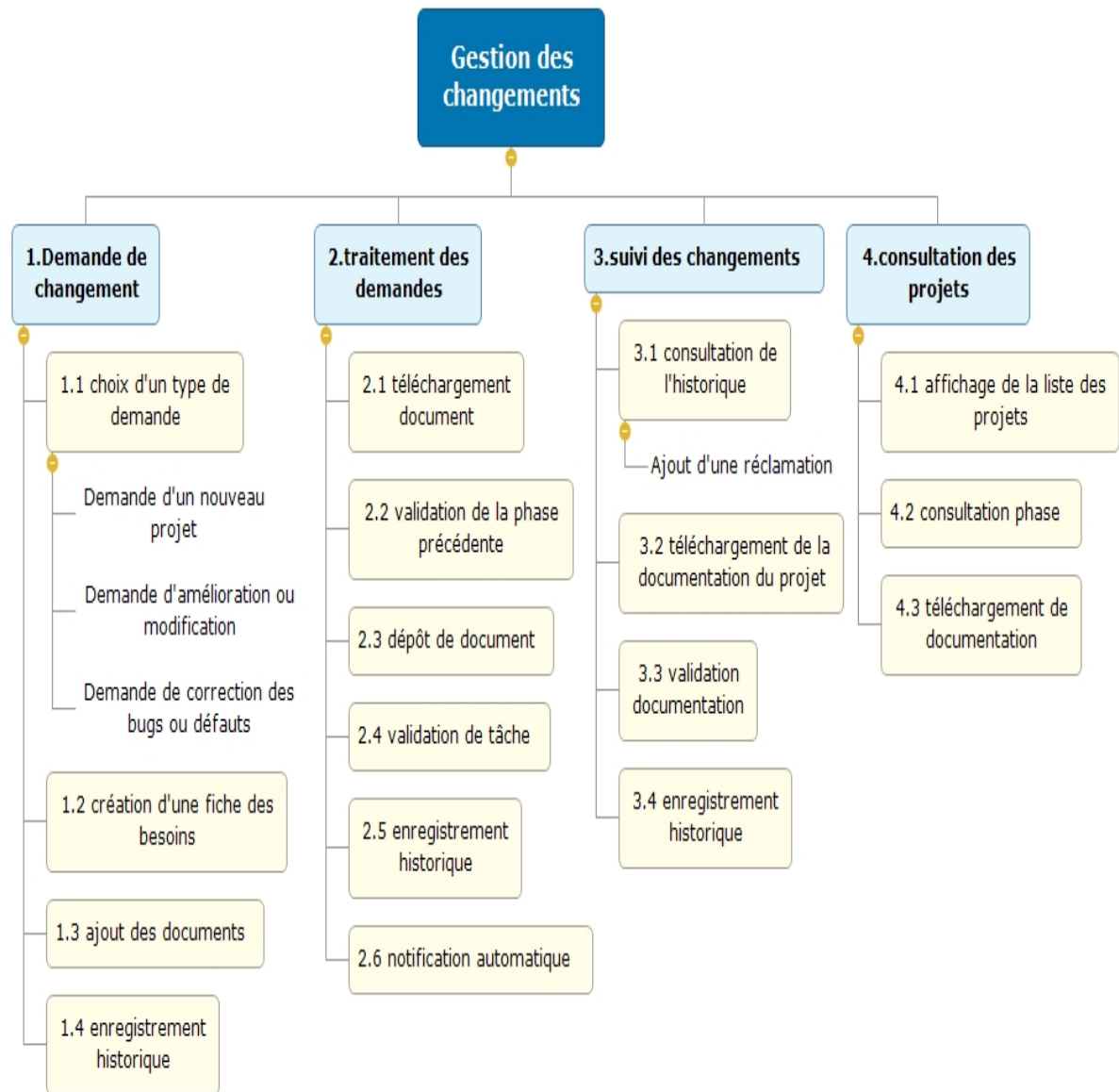


FIGURE 2.4: Organigramme de la décomposition en tâches WBS

## II.3 Enchaînement des tâches

La figure 2.5 illustre le diagramme de GANTT théorique présentant le planning du développement des différents modules et des tâches qui les composent.



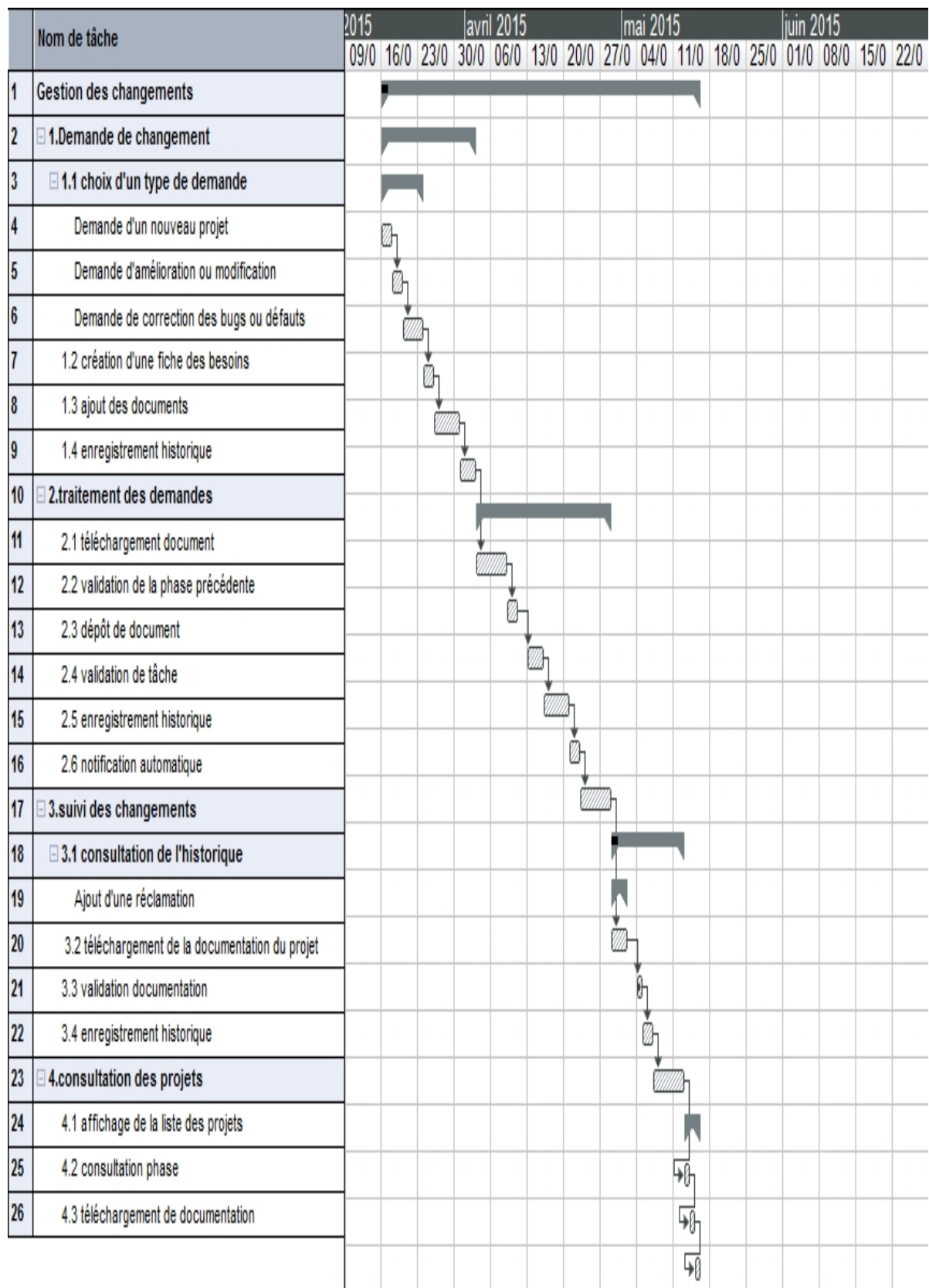


FIGURE 2.5: Diagramme de Gantt théorique

D'après la figure présentant le diagramme de GANTT théorique, nous pouvons constater que la plupart des tâches de notre application se déroulent d'une manière séquentielle.

## Conclusion

Dans ce chapitre, nous avons présenté la démarche de découpage d'un projet nécessaire pour la planification du travail. Nous avons abordé les différentes normes assurant un suivi efficace du projet. Et finalement, nous avons appliqué cette décomposition dans notre projet qui nous a permis de déduire les sprints de notre application.

Le prochain chapitre est consacré à la définition du backlog du produit ainsi qu'à la présentation des besoins fonctionnels et non fonctionnels. Et nous terminons par une analyse de ces besoins en se basant sur les diagrammes de cas d'utilisation d'UML.

## Chapitre 3

### Analyse et spécification des besoins

### Introduction

Ce chapitre est consacré à l'analyse et la spécification des besoins. Nous commençons par définir notre backlog du produit, par la suite nous spécifions les besoins fonctionnels et non fonctionnels.

L'analyse de ces besoins nous a permis d'identifier les différentes fonctionnalités et les acteurs du système. Les Diagrammes UML résultant de cette analyse sont produits dans la dernière section du chapitre.

## I Le Backlog du produit

Nous commençons par définir le backlog qui consiste en une liste de fonctionnalités à réaliser. Ces fonctionnalités, exprimées sous forme de besoins, sont priorisées par le Product Owner ce qui permet d'établir un ordre à respecter lors de la réalisation de l'application.

La figure 3.1 montre le backlog du produit de notre solution :

<b>Id</b>	<b>Tâche</b>	<b>Ordre de priorité</b>	<b>Durée (jour)</b>	<b>Prédécesseur</b>
<b>1. Demande de changement</b>				
A1	1.1 choix d'un type de demande	1	6	-
B1	1.2 création d'une fiche des besoins	1	2	A1
C1	1.3 ajout des documents	2	3	B1
D1	1.4 enregistrement historique	2	3	C1
<b>2. traitement des demandes</b>				
A2	2.1 téléchargement document	2	4	D1
B2	2.2 validation de la phase précédente	3	2	A2
C2	2.3 dépôt de document	2	3	B2
D2	2.4 validation de tâche	3	3	C2
E2	2.5 enregistrement historique	1	2	D2
F2	2.6 notification automatique	3	4	E2
<b>3. suivi des changements</b>				
A3	3.1 consultation de l'historique	1	6	F2
B3	3.2 téléchargement de la documentation du projet	1	1	A3
C3	3.3 validation documentation	3	2	B3
D3	3.4 enregistrement historique	2	4	C3
<b>4. consultation des projets</b>				
A4	4.1 affichage de la liste des projets	1	1	D3
B4	4.2 consultation phase	1	1	A4
C4	4.3 téléchargement de documentation	2	1	B4

FIGURE 3.1: Backlog de notre produit

## II Identification des acteurs

L'analyse d'une application débute par la détermination de ses différents acteurs. Une étude de l'interaction du système avec son environnement extérieur a permis de dégager principalement :

- L'utilisateur : C'est un employé de AMEN BANK, il rédige en langage naturel le

documents des fonctionnalités attendues d'un projet ainsi que les contraintes non fonctionnelles.

- La direction centrale de l'organisation et du système informatique : Ce sont les employés du département informatique d'AMEN BANK qui traitent les demandes des utilisateurs selon leurs rôles et leurs privilèges. Cette direction est constitué de :
  - Chef de division :
    - C'est le responsable du département informatique peut accéder à tous les documents
  - Les développeurs :
    - Rattachent les documents contenant le code du logiciel, les documents de test,
    - Téléchargent le cahier des charges, les documents de conception et le document d'expression des besoins.
  - La division architecture et intégration :
    - Rattache les documents de conception,
    - Télécharge le cahier des charges et le document d'expression des besoins.
  - La division des cahiers des charges :
    - Rattache le cahier des charges,
    - Télécharge le document d'expression des besoins.
  - La division qualités :
    - Rattache le document de test,
    - Télécharge le cahier des charges, le document d'expression des besoins, le script et les documents de conception.
  - La division système :
    - Rattache la documentation,
    - Télécharge le cahier des charges, le document d'expression des besoins et le script.



## III Spécification des besoins

### III.1 Besoins fonctionnels

Cette étape consiste à répondre à la question à quoi sert notre système. Nous regroupons ces besoins dans les points suivants :

- Permettre aux employés d'AMEN BANK d'effectuer des demandes de changement, en déposant un document d'expression des besoins,
- Classer les demandes en trois catégories : les nouveaux projets, les améliorations ou modifications et les corrections de défaut ou bug,
- Notifier les employés du département informatique, s'il y a une nouvelle demande de changement ou s'il y a un nouveau document rattaché,
- Permettre aux membres de département informatique de déposer et télécharger les documents appropriés selon leurs rôles et leurs privilèges,
- Permettre aux employés de la direction informatique de valider les différentes phases selon leurs rôles,
- Permettre aux clients de notre système, qui sont des employés d'AMEN BANK, de télécharger un prototype illustrant leurs requêtes,
- Notifier tous les utilisateurs de notre application que le projet est mis en place par l'équipe de la division système,
- Permettre à tous les utilisateurs de notre système de faire des réclamations concernant une tâche particulière,
- Permettre à tous les utilisateurs de notre système de consulter les différentes réclamations effectuées concernant une phase donnée,
- Visualiser l'avancement des requêtes par la consultation de l'historique des demandes de changement,
- Garder une trace de toutes les demandes de changements,
- Permettre aux employés du département informatique de consulter la liste des projets réalisés ainsi que leurs documents,
- Permettre aux employés d'AMEN BANK de consulter la liste des projets réalisés par le département informatique.

### III.2 Besoins non fonctionnels

Outre que les besoins fonctionnels cités précédemment, pour bien répondre aux exigences des utilisateurs, le système devra pouvoir assurer les besoins non fonctionnels suivants :

- L’ergonomie et la convivialité : L’application fournira une interface conviviale et simple à utiliser et qui ne requiert aucun pré requis, donc elle pourra être exploitable par tout type d’utilisateurs (même les non informaticiens).
- La sécurité : L’accès aux informations n’est possible qu’après vérification des privilèges et des droits d’accès. Ainsi tout utilisateur passera par une phase d’authentification pour pouvoir consulter les services offerts par l’application.
- L’extensibilité : L’architecture de l’application permettra l’évolution et la maintenance (ajout ou suppression ou mise à jour) au niveau de ses différents modules d’une manière flexible.
- Contraintes humaines : Notre solution doit tenir compte des besoins de chaque acteur participant dans la gestion des changements. Elle doit leur faciliter la tâche et assurer leur activité dans les bonnes conditions. Donc, notre application doit être facile à utiliser en présentant des interfaces ergonomiques.

## IV Analyse des besoins

Dans cette partie, nous présentons tout d’abord le diagramme de cas d’utilisation général. Cette phase représente la vue « fonctionnelle » de l’architecture du système.

Dans ce qui suit, nous identifions les cas d’utilisation principaux de notre application. Cette identification sera par la suite suivie d’une description détaillée de chacun d’eux.

### IV.1 Diagramme de cas d’utilisation du module «Gestion des changements»

Le diagramme de cas d’utilisation général du module gestion des changements, présenté dans la figure 3.2, illustre une vue globale de notre solution.

