



Université Cadi Ayyad
École Nationale des Sciences Appliquées de Marrakech

Rapport DE PROJET DE FIN D'ÉTUDES

Réalisation d'une API emailing pour la gestion des campagnes publicitaires (E-News) des établissements de formation (d'expertises comptables)

Présenté Par :

Chanaa Abdessamad

Réalisé à :

TamTam Internationale.

Encadré Par :

M.Nejeoui Abderrazzak(ENSA)
M.Elhachmi Mohamed (TamTam)

SOUTENU LE 13/07/2017 DEVANT LE JURY :

Membre du jury: Mr. Mustapha Ameur.

Membre du jury: Mme. Sara Bouzid.

Encadrant (ENSA): Mr. Nejeoui Abderrazzak.

Encadrant (Entreprise): Mr. Elhachmi Mohamed.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿ يَرْفَعُ اللَّهُ الَّذِينَ ءَامَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا
الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَيْرٌ ﴾ ١١

سورة اطهادلة

Dédicace

Au Dieu tout puissant mon créateur.

A ma mère,

Ma raison d'être, ma raison de vivre, la lanterne qui éclaire mon chemin et m'illumine de douceur et d'amour.

A mon père,

En signe de reconnaissance et de gratitude pour tous les soutiens et les sacrifices dont il a fait preuve à mon égard.

A mes frères,

J'espère avoir atteint le seuil vos espérances. Je vous remercie pour le soutien moral et l'encouragement que vous m'avez accordé.

A mes professeurs,

A mes professeurs de l'ENSA qui doivent voir dans ce travail la fierté d'un savoir bien acquis.

A mes ami(e)s,

Qui ont su m'apporter aide et soutien aux moments propices, je vous dédie ce travail en étant reconnaissant et en vous remerciant chaleureusement.

Remerciements

Je tiens à adresser mes vifs remerciements à l'entreprise **TamTam International**, plus particulièrement au directeur **M. Degrève Emmanuel** pour m'avoir accepté au sein de cette entreprise pour effectuer ce stage.

Aussi je tiens à présenter mes reconnaissances et mes remerciements à mon professeur encadrant **M. Nejeoui Abderrazzak** pour le temps consacré aux réunions qui ont rythmées les différentes étapes de mon projet de fin d'étude.

Les discussions que nous avons partagées ont permis d'orienter mon travail d'une manière pertinente. Je le remercie aussi pour sa disponibilité à encadrer ce travail à travers ses critiques constructives et ses propositions d'amélioration.

Je formule ma gratitude à **M. Elhachmi Mohamed** l'encadrant à TamTam International pour la confiance qu'il m'a accordé en m'attribuant des responsabilités de gestion et de supervision.

Il a toujours montré l'intérêt pour mes travaux, tout en gardant l'œil critique et avisé pour me donner des conseils appropriés afin d'améliorer mes résultats.

Je tiens à remercier aussi tous les membres du jury : **M. Ameur Mustapha et Mme. Bouzid Sara** qui m'ont fait l'honneur d'accepter de juger mon travail.

Mes remerciements vont aussi pour **Mlle. Fatima Ezzahra Lridrissi** et tous l'équipe de TamTam pour leur disponibilités inconditionnelles.

Comme je tiens à remercier profondément l'ensemble du corps professoral qui a contribué à ma formation au sein de l'ENSA de Marrakech, veillé au bon déroulement des études dans des conditions favorables permettant une formation d'ingénieur de haut niveau.

Enfin, que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de mes remerciements et mes considérations.

Résumé

Le présent document est le fruit du travail réalisé dans le cadre du projet de fin d'études pour l'obtention du diplôme d'ingénieur, en génie informatique à l'école national des sciences appliquées (ENSA) de Marrakech, Université Cadi Ayyad. Ce projet a été effectué au sein de Tamtam international. Il s'agit de la conception et réalisation d'une interface de programmation applicative (API).

Le projet nommé «Emailing» est une solution qui est une sorte d'API indépendante de tous autre système, qui va essentiellement servir à la gestion des campagnes publicitaires (E-News) des établissements de formation (d'expertise comptable) - fiduciaires - et les d'autres qui opèrent dans le domaine (Finance/économique).

L'objectif de mon projet de fin d'études consiste à améliorer l'interaction entre les clients (organisateurs des formations des experts comptables) et notre plateforme toute en offrant une API proche de l'utilisateur et toujours disponible.

Ce rapport est organisé en six chapitres, le premier chapitre donne une brève présentation sur l'organisme d'accueil suivi du contexte général du projet, succédé par un chapitre qui illustre l'étude détaillée de l'algorithme utilisé. Les deux chapitres suivants présentent l'analyse fonctionnelle et technique du projet, le cinquième chapitre est consacré à la conception de notre système, finalement le dernier chapitre montre les étapes que nous avons suivies pour la réalisation du projet.

Abstract

This paper presents the results of the work carried out within the final project for the graduation of an engineer, in computer science engineering at the National School of Applied Sciences (ENSA) in Marrakech, Cadi Ayyad University. This project was carried out within «Tamtam» International. This is the conception and implementation of an Application Programming Interface (API).

The project called "Emailing" is a solution that is a kind of an API independent of any other system, which will essentially serve for the management of advertising campaigns (E-News) of training institutions (fiduciary) And others who operate in the field (Finance / Economics).

The objective of my graduation project is to improve the interaction between the customers (organizers of the trainings of the accounting experts) and our platform while offering a user-friendly and always available API.

This report is organized into six chapters. The first chapter gives a short presentation of the host organization followed by the general context of the project. After that a chapter that illustrates the detailed study of the algorithm used. The next two chapters present the functional and technical analysis of the project, the fifth chapter devoted to the conception of our system, and finally the last chapter shows the steps we have taken to carry out the project.

ملخص

هذه الوثيقة ، هي نتيجة العمل في إطار مشروع التخرج للحصول على شهادة مهندس دولة في الهندسة المعلوماتية في المدرسة الوطنية للعلوم التطبيقية (ENSA) بمراكش، جامعة القاضي عياض، و تم تنفيذ المشروع في شركة (TAMTAM) الدولية . وهو تصميم وتنفيذ لبرمجة واجهة التطبيقات.

المشروع هو نوع من برمجة واجهة التطبيقات مستقلة عن كل نظام آخر ، والتي سوف تكون أساسا يستخدم لإدارة الحملات الإعلانية والأخبار وادارة مؤسسات التدريب بالنسبة للمحاسين وغيرهم من الذين يعملون في الميادين المالية والاقتصادية .

يهدف هذا المشروع إلى تحسين التفاعل بين العملاء (منظمي التدريب للمحاسين) ، والتطبيق الكامل وذلك من خلال توفير تطبيق قريب للمستخدم ومتاح دائمًا .

و يتم تنظيم هذا التقرير إلى ستة فصول، ويعطي الفصل الأول عرضاً موجزاً عن الشركة المصنفة متبعاً بالسوق العام للمشروع. بعد ذلك يأتي الفصل الذي يوضح دراسة مفصلة حول الخوارزمية المستخدمة. الفصلين التاليين يظهران التحليل الوظيفي والتقيي، الفصل الخامس من المشروع يكرس لتصميم نظامنا وأخيراً بين الفصل الأخير الخطوات المتبعة لإنجاز المشروع.

Liste des acronymes

| | | |
|---|-------|---|
| | ADMIN | Administrateur |
| A | API | Application Programming Interface |
| | AWS | Amazon Web Service |
| B | BSD | Berkeley Software Distribution |
| C | CSS | Cascading Style Sheets |
| | CSV | Comma Separated Values |
| D | DAO | Data Access Object |
| G | GUI | Graphical User Interface |
| H | HTML | HyperText Markup Language |
| | HTTP | Hypertext Transfer Protocol |
| I | ID | Identifiant |
| | IEC | Institut des Experts-comptables et des Conseils fiscaux |
| | IHM | Interactions Homme Machines |
| J | JS | Javascript |
| | JSON | JavaScript Object Notation |
| M | MVC | Modèle Vue Contrôleur |
| O | ORM | Object Relational Mapping |
| | OS | Operating System |
| P | PDF | Portable Document Format |
| R | REST | Representational state transfer |
| | RUP | Rational Unified Process |
| S | SASS | Syntactically Awesome Stylesheets |
| | SES | Simple Email Service |
| | SMTP | Simple Mail Transfer Protocol |
| | SNS | Simple Notification Service |
| | SQL | Structured Query Language |
| | SSL | Secure Socket Layer |
| T | TDD | Test Driven Development |
| | TTP | TamTam Pro |
| U | UML | Unified Modeling Language |
| | URL | Uniform Resource Locator |
| X | XML | Extensible Markup Language |
| | XP | eXtreme Programming |

Tables des matières

| | |
|--|-----------|
| Dédicace..... | 3 |
| Remerciements..... | 4 |
| Résumé..... | 5 |
| Abstract..... | 6 |
| ملخص..... | 7 |
| Liste des acronymes..... | 8 |
| Tables des matières..... | 9 |
| Liste des figures..... | 12 |
| Liste des Tableaux..... | 15 |
| Introduction générale | 16 |
| Chapitre 1 : Contexte générale du projet..... | 17 |
| Introduction..... | 18 |
| 1.1. Présentation de l'entreprise TamTam..... | 19 |
| 1.1.1 Fiche signalétique TamTam..... | 20 |
| 1.2. Présentation du projet..... | 21 |
| 1.3. Méthodologie de travail..... | 22 |
| 1.3.1 Étude préliminaire..... | 22 |
| 1.3.2 Choix de méthodologie de travail..... | 23 |
| 1.3.3 Répartitions des rôles dans Scrum..... | 23 |
| 1.3.4 développement piloté par les tests..... | 24 |
| 1.4. Planning..... | 25 |
| 1.4.1 diagramme de gantt..... | 25 |
| 1.4.1 description de diagramme de gantt..... | 26 |
| Conclusion..... | 27 |
| Chapitre 2 : État d'art..... | 28 |
| Introduction..... | 28 |
| 2.1. Présentation générale..... | 30 |
| 2.1.1. Introduction..... | 30 |
| 2.1.2. Contexte et problématique..... | 30 |
| 2.2. Les échecs transitoires..... | 31 |

| | |
|--|-----------|
| 2.2.1. Introduction..... | 31 |
| 2.2.2. Gestion des erreurs transitoires..... | 31 |
| 2.2.3. Problématique..... | 32 |
| 2.3. Les algorithmes du backoff..... | 33 |
| 2.3.1. Questions et considérations..... | 33 |
| 2.3.2. Quand utiliser ce pattern ?..... | 34 |
| 2.3.3. les différentes phase du backoff..... | 34 |
| 2.3.4 . Fonctions retry backoff..... | 36 |
| 2.4. Exponentielle backoff..... | 37 |
| 2.4.1. Introduction..... | 37 |
| 2.4.2. Formule mathématique..... | 37 |
| Conclusion..... | 38 |
| Chapitre 3 : Étude Fonctionnelle..... | 39 |
| Introduction..... | 40 |
| 3.1. Spécification des Exigences fonctionnelles..... | 41 |
| 3.1.1. Domaine d'activité..... | 41 |
| 3.1.2. Client..... | 41 |
| 3.2. Capture des besoins fonctionnels..... | 42 |
| 3.2.1. fonctionnalité souhaité..... | 42 |
| 3.2.2. Définition des acteurs..... | 42 |
| 3.2.3. Identification des exigences fonctionnelles..... | 43 |
| 3.3. Analyse des spécifications fonctionnelles..... | 44 |
| 3.3.1. Diagrammes des cas d'utilisation..... | 44 |
| 3.3.2. Diagrammes de séquence..... | 52 |
| 3.4. Exigences Non Fonctionnelles..... | 57 |
| Conclusion..... | 58 |
| Chapitre 4 : Étude Technique..... | 59 |
| Introduction..... | 60 |
| 4.1. Architecture physique | 61 |
| 4.2. Architecture logique | 63 |
| 4.2.1. Le langage de programmation..... | 63 |
| 4.2.2. Outils de développement..... | 63 |
| 4.2.3. Frameworks et technologies de développement | 65 |
| 4.2.4. Architecture de la solution..... | 69 |
| 4.2.5. Design Pattern..... | 70 |
| 4.3. Amazon Web Services..... | 73 |
| 4.3.1. Amazon web service SES..... | 73 |

| | |
|--|------------|
| 4.3.2. Amazon SNS Pub/Sub Messaging..... | 74 |
| 4.3.3. Amazon Relational Database Service (RDS)..... | 74 |
| 4.4. Pixel espion de tracking..... | 75 |
| Conclusion..... | 76 |
| Chapitre 5: Conception..... | 77 |
| Introduction..... | 78 |
| 5.1. Prototypage..... | 79 |
| 5.1.1. Les maquettes réalisées..... | 79 |
| 5.2. Conception du système..... | 83 |
| 5.2.1. Diagramme de Classe Métier..... | 83 |
| 5.2.2. Diagrammes état de transition..... | 85 |
| 5.2.3. Diagrammes d'Activité..... | 87 |
| 5.3. Conception et documentation de l'API REST..... | 89 |
| Conclusion..... | 91 |
| Chapitre 6: Réalisation..... | 92 |
| Introduction..... | 93 |
| 6.1. Organisation du Travail..... | 94 |
| 6.2. Environnement du travail..... | 95 |
| 6.3. Structure du projet..... | 96 |
| 6.3.1. coté serveurs..... | 98 |
| 6.3.2. coté client..... | 99 |
| 6.4. Les interfaces graphiques..... | 98 |
| 6.4.1. les interfaces de configuration d'Amazon SNS..... | 98 |
| 6.4.2. les interfaces de réalisation..... | 99 |
| Conclusion..... | 108 |
| Chapitre 7: Conclusion Générale..... | 109 |
| Conclusion Générale..... | 110 |
| WeboGraphie..... | 111 |
| Annexe I. Choix de Docker..... | 112 |
| Annexe II. Choix de PHP 7.0..... | 114 |
| Annexe III. Choix de React..... | 116 |
| Annexe IV. Envoi via Amazon..... | 118 |

Liste des figures

| | |
|---|----|
| Figure1-1 : logo de Tamtam..... | 19 |
| Figure 1-2 : produits de tamtam..... | 19 |
| Figure 1-3 : démarche de développement Scrum..... | 24 |
| Figure 1-4 : démarche de TDD | 24 |
| Figure 1-5 :Taches à réaliser dans le cadre du projet..... | 25 |
| Figure 1-6 : l'ordonnancement des taches..... | 25 |
| Figure 2-1 : processus d'échec transitoire | 31 |
| Figure 2-2 : diagramme d'activité du simple retry..... | 32 |
| Figure 2-3 : diagramme d'activité du backoff retry..... | 33 |
| Figure 2-4 : les différentes phases d'un algorithme backoff..... | 35 |
| Figure 2-5 : délai constant entre les tentatives..... | 35 |
| Figure 2-6 : délai croissant entre les tentatives..... | 36 |
| Figure 2-7 : graphe de comparaison entre les fonctions backoff..... | 36 |
| Figure3-1 :Diagramme du contexte du projet..... | 43 |
| Figure 3-2 :Diagramme cas d'utilisation de gestion des groupes..... | 44 |
| Figure 3-3 :Lister les campagnes..... | 46 |
| Figure 3-4 :Créer une campagne..... | 48 |
| Figure 3-5 : diagramme cas d'utilisation pour suspension d'envoi de campagne..... | 51 |
| Figure 3-6 :diagramme cas d'utilisation pour la consultation des statistiques..... | 52 |
| Figure 3-7 :diagramme de séquence de l'envoi d'une campagne..... | 53 |
| Figure 3-8 : diagramme de séquence d'affectation d'un groupe à un contact..... | 54 |
| Figure 3-9 : diagramme de séquence de consultation des statistiques..... | 54 |
| Figure 3-10 : diagramme de séquence de gestion des tentatives d'envoi..... | 55 |
| Figure 3-11 : diagramme de séquence de gestion des tentatives d'envoi..... | 56 |
| Figure 3-12 : diagramme de séquence de gestion des prospects..... | 56 |
| Figure 4-1 : Architecture physique de l'application..... | 61 |
| Figure 4-2 : Logo Docker..... | 61 |
| Figure 4-3 : Machine virtuelle Vs conteneur Docker..... | 62 |
| Figure 4-5 : Les conteneurs Docker du projet..... | 62 |
| Figure 4-6 : Logo php7..... | 63 |
| Figure 4-7 : Logo Jira..... | 63 |
| Figure 4-8 : Logo Github..... | 63 |
| Figure 4-9 : Logo Wercker..... | 64 |
| Figure 4-10 : plateforme de déploiement continu pour builder et de déployer les applications..... | 64 |
| Figure 4-11 : workflow de déploiement continu. | 64 |
| Figure 4-12: Logo Symfony..... | 65 |
| Figure 4-13 :Architecture de base de Symfony. | 65 |
| Figure 4-14: Logo phpununit..... | 65 |
| Figure 4-15 : Logo React..... | 66 |
| Figure 4-16 : Logo Sass..... | 66 |
| Figure 4-17 : Logo DraftJs..... | 66 |
| Figure 4-18 : Logo Foundation..... | 67 |
| Figure 4-19: Logo phantomJs..... | 67 |
| Figure 4-20 : Logo SwiftMailer..... | 68 |

| | |
|---|-----|
| Figure 4-21 : Logo RestApi..... | 68 |
| Figure 4-22 : architecture de projet avec le desein «Api Rest»..... | 68 |
| Figure 4-23 : Logo doctrine..... | 69 |
| Figure 4-24 :Architecture et outils de la solution..... | 69 |
| Figure 4-26 : diagramme MVC..... | 70 |
| Figure 4-27: Logo Redux..... | 71 |
| Figure 4-28 : workflow of redux..... | 71 |
| Figure 4-29 : architecture de redux liant le coté serveur au coté client..... | 71 |
| Figure 4-30 : diagramme de classe de deseign pattern adaptateur..... | 72 |
| Figure 4-31 : logo amazon web service..... | 73 |
| Figure 4-32 : logo amazon SES..... | 73 |
| Figure 4-33 : Envoi de mail par SES..... | 73 |
| Figure 4-34 : logo amazon SNS..... | 74 |
| Figure 4-35 : Diagramme SNS..... | 74 |
| Figure 4-36 : logo amazon RDS..... | 74 |
| Figure 4-37 :schéma explicative de procédure de pixel de track..... | 75 |
| Figure 5-1 :création d'un nouveau prospect..... | 79 |
| Figure 5-2 : édition d'un prospect existant..... | 80 |
| Figure 5-3 : Affecter contact à un groupe..... | 80 |
| Figure 5-4 : Lister les campagnes..... | 81 |
| Figure 5-5 : création d'un campagne markdown..... | 81 |
| Figure 5-6 : création d'une campagne web..... | 82 |
| Figure 5-7 : statistique sur un emailing..... | 82 |
| Figure 5-8 : diagramme de classe métier..... | 83 |
| Figure 5-9 : diagramme état transition campagne..... | 85 |
| Figure 5-10: diagramme état transition de campagneRecipient..... | 86 |
| Figure 5-11 : diagramme d'activité de gestion des tentatives..... | 87 |
| Figure 5-12 : diagramme d'activité d'envoi de campagne..... | 88 |
| Figure 5-13 : création des urls..... | 89 |
| Figure 5-14: appel en utilisant restApi..... | 89 |
| Figure 5-15 : reponse json de l'api..... | 90 |
| Figure 6-1 : workflow on Jira..... | 94 |
| Figure 6-2 : structure de base du backend..... | 96 |
| Figure 6-3 : structure des dossier dans Phpstorm..... | 96 |
| Figure 6-4 : workflow de l'architecture en front en liason avec l'API..... | 97 |
| Figure 6-5 : structure des dossiers dans Webstorm..... | 97 |
| Figure 6-6 : spécification du topic..... | 98 |
| Figure 6-7 : spécification de l'end-pointe..... | 98 |
| Figure 6-8 : validation de l'end-pointe..... | 99 |
| Figure 6-9 : l'accueil de l'application..... | 99 |
| Figure 6-10:l'authentification de l'application..... | 100 |
| Figure 6-11 : l'accueil après authentification de l'application..... | 100 |
| Figure 6-12 : lister les campagnes..... | 101 |
| Figure 6-13 : lister les campagnes selon un filter..... | 101 |
| Figure 6-14 : campagne web visualisé..... | 102 |
| Figure 6-15 : étape 1 de l'envoi..... | 102 |

| | |
|--|-----|
| Figure 6-16 : étape 2 de l'envoi..... | 103 |
| Figure 6-17 : étape 3 de l'envoi..... | 103 |
| Figure 6-18 : étape 4 de l'envoi..... | 104 |
| Figure 6-19 : validation de l'envoi..... | 104 |
| Figure 6-20 : La campagne envoyées..... | 105 |
| Figure 6-21 : état de la campagne..... | 105 |
| Figure 6-22 : campagne reçue..... | 105 |
| Figure 6-23 :statistique d'une campagne..... | 106 |
| Figure 6-24 : page d'erreur..... | 106 |
| Figure 6-25 : création de prospect..... | 107 |
| Figure annexe 1-1 : structure docker..... | 113 |
| Figure annexe 2-1: comparaison de performance entre les langages de programmation..... | 115 |
| Figure annexe 3-1 : graphe de comparaison de temps d'exécution des frameworks js..... | 117 |
| Figure annexe 4-1 : l'envoi de message via amazon SES..... | 118 |
| Figure annexe 4-2 : envoi réussie..... | 120 |
| Figure annexe 4-3 : hard bounce..... | 120 |
| Figure annexe 4-5 : soft bounce..... | 121 |
| Figure annexe 4-5 : plainte..... | 121 |

