

RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du

Diplôme National d'Ingénieur en Sciences Appliquées et Technologiques

Spécialité : Génie Logiciel et Systèmes d'Information

Par

Prénom NOM

Titre du projet

Encadrant professionnel : Monsieur/Madame Prénom NOM Ingénieur R&D

Encadrant académique : Monsieur/Madame Prénom NOM Maître Assistant(e)

Réalisé au sein de Tunisie Télécome



RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du

Diplôme National d'Ingénieur en Sciences Appliquées et Technologiques

Spécialité : Génie Logiciel et Systèmes d'Information

Par

Prénom NOM

Titre du projet

Encadrant professionnel : Monsieur/Madame Prénom NOM Ingénieur R&D

Encadrant académique : Monsieur/Madame Prénom NOM Maître Assistant(e)

Réalisé au sein de Tunisie Télécome



J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant professionnel, **Monsieur/Madame Prénom NOM**

Signature et cachet

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant académique, **Monsieur/Madame Prénom NOM**

Signature

Dédicaces

Je dédie ce travail à :

Monsieur **Monsieur/Madame Prénom NOM**, Monsieur **Monsieur/Madame Prénom NOM** pour m'avoir encadré et fait de leurs mieux afin de m'aider.

etc.

Prénom NOM

Remerciements

Je remercie

Je suis reconnaissant

J'exprime ma gratitude

Table des matières

Introduction générale	1
1 Contexte général	2
1.1 Présentation de l'entreprise	3
1.1.1 Le groupe Sofrecom	3
1.1.2 Sofrecom Tunisie	3
1.2 Contexte du projet	4
1.2.1 Présentation de l'application Panoramix	4
1.2.2 Limites et critiques de l'existant	4
1.3 Solution proposée	5
1.4 La méthodologie : SCRUM	6
2 Analyse préliminaire	7
2.1 Spécification des besoins	8
2.1.1 La présentation des acteurs	8
2.1.2 Les besoins fonctionnels	9
2.1.3 Diagramme de cas d'utilisation global	10
2.1.4 Les besoins non fonctionnels	10
2.2 Structure et découpage du projet	11
2.2.1 Identification des rôles dans l'équipe SCRUM	11
2.2.2 Planification d'un projet par Scrum	11
2.2.3 Le Product Backlog du produit	12
2.2.4 Planification des sprints	13
2.3 Environnement de travail	14
2.3.1 Environnement matériel	14
2.3.2 Environnement de développement	14
2.3.3 Environnement logiciel	16
2.4 L'architecture de la solution	19
2.4.1 L'architecture logique	19

2.4.2	L'architecture logicielle	21
2.5	Conclusion	22
3	Conception et choix technologiques	23
3.1	Conception détaillée	24
3.1.1	Diagramme des classes	24
3.1.2	Diagramme des séquences	24
3.2	Conception architecturale	24
3.3	Choix technologiques	24
4	Réalisation	25
4.1	Choix techniques	26
4.2	Travail réalisé	26
4.3	Planning réel du projet	26
	Conclusion générale	27
	Bibliographie	28
	Annexes	29
	Annexe 1. Exemple d'annexe	29
	Annexe 2. Entreprise	30

Table des figures

2.1	Les acteurs de système	8
2.2	Diagramme de cas d'utilisation générale	10
2.3	Description de processus SCRUM	12
2.4	Logo Eclipse IDE	14
2.5	Logo Visual studio Code	15
2.6	Logo Laragon	15
2.7	Logo Apache	15
2.8	Logo MariaDB	15
2.9	Logo HeidiSQL	16
2.10	Logo SeleniumLibrary	16
2.11	Logo Robot Framework	16
2.12	Logo Jenkins	17
2.13	Logo Javascript	17
2.14	Logo jQuery	18
2.15	Logo Axios	18
2.16	Logo HTML5	18
2.17	Logo PHP	18
2.18	Logo Orange framework & tools	19
2.19	Logo Boosted	19
2.20	L'architecture logique des tests de non régression	20
2.21	L'architectures 3 tiers du Web	21
2.22	L'architecture logicielle de l'application	22
Annexe 2.1	Logo d'entreprise	30

Liste des tableaux

2.1	Les tâches des acteurs de système	9
2.2	Backlog de produit	13
2.3	Planification des sprints	14
Annexe 1.1 Exemple tableau dans l'annexe		29

Liste des abréviations

- **DSI** = Direction des Systèmes d'information
- **OLS** = Orange Labs Services

Introduction générale

Exemple d'utilisation de la bibliographie utilisée [1]. Le style utilisé est IEEE [2].

Une introduction d'une à 3 pages où vous poserez clairement le problème auquel vous allez tenter d'apporter une solution. L'introduction se rédige à la fin de votre travail de rédaction. Avant de rédiger l'introduction, structurez TOUT le PFE. L'introduction peut se faire en même temps que la conclusion.

L'introduction sert trois objectifs :

- elle introduit le sujet. Ceci signifie qu'il faut présenter succinctement le contexte général du travail accompli, par exemple l'environnement professionnel et l'entreprise pour un rapport de stage, puis définir le sujet en termes précis et concis ;
- elle énonce ensuite succinctement les objectifs du travail personnel, et les moyens mis en œuvre pour tenter de les atteindre ;
- elle s'achève sur une présentation claire du plan adopté pour la suite du corps du rapport.

L'annonce du plan se fait au futur et doit être rédigée en entier.

L'introduction générale doit développer les points suivants :

- la présentation du contexte du projet (domaine exemple : télécommunication, sécurité, automate etc.) ;
- la présentation brève de l'entreprise d'accueil et de son domaine ;
- la description des objectifs du PFE/ Mémoire : justifier le sujet et poser le problème à résoudre ; indiquer la manière dont il sera traité en terme d'outils et de méthodes ; donner les raisons qui président à ce choix ; exposer les intérêts du sujet et sa problématique ;
- l'annonce du plan du rapport sans trop détailler. Il est recommandé, à partir de l'introduction générale, de recourir au « nous » de modestie.