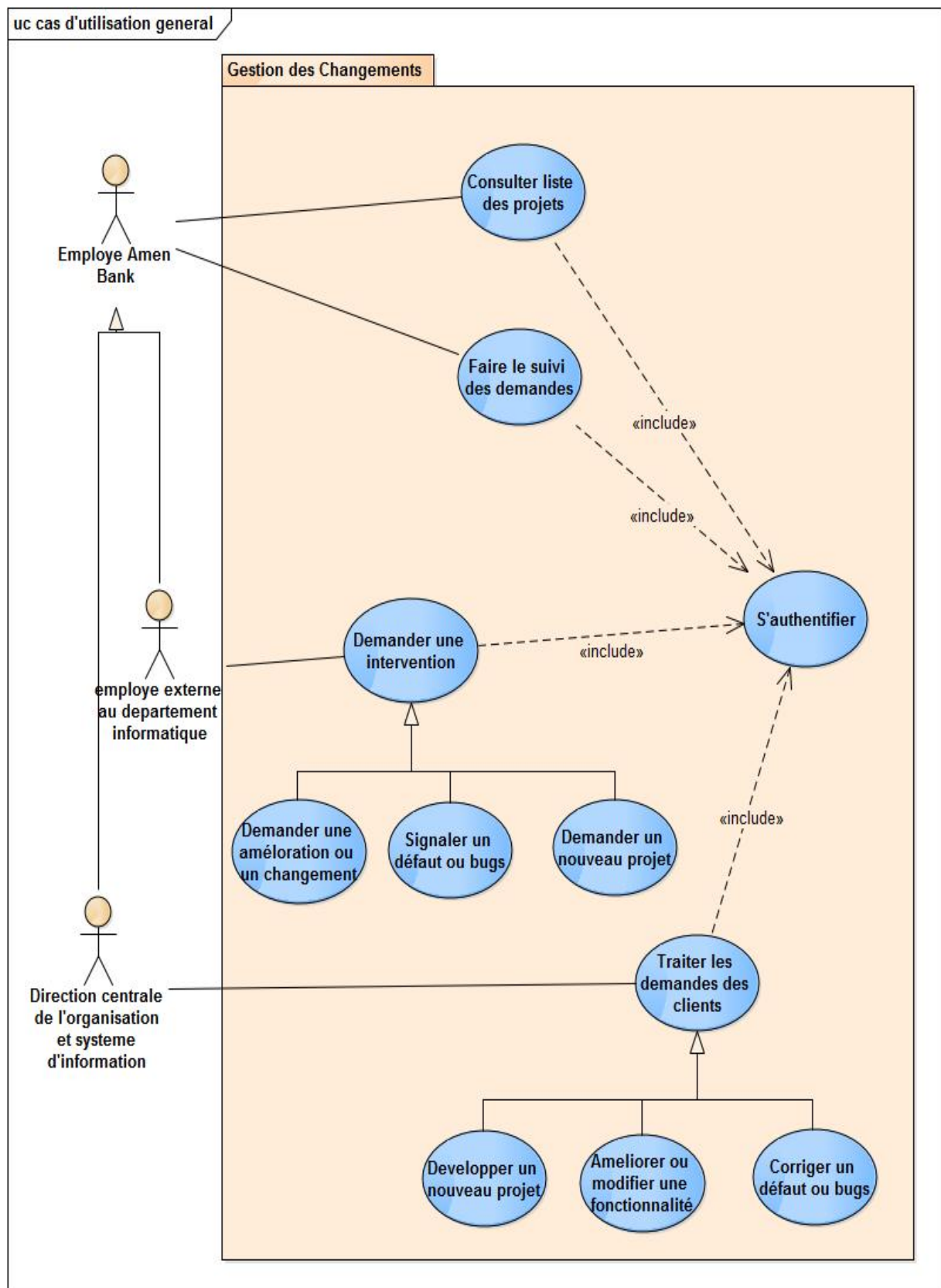


FIGURE 3.2: Diagramme de cas d'utilisation «Gestion des changements »

#### IV.1.1 Diagramme de cas d'utilisation détaillé du module «Faire le suivi des demandes»

La figure 3.3 présente le diagramme de cas d'utilisation détaillée du module du suivi de déroulement des tâches d'une demande effectuée par un client.

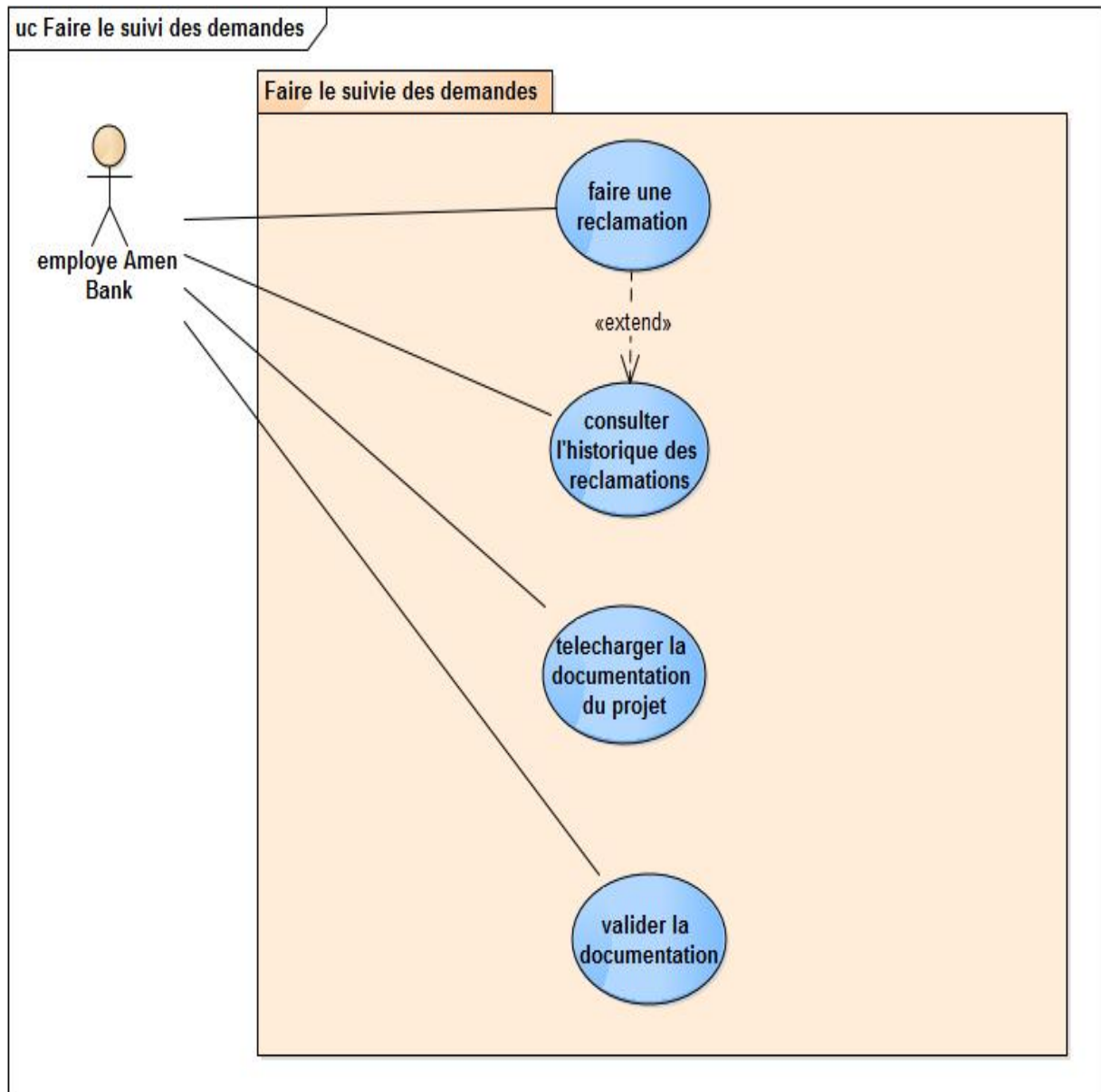


FIGURE 3.3: Diagramme de cas d'utilisation du module « Faire le suivi des demandes»

TABLE 3.1: Description textuelles du cas d'utilisation «Faire le suivi des demandes»

Résumé	<ul style="list-style-type: none"> <li>- <b>Titre</b> : Faire le suivi des demandes</li> <li>- <b>Acteur</b> : Direction centrale de l'organisation et système d'information et les employés ayant effectués des demandes</li> <li>- <b>Résumé</b> : ce cas d'utilisation définit le suivi de la demande de l'utilisateur, il présente la procédure décrivant le déroulement de cycle de vie de la requête cliente.</li> </ul>
Pré condition	<ul style="list-style-type: none"> <li>- Tous les utilisateurs doivent être authentifiés à l'application .</li> </ul>
Enchaînements nominal	<ul style="list-style-type: none"> <li>- Si l'utilisateur est un client il pourra voir dans quelle phase est sa demande</li> <li>- Pour une phase donnée, chaque utilisateur de système a la possibilité de faire une réclamation comme il peut consulter l'historique des commentaires échangés entre les autres utilisateurs</li> <li>- Le client télécharge la documentation du projet s'il est mis en exploitation</li> <li>- Ce document doit être soit validé soit rejeté.</li> </ul>

Enchaînement alternatif	<ul style="list-style-type: none"> <li>- Si le mot de passe et/ou le login sont incorrects alors un message d'erreur apparaît pour l'utilisateur pour qu'il saisisse des valeurs correctes</li> <li>- Si l'un des champs ou les deux sont vides un message apparaît aussi indiquant que le champ login et le champ mot de passe ne doivent pas être vide</li> </ul>
Post condition	<ul style="list-style-type: none"> <li>- Si l'acteur décide de faire une réclamation alors il sera redirigé vers la section des commentaires</li> <li>- L'historique des commentaires et des tâches sera modifié et enregistré</li> </ul>

#### IV.1.2 Diagramme de cas d'utilisation détaillé du module « Demande d'intervention »

La figure 3.4 représente le module détaillé de la demande d'intervention d'intervention effectuée par les employés d'AMEN BANK :

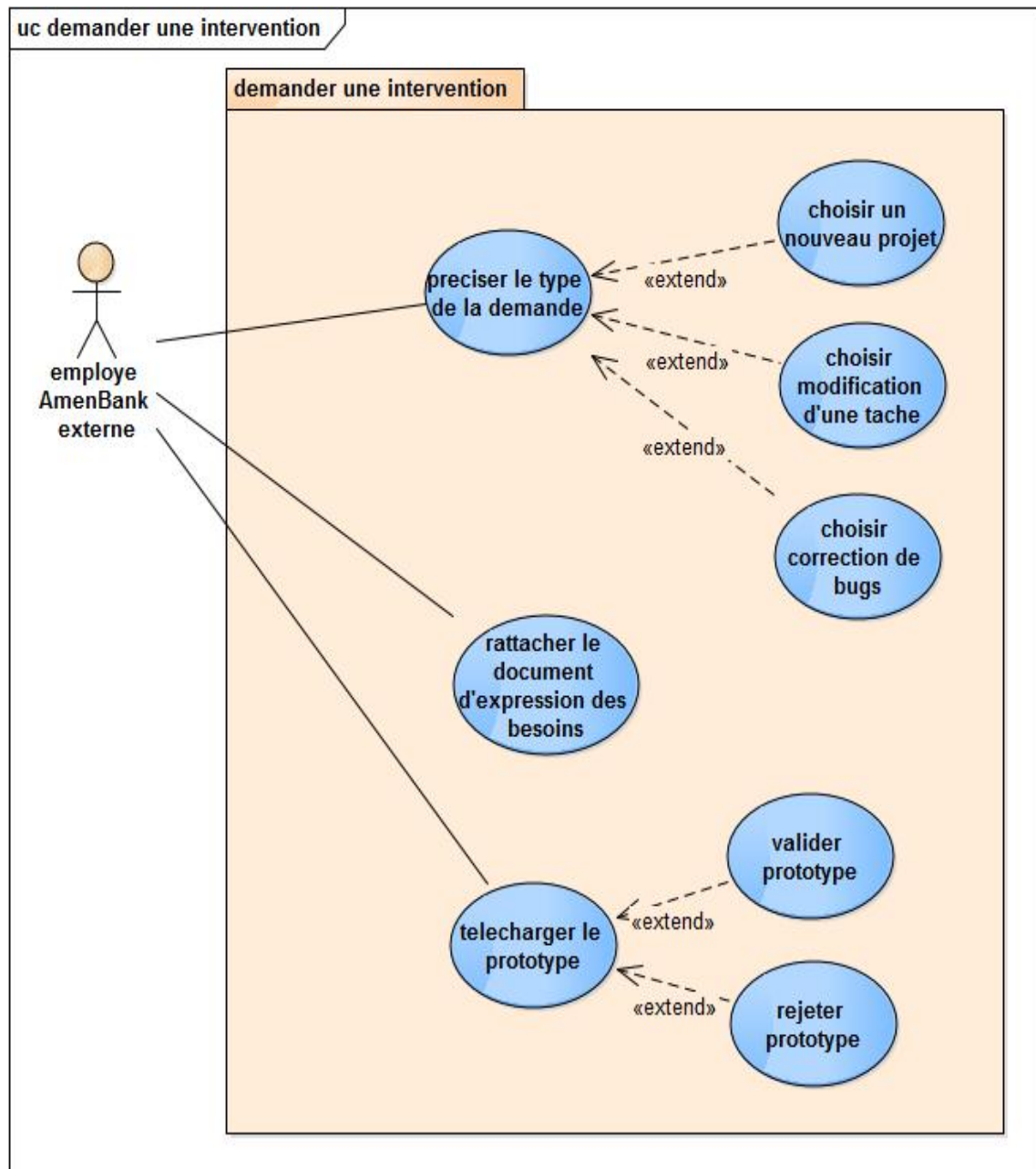


FIGURE 3.4: Diagramme de cas d'utilisation du module « Demande d'interventions»

TABLE 3.2: Description textuelles du cas d'utilisation «Demande d'interventions»

Résumé	<ul style="list-style-type: none"><li>- <b>Titre</b> : Demande d'intervention</li><li>- <b>Acteur</b> : les utilisateurs externes au département informatique</li><li>- <b>Résumé</b> : ce cas d'utilisation présente la demande de l'utilisateur qui peut être une demande d'un nouveau projet, d'amélioration ou modification ou correction de défaut ou bug</li></ul>
Pré condition	<ul style="list-style-type: none"><li>- Tous les utilisateurs doivent être authentifiés à l'application.</li></ul>
Enchaînement nominal	<ul style="list-style-type: none"><li>- Dans le cas où le login et le mot de passe sont corrects, l'utilisateur ayant une demande doit choisir si sa requête s'agit d'une demande d'un nouveau projet, une demande d'amélioration ou de modification d'une tâche ou bien correction de bugs</li><li>- Ensuite, il rattache le document d'expression des besoins, comme il peut consulter la liste des projets déjà réalisés et leurs historiques</li></ul>
Enchaînement alternatif	<ul style="list-style-type: none"><li>- Si le mot de passe et/ou le login sont incorrects alors un message d'erreur apparaît pour l'utilisateur pour qu'il saisisse des valeurs correctes</li><li>- Si l'un des champs ou les deux sont vides un message apparaît aussi indiquant que le champ login et le champ mot de passe ne doivent pas être vide</li></ul>



Post condition	<ul style="list-style-type: none"><li>- Le système enregistre la demande et affiche une interface d'acceptation de demande pour le client</li><li>- La chef de la division rédige le cahier de charge et l'envoie à l'utilisateur pour qu'il le valide</li><li>- Le système affiche une interface de validation des documents pour les employés du département où ils peuvent aussi rattacher d'autres documents</li><li>- Des notifications des tâches suivantes pour les acteurs appropriés</li></ul>
----------------	---

#### **IV.1.3 Diagramme de cas d'utilisation détaillé du module « Traiter les demandes des clients »**

Le diagramme suivant présenté par la figure 3.5 illustre le module du traitement des requêtes clientes :

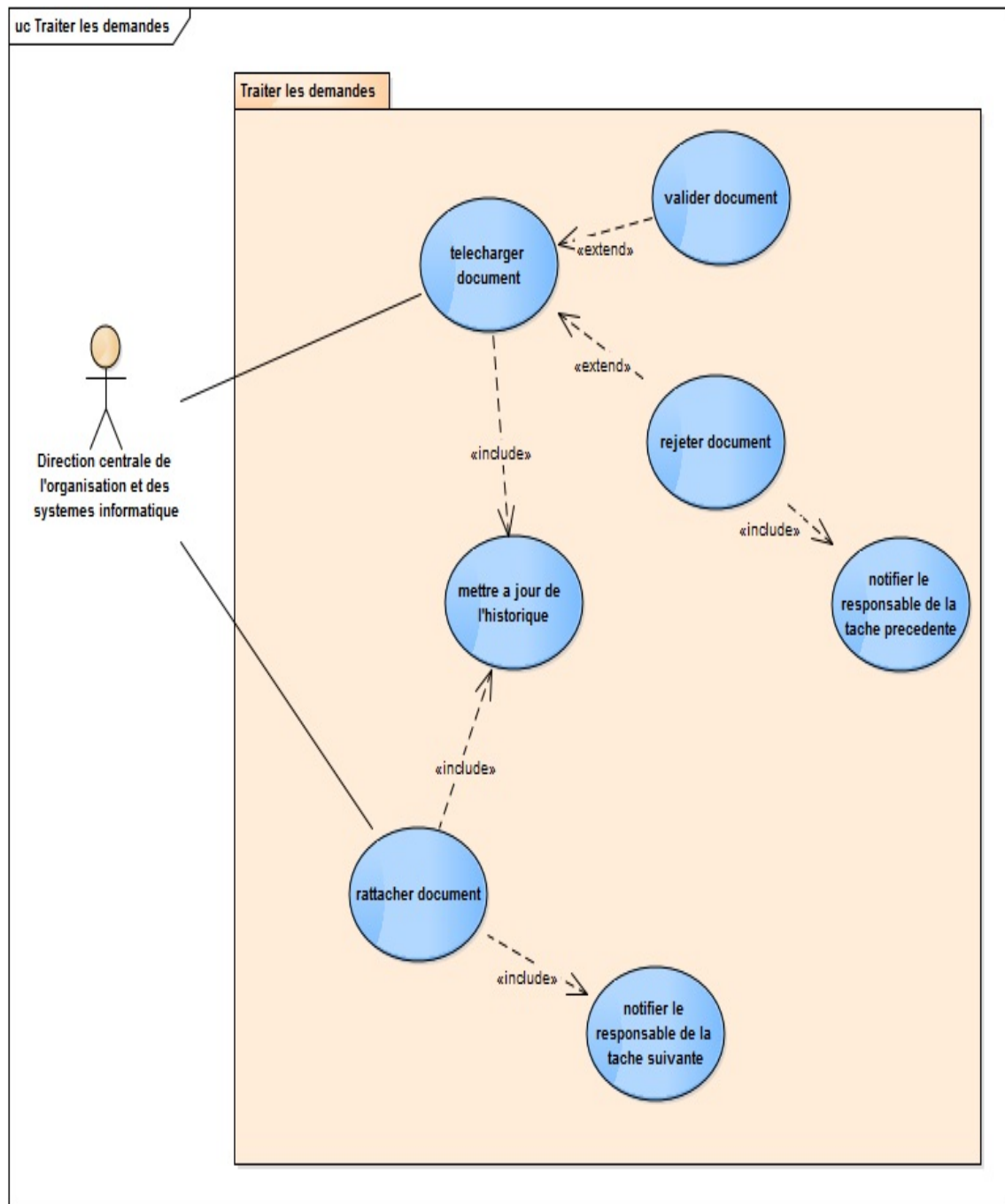


FIGURE 3.5: Diagramme de cas d'utilisation du module «Traiter les demandes»