Liste des Tableaux

| | |
|--|-----|
| Tableau 1-1 : Fiche signalétique TamTam..... | 20 |
| Tableau 1-2 : comparaison entre les méthodologies de travail | 22 |
| Tableau 1-3 : Description de diagramme de gant..... | 26 |
| Tableau 3-1 :Les différents acteurs..... | 43 |
| Tableau 3-2 :Les fonctionnalités du système..... | 43 |
| Tableau 3-3 : Création d'un groupe contact..... | 44 |
| Tableau 3-4 :Lister les campagnes..... | 45 |
| Tableau 3-5 :création prospect..... | 46 |
| Tableau 3-6 :suppression prospect..... | 46 |
| Tableau 3-7 :Lister les campagnes..... | 47 |
| Tableau 3-8 :Chercher une campagne..... | 47 |
| Tableau 3-9 : supprimer campagne..... | 47 |
| Tableau 3-10 :éditer une campagne..... | 48 |
| Tableau 3-11 : Enregistrer une campagne..... | 49 |
| Tableau 3-12 : tester une campagne..... | 49 |
| Tableau 3-13 :Envoyer une campagne..... | 49 |
| Tableau 3-14 :Éditer contenus d' une campagne..... | 51 |
| Tableau 3-15 : Suspendre l'envoi d'une campagne..... | 51 |
| Tableau 3-16 : consultation des statistiques..... | 52 |
| Tableau 4-1 : Les composants serveurs..... | 61 |
| Tableau 5-1 : description des packages de diagramme classe..... | 84 |
| Tableau 5-2 : description des classes de diagramme classe..... | 85 |
| Tableau 5-3: description de l'état de transition d'une campagne..... | 86 |
| Tableau 5-4 : description de l'état de transition d'une campagneRecipient..... | 87 |
| Tableau 5-5 : tableau de spécification pour API REST..... | 90 |
| Tableau 6-1 : outil de développement utilisée..... | 95 |
| Tableau annexe 2-1 : comparaison entre les langage de programmatio..... | 114 |

Introduction générale

Les newsletters sont généralement utilisées par les entreprises. C'est le moyen d'informer les adhérents par des nouveautés parvenues sur le site internet de l'entreprise ou via des promotions. À travers cette méthode, l'entreprise cherchant donc à fidéliser et rester en contact avec la clientèle, on cherche à stimuler la curiosité du client en effectuant l'action souhaitable.

Cet atout marketing est considéré une source d'expertise et d'information importante. Les newsletters portent un sujet, surtout une actualité d'entreprise, son domaine d'activité, ses offres de produits et services proposés. Les newsletters doivent être précises en répondant sur les attentes des clients abonnés afin d'éviter le désabonnement.

La newsletter est la dualité du support d'information et le relais qui guide le client vers le site à travers le lien proposé. C'est grâce au résumé présenté, le lecteur sera simulé et orienté à lire la totalité de l'article. L'utilisation de cet outil développe l'offre commerciale, augmente le nombre de visiteurs, augmenter les achats est donc le chiffre d'affaires d'entreprise.

Le présent rapport illustre, le travail effectué pour l'élaboration et la mise en œuvre de notre solution, il est organisé sous forme de 6 chapitres.

- Le premier consiste à décrire le contexte général du projet, en présentant l'organisme d'accueil, la problématique, l'objectif du projet, et la démarche suivie pour la conduite du projet utilisé ainsi que la méthodologie du travail.
- Le deuxième chapitre portera sur l'état d'art, il représente dans un premier axe le fruit d'un travail de recherche qui donne d'une manière simplifiée une problématique trouvée dans notre système, afin d'entamer un deuxième axe qui va nous mener à la description de notre système et de l'algorithme qu'on a choisi d'utiliser.
- Le troisième chapitre est consacré pour l'étude fonctionnelle, qui décrira dans un premier lieu les exigences fonctionnelles et non-fonctionnelles de ma solution, suivi ensuite de l'analyse fonctionnelle qui a été effectuée, vers la fin, il va lister les exigences non-fonctionnelles.
- Le quatrième est dédié à l'étude technique, on va présenter les technologies utilisées ainsi qu'on va justifier le choix des frameworks.
- Le cinquième chapitre est consacré pour l'étude conceptuelle dans laquelle on va essayer d'expliquer la conception faite lors de la réalisation du projet mené des interfaces IHM élaborées dans le procès de la conception.
- Le dernier chapitre portera sur la réalisation de notre solution, il va décrire les majeurs problématiques rencontrées ainsi que les solutions proposées puis des captures d'écran de l'application.

À la fin du rapport, une présentation des perspectives est importante pour démontrer que la solution réalisée est ouverte et extensible.

Le but de ce rapport n'est pas de faire uniquement une présentation exhaustive de tous les aspects techniques que nous avons pu apprendre ou approfondir, mais aussi, de manière synthétique et claire, de faire un tour d'horizon des aspects techniques et humains auxquels nous avons été confronté.

Chapitre I :

Contexte générale du projet

Introduction

1.1. Présentation de l'entreprise TamTam

1.1.1 Fiche signalétique TamTam

1.2. Présentation du projet

1.3. Méthodologie de travail

1.3.1 Étude préliminaire

1.3.2 Choix de méthodologie de travail

1.3.3 Répartitions des rôles dans Scrum

1.3.4 développement piloté par les tests

1.4. Planning

1.4.1 Diagramme de gantt

1.4.2 Description de diagramme de Gantt

Conclusion

Introduction

Dans ce chapitre nous allons situer le projet dans son contexte général, la première section sera dédiée à la présentation de l'organisme d'accueil, à savoir TamTam international, suivi d'une description détaillée de la problématique du projet, ses objectifs et son déroulement. La dernière section contiendra la démarche adoptée ainsi que le planning du projet.

Présentation de l'entreprise

Chaque organisation possède ses propres spécificités et se distingue des autres structures qui l'entourent. Il y a donc lieu de la présenter sous ses différents aspects organisationnels et fonctionnels afin d'avoir une idée précise sur la nature de ses activités, ses clients, et ses expériences.

TamTam international est une entreprise de développement et d'intégration de produits et solutions Web et mobile fondée en 2013 par Mr. Emanuel degréve à Solvay, Belgique et a ouvert son bureau à Marrakech en 2015.

TamTam international se spécialise dans le développement et la création de solutions techniques et informatiques; offshoring informatique ; la commercialisation et la vente de solutions techniques.

TamTam internationale se spécialise dans le conseil et le développement d'applications Web, desktop et mobile et opère sur plusieurs pôles liés entre eux:



Figure 1-1 : logo de Tamtam

efff-viewer : Application desktop pour la lecture et la validation des factures électroniques en format e-fff, ce format est né le 7 juillet 2012. Basé sur l'UBL 2.0, il limite considérablement les champs (réduits à 200 potentiels) et consacre une même interprétation de toutes les maisons de softwares. Un langage commun voit le jour et permet dorénavant à la majorité des logiciels comptables et de gestion (80 % du marché) d'émettre et de recevoir sous un même format une facture électronique.

Flux : Réseau social dédié pour les comptables et fiduciaires, c'est le produit qui rassemble tous les puzzles des autres produits.

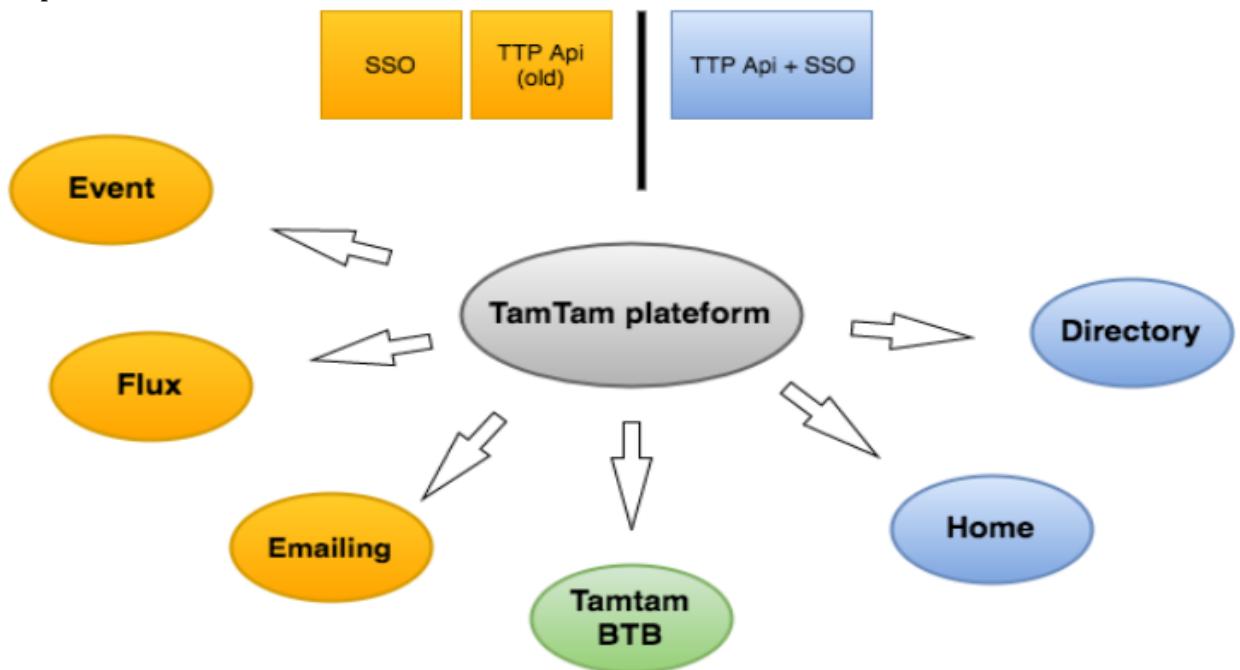


Figure 1-2 : produits de tamtam

Blog-ezine : C'est un outil de construction d'une newsletter à partir de n'importe quel article sur le web, il analyse le titre, l'image et les mots pertinents, qui deviennent des mots-clés par la suite, en utilisant la technique de drag and drop en génère des newsletters web sous différent format graphique en utilisant React

Directory : Envoi des documents aux clients (fiduciaires) en suivant l'envoi de document par un track de log (parse, validation, backup, envoi...).

Event : Application web et mobile consiste à organiser et programmer des événements de formations pour les expertises comptables Fiduciaires et les autres personnes qui opèrent dans le domaine de finance et économie, le client peut bien consulter le planning de l'événement, les heures de formation, les orateurs (Ministres, administrateurs ou membres de IEC ...) et les places disponibles, tous en ayant la possibilité de s'inscrire à ces formations en précisant une formule et leurs options qui lui convient.

Home : l'application web d'authentification pour les autres modules de l'organisation. Il englobe la partie administration de tous les produits.

1.1.1 Fiche signalétique TamTam

| | |
|--------------------|--|
| Raison sociale | TamTam international |
| Directeur générale | Emmanuel Degrève |
| IT spécialiste | Bruno de l'Escaille |
| Chef technique | Elhachmi Mohamed |
| Adresse | Résidence El Batoul, Rue Mohamed El Bequal Bloc B Appt N°20 - Marrakech-Médina |
| RC | 53375 (Tribunal de Marrakech) |
| Forme juridique | Société à Responsabilité Limitée à Associé Unique |
| Capitale | 99 000 Dh |
| Date de Création | 2013 |
| Émail | tamtampro.it@gmail.com |
| Téléphone | 05 24 15 63 12 |
| Nombre d'employé | 14 personnes |

Tableau 1 : Fiche signalétique TamTam

Présentation du projet

La volonté de TamTam international était de pouvoir proposer à ses clients un système efficace de gestion de lettre d'information électronique «NewsLetters».

Les solutions présentent sur le marché tel que : Mailchimp, SendinBlue et Sarbacane... Ne répond pas aux besoins compliqués de l'entreprise, surtout que nos exigences fonctionnelles demandent une intégration totale avec notre api, ces solutions sont chères (jusqu'aux 2290 Euro/An) pour chaque client, aussi limitées à un nombre des E-mail précis par jour.

L'objectif de notre travail est de développer une application de mailing de masse permettant un envoi rapide de campagnes d'e-mails personnalisés à chaque destinataire des différentes communautés et aux acteurs de nos produits (orateurs, exposants, participants ...), elle doit permettre aussi d'importer des destinataires à partir d'un fichier, et de générer des campagnes d'e-mails personnalisés à partir de modèles.

Le deuxième axe, c'est que les images et les liens contenus dans les Newsletters doivent également être encodés. En effet, tous les clics et les ouvertures des lettres doivent être comptabilisés, cela s'appelle le tracking et cela permet à l'entreprise de mieux connaître son lectorat et donc sa clientèle, en ayant ainsi conscience des comportements et des attentes de ses lecteurs, l'entreprise pourra orienter les sujets, de revoir la disposition des formations et orateurs présentés.

Méthodologie de travail

1.3.1 Étude préliminaire

| | Description | Points forts | Points faibles |
|-------------------------------------|--|---|--|
| Cascade | <ul style="list-style-type: none"> Propose de dérouler les phases projet de manière séquentielle Cité pour des raisons historiques | <ul style="list-style-type: none"> Distingue clairement les phases projet | <ul style="list-style-type: none"> Non itératif Ne propose pas de modèles de documents |
| RUP 'Rational Unified Process' | <ul style="list-style-type: none"> Promu par Rational. Le RUP est à la fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans un référentiel Web) Cible des projets de plus de 10 personnes | <ul style="list-style-type: none"> Itératif Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes... Propose des modèles de documents, et des canevas pour des projets types | <ul style="list-style-type: none"> Coûteux à personnaliser Très axé processus, au détriment du développement : peu de place pour le code et la technologie |
| XP 'eXtreme Programming' | <ul style="list-style-type: none"> Ensemble de « Bests Practices » de développement (travail en équipes, transfert de compétences...) Cible des projets de moins de 10 personnes | <ul style="list-style-type: none"> Itératif Simple à mettre en œuvre Fait une large place aux aspects techniques : prototypes, règles de développement, tests... Innovant: programmation en duo, kick-off matinal debout ... | <ul style="list-style-type: none"> Ne couvre pas les phases en amont et en aval au développement : capture des besoins, support, maintenance, tests d'intégration... Elude la phase d'analyse, si bien qu'on peut dépenser son énergie à faire et défaire Assez flou dans sa mise en œuvre: quels intervenants, quels livrables ? |
| 2TUP 'Two Track Unified Process' | <ul style="list-style-type: none"> S'articule autour de l'architecture Propose un cycle de développement en Y Détailé dans "UML en action" (voir références) Cible des projets de toutes tailles | <ul style="list-style-type: none"> Itératif Fait une large place à la technologie et à la gestion du risque Définit les profils des intervenants, les livrables, les plannings, les prototypes | <ul style="list-style-type: none"> Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement... Ne propose pas de documents types |
| Scrum | <ul style="list-style-type: none"> Se base sur des itérations dites sprints de développement. | <ul style="list-style-type: none"> Donne toute confiance aux développeurs et les laisser faire leur travail. | <ul style="list-style-type: none"> La mise en œuvre du développement n'est pas précisée. |

Tableau 1-2 : comparaison entre les méthodologies de travail

Nous avons cité la méthode en cascade dans le tableau précédent pour des raisons historiques. Les autres méthodes proposent toutes un déroulement itératif et incrémentale.

Suivre une approche itérative et incrémentale permet de réduire la complexité de la réalisation des différentes phases du projet. Elle permet aussi d'impliquer l'utilisateur et lui présenter rapidement des éléments de validation. De plus, l'itératif permet une gestion efficace des risques, en abordant dès les premières itérations les points difficiles.

RUP couvre l'ensemble du processus en spécifiant les interactions entre chacune des phases. Elle est connue par sa difficulté d'être personnalisée. Elle néglige les contraintes techniques.

XP se concentre sur la phase de développement et néglige la phase de capture de besoins fonctionnels et techniques et la phase de conception.

1.3.2 Choix de méthodologie de travail

En termes de méthodes de travail, l'équipes de «TamTam» s'efforcent de suivre un ensemble de principes jugés pragmatiques pour développer et gérer les différents projets en cours. Ces principes découlent de quatre valeurs: l'équipe, l'application, la collaboration et l'acceptation du changement. Il s'agit des méthodes Agiles.