CONTEXTE GÉNÉRAL

Plan

1	Présentation de l'entreprise	3
2	Contexte du projet	4
3	Solution proposée	5
4	La méthodologie : SCRUM	6

Introduction

Dans ce premier chapitre introductif, nous présentons l'organisme d'accueil Sofrecom et sa filiale Sofrecom Tunisie. Ensuite, nous introduisons le contexte du sujet, la problématique et un diagnostic technique de la solution existante. Enfin, nous présentons la méthode de développement choisie pour la réalisation de notre solution.

1.1 Présentation de l'entreprise

1.1.1 Le groupe Sofrecom

Sofrecom, filiale d'Orange, développe depuis 50 ans un savoir-faire unique dans les métiers de l'opérateur, ce qui en fait un leader mondial du conseil et de l'ingénierie télécom. Ces dernières années, plus de 200 acteurs majeurs, dans plus de 100 pays, ont confié à Sofrecom la conduite de leurs projets stratégiques et opérationnels. Le Know-How Network de Sofrecom, c'est aussi la garantie d'un transfert de savoir-faire, de compétences et d'expertises pour une transformation durable s'appuyant sur des méthodologies certifiées au niveau international.

1.1.2 Sofrecom Tunisie

Sofrecom Tunisie créée en Octobre 2012, considérée comme la filiale la plus jeune et la plus importante du groupe Sofrecom en zone Afrique et Moyen Orient. Au court de 5 ans, elle a pu se positionner en tant qu'un acteur majeur d'ingénierie en télécommunications et du conseil.

Sofrecom Tunisie compte aujourd'hui plus que 560 experts, et deux clients majeurs qui font partie du groupe Orange : DSI France et OLS. Sofrecom Tunisie propose à ses clients une large gamme des services autour de huit spécialités :

- Ingénierie
- Architecture
- Support et maintenance
- Sécurité informatique
- Expertise technique
- Développement
- Innovation

— Consulting

Notre projet concerne le métier du **développement**.

1.2 Contexte du projet

Dans cette section, nous commencerons, dans un premier temps, par présenter l'application Panoramix. Nous citerons les différents manques, les points faibles et les problèmes, puis nous allons proposer notre solution.

1.2.1 Présentation de l'application Panoramix

Panoramix est une application web, développé par Sofrecom depuis 2016, qui servira de point d'entrée unique pour les positions de travail de la vente sur le segment Pro PME (entreprise de taille inférieure à 100 employées).

Ce portail adapté à chaque position de travail, permettra de traiter toutes les demandes client en simplifiant et fluidifiant le parcours des conseillers.

L'introduction de parcours guidés, là où il y avait des applications et des process à mémoriser, facilitera la montée en compétence d'une nouvelle recrue et masquera la complexité afin de se concentrer sur comment répondre au mieux à la demande client. Le conseiller réactif pourra lui aussi organiser son activité, grâce à une meilleure visibilité sur les dossiers client qu'il gère ou qui lui sont affectés. Au centre du portail, sera la vue 360 du client avec ses interlocuteurs, son parc, son historique. . . . Enfin, Panoramix est un moyen sûr d'être dans la posture adéquat pour respecter les règles de saine concurrence.

1.2.2 Limites et critiques de l'existant

Après chaque itération le projet Panoramix, subit des améliorations. Des fonctionnalités s'ajoutent pour s'aligner plus au besoin du client. Le projet a commencé depuis des années, et selon son plan d'évolution, il continuera à évoluer encore pour quelques années. Ceci a généré plusieurs défis. Et parmi les nouveaux défis, la fusion entre deux métiers différents dans le même socle :

- Le métier de Centre d'appel Orange ou bien 3901
- Le métier de Boutique Orange ou bien CPRO
 - Boutique GDT (générale de téléphone)
 - Boutique AD (Boutique Orange)

qui sont deux métiers séparés mais ils ont des grandes parties communes et ils ont le même client cible(Pro PME).

Cette différence ne concerne pas seulement le niveau d'interaction avec le client mais la différence au niveau métier aussi tel que les boutiques ne peuvent pas accéder aux mêmes applications que centre d'appel : l'accès aux applications est limité selon le type et l'emplacement de la boutique, par contre le centre d'appel à l'accès à toutes les applications.

Cette mise à jour permet à Panoramix d'ajouter entre 9000 et 12000 utilisateurs aux 2650 utilisateurs existants déjà et de maintenir environ 2500 utilisateurs actifs mais la solution actuelle n'est pas assez performante et l'infrastructure actuelle ne peut pas supporter ce nombre des utilisateurs. Et par conséquence, certains problèmes s'imposent. En premier lieu, nous remarquons que les informations ne sont pas centralisées, en deuxième lieu il y'a une différence d'interprétation de la fiche client et des applications et en plus le centre d'appel et les boutiques n'ont pas la même historique des réclamations de clients.