TABLE 3.3: Description textuelles du cas d'utilisation «Traiter les demandes»

Résumé	<ul style="list-style-type: none"> <li>- <b>Titre</b> : Traitement des demandes</li> <li>- <b>Acteur</b> : les employés la direction centrale de l'organisation et système informatique</li> <li>- <b>Résumé</b> : ce cas d'utilisation définit la gestion de la demande de l'utilisateur, il présente la procédure décrivant le déroulement de le processus de gestion d'une demande d'intervention</li> </ul>
Pré condition	<ul style="list-style-type: none"> <li>- Tous les employés du département informatique doivent être authentifiés à l'application. Une demande d'un nouveau projet est reçue de la part de l'utilisateur</li> </ul>
Enchaînement nominal	<ul style="list-style-type: none"> <li>- Les employés du département informatique chacun accède aux documents autorisés à télécharger. Selon leurs rôles, les utilisateurs reçoivent des notifications des nouvelles tâches</li> <li>- Lors de cette phase, une étude des besoins doit être effectuée par la CED. Elaboration de cahier de charge par la Division des Cahier des Charges à partir de document d'expression des besoins. Contrôle de la qualité du cahier de charge par le Chef de Division. Elaboration des documents de conception par le chef de la division et la division architecture et intégration.</li> </ul>

	<ul style="list-style-type: none"><li>- Les développeurs commencent le codage après avoir téléchargé le cahier de charge et les documents de conception. Le chef de la division et les développeurs effectuent les tests unitaire et global. La division système met en place le logiciel développé et élabore le circulaire décrivant le mode d'emploi</li><li>- Chaque étape déclenche l'étape suivante en affichant aux acteurs associés les documents qu'ils doivent utiliser. Les utilisateurs déposent les documents modifiés ou ajoutés</li><li>- Chaque tâche doit être acceptée ou déniée pour poursuivre le processus</li></ul>
Post condition	<ul style="list-style-type: none"><li>- Cette phase se termine par la validation de l'opération effectuée et par l'enregistrement de l'historique</li></ul>

## V Diagramme d'activité du système gestion des changements

Après avoir étudié les fonctionnalités de notre système, nous présentons le diagramme d'activité de notre application, présenté par la figure 3.6, qui décrit le processus de gestion des changements.

Ce diagramme décrit le circuit de traitement des tâches à accomplir, ainsi que leur validation entre les différents acteurs de notre système.

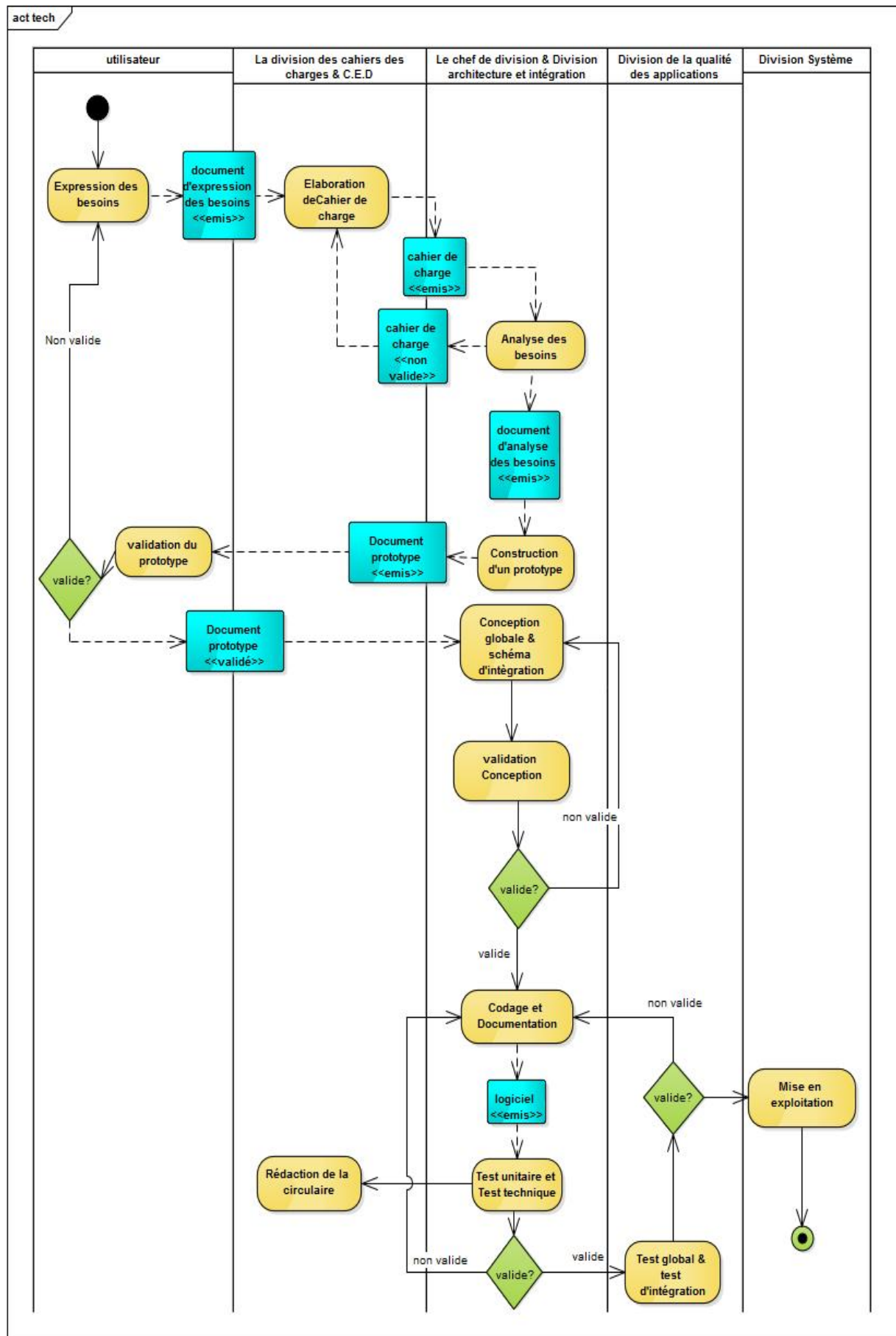


FIGURE 3.6: Diagramme d'activité «Gestion des changements»

### **Conclusion**

Nous avons présenté le backlog du produit en spécifiant les différentes fonctionnalités qui le composent. Nous avons aussi détaillé les besoins fonctionnels et non fonctionnels et nous les avons illustrés par des diagrammes de cas d'utilisation.

Le prochain chapitre est consacré pour la partie de la réalisation de notre solution en se basant sur la méthodologie AGILE adaptée qui est SCRUM.

# Chapitre 4

## Réalisation

## Introduction

Dans ce chapitre, nous présentons l'environnement logiciel qui nous a permis de le réaliser notre système. Nous détaillons ensuite le diagramme de classe de notre solution et nous abordons les différentes configurations effectuées avant de commencer la réalisation de notre application.

Sans oublier, l'exposition de chacun des sprints en précisant les résultats obtenus et les problèmes rencontrés. Et pour finir, nous présentons un exemple que nous avons réalisé pour montrer l'extensibilité de notre application.

## I Architecture logicielle de notre solution

Notre système de gestion des changements est caractérisé par son architecture MVC (Modèle - Vue - Contrôleur) qui est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface mais aussi elle offre un cadre pour structurer une application.

Dans l'architecture MVC, les rôles des trois entités sont les suivants :

- modèle : données (accès et mise à jour)
- vue : interface utilisateur (entrées et sorties)
- contrôleur : gestion des événements et synchronisation

Cette architecture est implémentée par Struts 2 de la façon suivante :



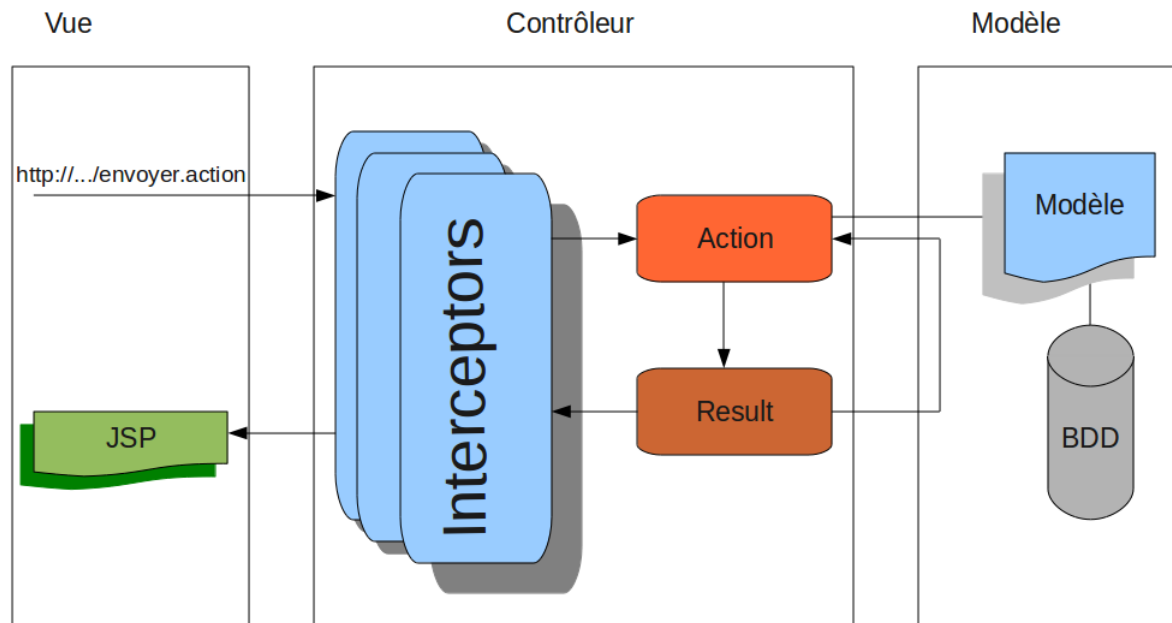


FIGURE 4.1: Fonctionnement de Struts2

Le traitement d'une demande d'un client se déroule de la façon suivante :

Les Url demandées sont de la forme `http://machine:port/contexte/rep1/rep2/.../Action`. Le chemin `[/rep1/rep2/.../Action]` doit correspondre à une action définie dans un fichier de configuration de Struts 2, sinon elle est refusée. Une action est définie dans un fichier Xml.

Les étapes suivantes sont alors exécutées :

- Demande :** le client navigateur fait une demande au contrôleur `[FilterDispatcher]`. Celui-ci voit passer toutes les demandes des clients. C'est la porte d'entrée de l'application. C'est le C de MVC.
- Traitement :** le contrôleur C consulte son fichier de configuration et découvre que l'action `actions/Action1` existe. Le namespace concaténé avec le nom de l'action définit l'action `actions/Action1`. Le contrôleur C instancie une classe de type `[actions.Action1]`. Le nom et le package de cette classe peuvent être quelconques. Il faut donc que la classe `[actions.Action1]` ait ces méthodes `setParami` pour chacun des paramètres attendus. Le contrôleur C demande à la méthode de signature `[String execute()]` de la classe `[actions.Action1]` de s'exécuter. Celle-ci peut alors exploiter les paramètres parami que la classe a récupérés. Dans le traitement de la demande de l'utilisateur, elle peut avoir besoin de la couche `[metier]`. Une fois

la demande du client traitée, celle-ci peut appeler diverses réponses. Un exemple classique est :

Une page d'erreurs si la demande n'a pu être traitée correctement une page de confirmation sinon la méthode exécute rend au contrôleur C un résultat de type chaîne de caractères appelée clé de navigation qui va mettre à jour le modèle M que va exploiter la page JSP qui va être envoyée en réponse à l'utilisateur. Ce modèle peut comporter des éléments de : la classe [actions.Action1] instanciée la session de l'utilisateur.

- **Réponse** : le contrôleur C demande à la page JSP correspondant à la clé de navigation de s'afficher. C'est la vue, le V de MVC. La page JSP utilise un modèle M pour initialiser les parties dynamiques de la réponse qu'elle doit envoyer au client.[w1]

## II Environnement de développement logiciel

L'environnement logiciel de notre solution comprend :

- Le système de gestion de base de données relationnelle MySQL,
- Eclipse Kepler pour la réalisation de notre application,
- Apache Tomcat 6 un conteneur web libre de servlets pour l'IDE Eclipse,
- TeXnicCenter Environnement pour la rédaction du rapport en langage LATEX,
- L'outil de modélisation UML Enterprise Architect,
- MatchWare Mindview pour la planification de projet.

## III Choix technologiques

Dans cette partie, nous nous intéressons aux langages, aux bibliothèques et aux techniques de programmation utilisées tout au long de la réalisation de notre application en justifiant notre choix.

### III.1 Plateforme JAVA EE

**Java Enterprise Edition** est une spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise. Ces applications sont

considérées dans une approche multi-niveaux. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au framework Java standard afin de faciliter notamment la création d'applications réparties.[w10]

### III.2 Le framework Struts 2 pour l'implémentation de la couche présentation

**Apache Struts 2** est un framework Java, pour le développement d'applications Web. Ce n'est pas une extension d'Apache Struts 1. Struts 2 regroupe les avantages de deux précédents outils, WebWork et Struts 1, mais c'est une refonte complète. Cette seconde génération de framework MVC utilise les notions suivantes : intercepteurs, annotations, langage d'expression OGNL, l'intégration d'outils comme JSTL ou Spring framework.[w1]

L'avantage majeur de cette plateforme est qu'elle est très vaste, et propose de très nombreux outils qui adressent toutes les problématiques de l'informatique d'entreprise : exposer des fonctionnalités avancées sur un site web, accès à des bases de données, communications distantes entre les applications Java ou même non Java. Ceci est réalisé grâce à des technologies "standard" donc connues d'un grand nombre de développeurs et de nombreux "frameworks" (Spring, Hibernate, ect) qui rendent la programmation plus simple.

### III.3 Hibernate pour l'implémentation de la couche d'accès au données

**Hibernate** est une solution open source de type ORM (Object Relational Mapping) qui nous a permis de faciliter le développement de la couche persistance de notre application. Hibernate nous a permis de représenter notre base de données en objets Java et vice versa. Hibernate nous a facilité la persistance et la recherche de données dans notre base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant. Nous avons utilisé Hibernate notamment grâce à ses bonnes performances et de son ouverture à de nombreuses bases de données.

### III.4 HTML5, CSS3, jQuery et JavaScript pour la construction des interfaces riches

**HTML5** a simplifié certaines balise afin d'alléger le code. Il introduit également un ensemble de nouvelles balises afin de donner plus de sémantique à nos pages. HTML 5 nous a permis une utilisation plus propre, code plus propre et nous pouvons, ainsi, éliminer la plupart des balises div et les remplacer par des éléments HTML 5 sémantiques.