Pour «TamTam», la méthode agile la plus répandue est Scrum.

Scrum: D'avantage qu'une méthode formelle, Scrum peut être vu comme un Framework méthodologique, dont l'implémentation doit être ajustée en fonction des caractéristiques techniques, organisationnelles et culturelles des projets qui souhaitent la mettre en œuvre.

1.3.3 Répartitions des rôles dans Scrum

Scrum identifie trois acteurs :

- ◆ **Product Owner** (Directeur de Produit), qui possède l'expertise fonctionnelle et qui réalise les arbitrages nécessaires à la priorisation des développements. Son rôle est absolument essentiel, et son respect des règles du jeu est la pierre angulaire du succès d'un projet agile, dans notre cas (IEC, FFF, UP ...)
- ◆ **Scrum Master**, membre de l'équipe, et dont la tâche principale est d'optimiser la capacité de production de l'équipe en l'aidant à travailler de façon autonome et à s'améliorer constamment. Il est également le garant de la bonne implémentation de Scrum.
- ◆ **l'Équipe**, qui prend en charge le développement du produit (planification, conception, codage, tests, documentation) sans spécialisation des rôles. La particularité d'une Équipe Scrum est d'être « auto-organisée », et donc dépourvue de hiérarchie.

L'unité de temps, dans Scrum, est le Sprint. Un sprint est une itération courte (de l'ordre de 2 semaines pour «TamTam») dont le périmètre est garanti et défini lors d'une cérémonie de planification initiale.

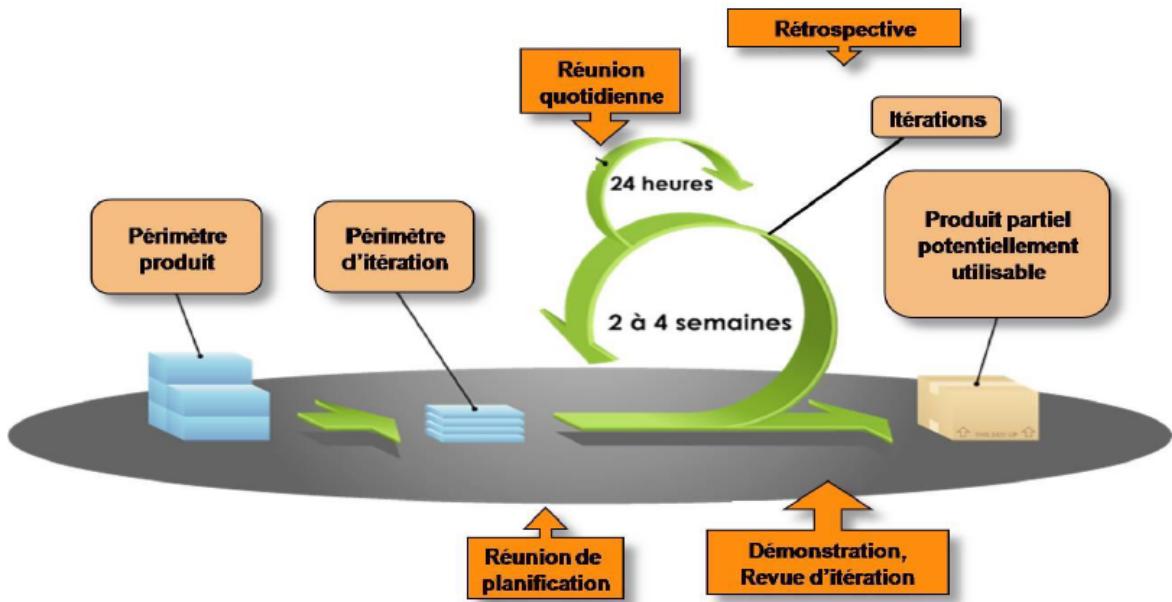


Figure 1-3 : démarche de développement Scrum

1.3.2 développement piloté par les tests

le développement piloté par les tests, ou Le «test-driven development» (TDD) est une démarche de développement qui préconise d'écrire les tests unitaires avant d'écrire le code source d'un logiciel.

Le cycle préconisé par TDD comporte cinq étapes :

1. écrire un premier test.
2. vérifier qu'il échoue (car le code qu'il teste n'existe pas), afin de vérifier que le test est valide.
3. écrire juste le code suffisant pour passer le test.
4. vérifié que le test passe.
5. puis refactoriser le code, c'est-à-dire améliorer le tout en gardant les mêmes fonctionnalités.

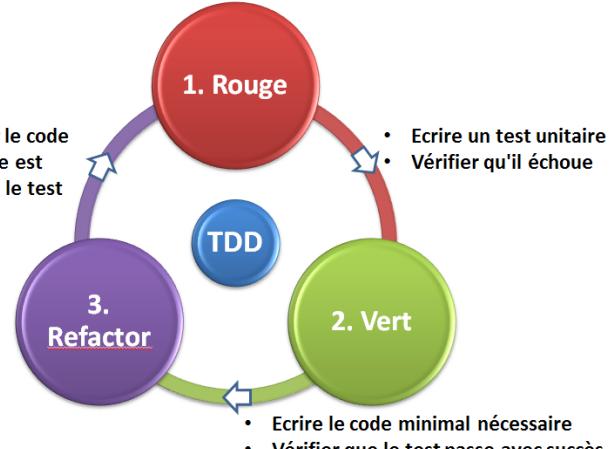


Figure 1-4 : démarche de TDD

L'approche classique du test consiste à coder puis tester à la fin (cycle de développement en cascade). Mais l'inconvénient majeur qu'on risque d'écrire un test faux qui valide un code faux .

Planning

1.4.1 diagramme de gantt

Pour un meilleur ordonnancement des tâches du projet on a choisi de le présenter à l'aide du diagramme de Gantt suivant :

| Diagramme de Gantt \ Diagramme des Ressources \ | | | |
|--|---------------|-------------|--|
| Nom | Date de début | Date de fin | |
| Formation sur la structure de l'entreprise et les outils ... | 08/03/17 | 09/03/17 | |
| Description de projet et étude de l'existant | 10/03/17 | 11/03/17 | |
| Formation sur symfony | 12/03/17 | 16/03/17 | |
| Formation sur les test unitaires et fonctionnelles | 17/03/17 | 19/03/17 | |
| Sprint: réalisation des ressources CRUD en backend | 20/03/17 | 06/04/17 | |
| Sprint: réalisation des fonctionnalités en backend | 07/04/17 | 23/04/17 | |
| Formation en React/Redux | 24/04/17 | 29/04/17 | |
| Sprint : réalisation des interfaces web de base | 30/04/17 | 19/05/17 | |
| Sprint : réalisation des interfaces web d'envoi de ca... | 20/05/17 | 04/06/17 | |
| Etude de l'algorithme utilisé | 01/06/17 | 10/06/17 | |
| Sprint : réalisation du partie statistique | 05/06/17 | 24/06/17 | |
| Sprint : implémentation de l'algorithme | 26/06/17 | 01/07/17 | |

Figure 1-5 : Tâches à réaliser dans le cadre du projet

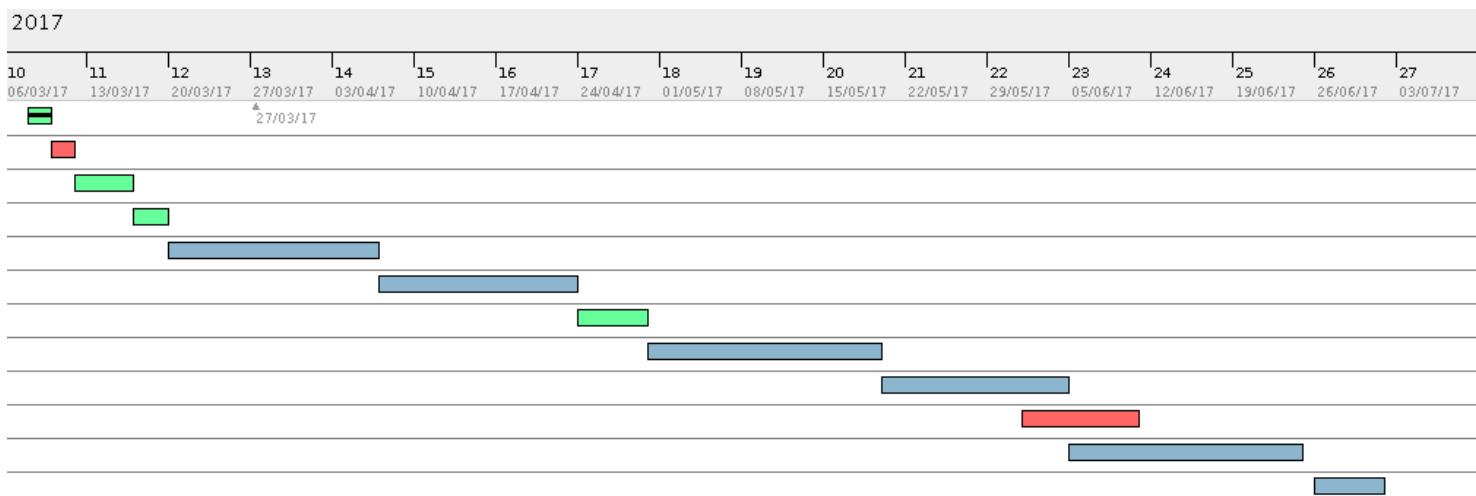


Figure 1-6 : l'ordonnancement des tâches

1.4.1. Description de diagramme de Gantt

| | | |
|-----------|---|--|
| Formation |  | Dans cette section, on a opté de tout ce qui est formation sur la structure de l'entreprise et sur les technologies utilisées. |
| Étude |  | Dans une première partie, on a fait une description générale du projet, et une étude de l'existant, après on a fait une étude détaillée sur l'algorithme utilisé lors du projet, cette étude était faite en parallèle avec la réalisation de projet puisqu'elle n'était pas inclus dans le cahier de charge. |
| Sprint |  | Cette partie est conçue pour la réalisation des sprints, au début de chaque sprint, on réalise des réunions où on se met d'accord sur l'étude fonctionnelle du projet et la conception, lors de réalisation, on procède avec les tests selon la procédure de TDD puis la réalisation des fonctionnalités souhaitées, ainsi que la réalisation des maquettes. A la fin de chaque sprint, on procède à une réunion d'état d'avancement du projet, avec une réalisation des minis sprints de correction convenable aux exigences du client. |

Tableau 1-3 : Description de diagramme de gantt

Conclusion

Après avoir présenté l'organisation d'accueil, le thème du projet et sa problématique. Nous sommes passés à une description de la méthodologie de travail. Nous avons pu ensuite définir les principales tâches du projet, et de les ordonner à l'aide du diagramme de Gantt.

Chapitre II :

État d'art

Introduction

2.1. Présentation générale

2.1.1. Introduction

2.1.2. Contexte et problématique

2.2. Les échecs transitoires

2.2.1. Introduction

2.2.2. Gestion des erreurs transitoires

2.2.3. Problématique

2.3. Les algorithmes du backoff

2.3.1. Questions et considérations

2.3.2. Quand utiliser ce pattern ?

2.3.3. les différentes phase du backoff

2.3.4 . Fonctions retry backoff

2.4. Exponentielle backoff

2.4.1. Introduction

2.4.2. Formule mathématique

Conclusion

Introduction

Ce chapitre a comme but de présenter les résultats de la première phase de mon stage de fin d'étude qui est l'état d'art sur les méthodes d'envoi de mail afin d'analyser les différents problèmes qui peuvent apparaître lors de l'envoi, avant de passer à développer notre propre système adapté à nos exigences.

Présentation générale

2.1.1 Introduction

Dans ce chapitre, nous discuterons l'importance du «Retry Pattern» et de la manière de l'appliquer efficacement dans nos applications. Nous discuterons également la façon dont l'algorithme «Exponentiel Backoff» peut être utilisé avec le «Retry Pattern».

2.1.2 Contexte et Problématique

La plupart d'entre nous ont mis en place un mécanisme de réactivation à un moment ou à un autre dans nos applications. Pourquoi devons-nous appeler cela un patron de conception est la première question qui vient à l'esprit. Pour comprendre cela, nous devons d'abord comprendre le concept des «échecs transitoires» et pourquoi les échecs transitoires sont considérée d'avantage à notre application.

Une application qui communique avec les éléments en cours d'exécution doit être sensible aux défauts transitoires pouvant survenir dans cet environnement. Les défauts incluent la perte momentanée de la connexion réseau aux composants ou aux services, l'indisponibilité temporaire d'un service ou les délais d'attente lorsqu'un service est occupé.

Ces défauts sont généralement auto-corrigés, et si l'action déclenchée par un défaut est répétée après un délai approprié, il est susceptible de réussir. Par exemple, notre service de base de données qui traite un grand nombre de demandes d'envoi de mail simultanées peut implémenter une stratégie d'étranglement qui rejette temporairement toute autre requête jusqu'à ce que sa charge de travail se diminu. Un client qui tente accéder à la base de données et envoyer le mail peut échouer à se connecter, mais si elle tente à nouveau après un délai, elle peut réussir.

Les échecs transitoires

2.2.1 Introduction

Les échecs transitoires (Transient Failures) sont les pannes qui se produisent lors de la communication avec le composant ou le service externe, ou bien ce dernier peut ne pas être disponible. Mais cette indisponibilité ou l'absence de connexion ne sont pas dues à un problème dans le service mais en raison de certains problèmes tels que l'échec du réseau ou la surcharge du serveur. De tels problèmes sont éphémères. Si nous appelons le service à nouveau, il est probable que notre appel réussira. Un tel type de défaillance s'appelle des échecs transitoires.

Traditionnellement, nous avons eu de telles erreurs dans les connexions à la base de données et les appels de service. Mais dans notre cas, les chances d'obtenir de telles erreurs ont augmenté, car notre application elle-même contient certains éléments/composantes dans le serveur.

Ainsi, les échecs comme la perte momentanée du réseau, l'indisponibilité du service et les délais d'attente sont devenus plus importants.

Ce type d'échecs peut facilement être contourné en appelant simplement le service après un délai d'attente.



Figure 2-1 : processus d'échec transitoire

2.2.2 Gestion des erreurs transitoires

Avant de commencer à parler de la façon de gérer les défauts transitoires, la première tâche devrait être d'identifier les défauts transitoires. La façon de le faire est de vérifier si la faute est un message que le service cible envoie et présente un certain contexte du point de vue de l'application. Si c'est le cas, nous savons que ce n'est pas une faute transitoire puisque le service nous envoie une faute. Mais si nous recevons une faute qui ne vient pas du service et qui provient peut-être d'autres raisons, comme les problèmes d'infrastructure, et la faute semble être une faute qui peut être résolu simplement par le service appelant, nous pouvons le classer comme un faute transitoire.

Une fois que nous avons identifié la faute en tant que défaut transitoire, nous devons mettre une nouvelle logique de retry afin que le problème soit résolu simplement en appelant à nouveau le service.

La façon typique de mettre en œuvre le diagramme d'activités suivant:

- 1-Identifier si le défaut est un défaut transitoire.
- 2-Définir le nombre maximum des nouvelles tentatives(selon le type d'erreur).
- 3-Définir le délai d'attente fixe avant de ré-appeler le service
- 4-Réessayez l'appel de service et augmenter le nombre de nouvelles tentatives.
- 5-Si les appels réussissent, renvoie le résultat à l'appelant.
- 6-Si nous obtenons toujours la même faute, garder jusqu'à ce que le nombre maximum de nouvelles tentatives est atteint.
- 7-Si l'appel échoue même après un maximum de relances, avisez l'appelant que le service cible est indisponible.

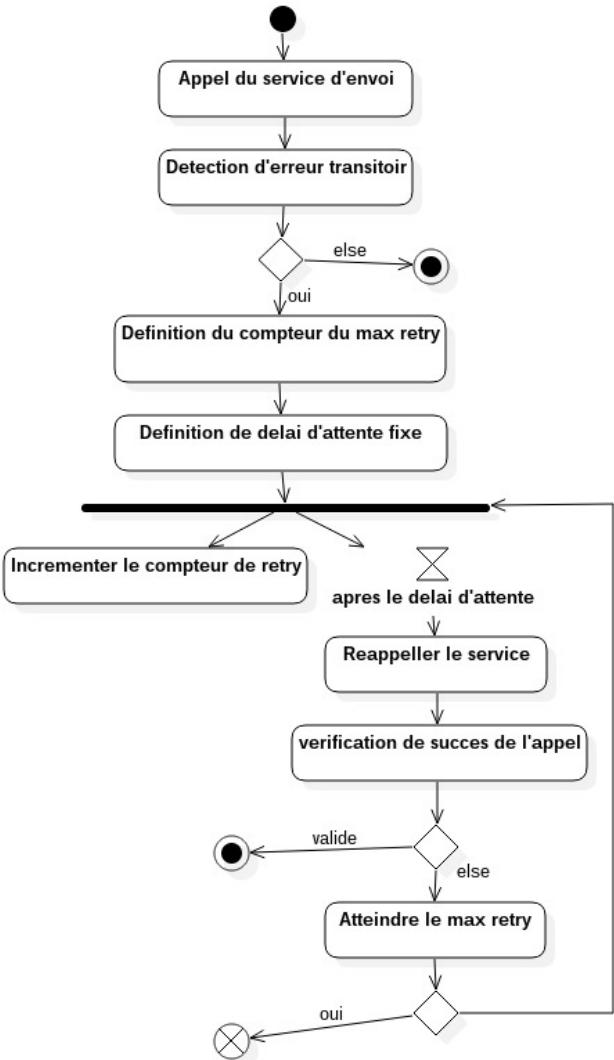


Figure 2-2 : diagramme d'activité du simple retry

2.2.3 Problématique

Le mécanisme du simple retry dont nous avons discuté est assez simple et on pourrait penser pourquoi nous discutons même une chose aussi simple. La plupart utilise ce mécanisme dans ses applications. La raison pour laquelle nous discutons, c'est parce qu'il y a un petit problème dans le mécanisme du simple retry mentionné ci-dessus.

Pour comprendre le problème, imaginez un scénario où la panne transitoire se produite car le service est surchargé ou un certain étranglement est mis en place à la fin du service. Ainsi, ce service rejette de nouveaux appels.

Il s'agit d'un défaut transitoire comme si nous appelions le service après un certain temps, notre appel pourrait réussir. Mais il pourrait également être possible que nos demandes de nouvelle tentative soient ajoutées à la surcharge du service occupé. Ce qui signifie que s'il y a plusieurs instances de nos applications qui tentent de réessayer pour le même service, le service sera plus longtemps dans l'état surchargé et prendra plus de temps pour se remettre de cet état.