1.3 Solution proposée

Dans le souci d'apporter une valeur ajoutée et un meilleur service technologique aux clients et au groupe lui-même, nous envisageons de :

- Fournir **un point d'accès unique aux outils et aux informations** du quotidien permettant de traiter toutes les demandes client inhérentes à une position de travail pour les deux types des utilisateurs (Boutique et centre d'appel)
- Assurer **une ergonomie optimisée et homogène**
- **Interconnecter les outils** pour supprimer les ressaisies et les ruptures de processus
- **Personnaliser et filtrer les informations** dont l'utilisateur métier a besoin
- **Un niveau de performance optimal** : l'application doit être rapide, sans temps de latence
- Un accès aux informations et outils **dont l'utilisateur a besoin, et uniquement celles nécessaires**, pour assurer son métier au quotidien
- **Haute disponibilité** même avec la différence horaire avec les pays DOM-TOM

Ces promesses seront bénéfiques pour :

- **Orange** : Réduire la complexité des projets et fusion deux métiers différents dans une seule application qui est "Panoramix"
- **Conseiller client** : mieux guidé, plus de confort, accès plus rapide aux infos
 - Réduction du temps de traitement
 - Plus efficace, plus à l'écoute du client
- **Client** : réduction du délai d'attente, baisse des réitérations, baisse du taux de transfert

1.4 La méthodologie : SCRUM

Dans le contexte de notre projet, les dimensions de notre produit ne sont pas fixes dès le début et en plus nous avons besoin de dialoguer et collaborer avec le reste des membres de l'équipe en quotidien pour pouvoir réussir toutes les étapes de production et de déploiement de notre projet. Donc l'utilisation d'une méthode agile est une priorité pour pouvoir réussir la mission dans les meilleures conditions.

Nous avons choisi d'adapter **la méthode Scrum**, utilisée par l'équipe de Panoramix, qui présente une implémentation de l'approche agile.

Conclusion

Dans le premier chapitre, nous avons présenté notre cadre de travail et la méthode de conception et de développement des tâches requises. Nous pouvons passer au chapitre suivant qui est réservé à l'analyse préliminaire.

ANALYSE PRÉLIMINAIRE

Plan

1	Spécification des besoins	8
2	Structure et découpage du projet	11
3	Environnement de travail	14
4	L'architecture de la solution	19
5	Conclusion	22

Introduction

Dans ce chapitre, nous ferons référence aux objectifs de notre application, ce qui nous amène à identifier les possibilités du système et les besoins des utilisateurs que nous essayerons de projeter dans des diagrammes de cas d'utilisations globales.

2.1 Spécification des besoins

Dans cette partie du rapport, nous présenterons les différents acteurs du système, les besoins fonctionnels ainsi que les besoins non-fonctionnels.

2.1.1 La présentation des acteurs

Les acteurs représentent les personnes ou des composants logiciels ou matériels qui interagissent directement avec le système.

Dans notre projet, il y'a 4 acteurs principaux qui manipulent notre site comme l'indique la figure suivant :

- Conseiller client réactif (Soit conseiller client de centre d'appel ou conseiller client de boutique)
- Administrateur des habilitations
- Administrateur
- Système



Figure 2.1: Les acteurs de système

La table 2.1 représentent les tâches de chaque acteur de notre système

Acteur	Rôle
Conseiller client réactif	Accès à l'application Consulter les PEF au fiche client
Administrateur des habilitations	Accès à l'application Gestion des utilisateurs
Administrateur	Accès à l'application Consulter les PEF au fiche client Gérer toute la partie opérationnelle du projet
Système	Assure le déclenchement des tâches automatiques de l'application

Tableau 2.1: Les tâches des acteurs de système

2.1.2 Les besoins fonctionnels

Notre application doit fournir un ensemble de fonctionnalités qui répondent aux exigences des acteurs. Les principales exigences fonctionnelles de notre outil peuvent être résumées comme suit le suivant :

- Recevoir des rapports hebdomadaires des tests de non régression
- Gestion les utilisateurs
- Gestion les rôles
- Consultation des données boutiques
- Affectation des utilisateurs aux boutiques
- Gestion des catégories
- Gestion des pefs
- Visibilité des PEF aux boutiques
- Simulation SRCD
- Consultation log
- Traçabilité

2.1.3 Diagramme de cas d'utilisation global

Dans cette sous-section, nous exposons le diagramme de cas d'utilisation global qui permet de donner une vision globale du comportement fonctionnel de notre système.

La figure 2.2 représente le diagramme de cas d'utilisation global.

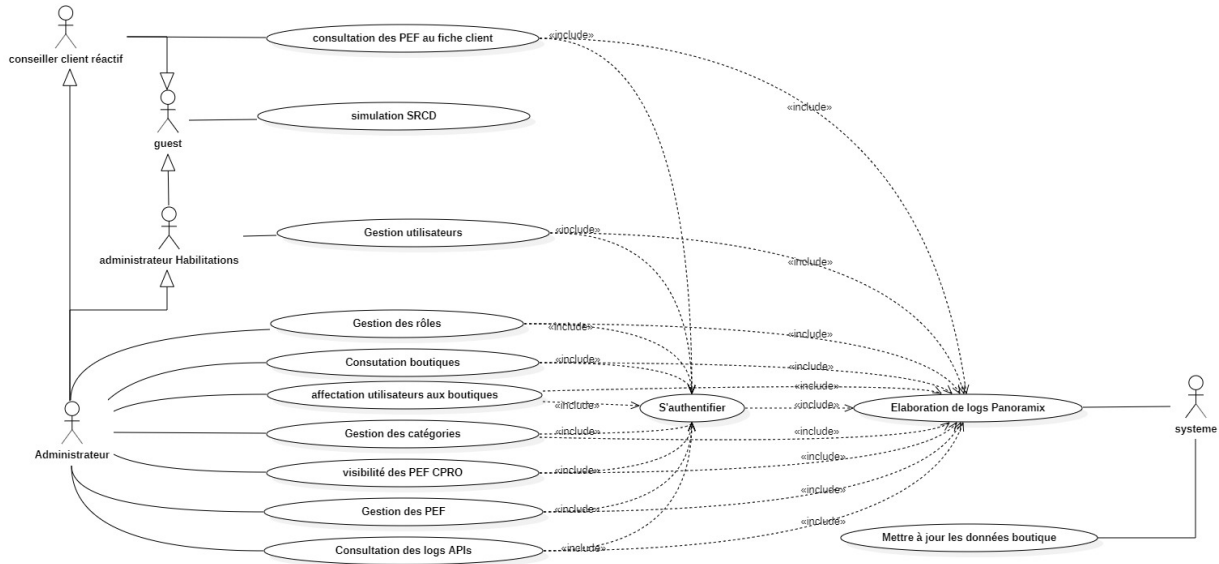


Figure 2.2: Diagramme de cas d'utilisation générale

2.1.4 Les besoins non fonctionnels

Outre des besoins fonctionnels, le futur système doit également répondre aux contraintes suivantes :