**CSS3** nous a permis d'automatiser certains effets visuels qui nécessitaient jusqu'à présent l'utilisation d'images, de scripts ou de modifications du code HTML : ombres portées, coins arrondis, opacité, arrière plans multiples, dégradés complexes, multicolonage, effets textuels... L'avantage majeur et bien entendu une optimisation sur le temps de chargement des pages, car moins d'images, moins de code, moins de fichiers flash ect.

**JavaScript** est un langage de programmation Web distinct du HTML qui nous a permis d'ajouter de l'interactivité à nos pages Web en accédant directement aux éléments de la page HTML et en les manipulant. Il est, sans aucun doute, un des langages les plus populaires et les plus utilisés sur Internet, surtout qu'il fonctionne sur bon nombre de navigateurs différents.

**jQuery** est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. Il nous a permis de mieux développer nos pages web puisqu'il est :

- Stable et robuste
- Facile à apprendre et à utiliser
- Syntaxe claire et concise réduisant le nombre de lignes de code à écrire
- Support des différences entre les navigateurs

## IV Diagramme de classe

Ce diagramme décrit les classes de notre système de gestion des changements :

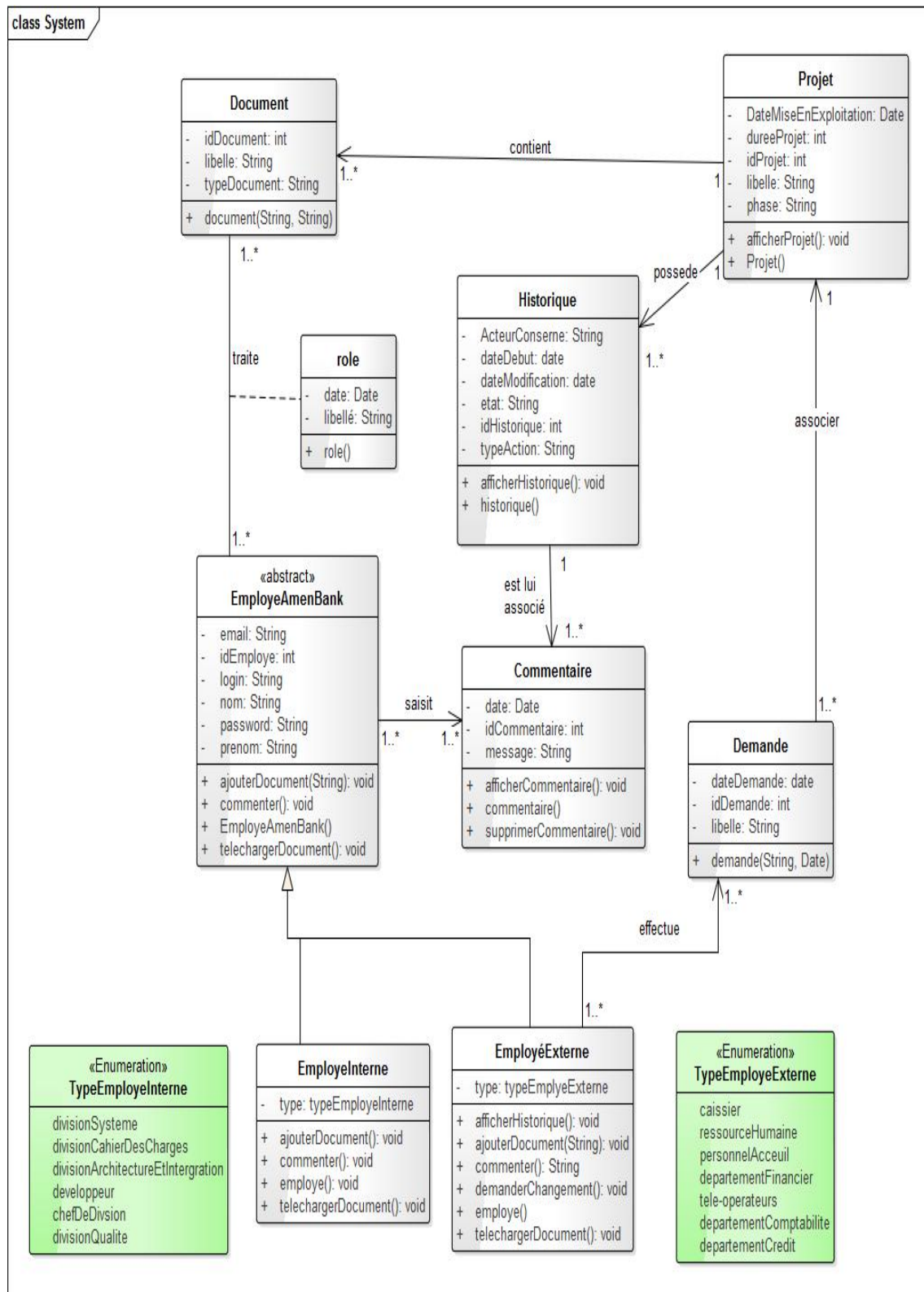


FIGURE 4.2: Diagramme de classe «Gestion de changement»

La figure 4.2 ci-dessus présente les classes qui ont servi à l'implémentation de l'application de notre application que nous pouvons les détaillées comme suit :

- Classe Employé : présente les employés d'Amen Bank, ils sont classés par deux catégories. La première catégorie regroupe un ensemble d'employés appartenant à la direction informatique ou ce qu'on appelle les employés internes au département informatique présenté par la table `EmployeInterne`, nous avons regroupé leurs rôles dans la table d'énumération `TypeEmployeInterne`.  
La deuxième catégorie regroupe les clients qui représentent tous employés d'Amen Bank qui n'appartiennent pas au département informatique que nous avons appelé les employés externes présentés par la table `EmployeExterne`, leurs rôles sont classés dans la classe d'énumération `TypeEmployeExterne`.
- Classe Demande : un client de notre application de gestion de changement a la possibilité de déposer sa demande qui peut être une demande d'un nouveau projet, amélioration d'une tâche ou bien correction d'un bug.
- Classe Projet : présente tous les projets réalisés par le département informatique d'AMEN BANK, un projet est caractérisé par une durée et sa phase qui désigne le cycle de vie d'une demande. Tout projet est créé lors de chaque une demande d'un nouveau projet.
- Classe Document : présente les documents échangés par les acteurs de notre système de gestion de changement. Chaque document peut être traité par plusieurs employés selon leurs privilèges. Ce document est soit validé soit rejeté. Chaque demande comporte un ou plusieurs documents et chaque phase exige au moins un document.
- Classe Historique : cette la classe enregistre tous les événements lors du traitement d'une demande. Plusieurs historique peuvent être associés à une seule demande.

- Classe commentaire : pour chaque historique d’une phase réalisée, il est possible qu’un utilisateur de notre application fait une réclamation.

## V Déroulement des Sprints

Avant de présenter les différents sprints qui composent notre backlog du produit, nous évoquons le planning de notre travail qui s’est étalé sur une période de quatre mois. Tout au long de ce chapitre, nous présentons une vue dynamique à travers les diagrammes de séquence ainsi les interfaces implémentées et pour terminer chaque nous présentons les tests à la fin. Durant cette période, le travail a été réparti ainsi :

Phase	Février				Mars				Avril				Mai			
	S1	S2	S3	S4	S1	S2	S3	S1	S1	S2	S3	S4	S1	S2	S3	S4
Elaboration du cahier de charge																
Familiarisation avec la plateforme Java EE et le framework Struts2																
Préparation de la conception générale																
Réalisation des Sprints									Sprint1	Sprint2	Sprint3	Sprint4				

FIGURE 4.3: Planning du travail

La décomposition du projet en sprints a été décrite par le backlog du produit. Il s’agit à chaque étape, de faire une réunion avec les collaborateurs pour :

- énoncer les objectifs d'un sprint (le produit du sprint),
- proposer des estimations temporelles en nombre de jour et décider de l'ordre de réalisation des tâches de chaque sprint,
- valider la réalisation à la fin du sprint en effectuant une démonstration du travail,
- modifier éventuellement certains objectifs définis au départ en fonction de nouvelles requêtes.

### **V.1 Sprint1– « Authentification et demande d'intervention »**

Après avoir entamé la conception de notre système informatique, nous pouvons maintenant nous lancer dans les travaux nécessaires pour produire le premier sprint qui sera composée de deux sous sprints, chacun ayant une vélocité d'une semaine.

#### **V.1.1 Conception détaillée du sprint 1**

Pour décrire les scénarios les plus importants dans la phase de d'un contrat, nous nous sommes basées sur une vue dynamique d'UML étant le diagramme de séquence.

La figure 4.3 ci-dessous montre le diagramme de séquence de la tâche «Authentification» :



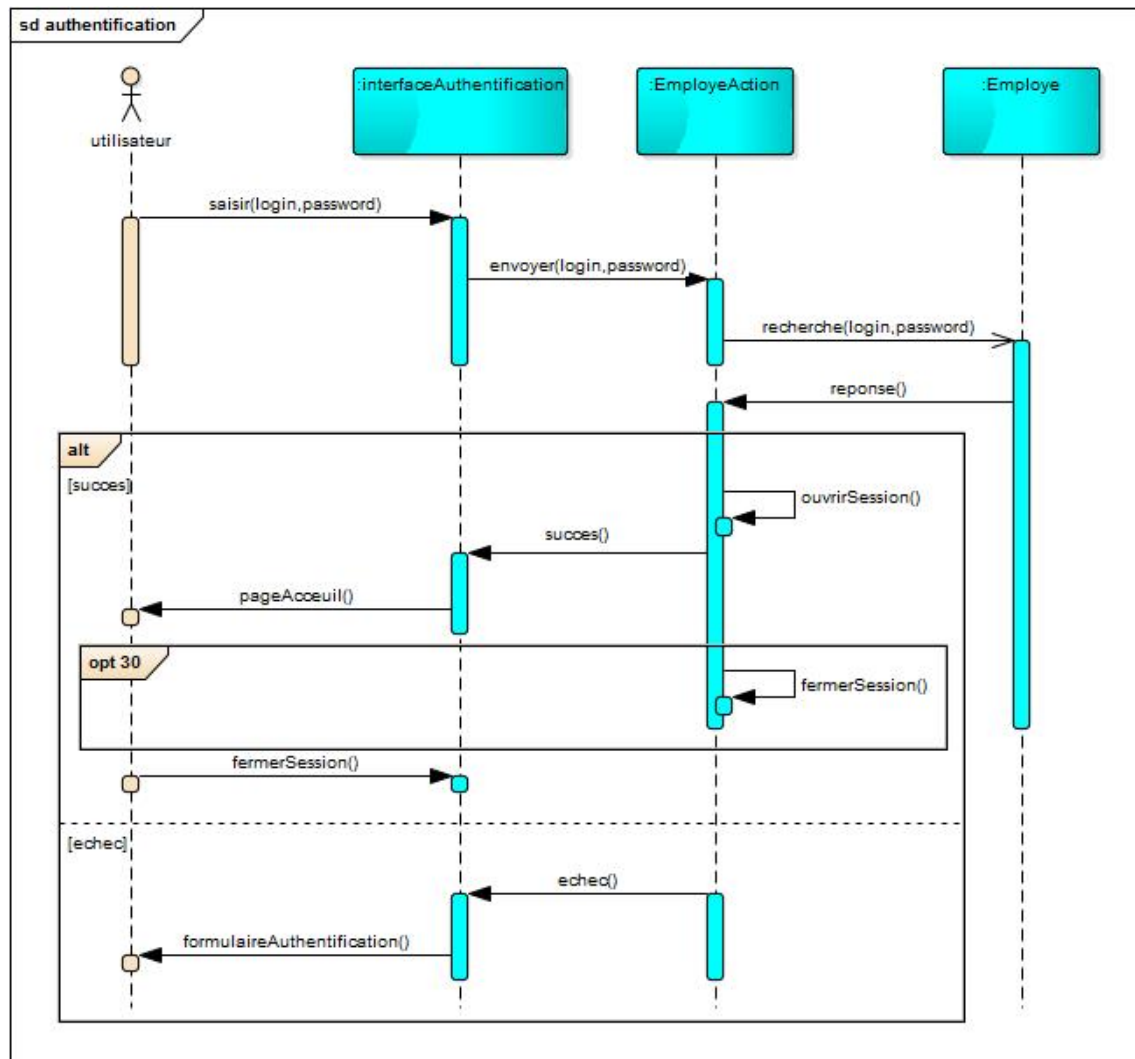


FIGURE 4.4: Diagramme de séquence « Authentification »

Pour s'authentifier, un utilisateur doit saisir son login et son mot de passe, si les données saisies sont correctes alors une session de sera ouverte pour lui et il sera redirigé automatiquement à la page d'accueil de l'application appropriée pour chaque type de d'utilisateur.

Si les données sont erronées alors un message d'erreur apparaîtra demandant à l'utilisateur de saisir de nouveau correctement le login et le mot de passe.

L'une des tâches les plus remarquables dans ce sprint est la demande d'intervention exposée dans les diagrammes de séquence suivants :

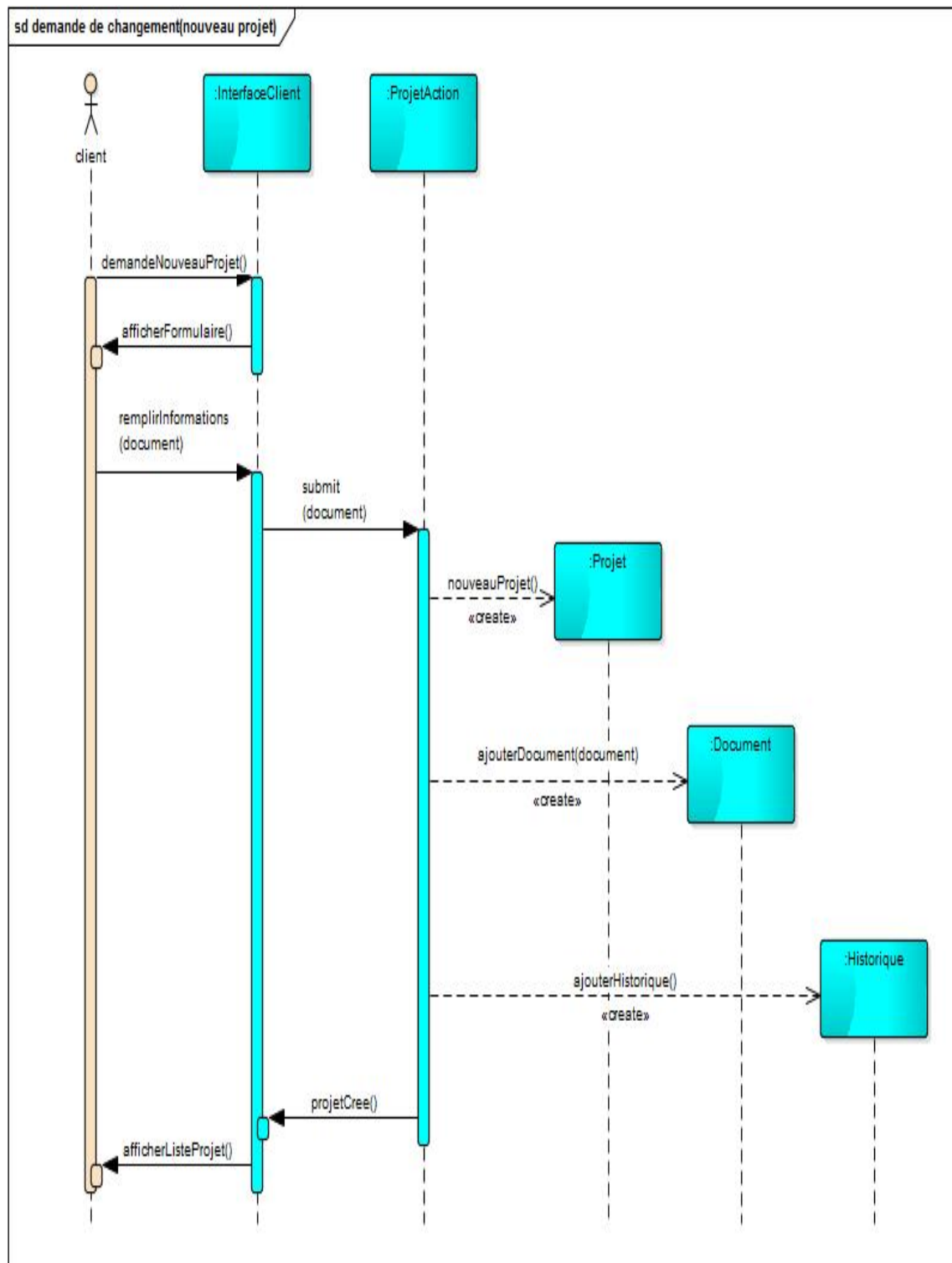


FIGURE 4.5: Diagramme de séquence « Demande d'un nouveau projet »