Donc, en quelque sorte, nos demandes contribuent davantage à la raison de la faute et, en fin de compte, c'est seulement notre application qui souffre en raison du temps de récupération plus long du service. Alors, comment pouvons-nous résoudre ce problème ?

Les algorithmes du backoff

L'idée derrière l'utilisation d'un retour «backoff» avec tentatives est que, au lieu de réessayer après avoir attendu un certain temps, nous augmentons le temps d'attente entre les tentatives après chaque échec de réessayer.

Par exemple, lorsque la demande échoue la première fois, nous réessayons après 1 seconde. Si elle échoue pour la deuxième fois, nous attendons 2 secondes avant la prochaine tentative. Si la deuxième tentative échoue, nous attendons 4 secondes avant la prochaine tentative. Nous augmentons de plus en plus le temps d'attente entre les demandes de retouches consécutives après chaque échec. Cela donne au service un certain temps de respiration afin que, si la faute est due à une surcharge de service, elle pourrait être résolue plus rapidement.

Le temps réel d'attente devrait dépendre de l'application et du service. Dans les scénarios pratiques, aucun utilisateur n'attendra plus de 7 secondes pour que sa demande soit traitée (comme dans l'exemple ci-dessus).

Donc, avec un retour backoff, notre algorithme ressemblera à ce qui suit:

- 1-Identifier si le défaut est un défaut transitoire.
- 2-Définir le nombre maximum des nouvelles tentatives(selon le type d'erreur).
- 3-Augmenter le temps d'attente puis Réessayez l'appel de service et augmenter le nombre de nouvelles tentatives.
- 4-Si les appels réussissent, renvoyez le résultat à l'appelant
- 5-Continuez à réessayer et augmentez la période de délai jusqu'à ce que le nombre maximal de répétition, ou le temps maximal d'attente soit atteint.

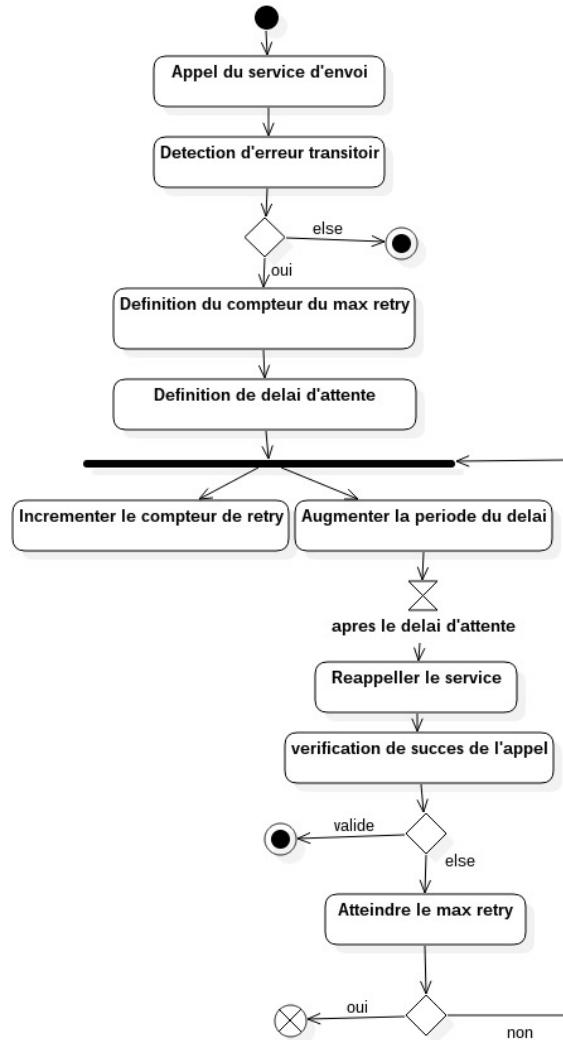


Figure 2-3 : diagramme d'activité du backoff retry

2.3.1 Questions et considérations

Une politique de retry agressive avec un délai minimal entre les tentatives et un grand nombre de tentatives, pourrait encore dégrader un service occupé qui se déroule à proximité de sa capacité. Cette politique de retry pourrait également affecter la réactivité de l'application si elle essaie continuellement d'effectuer une opération en panne.

Si une demande échoue après un nombre important de tentatives, il est préférable pour l'application d'empêcher d'autres requêtes d'accéder à la même ressource et de simplement signaler une panne immédiatement. Lorsque la période expire, l'application peut temporairement permettre une ou plusieurs demandes pour voir si elles sont couronnées de succès. Pour plus de détails sur cette stratégie, voir le modèle Pattern «Circuit Breaker».

Considérez si l'opération est idempotente (opération a le même effet qu'on l'applique une ou plusieurs fois). Si c'est le cas, il est naturellement sûr de réessayer. Sinon, les tentatives peuvent entraîner l'exécution de l'opération plus d'une fois, avec des effets secondaires involontaires. Par exemple, un service peut recevoir la demande, traiter la requête avec succès, mais ne pas envoyer de réponse. À ce stade, la logique de tentative pourrait renvoyer la demande, en supposant que la première requête n'a pas été reçue.

Une demande à un service peut échouer pour diverses raisons, élevant différentes exceptions en fonction de la nature de l'échec. Certaines exceptions indiquent un échec qui peut être résolu rapidement, tandis que d'autres indiquent que l'échec est plus long. Il est utile pour la politique de retry d'ajuster le temps entre les tentatives de ré-tentative en fonction du type d'exception.

Ne réessayer la logique que lorsque le contexte complet d'une opération défaillante est compris. Par exemple, si une tâche qui contient une politique de tentative invoque une autre tâche qui contient également une politique de retry, cette couche supplémentaire de tentatives peut ajouter de longs délais au traitement. Il pourrait être préférable de configurer la tâche de niveau inférieur pour échouer rapidement et signaler la raison de l'échec à la tâche qui l'a invoqué. Cette tâche de niveau supérieur peut alors gérer l'échec en fonction de sa propre stratégie.

Il est important de consigner toutes les pannes de connectivité qui provoquent une nouvelle tentative afin que les problèmes sous-jacents à l'application, aux services ou aux ressources puissent être identifiés. Enquêter sur les défauts les plus susceptibles de se produire pour un service ou une ressource pour découvrir si elles sont probablement durables ou terminales. Si c'est le cas, il est préférable de gérer la faute comme une exception. L'application peut signaler ou enregistrer l'exception, puis essayer de continuer soit en invoquant un autre service (si celui-ci est disponible), soit en offrant des fonctionnalités dégradées. Pour la façon de détecter et de gérer les défauts de longue durée il faut utiliser le pattern de «Circuit Breaker».

2.3.2 Quand utiliser ce pattern ?

Utilisez ce modèle lorsqu'une application pourrait subir des défauts transitoires car il interagit avec un service à distance ou accède à une ressource distante. Ces défauts devraient être de courte durée, et en répétant une demande qui a déjà échoué et qui pourrait réussir à une tentative ultérieure.

dans la pratique on peut Considérer les situations suivantes comme une tentative de livraison échouée.

-Status HTTP dans la gamme 500-599.

-Status HTTP en dehors de la plage 200-599.

-Expiration d'un délai d'attente de demande .

-Toute erreur de connexion telle que l'expiration de délai de connexion, endpoint inaccessible, mauvais certificat SSL, etc.

Ce pattern pourrait ne pas être utile:

-Lorsqu'un défaut est susceptible d'être durable, car cela peut affecter la réactivité d'une application. L'application perd du temps et des ressources à essayer de répéter une demande qui est susceptible d'échouer.

-Pour le traitement des défaillances qui ne sont pas dus à des défauts transitoires, comme les exceptions internes «internal exceptions» causées par des erreurs dans la logique métier d'une application.

-En alternative à la résolution des problèmes d'évolutivité dans un système. Si une application connaît des défauts occupés fréquents, il est souvent indiqué que le service ou la ressource auquel on accède devrait être étendu.

2.3.3 les différentes phases du backoff

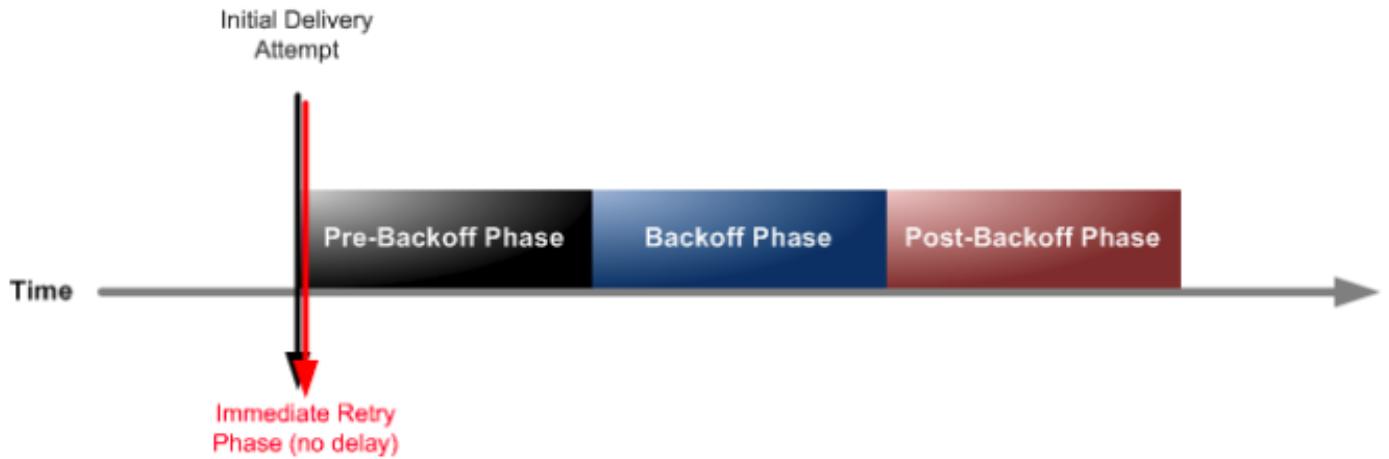


figure 2-4 : les différentes phases d'un algorithme backoff

Phase de retry immédiate : Aussi appelée phase sans délai, C'est une phase optionnelle, cette phase se produit immédiatement après la tentative de livraison initiale. La valeur qu'on définit pour cette phase, détermine le nombre de tentatives immédiatement après la tentative de livraison initiale. Il n'y a pas de délai entre les tentatives dans cette phase, Le nombre par défaut de tentatives pour cette phase est 0.

Phase de Pre-backoff : cette phase suit la phase de retry immédiate.C'est une phase aussi optionnelle, On l'utilise pour créer un ensemble de tentatives qui se produisent avant qu'une fonction backoff qui s'applique aux tentatives. On utilise le paramètre «Rappel de retard minimum» pour spécifier le nombre de tentatives dans la phase de pré-backoff. On peut contrôler le délai entre les tentatives de cette phase en utilisant le minimum de délai constant, ce paramètre affecte les tentatives en deux phases: elle s'applique à toutes les tentatives dans la phase de pré-backoff et sert de délai temporel initial pour les tentatives dans la phase de backoff. Le nombre par défaut de tentatives pour cette phase est 0.

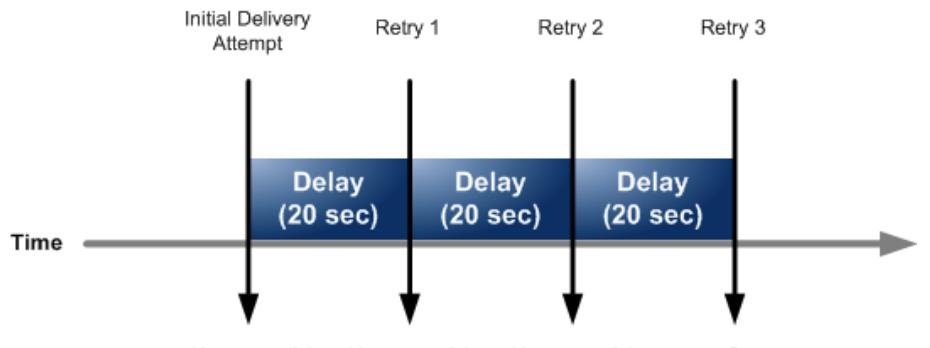


figure 2-5 : délai constant entre les tentatives

Phase de backoff : Cette phase s'appelle la phase backoff car on peut contrôler le délai entre les tentatives de cette phase en utilisant la fonction retry backoff, c'est la seule phase utilisée par défaut. Régler le délai minimum et le délai maximal, puis sélectionner une fonction «retry backoff» pour définir la rapidité avec laquelle le délai passe du délai minimum au délai maximal. Dans cette phase on définit le nombre de tentatives qui représente le nombre total de tentatives des trois phases.

La fréquence des tentatives est donc contrôlée par trois facteurs :

-**Délai minimum** : délai associé à la première tentative de la phase backoff.

-**Délai maximal** : délai associé à la dernière tentative de la phase backoff.

-**Fonction retry backoff** : La fonction retry backoff définit l'algorithme utilisé pour calculer les retards associés à toutes les tentatives de réessayer entre la première et la dernière tentative dans la phase de retour.

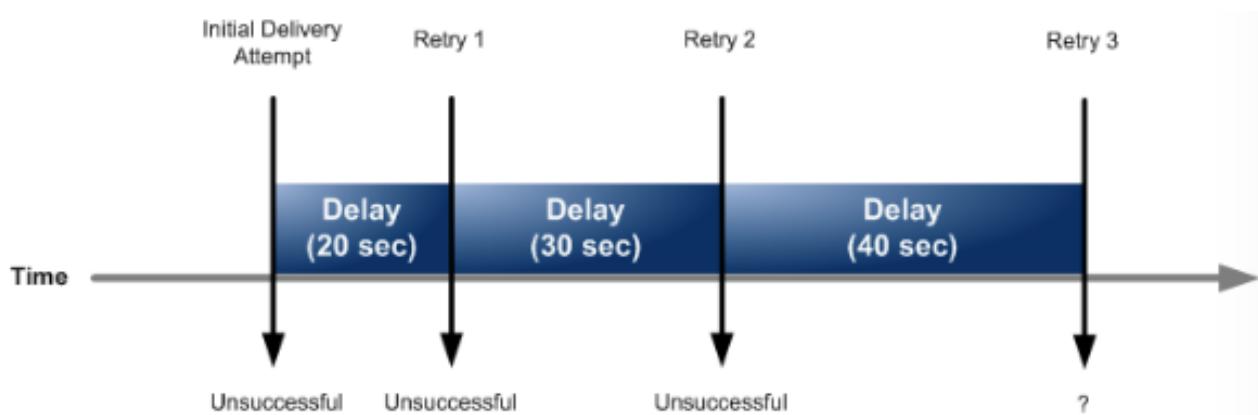


Figure 2-6 : délai croissant entre les tentatives

Phase post-backoff: La phase post-backoff est la phase finale. Elle est utilisée si on souhaite créer un ensemble d'une ou plusieurs tentatives qui se produisent après que la fonction backoff affecte le délai entre les tentatives. Dans cette phase, le temps entre les tentatives est constant et est égal au paramètre qu'on choisit pour le délai maximal. Le paramètre de délai maximal affecte les tentatives dans deux phases: elle s'applique à toutes les tentatives dans la phase post-backoff et sert de délai final pour les tentatives dans la phase de backoff. Le nombre par défaut de tentatives pour cette phase est 0.

2.3.4 Fonctions retry backoff

dans le choix du facteur fonction retry pattern dans la 3eme phase on peut choisir l'un des fonctions suivants :

- ◆ Linear
- ◆ Geometric
- ◆ Exponential
- ◆ Arithmetic

La figure suivante montre comment chaque fonction de retry backoff affecte le retard associé aux messages

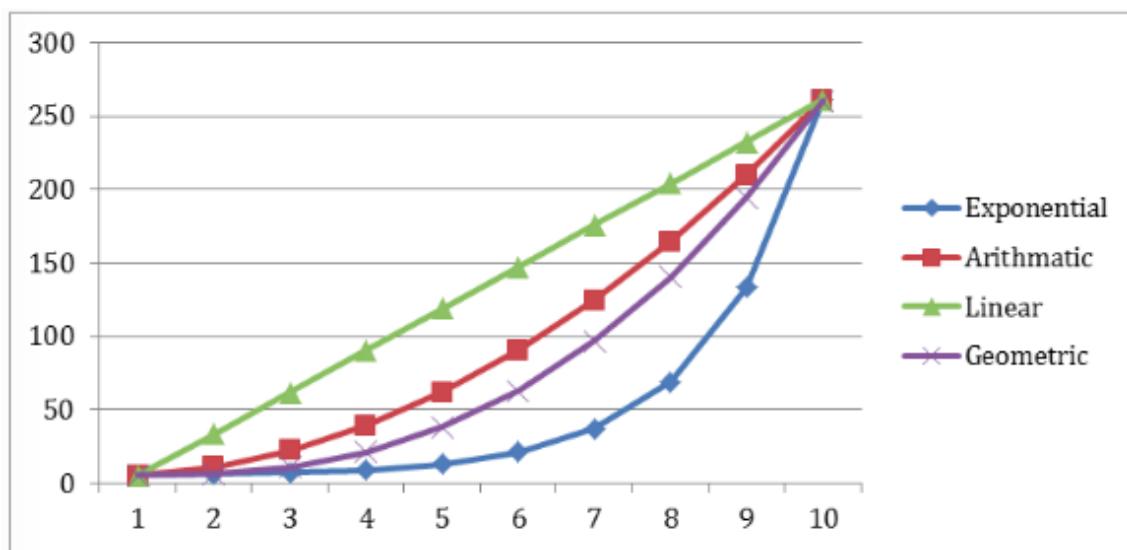


figure 2-7 : graphe de comparaison entre les fonctions backoff

pendant la période de backoff. L'axe vertical représente le retard en secondes associé à chacune des 10 tentatives. L'axe horizontal représente le numéro de tentative. Le délai minimum est de 5 secondes et le délai maximal est de 260 secondes.

Exponentielle backoff

2.4.1 Introduction

Le retour exponentiel est un algorithme qui utilise des feedback pour diminuer de manière multiple le taux de certains processus, afin de trouver progressivement un taux acceptable. Les tentatives augmentent de façon exponentielle et cessent d'augmenter lorsqu'un certain seuil est atteint.

On peut implémenter un algorithme d'exponentiel backoff pour un meilleur contrôle de flux. L'idée derrière un retour exponentiel est d'utiliser des attentes progressivement plus longues entre les tentatives pour des réponses d'erreur consécutives. On doit implémenter un intervalle de délai maximal, ainsi qu'un nombre maximal de tentatives. L'intervalle de délai maximal et le nombre maximal de tentatives ne sont pas forcément des valeurs fixes et doivent être définis en fonction de l'opération effectuée, ainsi que d'autres facteurs locaux, tels que la latence du réseau.

La plupart des algorithmes de retour exponentiel utilisent «retard aléatoire» pour éviter les collisions successives. Puisque nous essayons pas d'éviter ces collisions dans ces cas, nous n'avons pas vraiment besoin d'utiliser ce nombre aléatoire.