- **La rapidité de traitement** : Compte tenu du grand nombre de transactions par jour, le temps de traitement doit être le plus proche possible du temps réel.
- **La performance** : Nous utilisons les performances pour spécifier la durée pendant laquelle le système répond aux demandes d'entrée. Ce terme fait référence à la vitesse à laquelle le système effectue le traitement.
- **La disponibilité** : L'application devrait être opérationnelle d'une façon continue car l'utilisateur peut faire des réservations à tout moment. Le système doit être en permanence à la disposition de ses utilisateurs.
- **Ergonomie et Simplicité** : Cette fonctionnalité permet à l'utilisateur d'être à l'aise lors de l'utilisation ou de la consultation du site.

- **L’extensibilité** : Cela nous donne la possibilité d’ajouter, de modifier ou de supprimer des fonctionnalités.
- **Fiabilité** : Notre application doit être bien testé avant de l’héberger aux clients à fin d’éviter les éventuels des bugs.

2.2 Structure et découpage du projet

Nous présentons, dans la suite, les différents intervenants dans notre projet ainsi que le cycle de vie de la méthode Scrum et nous finissons par la présentation de notre product backlog.

2.2.1 Identification des rôles dans l’équipe SCRUM

Dans un projet SCRUM, l’équipe a un rôle fondamental : elle permet d’optimiser la productivité et la flexibilité. En effet, elle doit être auto organisée et multifonctionnelle.

Cette méthode agile intègre généralement la participation de plusieurs acteurs, dans notre contexte nous avons le "Product owner" qui est la personne qui porte la vision du produit à réaliser, et qui est responsable de la gestion du "backlog produit" et de l’interaction avec l’équipe de développement. Il est généralement un expert du domaine métier du projet. Mme « Nisrine ZIADIA » est le "Product owner" et Mme « Meriem OUEDERNI » joue le rôle de "Product owner Proxi. ".

Le "Scrum master" est la personne qui doit maîtriser la méthodologie SCRUM et s’assurer qu’elle est bien comprise et appliquée. M. « Mohamed Aymen FEKIRI » est le "Scrum master".

L’équipe traduit les exigences en fonctionnalités pour obtenir des incréments utilisables et livrables à la fin de chaque itération. Notre équipe est pluridisciplinaire et auto-organisée. Cette dernière comporte un seul stagiaire, moi-même, Lassad KEFI, étudiant en Ingénierie des systèmes intelligents à L’Ecole Nationale des Ingénieurs de Carthage, qui joue le rôle d’un développeur au sein de l’équipe Panoramix.

2.2.2 Planification d’un projet par Scrum

Pour appliquer correctement SCRUM, il faut comprendre le cycle de vie d’un sprint pendant un processus SCRUM. Le processus, illustré dans la figure 2.3, est décrit ci-dessous :

1. le Product owner crée le "product backlog" en identifiant et priorisant les user stories.
2. Pendant la planification du sprint, l’équipe choisit un ensemble de " user stories " les plus prioritaires à partir du "product backlog" pour construire le sprint Backlog.

3. L'équipe implémente les "users stories " pendant une période qui dure de 2 à 4 semaines.
4. Durant le sprint, l'équipe se réunit chaque jour, "Daily Scrum", pour synchroniser les tâches.
5. A la fin du sprint, le travail doit être achevé pour faire une démonstration au client.
6. Le sprint est clôturé par un "sprint review" pour discuter les prochaines étapes du projet et par un "sprint rétrospective" pour parler des manières à appliquer pour rendre l'équipe plus productive.

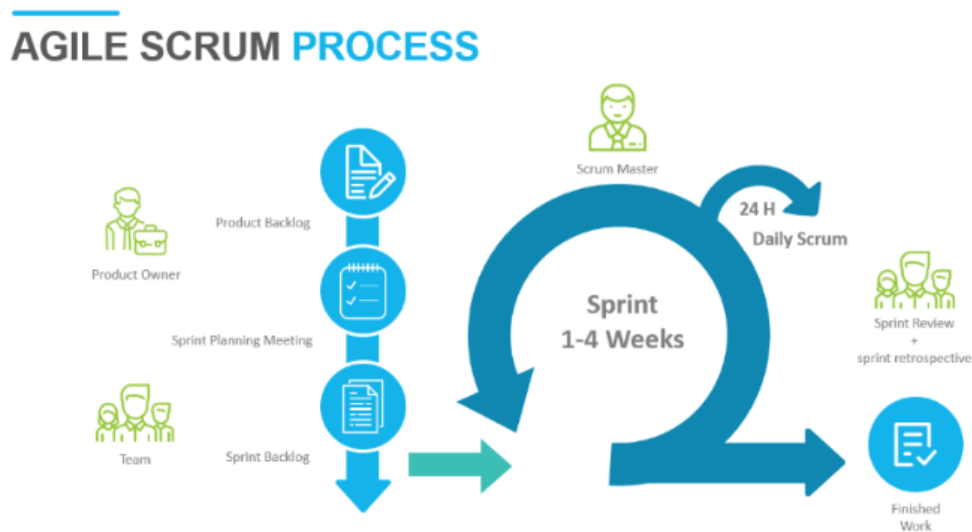


Figure 2.3: Description de processus SCRUM

2.2.3 Le Product Backlog du produit

Le backlog est élaboré avant le lancement des sprints, dans la phase de préparation. Il est utilisé pour la planification de la release, puis à chaque sprint, lors de la réunion de planification du sprint pour décider du sous-ensemble d'éléments. Les éléments y sont classés par priorité ce qui permet de définir l'ordre de réalisation. La table 2.2 représente notre backlog de produit :