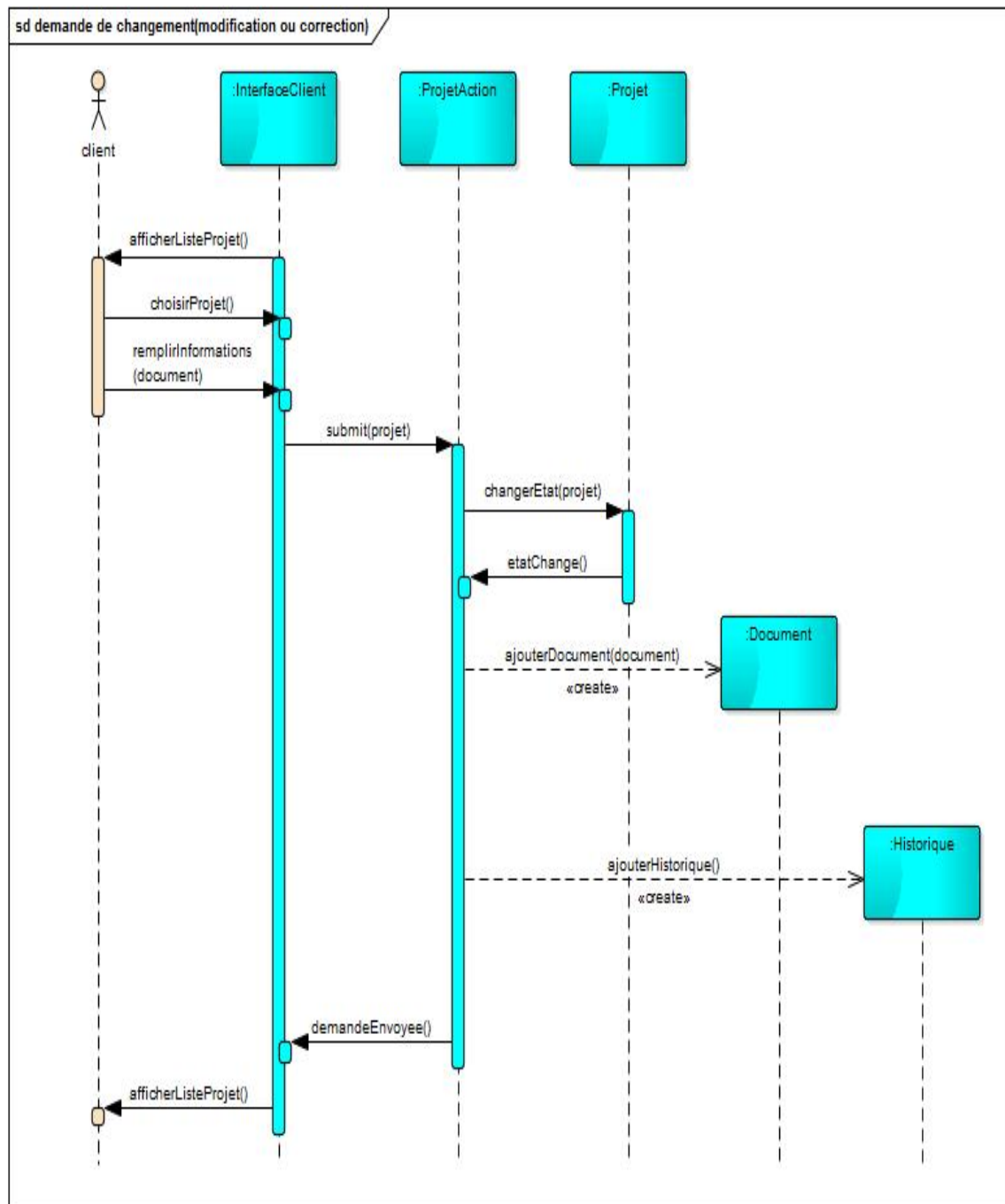


FIGURE 4.6: Diagramme de séquence « Demande de modification ou de correction »

Concernant la tâche de la demande d'intervention, si le client veut faire une demande d'un nouveau projet, il remplit les champs spécifiques du formulaire pour le type de cette requête et rattache le document d'expression des besoins.

Sinon si la demande concerne une demande d'amélioration de tâche ou correction de bug, le client choisi le projet qu'il désire qu'on y effectue des changements, remplit les champs nécessaires du formulaire et rattache aussi le document d'expression des

besoins.

La demande d'intervention engendre la création d'une nouvelle ligne dans la table projet et dans la table historique des projets. De plus, le document rattaché par l'utilisateur sera enregistré de même à la base de données.

### V.1.2 Développement du sprint 1

Après avoir préparé la conception du sprint 1, nous nous sommes concentrées sur la phase de codage pour élaborer les différentes interfaces de la partie d'authentification et demande d'intervention. Nous commençons par présenter les premières interfaces relatives à l'authentification. Les figures suivantes illustrent cette première tâche :



FIGURE 4.7: Interface d'authentification

Pour accéder à l'application, chaque employé doit en premier lieu s'authentifier en saisissant son login et son mot de passe. La figure 4.8 ci-dessous représente le cas où l'un des champs ou les deux sont vides.

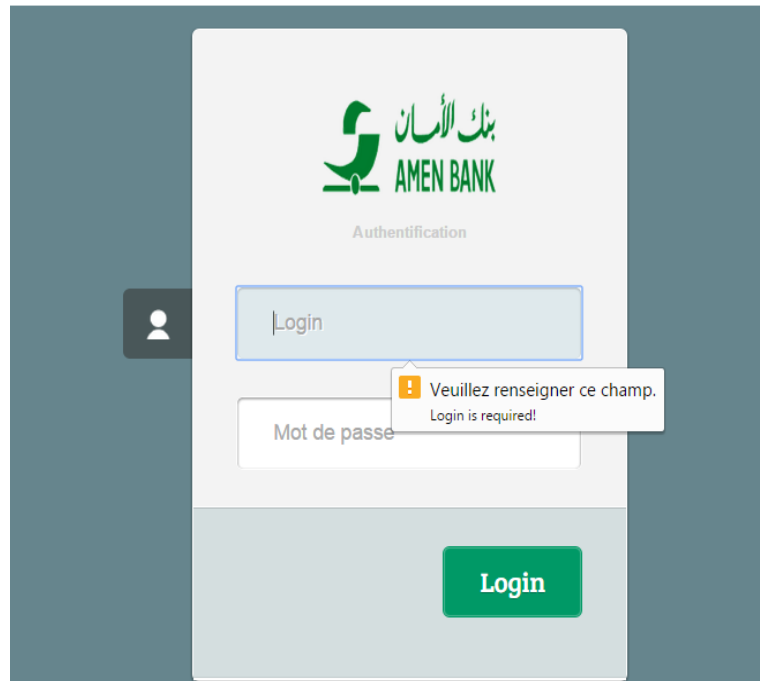


FIGURE 4.8: Interface d'authentification cas de champs vides

Si l'utilisateur ne remplit pas l'un des champs ou les deux alors un message apparaîtra pour chaque champ vide demandant à l'utilisateur de saisir son login et/ou mot de passe. La figure 4.9 suivante illustre le cas où les données saisies par l'employé sont fausses :

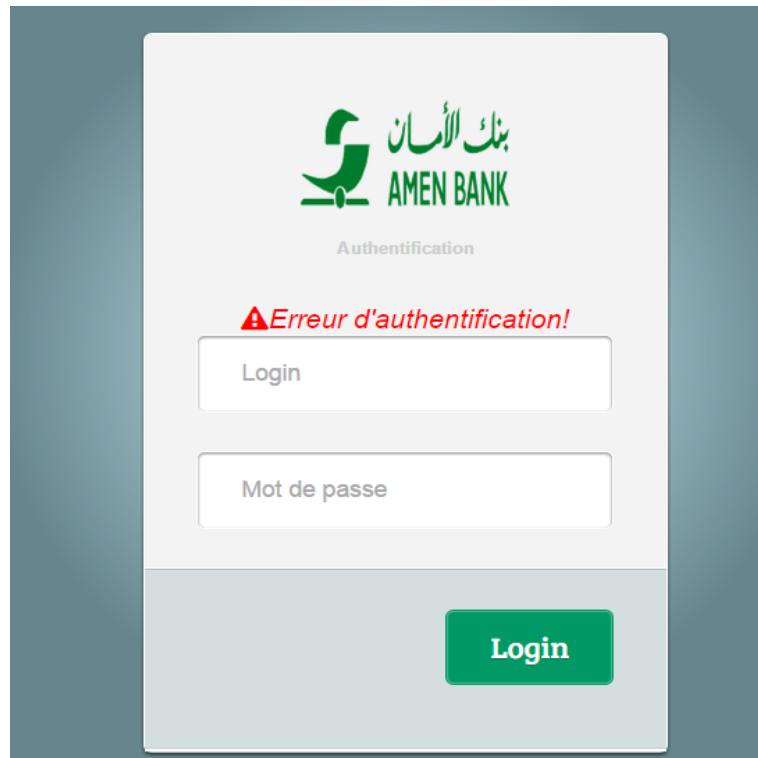


FIGURE 4.9: Interface d'authentification cas des données erronées

Si l'utilisateur saisit son login et/ou mot de passe erroné alors le formulaire d'authentification sera réaffiché de nouveau avec un message d'erreur indiquant une erreur dans les valeurs saisies. Les figures suivantes représentent la page d'accueil de notre application et la partie concernant les demandes des clients :



FIGURE 4.10: Interface d'accueil



FIGURE 4.11: Interface des demandes clientes

L'interface ci-dessus représente la partie concernant les demandes de changements effectuées par les clients. L'utilisateur doit choisir si sa requête est une demande d'un nouveau projet, amélioration ou modification d'une tâche d'un projet déjà réalisé ou la correction d'un bug dans l'une des applications d'Amen Bank.

La figure 4.12 suivante représente le cas d'une demande d'un nouveau projet :

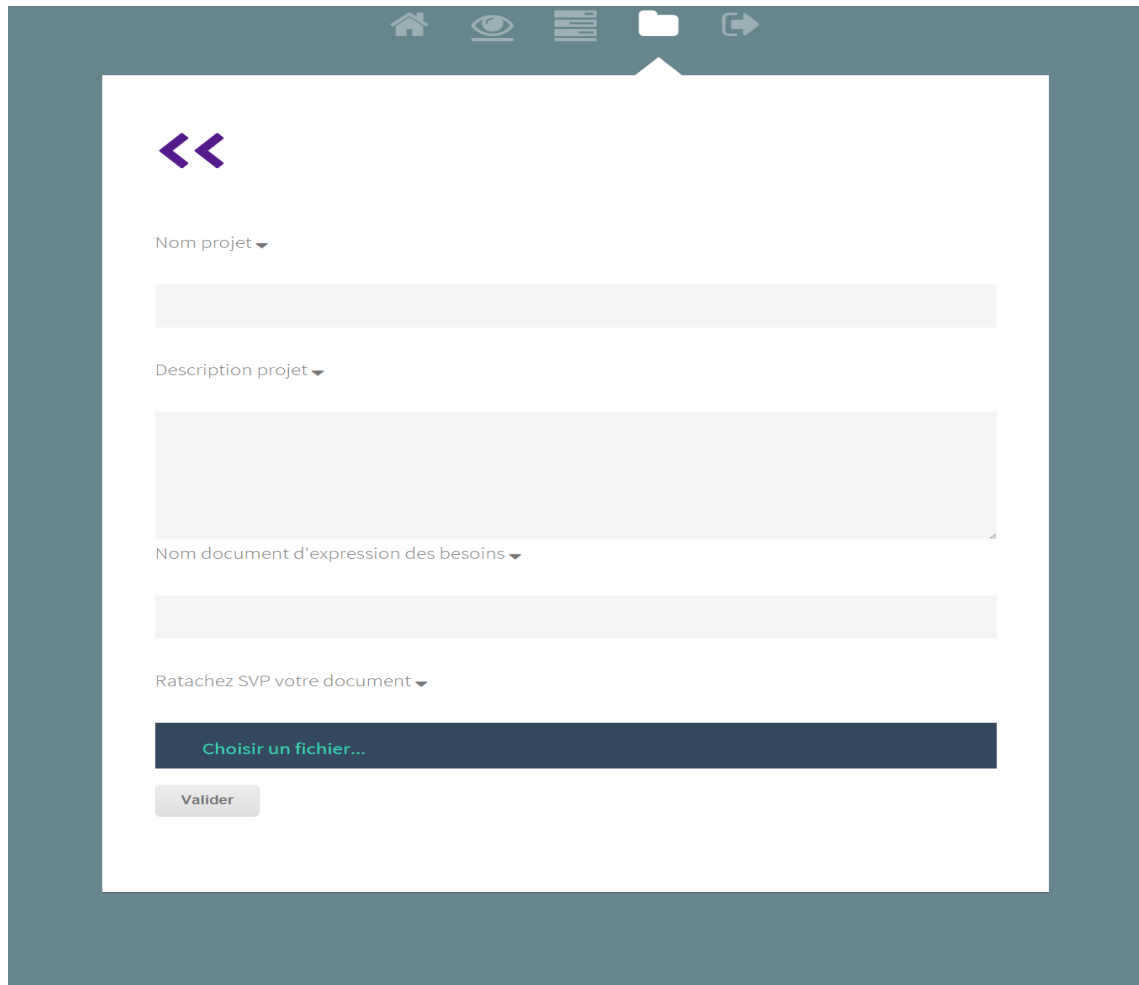
The image shows a web application interface for creating a new project. At the top, there is a navigation bar with icons for home, eye, list, folder, and share. Below this, a white form is centered on a dark teal background. The form starts with a purple double-left arrow icon. It contains four text input fields, each with a label and a dropdown arrow: 'Nom projet', 'Description projet', 'Nom document d'expression des besoins', and 'Ratachez SVP votre document'. The first three fields are empty, while the fourth contains a dark blue button with the text 'Choisir un fichier...'. At the bottom of the form is a grey button labeled 'Valider'.

FIGURE 4.12: Interface des demandes d'un nouveau projet

Cette interface représente le cas où le client désire demander un nouveau projet, alors il remplit les champs du formulaire : le nom du projet, sa description, le nom du document d'expression des besoins et rattache enfin son document.

D'une part une ligne dans la table demande et la table projet sera ajoutée, d'autre part, le document sera enregistré dans la base de données et un nouveau historique sera créé associer à ce nouveau projet. La figure 4.13 ci-dessous représente le cas d'une demande d'amélioration d'une tâche :



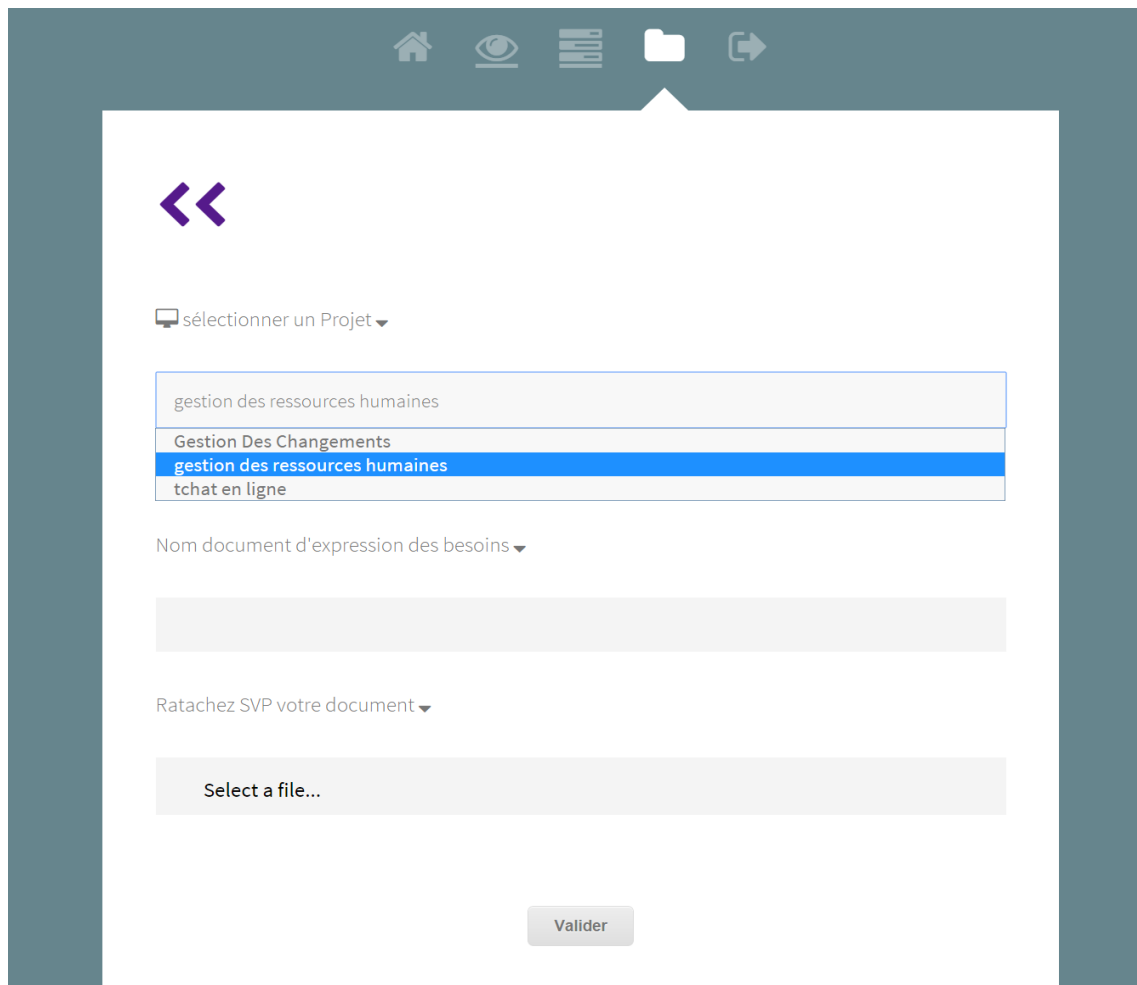


FIGURE 4.13: Interface de demande d'amélioration d'une tâche

Dans le cas où le client désire améliorer une tâche ou une partie d'un projet existant déjà, alors il doit sélectionner le nom du projet, saisir le nom du document d'expression des besoins et le rattacher.