2.4.2 Formule mathématique

Étant donné une répartition uniforme des temps de recul, le temps de retour attendu est le moyen des possibilités. C'est-à-dire, après un nombre de tentative c , le nombre de place backoff est en $[0,1,\dots,N]$, où

$$N=2^c-1 \text{ , et le temps de retour attendu est : } \frac{1}{N+1} \cdot \sum_{i=0}^N i$$

Puis calculer la moyenne des possibilités de temps de retour:

$$E(c)=\frac{1}{N+1} \cdot \sum_{i=0}^N i = \frac{N}{2} = \frac{2^c-1}{2}$$

Conclusion

Dans ce présent chapitre, nous avons principalement présenté la notion des erreurs transitoires qui sont des erreurs susceptibles d'être résolues si on ré-appeler le service, et que le simple retry utilisé pour ces problème n'est pas assez efficace, donc on a introduit le terme backoff en utilisant la fonction d'exponentielle backoff.

Maintenant, afin d'étudier en profondeur le projet, nous allons analyser ses besoins fonctionnels et techniques.

Chapitre III :

Étude Fonctionnelle

Introduction

3.1. Spécification des Exigences fonctionnelles

3.1.1. Domaine d'activité

3.1.2. Client

3.2. Capture des besoins fonctionnels

3.2.1. fonctionnalité souhaité

3.2.2. Définition des acteurs

3.2.3. Identification des exigences fonctionnelles

3.3. Analyse des spécifications fonctionnelles

3.3.1. Diagrammes des cas d'utilisation

3.3.2. Diagrammes de séquence

3.4. Exigences Non Fonctionnelles

Conclusion

Introduction

Ce chapitre englobe l'analyse fonctionnelle effectuée, les différents modules du système, les acteurs principaux , ainsi qu'une description détaillée des cas d'utilisations.

Dans ce chapitre nous allons se focaliser sur les besoins de l'utilisateur de l'application. Ces derniers doivent être bien définis car des exigences incorrectes peuvent conduire à des désastres, et influent négativement toutes les phases de développement. Par conséquent, les besoins doivent être définis en détails pour sortir avec un cahier des charges bien précis, en gérant les exigences relatives au domaine, exigences relatives à l'utilisateur et finalement les exigences relatives au système.

Spécification des Exigences fonctionnelles

L'intérêt de cette section est de comprendre tous ce que l'utilisateur de l'application veut du système. En outre, on va répondre aux questions suivantes :

- Que doit faire le système doit faire ?
- Comment le système doit le faire ?
- Existe-t-il plusieurs modes de fonctionnement ?
- Quelle sont les bonnes réactions aux stimulus possibles ?
- Quel doit être le format des données ?

3.1.1. Domaine d'activité

Le domaine d'activité correspond à un secteur du travail de finance et économie, plus précisément le domaine comptabilité et la fiduciaire. On s'adresse plus particulièrement aux expertises comptables.

3.1.2. Client

Le client peut être une personne physique ou morale, des organisateurs des formations des expertises comptables en Belgique, des personnes qualifiées, ou des organisations officielles telles que IEC (Institut des Experts-comptables et des Conseils fiscaux).

Capture des besoins fonctionnels

3.2.1. fonctionnalité souhaitée

L'application à produire doit répondre aux besoins suivants :

- ◆ Ordonner les subscribers de l'application sous forme de liste ou groupes .
- ◆ Envoyer des campagnes de mails aux utilisateurs des différentes communautés, aux acteurs de nos différents produits (par exemple: participants, orateurs, exposants d'Évent).
- ◆ Les mails envoyés pourront être particularisés en fonctions de données propres à nos différents produits.
- ◆ Gestion Multi-listes de diffusion.
- ◆ Gestion des fichiers joints ou incorporés .
- ◆ Prévisualisation des newsletters avant envoi.
- ◆ Chargement possible des modèles de newsletters par URL.
- ◆ Module d'importation de masse.
- ◆ Archivage des newsletters précédemment envoyées.
- ◆ Des statistiques de chaque campagne seront stockées et visualiser proprement.
- ◆ Les mails envoyés seront unilingues, mais il sera possible de choisir la langue de l'email. le choix de la langue limitera automatiquement la liste des destinataires aux destinataires parlant cette langue.
- ◆ La sélection des destinataires sera multi critère en fonction des « types » de destinataires (il sera par exemple possible d'envoyer l'email à tous les inscrits de tel événement, tous ceux qui n'ont pas encore payé, ou qui ont choisi telle formule ou qui participent à telle conférence, ou les orateurs de tel type, ou les contacts appartenant aux groupes a, b et c mais pas x,y ou z).
- ◆ Les mails seront visibles via le web et seront conservés pour donner accès à une « boîte personnelle ».
- ◆ Le système gérera également la dés-inscription et les rebonds.
- ◆ Tous les clics et les ouvertures des lettres doivent être comptabilisés.

3.2.2. Définition des acteurs

Dans notre système, on a un seul acteur qui interagit avec l'application, c'est le client de la plateforme avec les privilégiés d'administrateur, il organise les formations des expertises comptables.

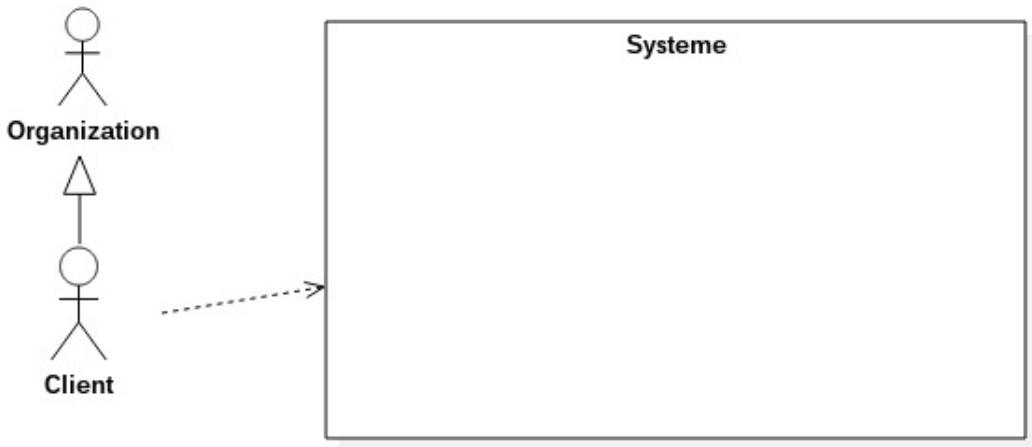


Figure3-1 :Diagramme du contexte du projet

Nous pouvons alors définir les différents droits de notre utilisateur comme suit :

| Utilisateur | Description |
|-------------|---|
| Client | L'utilisateur principal de l'application, il organise des formations des expertises comptables, une personne qualifiée dans le domaine fiduciaire, généralement un membre représente l'IEC (Institut des Experts-comptables et des Conseils fiscaux), il doit avoir les priviléges «Admin» l'application lui permet de notifier les différents contacts de la plateforme des formations et événements existants, aussi les orateurs, les exposants ainsi que les différents invités. l'application lui permet aussi de suivre les différentes statistiques sur ces e-mails (suivi des e-mails, nombre de vue, mails inactives, spams, retours...). |

Tableau 3-1 :Les différents acteurs

3.2.3. Identification des exigences fonctionnelle

| Fonctionnalité | Description |
|--------------------------------------|---|
| Authentification | Le client peut s'authentifier à la plateforme via son email d'inscription et son mot de passe. Et choisir le produit de l'Emailing parmi les autres produits de l'application. |
| Créer un nouveau groupe | Le client peut créer un nouveau groupe en précisant son nom ainsi les contacts appartenant à ce groupe. |
| Modifier un groupe | Le client peut modifier les informations d'un groupe déjà créé. |
| Gestion des prospects | Le client peut créer un prospect ou importer une liste. |
| Lister les campagnes | Le client peut lister les campagnes existantes, et leur différents états. |
| Supprimer une campagne | Le client peut supprimer une campagne de la liste. |
| Éditer une campagne | Le client peut éditer le contenu ou les destinataires d'une campagne, avant de la renvoyer. |
| Créer une nouvelle campagne | Créer une nouvelle campagne en spécifiant les groupes destinataires, en utilisant le langage de markdown, ou une page web avec la possibilité de visualiser le message avant l'envoi. |
| Tester une campagne | Créer une campagne et l'envoyer à un mail (groupe) de test. |
| Consulter statistique d'une campagne | Afficher les différentes statistiques sur les campagnes, et les destinataires ainsi leurs réactions avec les campagnes (nombre de vues, nombre de clics, nombre de retour, nombre de spam ...). |

Tableau 3-2 :Les fonctionnalités du système

Analyse des spécifications fonctionnelles

Pour mieux exploiter les fonctionnalités déjà mentionnées on va dessiner les diagrammes de cas d'utilisation pour mieux comprendre chaque tache.

3.3.1. Diagrammes des cas d'utilisation

3.3.1.1 Gestion groupe

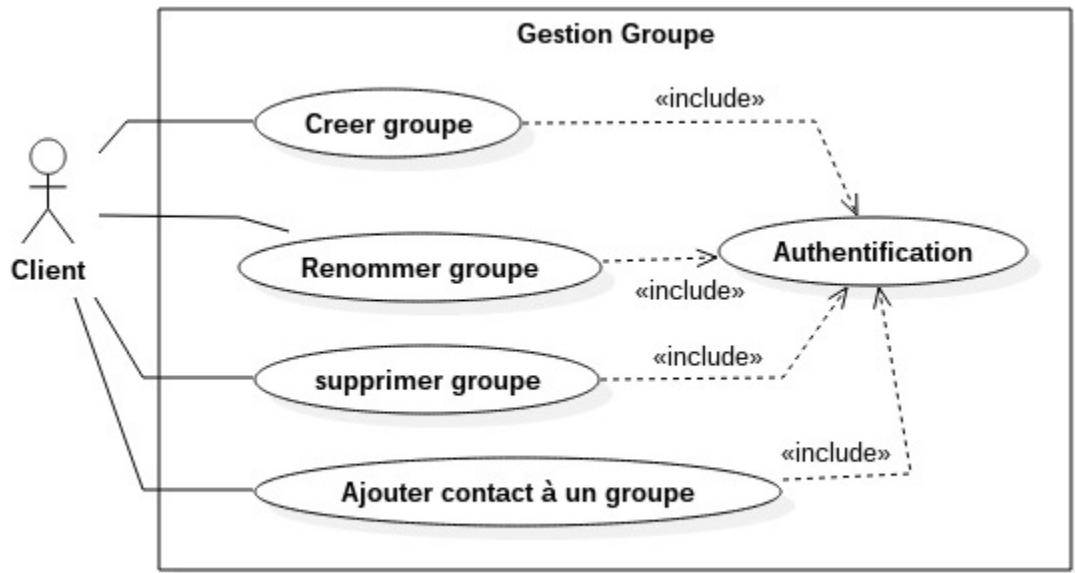


Figure 3-2 :Diagramme cas d'utilisation de gestion des groupes

a-Cas d'utilisation : création d'un groupe

| Titre | Création d'un groupe |
|----------------------|--|
| Objectif | Permet le client de gérer leur groupe d'envoi (création , modification suppression), et ajouter un contact un groupe existant. |
| Auteur | Client |
| Pré condition | -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing -Accès à la base de données . |
| Scénario nominal | -Lancer l'application. -Ajouter un nouveau groupe (nom du groupe). -Affecter les contacts à un groupe. |
| Scénario alternatif | Aucun |
| Scénario d'exception | «Problèmes dans le serveur des données». «Erreur Format JSON ». Aucune tache n'est effectuer. |

Tableau 3-3 : création d'un groupe contact

b-Cas d'utilisation : suppression d'un groupe

| | |
|-----------------------------|--|
| Titre | Suppression d'un groupe |
| Objectif | Permet le client de supprimer un groupe d'envoi. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . - privilège d'administrateur. -Activation du produit Emailing -Accès à la base de données. |
| Scénario nominal | <ul style="list-style-type: none"> -lancer l'application. -choix du groupe à supprimer. - les contacts appartenant au groupe supprimé ne seront pas affectés par cette suppression. |
| Scénario alternatif | Aucun |
| Scénario d'exception | <ul style="list-style-type: none"> «Problèmes dans le serveur des données». «Erreur Format JSON ». Aucune tâche n'est effectuée. |

Tableau 3-4 :Lister les campagnes

3.3.1.2 Gestion des prospects

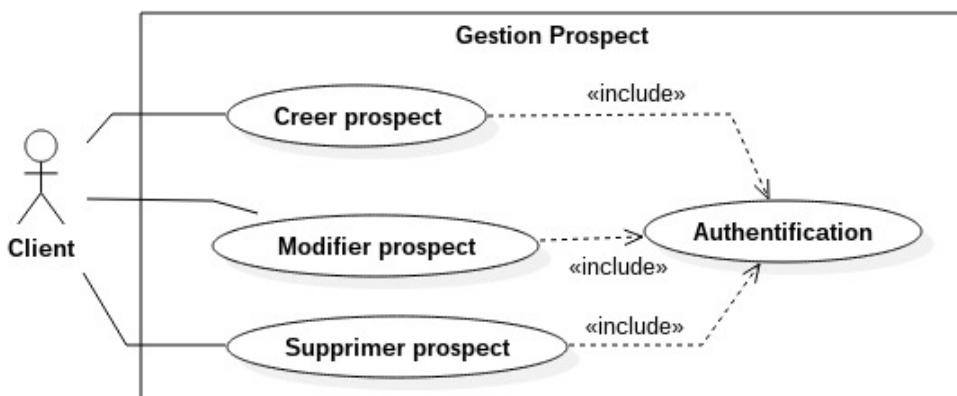


Figure 3-2 : Diagramme cas d'utilisation de gestion des groupes

a-Cas d'utilisation : Crédit de prospect

| | |
|----------------------------|--|
| Titre | Création de prospect |
| Objectif | Permet le client de créer les prospects de l'application. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing -Accès à la base de données . |
| Scénario nominal | <ul style="list-style-type: none"> -Lancer l'application. -Créer un prospect. -Spécifier les informations correspondantes à ce prospect telles que : nom, email, fonction ... |
| Scénario alternatif | <ul style="list-style-type: none"> -Lancer l'application. -Choisir d'ajouter une liste de prospects en important une liste déjà préparée . |

| | |
|-----------------------------|---|
| Scénario d'exception | «Problèmes dans le serveur des données». «Erreur Format JSON ». Aucune tache n'est effectuée. |
|-----------------------------|---|

Tableau 3-5 :création prospect

b-Cas d'utilisation : suppression de prospect

| | |
|-----------------------------|--|
| Titre | Suppression de prospect |
| Objectif | Permet le client de supprimer un ou plusieurs prospects. |
| Auteur | Client |
| Pré condition | -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing -Accès à la base de données des contacts. |
| Scénario nominal | -Lancer l'application. -Choix du prospect à supprimer. |
| Scénario alternatif | Aucun |
| Scénario d'exception | «Problèmes dans le serveur des données». «Erreur Format JSON ». Aucune tache n'est effectuée. |

Tableau 3-6 :suppression prospect

3.3.1.3 Lister les campagnes

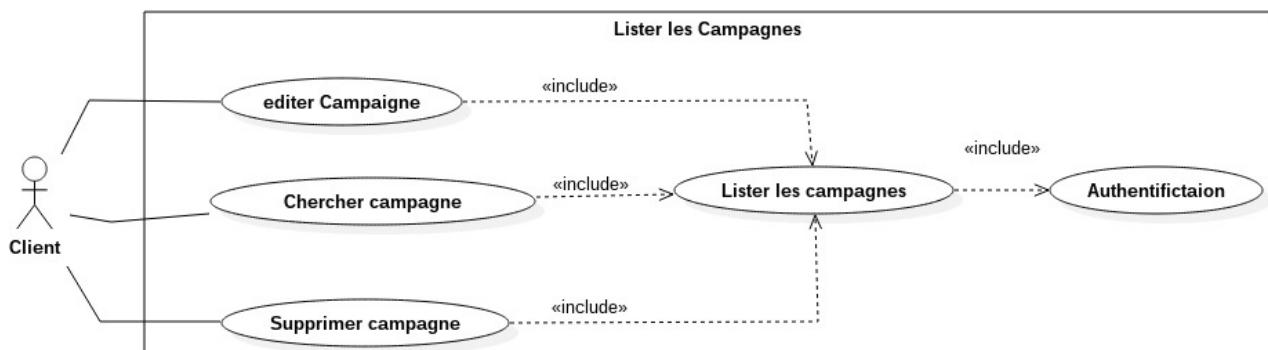


Figure 3-3 :Lister les campagnes

a. Cas d'utilisation : Lister les campagnes

| | |
|-------------------------|--|
| Titre | Lister les campagnes |
| Objectif | Permet le client de lister ses campagnes, leurs états ainsi que donner accès aux statistiques de chaque campagne, l'éditer ou la supprimer.. |
| Auteur | Client |
| Pré condition | -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. |
| Scénario nominal | -Lancer l'application. -Lister les campagnes existantes, avec leur état (créer, envoyées, tester , supprimer). |

| | |
|-----------------------------|---|
| Scénario alternatif | <ul style="list-style-type: none"> -Lancer l'application. -Lister les campagnes existantes. -Enregistrer le liste des campagnes en fichiers local (Csv, Pdf, Xls). -Filtrer les résultats des campagnes par nom, langages(Français, Anglais, Néerlandais), ou état de la campagne(Créer, Envoyer, Tester, Supprimer) -Ordonner la liste des campagnes selon le sujet de la campagne ou la date de création. |
| Scénario d'exception | <p>«Problèmes dans le serveur des données».</p> <p>«Erreur Format JSON ».</p> |

Tableau 3-7 :Lister les campagnes

b. Cas d'utilisation : Chercher Campagne

| | |
|-----------------------------|--|
| Titre | Chercher Campagne |
| Objectif | Permet le client de chercher une campagne selon leur sujet. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. |
| Scénario nominal | <ul style="list-style-type: none"> -Lancer l'application. -Lister la liste des campagnes existantes. -Saisir une chaîne de caractère dans la barre de recherche -Afficher les résultats si ils existent. |
| Scénario alternatif | Aucun |
| Scénario d'exception | <p>«Problèmes dans le serveur des données».</p> <p>«Erreur Format JSON ».</p> |

Tableau 3-8 :Chercher une campagne

c. Cas d'utilisation : Supprimer Campagne

| | |
|-----------------------------|--|
| Titre | Supprimer Campagne |
| Objectif | Permet le client de supprimer une campagne de la liste. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. |
| Scénario nominal | <ul style="list-style-type: none"> -Lancer l'application. -Lister la liste des campagnes existantes. -Supprimer une campagne de la liste. -Valider la suppression -La Suppression doit être une suppression logique et non pas physique. -Une redirection automatique vers la liste des campagnes avec une état «deleted». |
| Scénario alternatif | Aucun |
| Scénario d'exception | <p>«Problèmes dans le serveur des données».</p> <p>«Erreur Format JSON ».</p> |