Backlog du Produit	priorité	Estimation
Migration de l'environnement & automatisation des tests de non régression	1	Haute
Gestion des utilisateurs & des rôles	2	Haute
Intégration de module boutique & affectation utilisateurs aux boutiques	3	Moyenne
gestion catégories	4	Moyenne
Gestion des PEF	5	Moyenne
Simulation SRCD et consultation des logs API	6	faible
Elaboration des logs de l'application	7	faible

Tableau 2.2: Backlog de produit

2.2.4 Planification des sprints

Notre travail est divisé sur deux releases, le premier release contient la partie de migration de l'environnement, l'élaboration des tests de non régression et la gestion des utilisateurs et leurs rôles. Le deuxième release contient la consultation des boutiques, l'affectation des utilisateurs, la gestion des pefs, la simulation SRCD et la consultation des logs.

Le tableau 2.3 montre la répartition des sprints relative à notre système.

Répartition des sprints		
Les Releases	Les sprints	Les tâches
Release 1	Sprint 0	Migration PHP
		Migration SGBD
		Migration Framework
		Développement des tests de non régressions
		Automatisation des tests de non régressions
	Sprint 1	Gestion des utilisateurs
		Gestion des rôles
Release 2	Sprint 2	Intégration de module Boutique
		Affectation des utilisateurs aux boutiques
		Simulation SRCD
	Sprint 3	Gestion des catégories
		Gestion des pefs
		Visibilité pef boutique

Release 2	Sprint 3	Accès aux pef via la fiche client
		Consultation des logs API
		Elaboration des logs d'application Panoramix

Tableau 2.3: Planification des sprints

2.3 Environnement de travail

Dans cette partie, nous présenterons l'environnement de travail lors de la conception et la réalisation des tâches du projet.

2.3.1 Environnement matériel

Lors de la réalisation de notre application, nous avons utilisé un seul ordinateur dont les configurations sont les suivants :

- **PC** : DELL LATITUDE E5540
- **Processeur** : Intel i5-4210U
- **RAM** : 12 Go
- **Système d'exploitation** : Windows 10 Entreprise

2.3.2 Environnement de développement

- **Eclipse** : est un environnement de développement intégré (IDE) utilisé dans la programmation informatique. Il contient un espace de travail de base et un système de plugin extensible pour personnaliser l'environnement. Eclipse est principalement écrit en Java et son utilisation principale est le développement d'applications Java, mais il peut également être utilisé pour développer des applications dans d'autres langages de programmation via des plug-ins, notamment C, C++, C#, JavaScript, PHP, Python et autres.

**Figure 2.4:** Logo Eclipse IDE

- **Visual studio Code** : est un éditeur de code redéfini et optimisé pour la création et le débogage d'applications web et cloud modernes. Visual Studio Code est gratuit et disponible sur votre plateforme favorite Linux, macOS et Windows.



Figure 2.5: Logo Visual studio Code

- **Laragon** : est un environnement de développement universel, portable, isolé, rapide et puissant pour PHP, Node.js, Python, Java, Go, Ruby. Il est rapide, léger, facile à utiliser et facile à étendre. Laragon est idéal pour construire et gérer des applications web modernes. Il est axé sur la performance, conçu autour de la stabilité, de la simplicité, de la flexibilité et de la liberté.



Figure 2.6: Logo Laragon

- **Apache** : Un serveur HTTP créé et maintenu sur la fondation d'Apache. Jusqu'en avril 2019, c'était le serveur HTTP le plus populaire sur le World Wide Web. Il est distribué sous les termes de la licence Apache.



Figure 2.7: Logo Apache

- **MariaDB** : est un système de gestion de base de données publié sous licence GPL. Il s'agit d'une branche communautaire de MySQL : la gouvernance du projet est assurée par la Fondation MariaDB, et sa maintenance est assurée par Monty Program AB, le créateur du projet.



Figure 2.8: Logo mariaDB

- **HeidiSQL** : est un outil de gestion de base de données avec éditeur SQL et générateur de requêtes. Il a été développé et optimisé pour une utilisation avec le SGBD relationnel MySQL/MariaDB.



Figure 2.9: Logo HeidiSQL

2.3.3 Environnement logiciel

Dans cette partie, nous allons présenter les différents logiciels, framework et technologies web utilisés pour la réalisation des applications.

- **SeleniumLibrary based on Python** : SeleniumLibrary est une bibliothèque de test web pour Robot Framework qui utilise l'outil Selenium en interne. SeleniumLibrary fonctionne avec Selenium 3 et 4. Elle supporte Python 2.7 ainsi que Python 3.6 ou plus récent. En plus de l'interpréteur Python normal, elle fonctionne également avec PyPy et Jython.



Figure 2.10: Logo SeleniumLibrary

- **Robot Framework** : Robot Framework est un framework d'automatisation basé sur Python et extensible par mot-clé pour les tests d'acceptation, le développement piloté par les tests d'acceptation, le développement piloté par le comportement et l'automatisation des processus robotiques. Il peut être utilisé dans des environnements distribués et hétérogènes, où l'automatisation nécessite l'utilisation de différentes technologies et interfaces. Le cadre est entouré d'un riche écosystème constitué de diverses bibliothèques et outils génériques qui sont développés dans le cadre de projets distincts.



Robot Framework

Figure 2.11: Logo Robot Framework

- **Jenkins** : Jenkins est un outil d'intégration continue open source. Après les différends entre son auteur *Kohsuke Kawaguchi* et Oracle, Jenkins devient une branche des outils Hudson. Jenkins est écrit en Java et il peut fonctionner dans un conteneur de servlet (comme Apache Tomcat), ou il peut être utiliser un serveur Web intégré en mode autonome.