La demande sera enregistrée dans la base de données ainsi que le document qui sera sauvegardé dans la table document.

### V.1.3 Tests relatifs au sprint 1

Après la réalisation du premier sprint, nous avons effectué les tests pour savoir si nous avons pu atteindre les objectifs définis. Et pour cela, nous nous basons sur la rétrospective du sprint :

- Ce qui s'est bien passé :
  - Les tâches de ce module se sont bien déroulées. Il n'y a pas eu des problèmes dans ce sprint.

- Ce qui peut être mieux fait :
  - la présentation des interfaces peut être améliorée,
  - la validation du formulaire l'authentification peut être enrichie d'avantage.
- Améliorations :
  - l'affichage de l'interface de la demande des clients.

### **V.2 Sprint 2 - « Traitement des requêtes des clients »**

Dans cette partie, nous nous intéressons à une tâche principale du sprint 2 étant la gestion et le traitement des demandes effectuées par les utilisateurs de notre système.

#### **V.2.1 Conception détaillée du sprint 2**

En effet, pour planifier des événements liés à la gestion des requêtes clientes, nous avons construit deux diagrammes de séquence le premier est conçu pour le rattachement et le deuxième est conçu pour le téléchargement des documents des autres utilisateurs de notre système.

Les diagramme de séquence suivants explique l'enchaînement des tâches rattachement et téléchargement des documents :

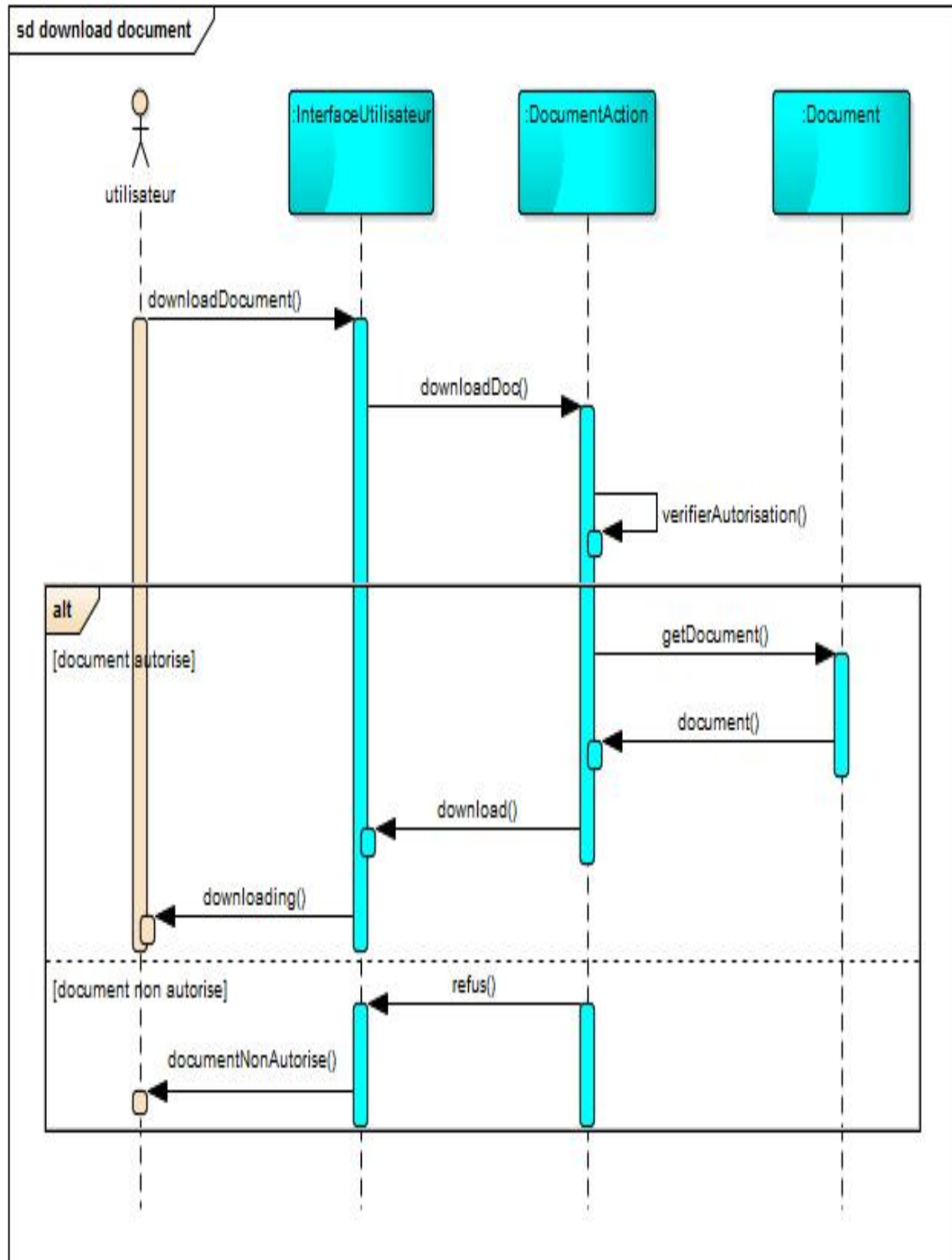


FIGURE 4.14: Diagramme de séquence « Téléchargement des documents »

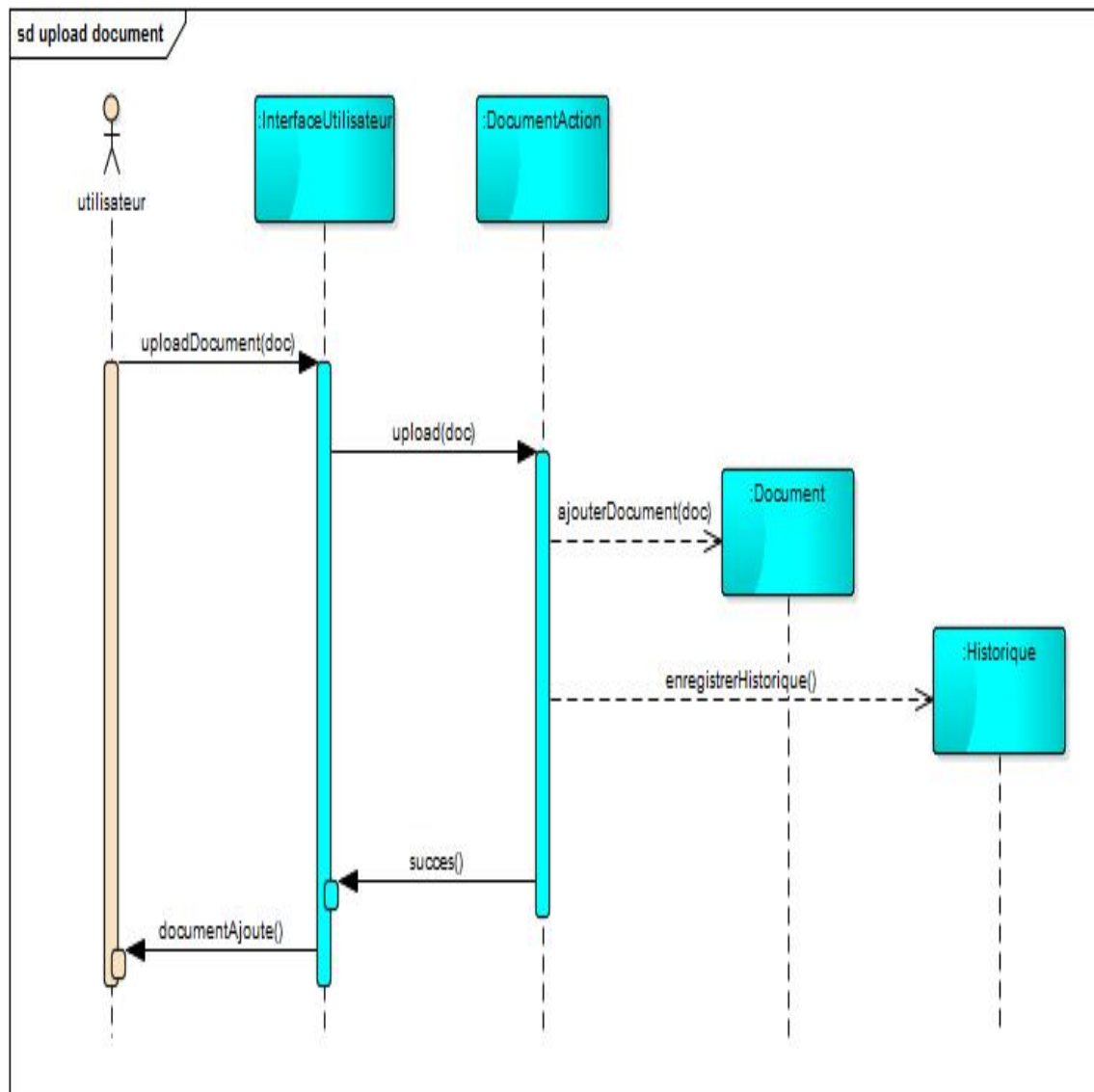


FIGURE 4.15: Diagramme de séquence « Rattachement des documents »

Notre application repose sur le rattachement et le téléchargement des documents, de ce fait, chaque utilisateur a la possibilité de joindre un ou plusieurs documents selon son rôle et selon la tâche qu'il est chargé de réaliser.

L'employé choisi le fichier qu'il désire rattacher, ce dernier sera sauvegardé dans la table document de notre base de données et l'acteur responsable de la tâche suivante sera notifié. De plus la table historique de notre base de données sera mise à jour.

Dans le cas où l'utilisateur de notre système désire télécharger un document qu'il lui sera peut être utile dans d'autres tâches, alors le contrôleur vérifie si cet utilisateur est autorisé ou non de le downloader.

S'il est autorisé, alors l'utilisateur va recevoir ce document dans son poste. Sinon il

sera averti qu'il n'est pas autorisé de télécharger ce fichier.

### V.2.2 Développement du sprint 2

La figure 4.16 ci-dessous présente la partie concernant les employés du département informatique.



FIGURE 4.16: Interface des utilisateurs du département informatique d'AMEN BANK

Pour commencer la phase de gestion des requêtes clientes, les employés du département informatique d'AMEN BANK consultent s'il y a des notifications concernant les demandes de changement des clients. La figure 4.17 ci-dessous illustre cette partie :

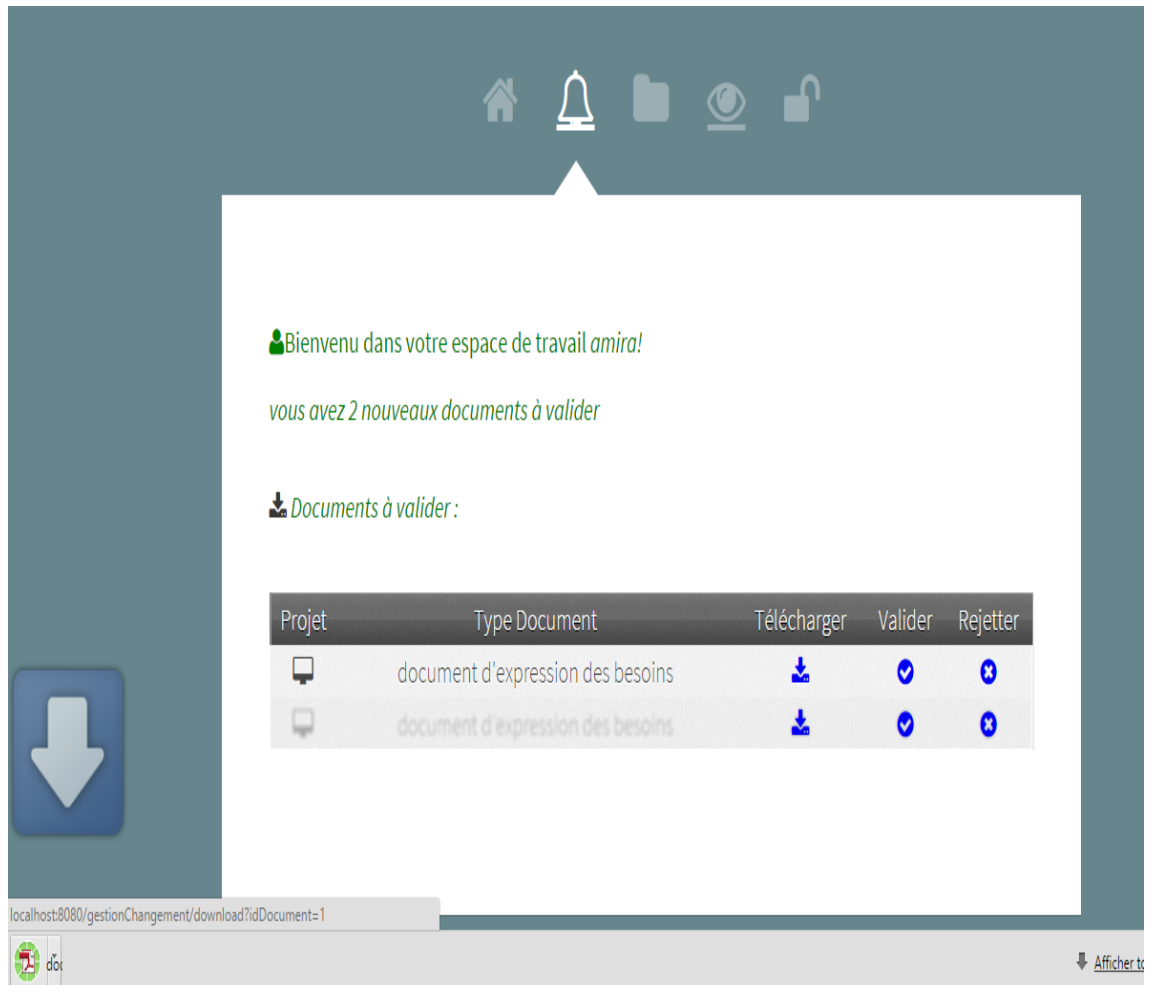


FIGURE 4.17: Interface de consultation des notifications

Cette interface illustre la phase de validation des différents documents par l'employé chargé de la tâche actuelle.

FIGURE 4.18: Interface de validation des documents

Après avoir télécharger un document concernant la tâche précédente d'une requête cliente, L'employé concerné de la phase actuelle, doit valider sa tâche en rattachant les documents nécessaires.

### V.2.3 Tests relatifs au sprint 2

Connaître le résultat des tests est très important pour déduire les défauts de ce que nous avons réalisé :

- Ce qui s'est bien passé :
  - Toutes les tâches ont été réalisées avec succès malgré que nous avons eu des challenges au début avec le téléchargement et rattachement des documents en utilisant le framework Struts2. Nous avons également eu des challenges surtout avec le CSS.