Tableau 3-9 : supprimer campagne

d. Cas d'utilisation : Éditer une campagne

| | |
|-----------------------------|---|
| Titre | Éditer une campagne |
| Objectif | Permet le client d'éditer une campagne de la liste, on peut changer le sujet, le contenu, ainsi que les groupes destinataires. |
| Auteur | Client |
| Pré condition | -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. |
| Scénario nominal | -Lancer l'application. -Lister la liste des campagnes existantes. -Éditer une Campagne. -Redirection vers la page de création de campagne -Éditer les données de la page (le contenu, les groupes de destination ...) -la campagne s'afficher dans la liste des campagne selon l'état choisis |
| Scénario alternatif | Aucun |
| Scénario d'exception | «Problèmes dans le serveur des données». «Erreur Format JSON ». |

Tableau 3-10 :éditer une campagne

3.3.1.4 Créer une campagne

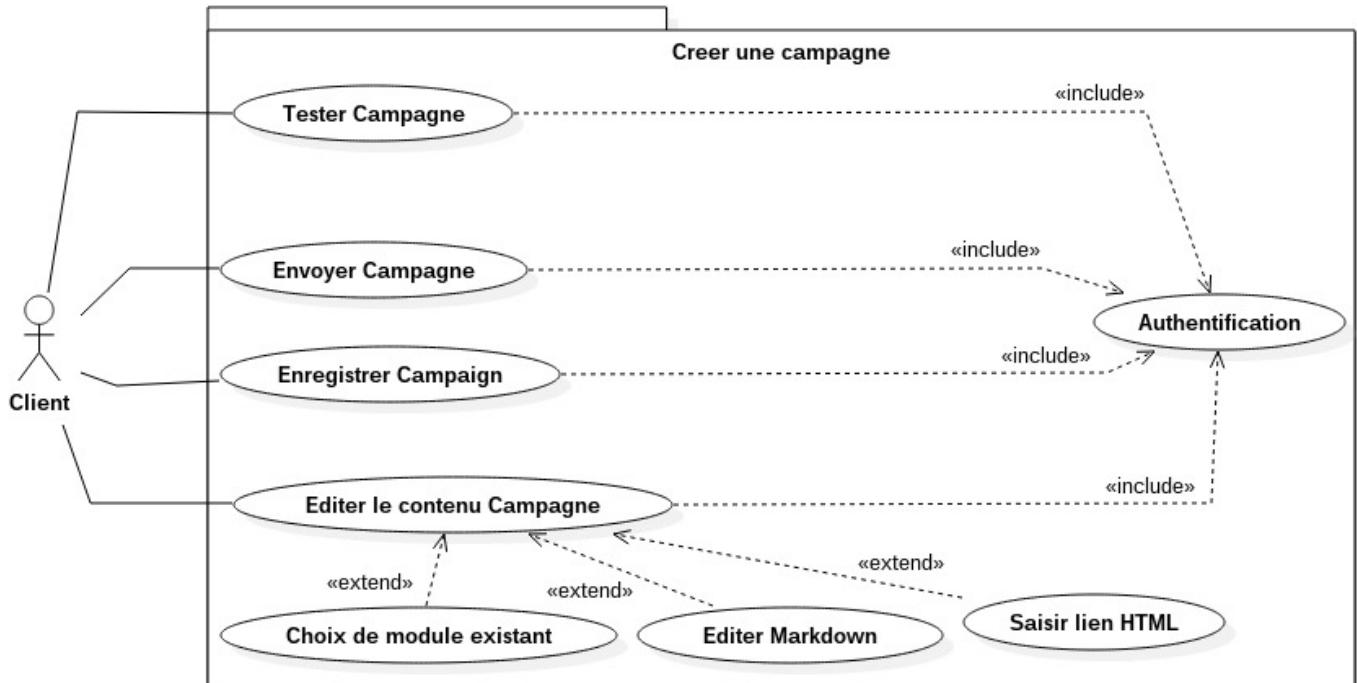


Figure 3-4 :Créer une campagne

a. Cas d'utilisation : Enregistrer une campagne

| | |
|-----------------------------|--|
| Titre | Enregistrer une campagne |
| Objectif | Permet le client d'enregistrer une campagne pour un usage ultérieur. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. |
| Scénario nominal | <ul style="list-style-type: none"> -Lancer l'application. -Choix type de campagne. -Enregistrer la campagne dans la liste des campagnes avec un état «Draft». -Redirection vers la liste des campagnes. |
| Scénario alternatif | <ul style="list-style-type: none"> -Lancer l'application. -Choix type de campagne. -Choix du sujet et la langue de la campagne. -Choix de groupe de destination. -Éditer le contenu de la campagne. -Enregistrer la campagne dans la liste des campagnes avec un état «Draft». -Redirection vers liste des campagnes. |
| Scénario d'exception | <ul style="list-style-type: none"> «Problèmes dans le serveur des données». «Erreur Format JSON ». |

Tableau 3-11 : Enregistrer une campagne

b. Cas d'utilisation : Tester une campagne

| | |
|-----------------------------|--|
| Titre | Tester une campagne |
| Objectif | Permet le client d'envoyer une campagne de test vers un email de test. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. |
| Scénario nominal | <ul style="list-style-type: none"> -Lancer l'application. -Choix type de campagne. -Choix du sujet et la langue de la campagne. -Éditer le contenu de la campagne. -Choix du mail de test. -Envoi de la campagne de test. -La campagne est enregistrée dans la liste des campagnes avec un état «Test». |
| Scénario alternatif | Aucun |
| Scénario d'exception | <ul style="list-style-type: none"> «Problèmes dans le serveur des données». «Erreur Format JSON ». «Erreur au serveur smtp». «Erreur Api Amazon». |

Tableau 3-12 :tester une campagne

c. Cas d'utilisation : Envoyer une campagne

| | |
|-----------------------------|---|
| Titre | Envoyer une campagne |
| Objectif | Permet le client d'envoyer une campagne. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. |
| Scénario nominal | <ul style="list-style-type: none"> -Lancer l'application. -Choix type de campagne. -Choix du sujet et la langue de la campagne. -Éditer le contenu de la campagne. -Choix du groupes inclus. -Valider l'envoi de la campagne. -La campagne est enregistré dans la liste des campagnes avec un état «SENT». -Redirection vers la liste des campagnes. |
| Scénario alternatif | <ul style="list-style-type: none"> -Lancer l'application. -Choix type de campagne. -Choix du sujet et la langue de la campagne. -Éditer le contenu de la campagne. -Choix du groupes inclus. -Choix du groupes exclus. -Exclude un destinataire spécifique. -Envoyer une campagne de test. -Valider l'envoi de la campagne. -La campagne est enregistré dans la liste des campagnes avec un état «SEND». -Redirection vers les statistiques. |
| Scénario d'exception | <ul style="list-style-type: none"> «Problèmes dans le serveur des données». «Erreur Format JSON ». «Erreur au serveur smtp». «Erreur Api Amazon». |

Tableau 3-13 :Envoyer une campagne

d. Cas d'utilisation : Éditer contenu campagne

| | |
|----------------------------|--|
| Titre | Éditer contenu de la une campagne |
| Objectif | Permet le client d'éditer le contenu à envoyer d'une campagne, selon le type de la campagne choisis. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. |
| Scénario nominal | <ul style="list-style-type: none"> -Lancer l'application. -Choix type markdown de la campagne. -Éditer un contenu markdown généralisé à tous les destinataires. |
| Scénario alternatif | <ul style="list-style-type: none"> -Lancer l'application. -Choix type web de la campagne. |

| | |
|-----------------------------|---|
| | -Choix d'un des modèles données par l'application et le visualiser, ces modèles sont offerts par le produit «Blog-Ezine». |
| Scénario d'exception | «Problèmes dans le serveur de donné». «Erreur Format JSON ». |

Tableau 3-14 : Éditer contenus d'une campagne

3.3.1.5 suspension d'envoi de la campagne

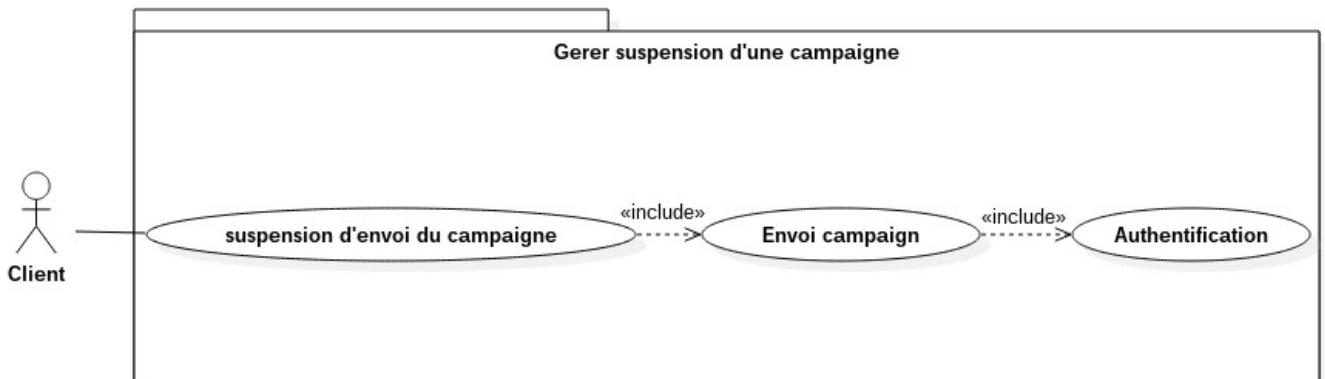


Figure 3-5 : diagramme cas d'utilisation pour suspension d'envoi de campagne

a. Cas d'utilisation : suspension d'envoi d'une campagne.

| | |
|-----------------------------|--|
| Titre | Suspendre l'envoi d'une campagne |
| Objectif | Permet le client de suspendre l'envoi de campagne. |
| Auteur | Client |
| Pré condition | -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. -Campagne en moment d'envoi. |
| Scénario nominal | -Lancer l'application. -Choix d'une campagne en état «Created» ou «Recipient_created». -Choix de suspendre l'envoi. |
| Scénario alternatif | Aucun |
| Scénario d'exception | «Problèmes dans le serveur des données». «Erreur Format JSON ». |

Tableau 3-15 : Suspendre l'envoi d'une campagne

3.3.1.6 Consulter statistique

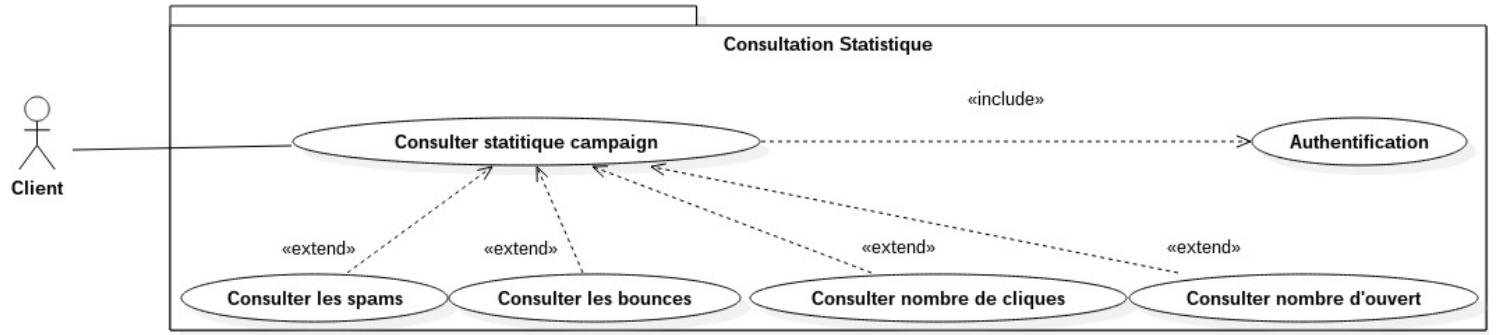


Figure 3-6 :diagramme cas d'utilisation pour la consultation des statistiques

a.Cas d'utilisation : suspension d'envoi d'une campagne.

| | |
|-----------------------------|---|
| Titre | Consultation des statistiques d'une campagne |
| Objectif | Permet le client de consulter les statistiques détaillées de chaque campagne. |
| Auteur | Client |
| Pré condition | <ul style="list-style-type: none"> -Authentification avec un compte client existant . -Privilège d'administrateur. -Activation du produit Emailing. -Accès à la base de données. -Campagne en état «SENT». |
| Scénario nominal | <ul style="list-style-type: none"> -Lancer l'application. -Afficher la liste des campagnes. -Consulter les statistiques d'une campagne en état «SENT». -Consulter le nombre total d'ouvertures de la campagne, ainsi que les destinataires qui l'ont ouvert. -Consulter le nombre total des clics aux liens de la campagne, ainsi que les destinataires qui les ont consultés. -Consulter le nombre total de de bounce (retour) de la campagne, ainsi que les destinataires qui sont concernés. -Consulter le nombre total de spam de la campagne, ainsi que les destinataires qui sont concernés. -Consulter le nombre total des mails délivrés, ainsi que les destinataires qui sont concernés. |
| Scénario alternatif | -Aucun. |
| Scénario d'exception | <ul style="list-style-type: none"> «Problèmes dans le serveur des données». «Erreur Format JSON ». «Erreur au serveur smtp». «Erreur Api Amazon». |

Tableau 3-16 : consultation des statistiques

3.3.2 Diagrammes de séquence

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique, dans cette partie on va définir les diagrammes de séquences pour mieux comprendre les cas d'utilisation des acteurs .

3.3.2.1 Création d'une campagne

Ce diagramme de séquence illustre les messages échangés entre : client , Système(qui inclut le backend l'interface graphique, et l'api Amazon SES) ainsi Contact et l'api Amazon SNS qui gère les statistiques lors d'un envoi d'une campagne à un contact, avec la possibilité de passer par une campagne de test avant l'envoi. Dans ce diagramme, on note aussi l'échange des données de statistique entre les différentes classes de notre système.

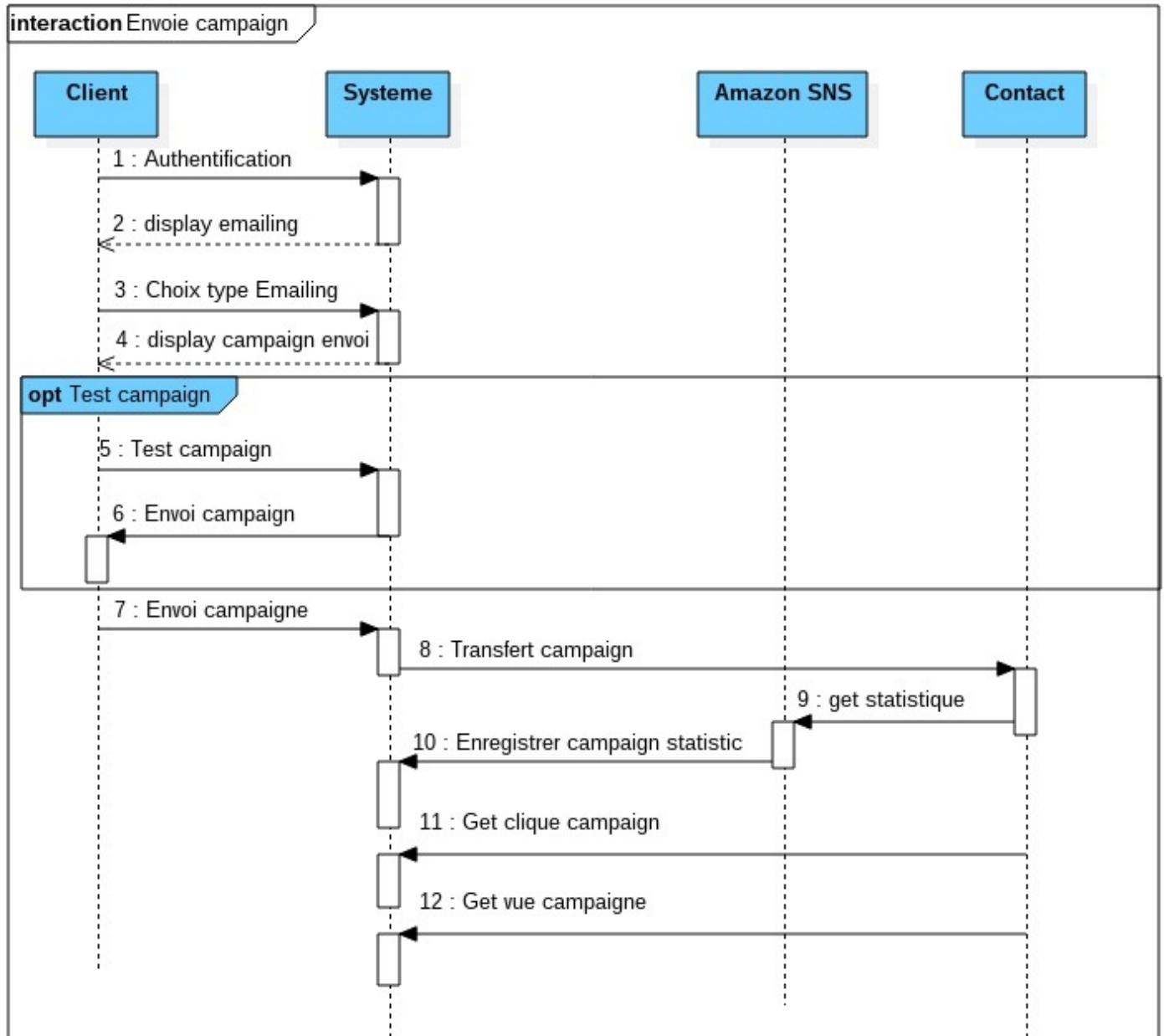


Figure 3-7 :diagramme de séquence de l'envoi d'une campagne

3.3.2.2 Affectation d'un contact à un groupe

Ce diagramme de séquence illustre les messages échangés entre les 3 classes,client, groupe et Contact lors d'affectation d'un contact à un groupe.