Figure 2.12: Logo Jenkins

- **GeckDriver** : Proxy pour l'utilisation de clients compatibles avec le W3C WebDriver pour interagir avec les navigateurs basés sur Gecko. Ce programme fournit l'API HTTP décrite par le protocole WebDriver pour communiquer avec les navigateurs Gecko, tels que Firefox. Il traduit les appels dans le protocole Marionnette à distance en agissant comme un proxy entre les extrémités locale et distante.
- **Xvfb plugin** : Xvfb ou X virtual framebuffer est un serveur d'affichage mettant en œuvre le protocole de serveur d'affichage X11. Contrairement aux autres serveurs d'affichage, Xvfb effectue toutes les opérations graphiques en mémoire sans afficher de sortie d'écran. C'est très utile si votre compilation nécessite un accès X11, par exemple pour effectuer des tests qui nécessitent une interface graphique.
- **Email Extension plugin** : Ce plugin vous permet de configurer tous les aspects des notifications par courrier électronique. Vous pouvez personnaliser le moment où un courriel est envoyé, qui doit le recevoir et ce que le courriel dit.
- **Javascript** : est un langage de script utilisé pour créer et contrôler le contenu dynamique d'un site web, c'est-à-dire tout ce qui bouge, rafraîchit ou change de quelque manière que ce soit sur votre écran sans vous obligez à recharger manuellement une page web. Des choses comme : des graphiques animés, des diaporamas de photos, des suggestions de texte à remplir automatiquement, des formulaires interactifs.



Figure 2.13: Logo Javascript

- **jQuery** : est une bibliothèque JavaScript multiplateforme gratuite, qui a été créée pour faciliter l'écriture de scripts côté client avec du code HTML sur les pages Web.



Figure 2.14: Logo jQuery

- **Axios** : Axios est un client HTTP populaire, basé sur la promesse, qui comporte une API facile à utiliser et peut être utilisé à la fois dans le navigateur et dans Node.js. Effectuer des requêtes HTTP pour récupérer ou enregistrer des données est l'une des tâches les plus courantes qu'une application JavaScript côté client devra effectuer.



Figure 2.15: Logo Axios

HTML5 : est un langage de balisage utilisé pour structurer et présenter le contenu sur le World Wide Web. Elle est la dernière révision majeure d'HTML. HTML5 spécifie deux syntaxes pour le modèle abstrait défini par le DOM : HTML5 et XHTML5



Figure 2.16: Logo HTML5

- **PHP** : préprocesseur hypertexte, connu sous son acronyme PHP, est un "langage de programmation" gratuit qui est principalement utilisé pour générer des pages Web dynamiques via un serveur HTTP, mais il peut également fonctionner comme n'importe quel langage interprété localement. PHP est un langage impératif orienté.



Figure 2.17: Logo PHP

- **Orange Framework & Tools** : est un squelette d'application PHP/MySQL prêt-à-l'emploi. Ce framework PHP est sécurisé et équipé d'outils de sécurité pour accompagner les développements, charté et respectueux de l'identité du Groupe #Boosted, performant et compatible avec toutes les offres d'hébergement #12factor, souple et adapté à la construction d'interfaces REST et HTML, équipé d'outils pour accélérer les développements et gérer les spécificités d'Orange et adapté aux débutants comme aux experts.

L'OFT est surtout un framework basé sur Symfony et Zend PHP, produit de réflexions sur ce qui est jugé le plus pertinent pour les projets du Groupe Orange puisqu'il est taillé selon les besoins de la communauté PHP d'Orange et qui suit les standards et réutilise les composants reconnus.



Figure 2.18: Logo Orange framework & tools

- **Boosted** : est représenté comme la bibliothèque HTML, CSS et JS d'Orange basée sur Bootstrap 4.5.0, la boîte à outils open source frontale la plus populaire au monde.

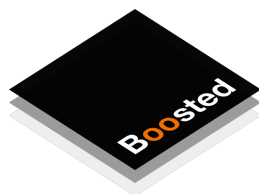


Figure 2.19: Logo Boosted

2.4 L'architecture de la solution

Dans cette partie, nous présenterons l'architecture logicielle et matérielle utilisée lors du développement de notre application.

2.4.1 L'architecture logique

2.4.1.1 L'architecture logique des tests de non régression

Nous présentons dans ce qui est suit l'enchaînement des tests de non régression de notre application. La figure 2.20 illustre la procédure des tests.

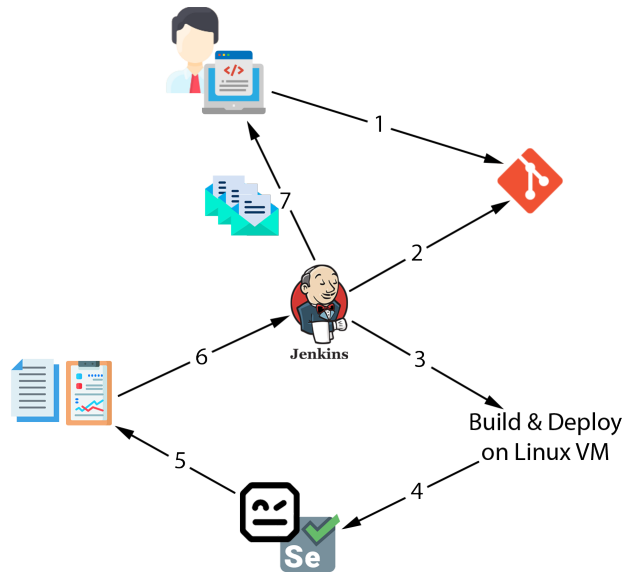


Figure 2.20: L'architecture logique des tests de non régression

1 - Le développeur de l'équipe Panoramix fait un « push » de code de robot Framework vers le repo git de « Orange Forge ».