- Ce qui peut être mieux fait :
  - Amélioration de l'interface de la gestion des demandes en proposant une structure plus flexible pour l'utilisateur.
- Améliorations :
  - Amélioration du style de l'interface des utilisateurs internes au département informatique et enrichissement de son contenu.

### **V.3 Sprint 3 – «Suivi des projets et consultation de leurs historiques»**

Tout au long de ce sprint, nous nous intéressons à une tâche importante dans notre système étant le suivi de la liste des projets réalisés et la consultation de l'historique des différentes tâches effectuées d'une demande donnée.

#### **V.3.1 Conception détaillée du sprint 3**

Nous présentons dans cette partie le scénario de la consultation de la liste des demandes de changement en cours de développement ainsi que leurs historiques.



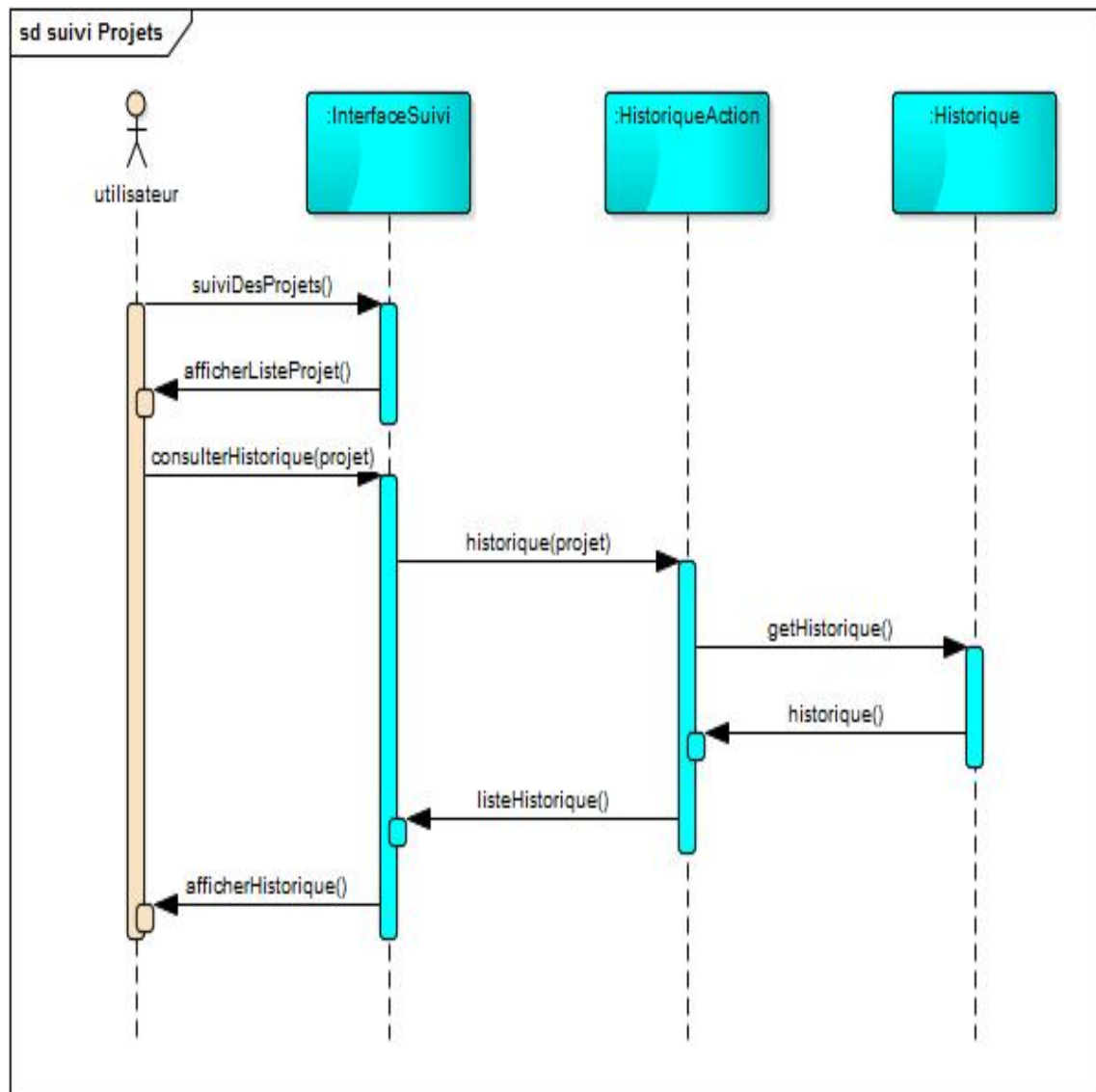


FIGURE 4.19: Diagramme de séquence « Suivi de la liste des demandes et de leurs historique»

La consultation de l'historique est une étape primordiale dans chaque phase du projet. Elle permet d'avoir une trace sur chaque activité, les acteurs impliqués, la date de modification et le déroulement de cycle de vie de la requête cliente.

Nous présentons dans cette partie le scénario des réclamations et du contact entre les utilisateurs de notre système.

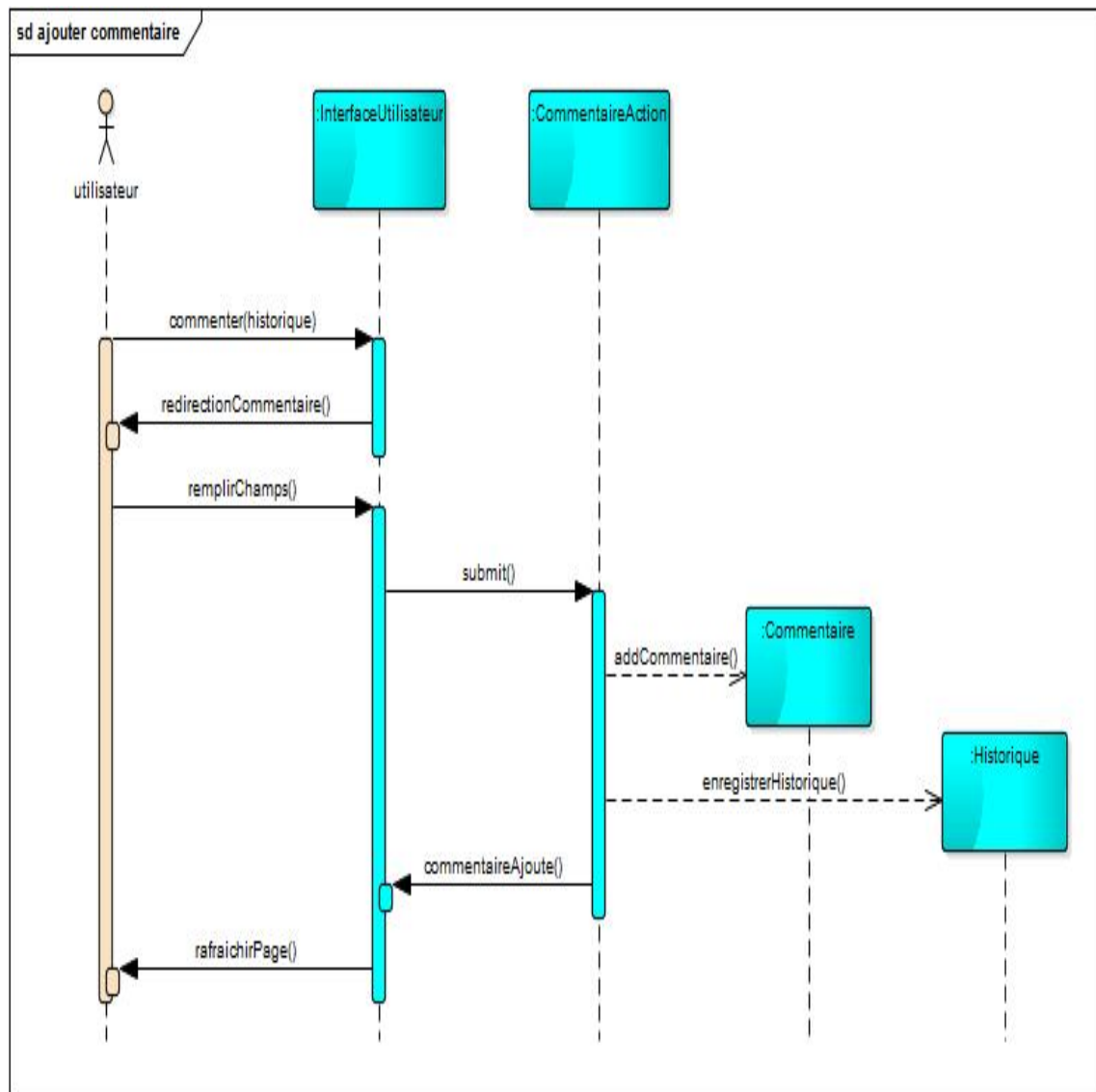


FIGURE 4.20: Diagramme de séquence « Ajout d'une réclamation »

Lorsqu'un utilisateur veut faire une réclamation concernant une tâche précise lors du suivi de l'historique, un lien le redirige vers la section destinée aux commentaires des utilisateurs. Il remplit les champs nécessaires puis il envoie sa réclamation.

Dans ce cas, la classe `CommentaireAction` enregistre le message dans la base de données et garde une trace dans l'historique de cette opération.

### V.3.2 Développement du sprint 3

Nous nous focalisons dans cette partie sur les différents types d'interfaces implémentées. Pour commencer, nous présentons l'interface de suivi des demandes en cours.

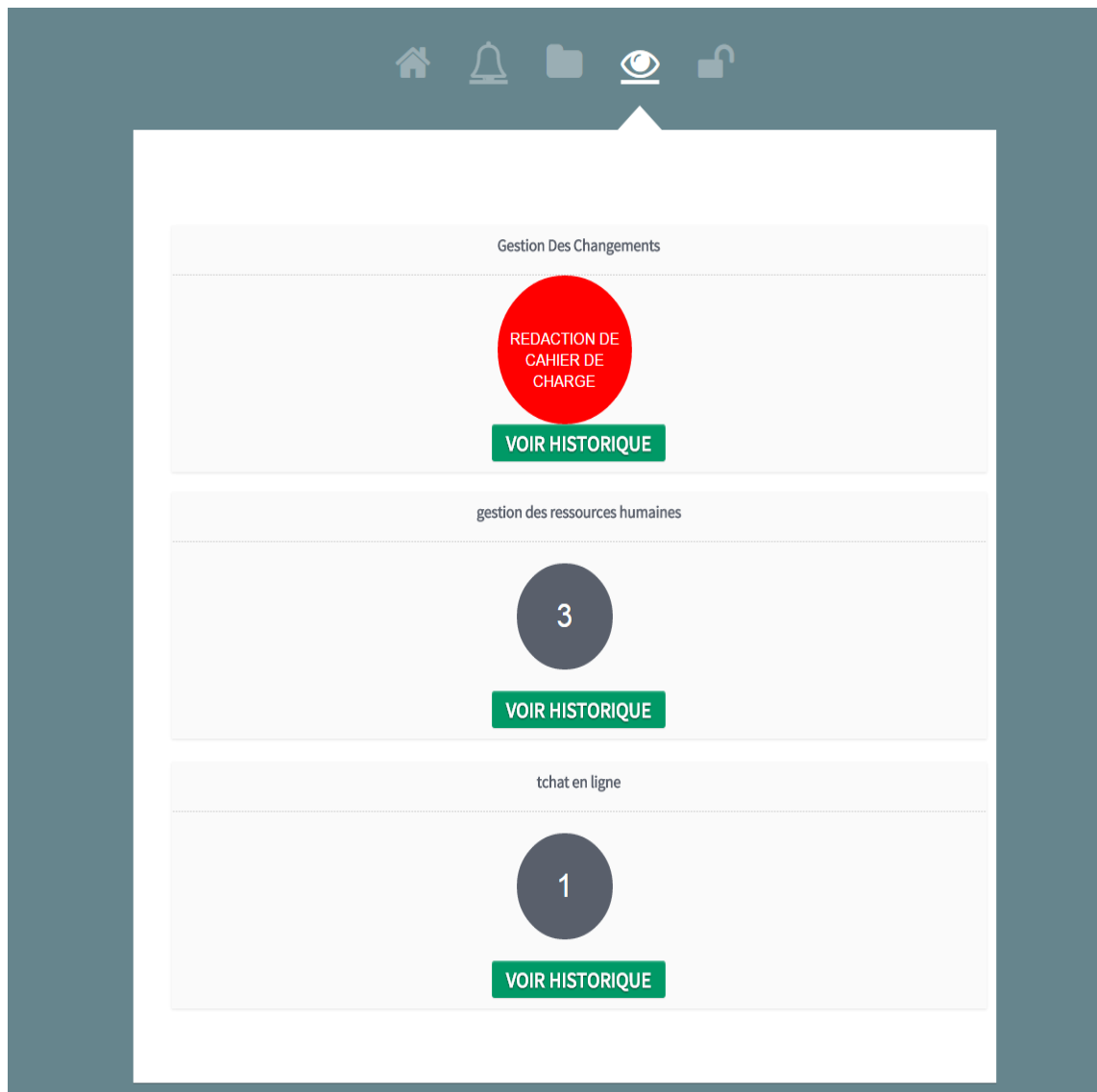


FIGURE 4.21: Interface de suivi des demandes clientes

Chaque utilisateur de notre système, par le biais de cette interface, peut faire le suivi de la liste des projets en cours de développement. Cette partie est importante surtout pour le client, elle lui permet de suivre pas à pas sa demande tout au long son cycle de vie et de voir dans quelle phase elle est, ainsi que les éventuelles modifications effectués et les acteurs impliqués à cette tâche.

Cette interface offre aussi aux utilisateurs de notre application la possibilité de faire des réclamations sur une ou plusieurs tâches bien déterminées, de consulter l'historique des commentaires des autres utilisateurs en les redirigeant vers la partie destinée aux réclamations et commentaires. La figure 4.22 représente un exemple de consultation d'une demande :

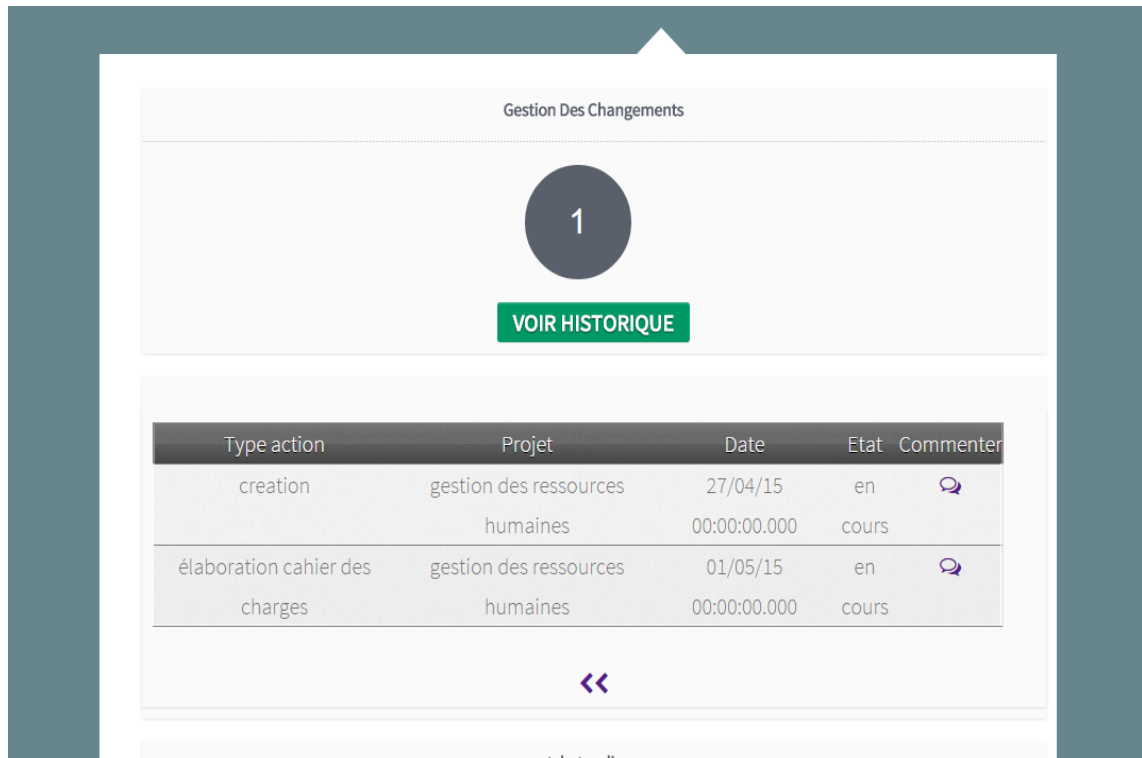


FIGURE 4.22: Interface de consultation des demandes en cours

Cette interface permet aux utilisateurs de notre système de consulter les réclamations des autres utilisateurs. Ils ont aussi la possibilité de faire une réclamation concernant une tâche précise :