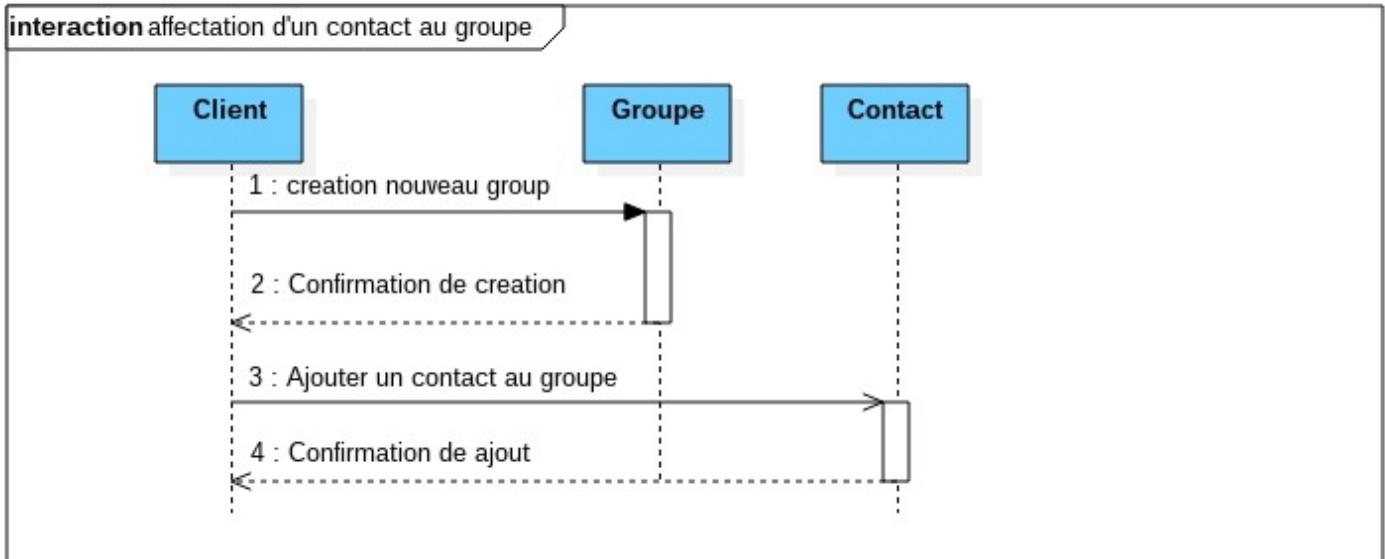


Figure 3-8 : diagramme de séquence d'affectation d'un groupe à un contact

3.3.2.3 Consultation des statistique d'une campagne

Ce diagramme de séquence illustre les messages échangés entre: client, l'interface graphique, API (backend) et le serveur des données affin de comptabiliser les statistiques enregistrées lors d'envoi de la campagne.

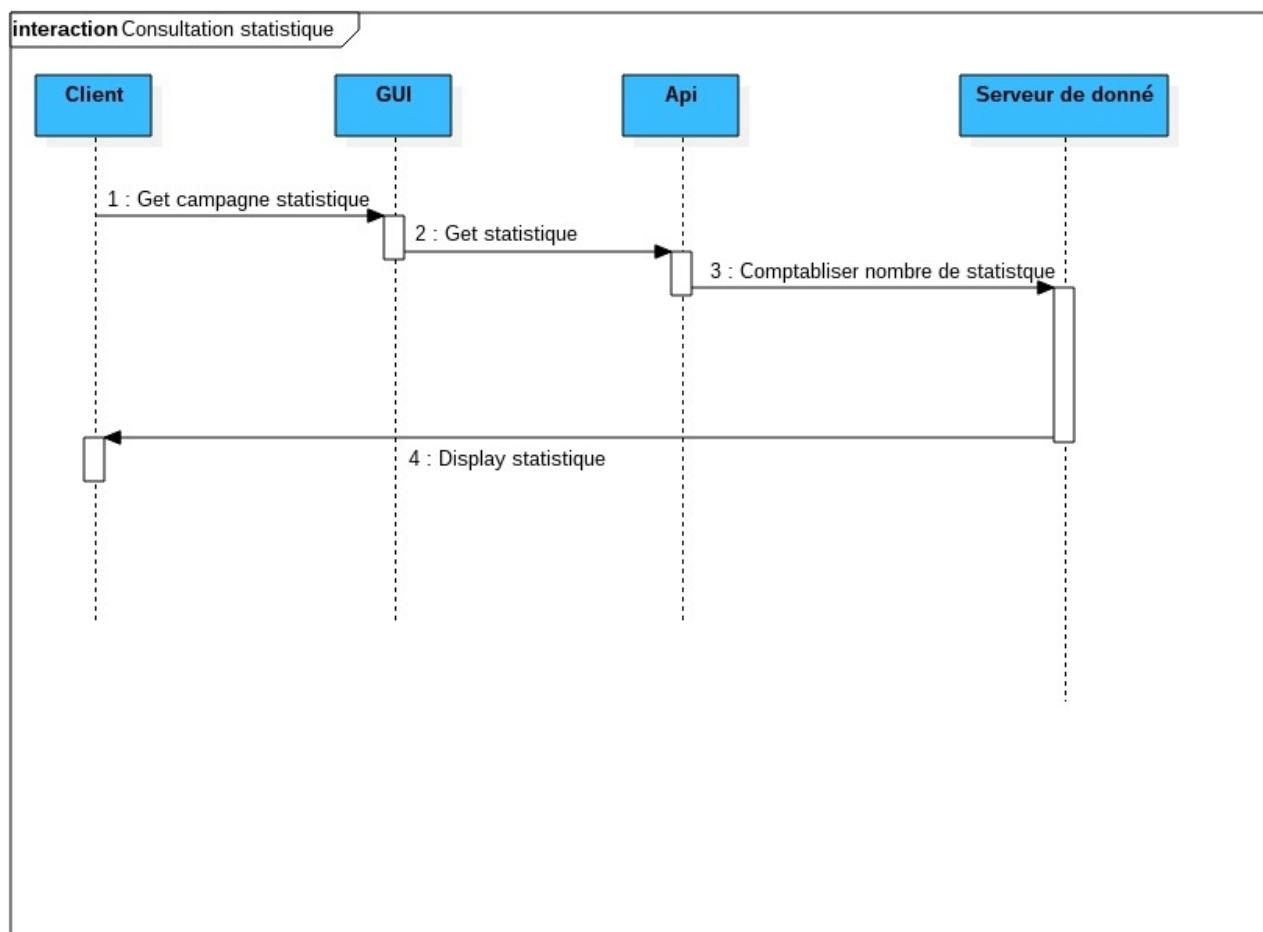


Figure 3-9 : diagramme de séquence de consultation des statistiques

3.3.2.4 Gestion des tentatives d'envoi de campagne

Ce diagramme de séquence explique les messages échangés entre notre api et l'api d'amazon SNS qui gère les statistiques selon l'algorithme d'exponentielle backoff, on détecte si l'erreur est transitoire, on set un temps maximale de réponse et on réessaye d'envoyer la campagne jusqu'au recevoir un accusé de réception ou atteindre le délai maximale.

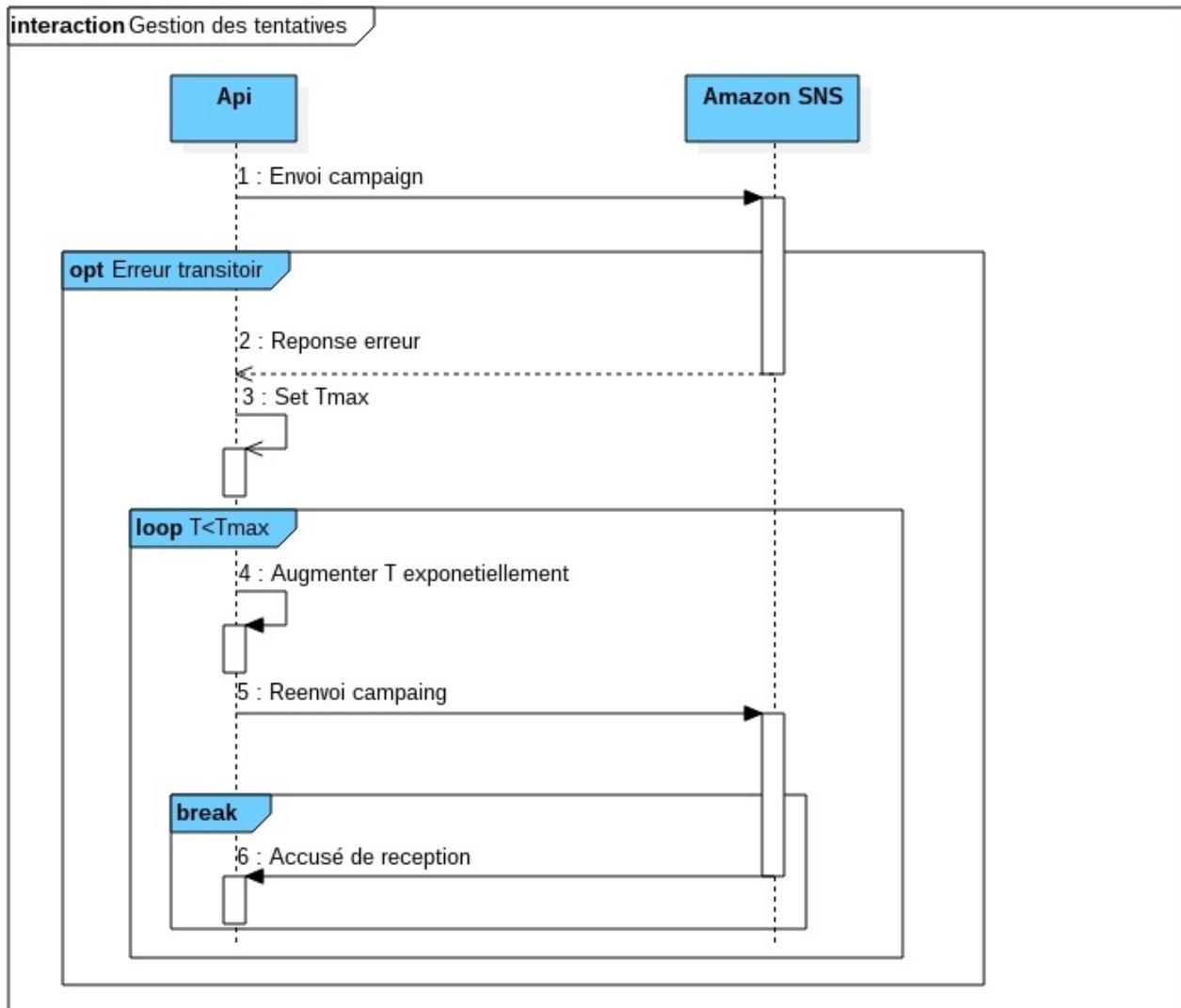


Figure 3-10 : diagramme de séquence de gestion des tentatives d'envoi

3.3.2.5 Choix de template générer par Blog-Ezine

Ce diagramme de séquence explique les messages échangés entre notre application Emailing et l'application Blog-Ezine, le produit Blog-Ezine génère des templates Html, notre application doit récupérer ces templates pour que le client choisisse une pour l'envoi de campagne.

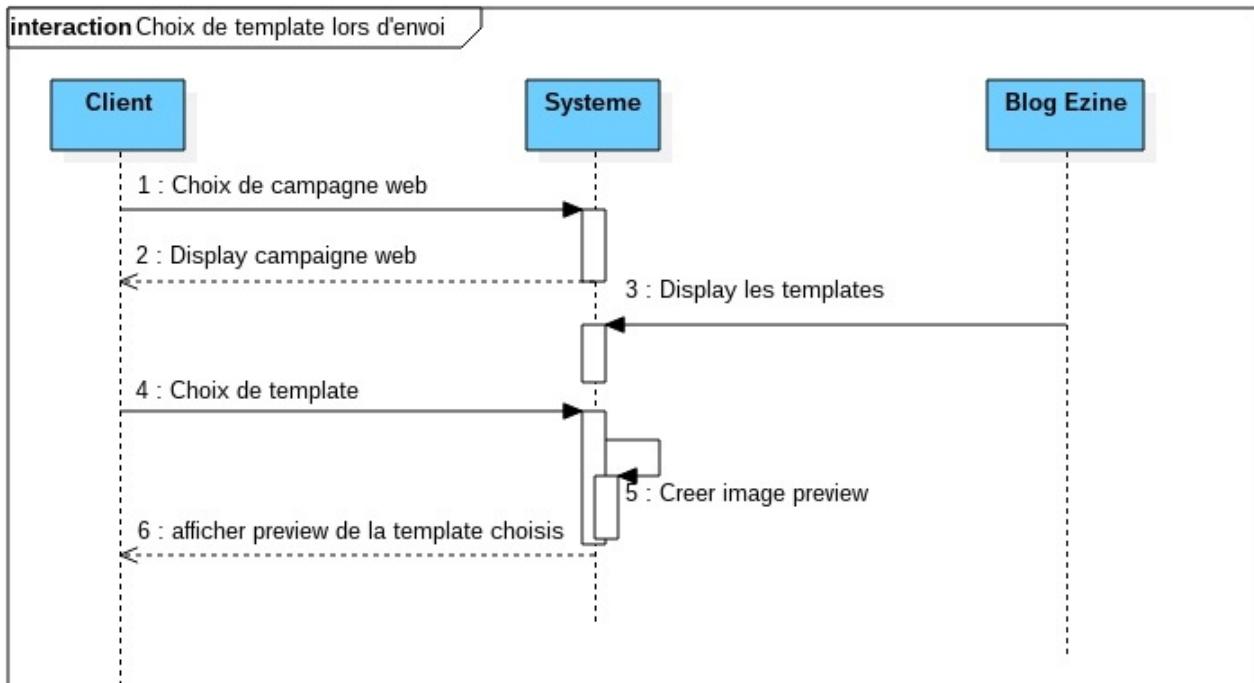


Figure 3-11 : diagramme de séquence de gestion des tentatives d'envoi

3.3.2.6 Gestion des prospects

Ce diagramme de séquence explique les messages échangés entre client et le système, les messages pour gérer les fonctionnalités CRUD du classe prospect.

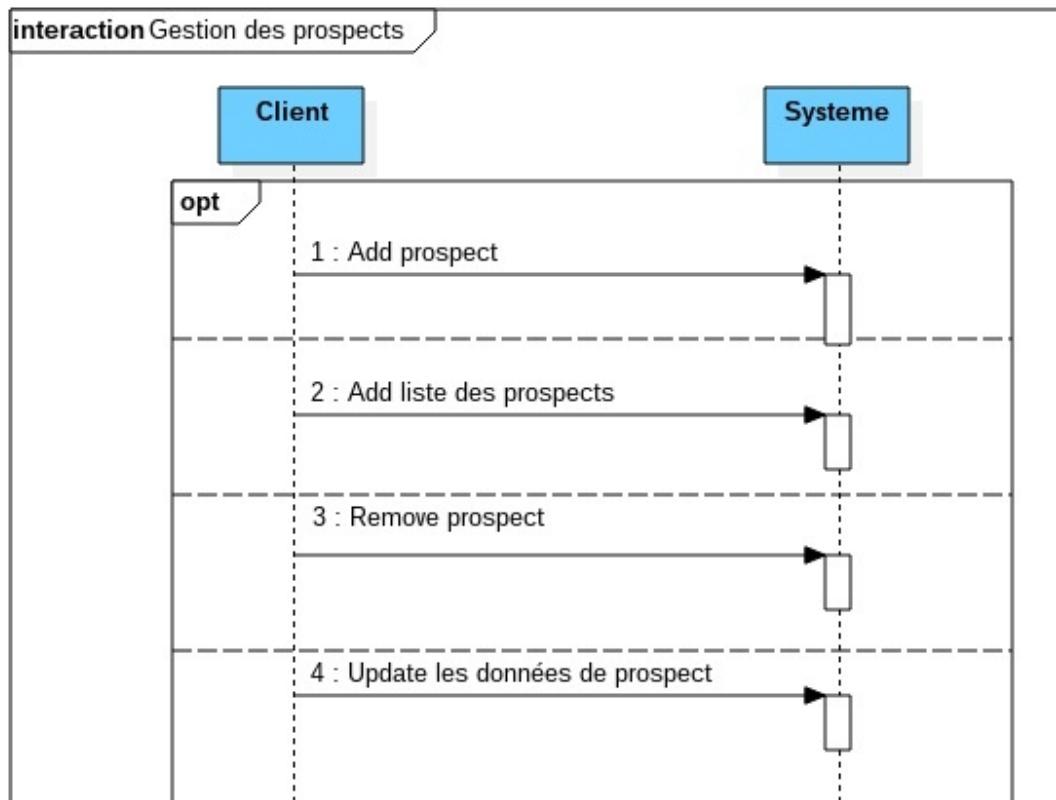


Figure 3-12 : diagramme de séquence de gestion des prospects

Exigences Non Fonctionnelles

Les besoins non-fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement.

En ce qui concerne notre application, nous avons dégagé les besoins suivants :

La disponibilité : l'application doit être disponible pour qu'elle soit utilisée à tous moments.

La fiabilité : les données fournies par l'application doivent être fiables et le système doit fonctionner sans défaillance.

Une solution ouverte et évoluée : l'application peut être améliorée par l'ajout d'autres modules pour garantir la souplesse, l'évolutivité et l'ouverture de la solution, ainsi, elle peut être améliorée en offrant un système de newsletters puissant.

Rapidité d'exécution : la rapidité d'un tel système est considérée primordiale, vu le grand nombre des mails envoyés par un client dans un temps très réduit, l'application doit aussi permettre de réduire le temps nécessaire pour créer une campagne, ainsi de réduire le temps de la recherche et des filtres de l'application.

Portabilité : l'application doit fonctionner sur toutes les plateformes, ainsi les navigateurs web les plus utilisés(Google Chrome, FireFox, Internet Explorer, Opera...).

Ré-utilisabilité : l'application peut être une base et un point de départ pour démarrer d'autre application, soit en utilisant les composantes, le code, le design....

Testabilité : les tests permettent d'augmenter la qualité de code, ainsi d'éliminer le risque de tomber sur des cas d'échecs ou l'application ne fonctionne plus correctement lors de la mise en production, il faut assurer des cas de test, des scénarios qui permettent de tester l'application avec toutes les façons possibles afin de détecter et corriger les bugs.

Ergonomie et convivialité des interfaces graphiques : l'application web doit fournir une interface conviviale et simple pour tout type d'utilisateur et vu qu'elle est une extension de la plateforme web, il faut donc développer des interfaces qui sont une projection de la plateforme pour que l'utilisateur se familiarise rapidement avec l'application.

Conclusion

Dans ce chapitre, nous avons présenté les différents acteurs du système, nous avons listé et expliqué chacune des fonctionnalités de l'application et on les a attribués aux acteurs, après nous avons dessiné les diagrammes des cas d'utilisation qui exploite ses données et les expliquent d'une façon explicite et pour chaque diagramme on a fait une description détaillée de ses objectifs et de ses scénarios.

Une fois les scénarios de l'application ont été définie, on a pu tracer un diagrammes de séquences des scénarios les plus pertinents.

Vers la fin de chapitre, on a listé les exigences non-fonctionnelles du système et dans le chapitre suivant nous allons présenter l'étude technique qui est fondamentale pour la mise en œuvre de notre application.