2+3 - Jenkins va cloner ce repo dans une VM Linux et prépare l'environnement pour lancer les tests

4 - les 54 tests de non régression s'exécutent test par test et chaque test simule un scénario d'utilisation de notre portail.

5 - A la fin des tests, Robot Framework génère 3 fichiers dont le fichier report.html et log.html sont les plus importants :

- **Log.html** : Les fichiers journaux contiennent des détails sur les cas de tests exécutés au format HTML. Ils ont une structure hiérarchique indiquant la suite de tests, le scénario de test et les détails des mots clés. Les fichiers journaux sont nécessaires presque à chaque fois que les résultats des tests doivent être examinés en détail. Même si les fichiers journaux contiennent également des statistiques, les rapports sont plus utiles pour obtenir une vue d'ensemble de haut niveau.
- **Report.html** : Les fichiers de rapport contiennent un aperçu des résultats de l'exécution des tests au format HTML. Ils contiennent des statistiques basées sur les balises et les suites de tests exécutés, ainsi qu'une liste de tous les cas de tests exécutés. Lorsque les rapports et les journaux sont générés, le rapport comporte des liens vers le fichier journal pour faciliter la navigation vers des informations plus détaillées. Il est facile de voir l'état général de l'exécution des tests à partir du rapport, car sa couleur de fond est verte, si

tous les tests critiques réussissent, et rouge vif dans le cas contraire.

6+7 - Jenkins récupère les fichiers générés par robot Framework et les envoyer en email vers les personnes concernés.

2.4.1.2 L'architecture logique de l'application

Dans notre application que nous avons effectuée, nous envisageons d'utiliser une architecture 3-tiers qui sera réparti en trois couches comme suit :

- **Couche 1 (Site) :** Cette couche contient les interfaces côté utilisateurs qui interagissent souvent avec l'application Panoramix.
- **Couche 2 (Serveur) :** Cette couche représente la partie traitement qui contient toutes nos APIs, cette partie sera réalisée avec Apache qui est un serveur HTTP.
- **Couche 3 (Base de Données) :** Cette couche représente le côté base de données de notre site qui sera une base de données MySQL.

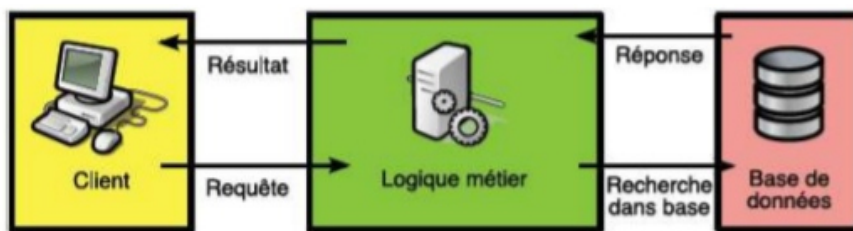


Figure 2.21: L'architectures 3 tiers du Web

2.4.2 L'architecture logicielle

2.4.2.1 L'architecture logicielle des tests de non régression

Les tests de non régression sont divisés en 3 grandes parties :

- Configuration globale : un dossier qui contient un fichier de configurations globales nécessaires dans la plupart des tests
- Les fonctions globales : ce dossier contient un fichier des fonctions générales utilisées dans tous les tests.
- TestSuite : un dossier contient tous les tests de non régression. Chaque test est divisé en 3 parties :
 - Fichier.robot : le script de test à exécuter

- Config : est un dossier contient les configurations nécessaires pour ce test
- Fonctions : est un dossier contient les fonctions nécessaires pour ce test

2.4.2.2 L'architecture logicielle de l'application

L'organisation du code source est assuré par le biais du modèle MVC. Mais son rôle est principalement de segmenter la logique du code en trois parties.

- Modèle(M) : Permet de récupérer des données brutes stockées dans notre base de données.
- Vue(V) : Cette partie inclut juste les fichiers html et CSS mais aussi quelques boucles et conditions PHP très simples qui se focalisent sur l'affichage.
- Contrôleur(C) : Ce dernier est assimilable à une passerelle entre les vues et le modèle. Il contient exclusivement que du PHP et gère notamment les droits d'accès de chaque utilisateurs.

La figure suivante schématise l'échange d'informations entre les éléments de l'architecture MVC

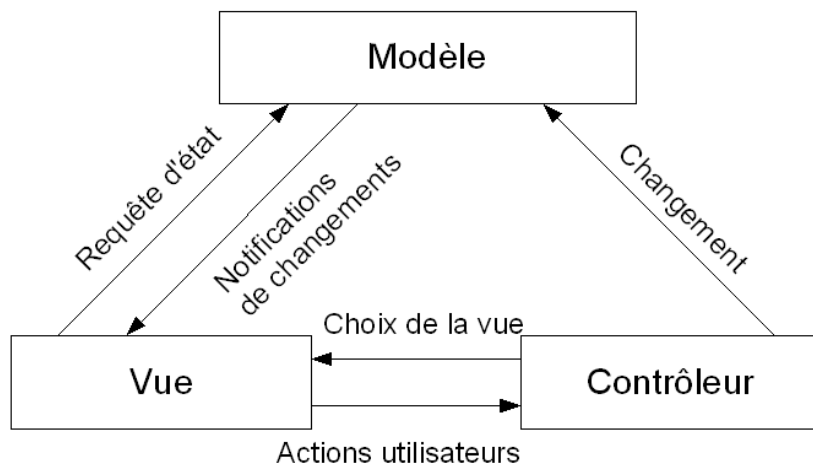


Figure 2.22: L'architecture logicielle de l'application

2.5 Conclusion

Dans ce chapitre, nous avons présenté les différents concepts nécessaires à la compréhension du projet. Nous avons également identifié les besoins fonctionnels, non fonctionnels ainsi que les acteurs. Par la suite nous avons réalisé une conception pour notre projet, présenté l'environnement de travail et l'architecture de la solution. Dans le chapitre suivant nous entamons le développement du premier release.