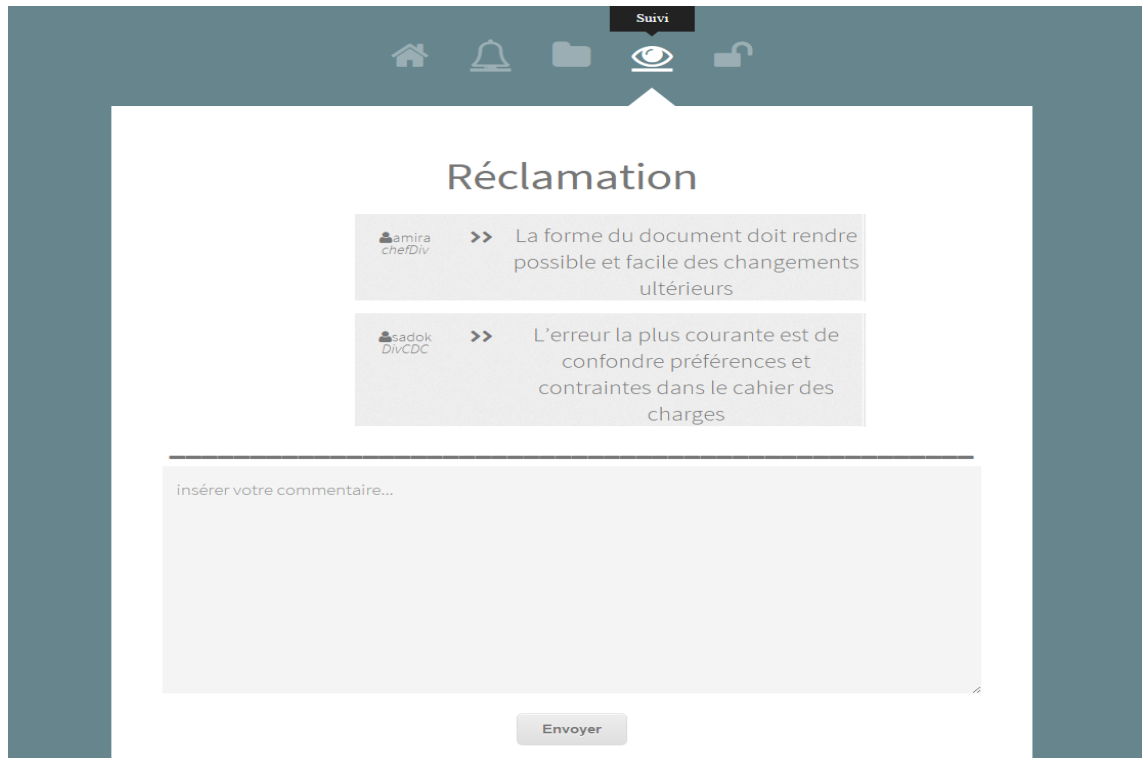


FIGURE 4.23: Interface des réclamations

Un utilisateur de notre système désire réclamer une tâche d'une demande de changement, remplit le formulaire des réclamations et envoie son commentaire. Il peut voir aussi tout les commentaires concernant cette tâche, effectués par d'autre utilisateur de notre application.

### V.3.3 Tests relatifs au sprint 3

Pour ce sprint, nous avons eu les résultats suivants :

- Ce qui s'est bien passé :
  - Nous avons eu des challenges pendant cette phase qui concerne l'insertion l'intervention à partir de l'interface de la liste des projets. Mais, ce problème a été réglé après un bon moment et nous avons pu répondre aux besoins d'Amen Bank.
- Ce qui peut être mieux fait :
  - Les vues de consultation peuvent être encore plus enrichies de critères afin de répondre aux besoins de n'importe quel utilisateur.
- Améliorations :

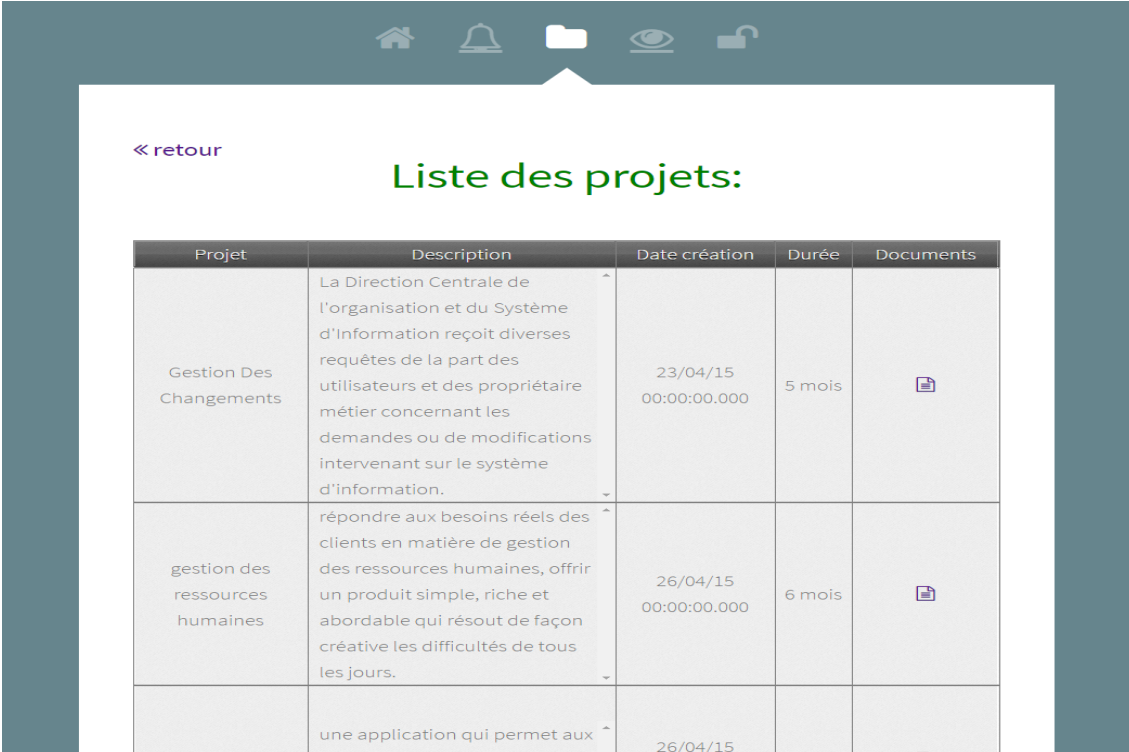
- Amélioration des interfaces déjà réalisées.

## V.4 Sprint 4 – «Consultation de la liste des documents disponibles»

Tout au long de ce sprint, nous nous intéressons à la phase de suivi de la liste des projets réalisés par le département informatique d'AMAN BANK, ainsi que la consultation des différents documents appartenant à une demande.

### V.4.1 Développement du sprint 4

La figure 4.24 suivante illustre la partie concernant la consultation des différentes demandes traitées par la direction centrale de l'organisation et du système d'information :






Projet	Description	Date création	Durée	Documents
Gestion Des Changements	La Direction Centrale de l'organisation et du Système d'Information reçoit diverses requêtes de la part des utilisateurs et des propriétaire métier concernant les demandes ou de modifications intervenant sur le système d'information.	23/04/15 00:00:00.000	5 mois	
gestion des ressources humaines	répondre aux besoins réels des clients en matière de gestion des ressources humaines, offrir un produit simple, riche et abordable qui résout de façon créative les difficultés de tous les jours.	26/04/15 00:00:00.000	6 mois	
	une application qui permet aux	26/04/15		

FIGURE 4.24: Interface de suivi des projets réalisés

Un acteur de notre système peut consulter la liste des documents d'une demande de changement déjà réalisée. La figure 4.25 ci-dessous représente une exemple d'historique de demande cliente :



Projet	Description	Date création	Durée	Documents
Gestion Des Changements	La Direction Centrale de l'organisation et du Système d'Information reçoit diverses requêtes de la part des utilisateurs et des propriétaire métier concernant les demandes ou de modifications intervenant sur le système d'information.	23/04/15 00:00:00.000	5 mois	
Liste des documents disponibles:				
<div>document d'expression des besoins document d'analyse des besoins cahier des charges</div>				
<div>télécharger</div>				
				

FIGURE 4.25: Interface de consultation des différents documents d'une demande de changement

#### V.4.2 Tests relatifs au sprint 4

Pour ce sprint, nous avons pu déduire les résultats suivants :

- Ce qui s'est bien passé :
  - Cette partie a pris énormément de temps mais elle s'est bien déroulée.
- Ce qui peut être mieux fait :
  - La présentation des données peut être mieux améliorée.
- Améliorations :

- Nous pouvons enrichir la partie de consultation des réclamations afin qu'elle soit en harmonie avec notre application

## VI Chronogramme

Ce travail a été réalisé durant une période de 4 mois. La répartition des tâches durant toute la durée de la réalisation de notre application est illustrée par la diagramme de Gantt réel de la figure suivante :

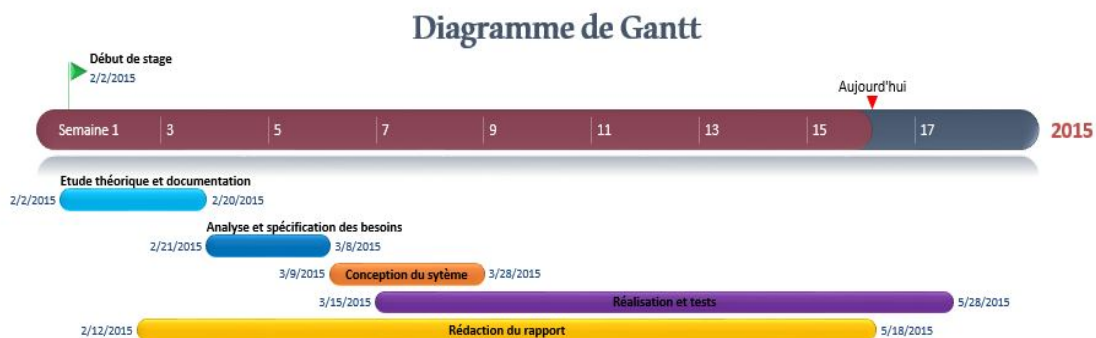


FIGURE 4.26: Diagramme de Gantt réel

## Conclusion

Dans ce chapitre, nous avons présenté l'architecture de notre système ainsi que l'environnement matériel et logiciel nécessaire pour notre travail. Nous avons ensuite abordé la conception générale de notre solution et les différentes configurations effectuées afin de préparer un milieu adéquat pour la réalisation de notre application.

Et pour finir, nous avons détaillé les différents sprints qui composent notre projet ainsi que les différentes interfaces qui implémentent notre projet. Nous évoquons dans la conclusion générale un résumé du travail effectué et nous présentons les perspectives ainsi que nos acquis sur le plan personnel et professionnel.



# Conclusion générale

Dans ce présent rapport, nous avons présenté, en premier lieu, le contexte général du projet qui s'est déroulé au sein d'AMEN BANK. Nous avons réalisé une étude de l'existant. Nous avons abordé aussi la méthodologie appliquée pour la bonne gestion de notre projet.

Par la suite, nous avons exposé la planification de notre travail qui nous a permis de détailler les différentes tâches de notre solution en précisant les délais et l'ordre de priorité.

Après, nous avons présenté le backlog du produit de notre solution ainsi que les besoins fonctionnels et non fonctionnels. Nous avons évoqué aussi l'analyse de ces besoins. Et à la fin, nous avons présenté notre application en détaillant chaque sprint.

La phase de l'élaboration du cahier des charges et de l'étude approfondie sur le cycle de vie d'une demande a requis un effort considérable pour la recherche des informations nécessaires pour déduire notre solution. La période que nous avons consacrée pour apprendre à manipuler la plateforme Java EE et le framework Struts 2 a été très critique vu qu'une telle plateforme et tel framework nécessitent un temps considérable afin de connaître ses différentes notions. Mais, nous avons pu en un temps très restreint de nous familiariser avec elle.

Grâce à notre expérience au sein d'AMEN BANK, nous avons appris, d'un point de vue personnel, à gérer notre projet de façon méthodique et organisée. Ce travail nous a été bénéfique dans la mesure où il nous a permis de mettre en pratique nos connaissances théoriques acquises tout au long de notre formation à l'Institut Supé-

rieur d'Informatique. Il nous a permis également d'approfondir nos connaissances et d'apprécier l'importance d'une méthodologie de gestion de projet. Notre projet est donc une source d'enrichissement technique, culturel, personnel et humain.

La phase de réalisation de notre solution a aussi nécessité un grand effort en vue de satisfaire les besoins d'AMEN BANK et de rendre un produit livrable dans les délais.

La communication était un facteur très important au niveau de cette phase pour bien préparer et réaliser l'application souhaitée. Nous avons rencontré certaines challenges pendant la phase de réalisation de notre application. Mais malgré tout, nous avons pu trouver des solutions au bon moment pour ne pas retarder le déroulement de notre projet.

Le travail en équipe et la notion de collaboration régnant au sein d'AMEN BANK nous a facilité notre intégration et notre reconnaissance des procédures de travail. La relation entre stagiaires et ingénieurs a été très enrichissante et conviviale. Le climat de confiance et de respect avec les supérieurs a été effectivement favorable pour aboutir à une meilleure productivité. Chaque étape de notre projet a nécessité un effort et une recherche approfondie.

Nous avons pensés à rajouter les notions de Workflow à notre application mais le product owner nous a dit que c'est en dehors du périmètre du projet qui s'agit d'automatiser uniquement le processus de gestion des changements.

Comme perspectives de travaux futurs, nous proposons d'enrichir cette application en s'intéressant à certains points. Pour étendre notre solution, nous proposons de présenter dynamiquement le cycle de vie d'une demande sous forme de Workflow permettant ainsi de donner une vue globale sur les états possibles d'une demande. Nous pouvons également offrir aux employés du département informatique une messagerie instantanée reliée aux demandes des changements.

# Bibliographie

- [1] The Standish Group Report, Chaos, 1995.
- [2] Eveleens et Verhoef The Rise and Fall of the Chaos Report Figures, IEEE Software, Chris Verhoef, 2010.
- [3] Formation sur SCRUM, Timwi Consulting, 2012.
- [4] Planification de projet : Fondements, méthodes et techniques, Jacques Boy, Christian Dudek, Sabine Kuschel, 2003.
- [5] Managing Successful Projects with PRINCE2, Office of Government Commerce, 2009.
- [6] Management des projets : techniques d'évaluation, analyse, choix et planification, Emmanuel Djuatio, 2001.
- [7] OBS : De l'ingénierie d'affaires au management de projet, Henri Georges Minyem, 2007.
- [8] JavaScript : Introduction et notions fondamentales, Luc Van Lancker, 2008.
- [9] Ajax, jQuery et PHP : 42 ateliers pour concevoir des applications Web 2.0, JeanMarie Defrance, 2013.
- [10] JavaScript : Gardez le meilleur, Douglas Crockford, 2008.

# Webographie

- [w1] <http://tahe.developpez.com/java/struts2/>
- [w2] <http://default.developpez.com/tutoriel/java/eclipse/hibernate/#L2>
- [w3] <http://www.amenbank.com.tn/>
- [w4] [http://varrette.gforge.uni.lu/download/polys/Tutorial\\_Latex.pdf/](http://varrette.gforge.uni.lu/download/polys/Tutorial_Latex.pdf/)
- [w5] [http://fr.wikipedia.org/wiki/Apache\\_Tomcat](http://fr.wikipedia.org/wiki/Apache_Tomcat)
- [w6] <http://fortawesome.github.io/FontAwesome/>
- [w7] [http://fr.wikipedia.org/wiki/Work\\_Breakdown\\_Structure](http://fr.wikipedia.org/wiki/Work_Breakdown_Structure)
- [w8] <http://forum.primefaces.org/viewtopic.php>
- [w9] <http://documentation.bonitasoft.com/5x/bos-57/gettingstarted/introduction-tutorials>
- [w10] [http://fr.wikipedia.org/wiki/Java\\_EE](http://fr.wikipedia.org/wiki/Java_EE)
- [w11] <http://eprint.insatoulouse.fr/135/1/CitlalihGutierrez.pdf>

# Annexes