CONCEPTION ET CHOIX TECHNOLOGIQUES

Plan

1	Conception détaillée	24
2	Conception architecturale	24
3	Choix technologiques	24

Introduction

Introduction partielle, qui annonce le chapitre.

3.1 Conception détaillée

Les diagrammes + descriptions textuelles

3.1.1 Diagramme des classes

3.1.2 Diagramme des séquences

3.2 Conception architecturale

architecture physique...

3.3 Choix technologiques

technologies à utiliser pendant la réalisation...

Conclusion

Conclusion partielle ayant pour objectif de synthétiser le chapitre et d'annoncer le chapitre suivant.

RÉALISATION

Plan

1	Choix techniques	26
2	Travail réalisé	26
3	Planning réel du projet	26

Introduction

Introduction partielle, qui annonce le contenu.

4.1 Choix techniques

4.2 Travail réalisé

4.3 Planning réel du projet

Conclusion

Conclusion partielle ayant pour objectif de synthétiser le chapitre et d'annoncer le chapitre suivant.

Conclusion générale

Rappel du contexte et de la problématique.

Brève récapitulation du travail réalisé et de la solution proposée.

La taille de la conclusion doit être réduite, une page de texte tout au plus. Il est important de souligner que la conclusion ne comporte pas de nouveaux résultats ni de nouvelles interprétations.

Le plus souvent, la conclusion comporte :

- un résumé très rapide du corps du texte ;
- un rappel des objectifs du projet ;
- un bilan professionnel qui indique clairement les objectifs annoncés dans l'introduction et en particulier ceux qui n'ont pu être atteints. Il présente et synthétise les conclusions partielles ;
- un bilan personnel qui décrit les principales leçons que vous tirez de cette expérience sur le plan humain ;
- les limites et les perspectives du travail effectué.

Bibliographie

- [1] N. AUTEUR. (Jan. 2015). « titre de l'article. » [Accès le 19-Janvier-2016], Organisme, adresse : http://www.exemple-lien.org/article/?id_art=124.
- [2] M. SHELL. (Jan. 2007). « IEEEtran Homepage. » [Accès le 5-Février-2016], adresse : <http://www.michaelshell.org/tex/ieeetran/>.

Annexes

Annexe 1. Exemple d'annexe

Les chapitres doivent présenter l'essentiel du travail. Certaines informations-trop détaillées ou constituant un complément d'information pour toute personne qui désire mieux comprendre ou refaire une expérience décrite dans le document- peuvent être mises au niveau des annexes. Les annexes, **placées après la bibliographie**, doivent donc être numérotées avec des titres (Annexe1, Annexe2, etc.).

Le tableau annexe 1.1 présente un exemple d'un tableau dans l'annexe.

Tableau annexe 1.1 : Exemple tableau dans l'annexe

0	0
1	1
2	2
3	3
4	4

Annexe 2. Entreprise

La figure annexe 2.1 présente le logo entreprise.



Figure annexe 2.1 : Logo d'entreprise

ملخص

يوضع الملخص باللغة العربية هنا... يوضع الملخص باللغة العربية هنا... يوضع الملخص باللغة العربية هنا... يوضع الملخص باللغة العربية هنا... يوضع الملخص باللغة العربية هنا... الرجاء أن يكون في حدود العشر أسطر... يوضع الملخص باللغة العربية هنا... الرجاء أن يكون في حدود العشر أسطر... يوضع الملخص باللغة العربية هنا... الرجاء أن يكون في حدود العشر أسطر... يوضع الملخص باللغة العربية هنا... الرجاء أن يكون في حدود العشر أسطر... يوضع الملخص باللغة العربية هنا... الرجاء أن يكون في حدود العشر أسطر... يمكنك أن تكتب كلمات بحروف لاتينية في وسط الملخص مثال Exemple ici يوضع الملخص باللغة العربية هنا...

كلمات مفاتيح : الرجاء عدم تحاوز الخمس كلمات

Résumé

Mettez le résumé en français ici... Mettez le résumé en français ici... Mettez le résumé en français
ici... Mettez le résumé en français ici... Merci de ne pas dépasser les dix lignes. Mettez le résumé
en français ici, merci de ne pas dépasser les dix lignes. Mettez le résumé en français ici, merci de
ne pas dépasser les dix lignes. Mettez le résumé en français ici, merci de ne pas dépasser les dix
lignes. Mettez le résumé en français ici, merci de ne pas dépasser les dix lignes. Mettez le résumé
en français ici, merci de ne pas dépasser les dix lignes. Mettez le résumé en français ici, merci de
ne pas dépasser les dix lignes. Mettez le résumé en français ici, merci de ne pas dépasser les dix
lignes. Mettez le résumé en français ici, merci de ne pas dépasser les dix lignes.

Mots clés : Merci de ne pas dépasser les cinq mots

Abstract

Put the English abstract here, put the English abstract here, put the English abstract here, put
the English abstract here, put the English abstract here, put the English abstract here... Please
don't exceed ten lines, Please don't exceed ten lines, Please don't exceed ten lines, Please don't
exceed ten lines. Put the English abstract here, please don't exceed ten lines. Put the English
abstract here, please don't exceed ten lines. Put the English abstract here, please don't exceed
ten lines. Put the English abstract here, please don't exceed ten lines. Put the English abstract
here, please don't exceed ten lines. Put the English abstract here, please don't exceed ten lines.
Put the English abstract here, please don't exceed ten lines. Put the English abstract here,
please don't exceed ten lines.

Keywords : Please don't use more than five keywords