Chapitre IV :

Étude Technique

Introduction

4.1. Architecture physique

4.2. Architecture logique

4.2.1. Le langage de programmation

4.2.2. Outils de développement

4.2.3. Frameworks et technologies de développement

4.2.4. Architecture de la solution

4.2.5. Design Pattern

4.3. Amazon Web Services

4.3.1. Amazon web service SES

4.3.2. Amazon SNS Pub/Sub Messaging

4.3.3. Amazon Relational Database Service (RDS)

4.4. Pixel espion de tracking

Conclusion

Introduction

Ce chapitre traite les besoins techniques. Il sera consacré exclusivement à l'étude des moyens technologiques dont nous devons disposer, afin de mettre en œuvre l'application, et à l'analyse des prérequis techniques capables d'assurer la totalité des besoins fonctionnels énoncés.

Cette partie du rapport est nécessaire avec les besoins fonctionnels, dans la mesure où elle permet de prévenir les risques qui peuvent persister. Ainsi que recenser toutes les contraintes sur les choix dimensionnant la conception technique du système, les outils et les matériaux sélectionnés ainsi que la prise en compte des contraintes lors de l'intégration avec l'existant.

Architecture physique

L'architecture physique (également nommée architecture technique) décrit l'ensemble des composants matériels supportant l'application. La figure ci-dessous présente l'architecture physique de notre application :

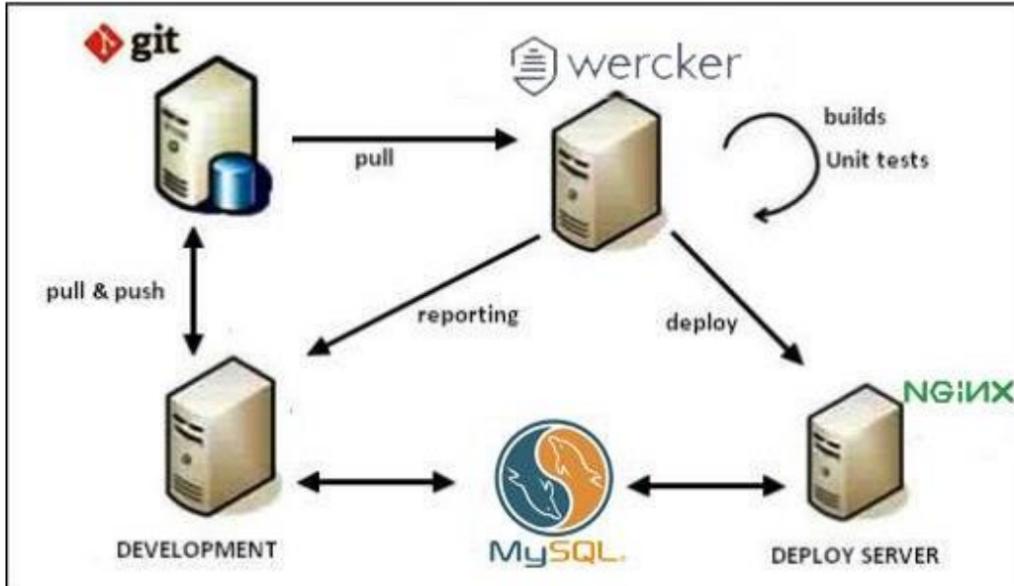


Figure 4-1 : Architecture physique de l'application

Ce tableau décrit les différents serveurs présents dans notre architecture physique :

| | |
|--|--|
| Serveur d'application PHP7(NGINX) | Contient l'application proprement dite et permet de communiquer avec les autres composants serveurs. |
| Serveur de base de données | Serveur qui contient la base de données de type MYSQL. |
| Wercker | permet d'automatiser les builds, le lancement des tests unitaires, les déploiements de manière périodique ou ponctuelle afin de repérer rapidement les problèmes |
| Serveur de gestion des sources | Garde en mémoire toutes les versions committées. Il offre ainsi la possibilité de revenir rapidement à un moment particulier du développement |

Tableau 4-1 : Les composants serveurs

La virtualisation est devenue un standard en informatique, mais une virtualisation légère avec DOCKER reste une approche différente des solutions classiques.

Docker: est un logiciel open source permettant la mise en œuvre de containers s'exécutant en isolation, un container Docker, à l'opposé de machines virtuelles traditionnelles, ne requiert aucun système d'exploitation séparé et n'en fournit aucun. Il s'appuie plutôt sur les fonctionnalités du système d'exploitation fournies par l'infrastructure sous-jacente. Il ne contient, au maximum, que les applications/librairies.

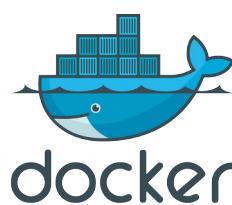


Figure 4-2 : Logo Docker

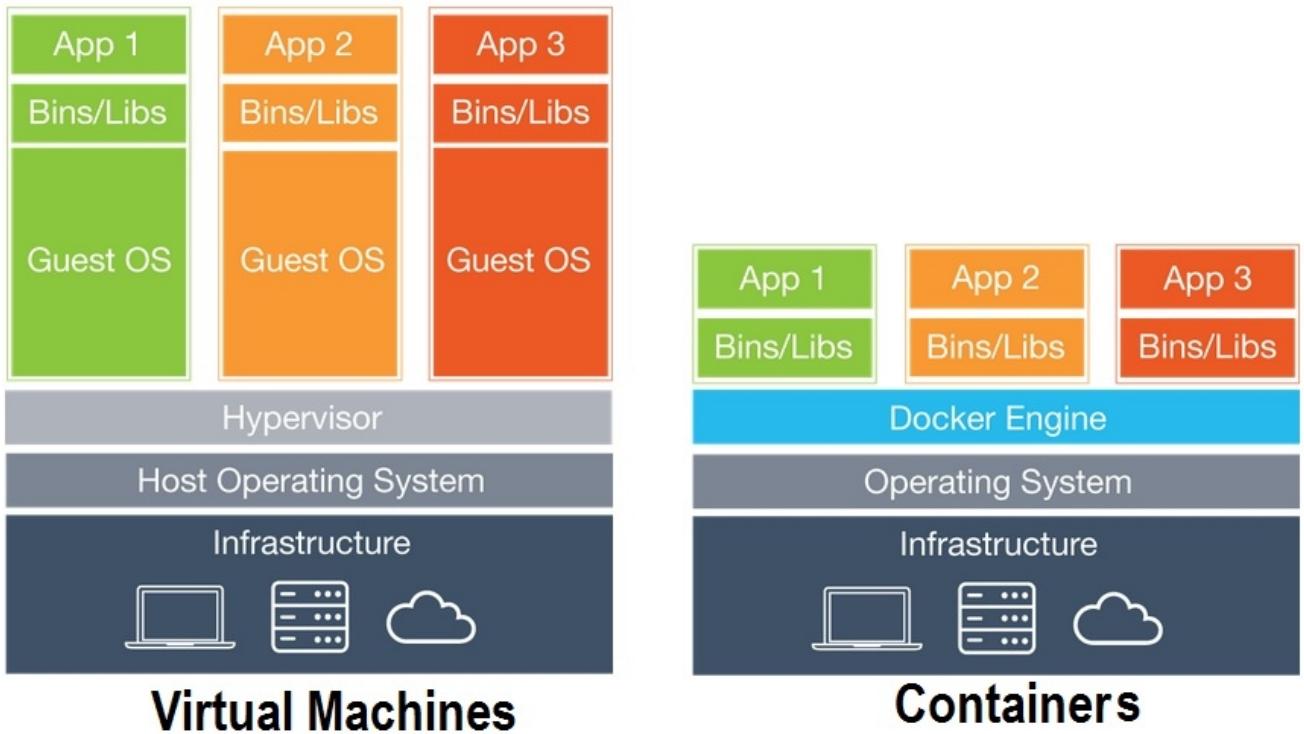


Figure 4-3 : Machine virtuelle Vs conteneur Docker

Le système hôte est ainsi géré directement par le Docker Engine (en bleu dans le schéma). On a ainsi une mutualisation qui permet de lancer les conteneurs de manière très rapide.

On dispose tout de même d'un environnement isolé (processus, système de fichier, ports réseau).

L'autre grand avantage de cette technologie est la portabilité, possibilité de déployer le conteneur qui est de taille relativement réduite sur son infrastructure de production physique ou virtuelle.

L'annexe I illustre plus d'informations sur cette technologie et l'avantage majeur d'exploiter cette solution dans notre projet .

Ci-dessous les conteneurs Docker pour notre projet :

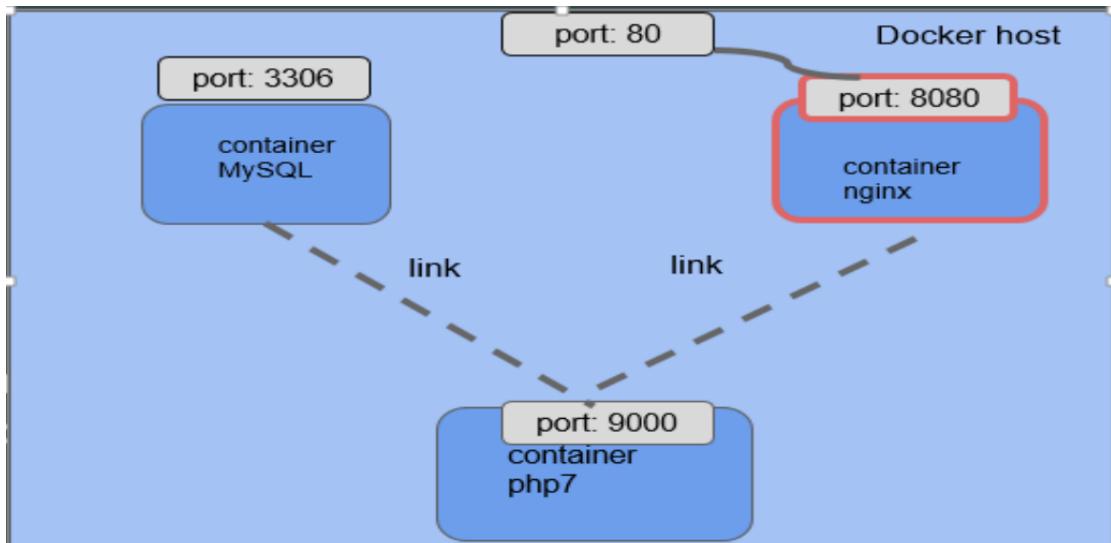


Figure 4-5 : Les conteneurs Docker du projet

Architecture logique

4.2.1 Le langage de programmation

Pour ce qui est des langages de programmation, la différence est très nette. En effet, PHP7 est plus rapide de 243% par rapport à Ruby 2.1, 400% par rapport à Python 2.7.8 et 740% par rapport à Perl 5.18.4.

Les gains en performances qu'apporte donc cette nouvelle version sont énormes, de plus, elle utilise beaucoup moins de ressources ce qui permettra d'héberger beaucoup plus de sites web sur le même serveur.



Figure 4-6 : Logo php7

On a décidé d'utiliser PHP7 comme langage de programmation pour notre système puisque c'est le plus adaptable à nos attentes et nos exigences déjà fixées.

L'annexe II illustre les raisons pour lesquelles on a abordées ce choix.

4.2.2 Outils de développement

On présente dans ce qui suit les différents Frameworks et technologies que nous avons utilisées :

4.2.2.1 Issues Tracker

Pour mettre en place la méthodologie de développement choisie (Scrum) On a décidé d'utiliser JIRA (projet développé par Atlassian) parce que c'est un produit mûr et très puissant puisque c'est parmi les premiers qui existent sur le marché, donc un support plus présent et une communauté assez réactive qui ne cesse d'améliorer son produit.

JIRA est un outil de gestion des projets agiles qui permet :



Figure 4-7 : Logo Jira

- ◆ La gestion du support et des activités de développement logiciel.
- ◆ La gestion des différents workflow d'une tâche (in progress, Pull Request, done/resolved...).
- ◆ Le suivi des anomalies.
- ◆ Le suivi d'activité.

4.2.2.2 Git Hub & Commit/Push

Git Hub est un dépôt social pour les projets open-source basé sur le système de contrôle de version Git, et sa fonction d'un point est de partager du code facilement et de collaborer sur des projets. Et d'autre point on évite toute sorte de conflits et on crée une traçabilité détaillée et atomique du travail.

Afin d'arranger l'ensemble des commit sur notre outils JIRA on a choisis De préférence que le message de commit doit être lié à une tâche bien précise selon son code sur notre interface JIRA.



Figure 4-8 : Logo Github

4.2.2.3 Wercker et L'approche DevOps

Le but est d'automatiser au maximum les tâches récurrentes, donc, après chaque push , il y'a une série de tâches à lancer (build) et c'est ça le fonctionnement de Wercker.

Wercker est une plateforme de déploiement continu qui permet de builder et de déployer des applications. En effet, Wercker est une plateforme ouverte, il est donc possible de définir un box, c'est à dire, la stack sur laquelle lancer le build. Dans notre projet notre box est TamTam/php7, pour mettre en place Il suffit d'ajouter un fichier wercker.yml dans la racine du projet .

Wercker vous fournit une plate-forme de cloud pour accélérer le processus de développement .



Figure 4-9 : Logo Wercker

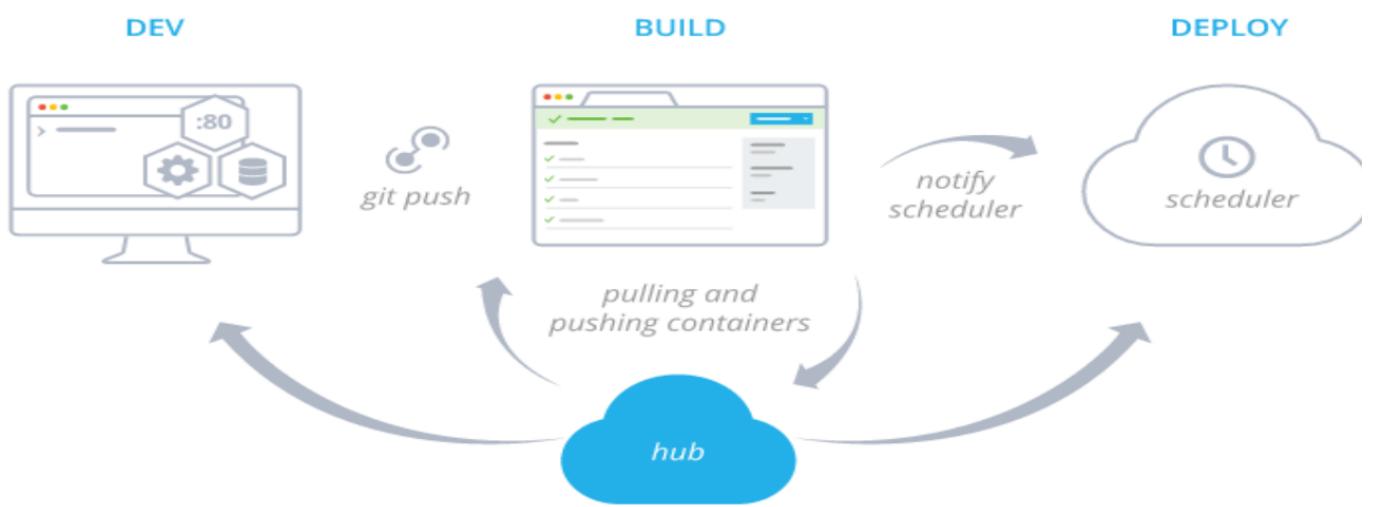


Figure 4-10 : plateforme de déploiement continu pour builder et de déployer les applications

4.2.2.4 Workflow des développement continu

Le schéma ci-dessous représente le workflow de déploiement continu avec wercker

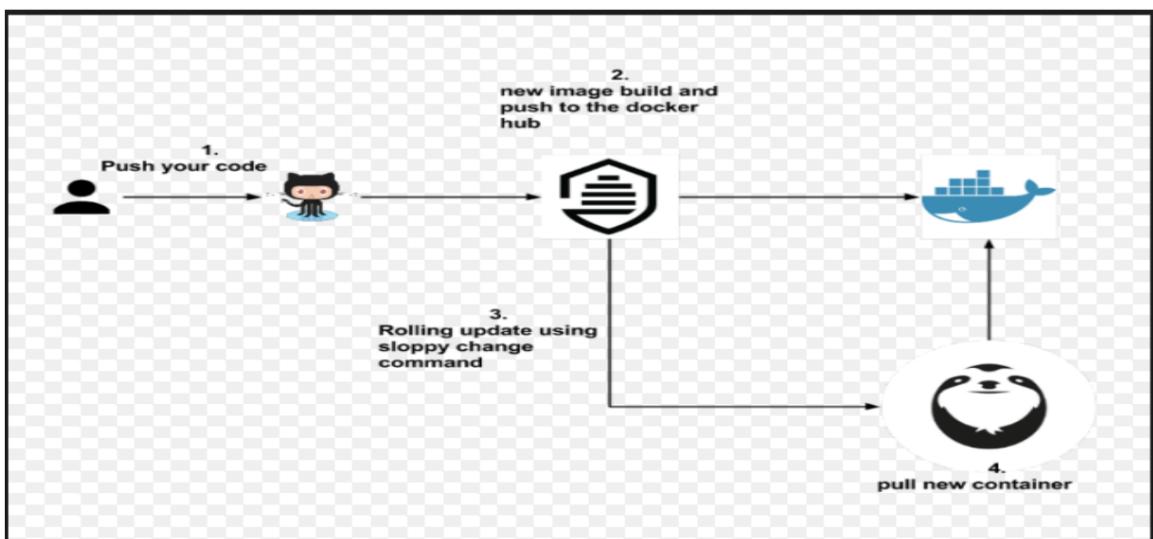


Figure 4-11 : workflow de déploiement continu

4.2.3 Frameworks et technologies de développement

On présente dans ce qui suit les différents Framework et technologies que nous allons utiliser :

4.2.3.1 Symfony 3.2

Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre créé par L'agence web française SensioLabs écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

Symfony est donc un Framework PHP. Bien sûr, il en existe d'autres ; pour ne citer que les plus connus : Zend Framework, CodeIgniter, CakePHP, Laravel..., Symfony 3 est l'un des plus flexibles et des plus puissants, il est très populaire. C'est un des Framework les plus utilisés dans le monde, notamment dans les entreprises. Il est utilisé par Dailymotion par exemple ! La première version de Symfony est sortie en 2005 et est aujourd'hui toujours très utilisée. Cela lui apporte un retour d'expérience et une notoriété exceptionnelles. Aujourd'hui, beaucoup d'entreprises dans le domaine de l'Internet recrutent des développeurs capables de travailler sous ce Framework. Ces développeurs pourront ainsi se greffer aux projets de l'entreprise très rapidement, car ils en connaîtront déjà les grandes lignes.

C'est un atout si vous souhaitez travailler dans ce domaine.

Architecture de base du Framework

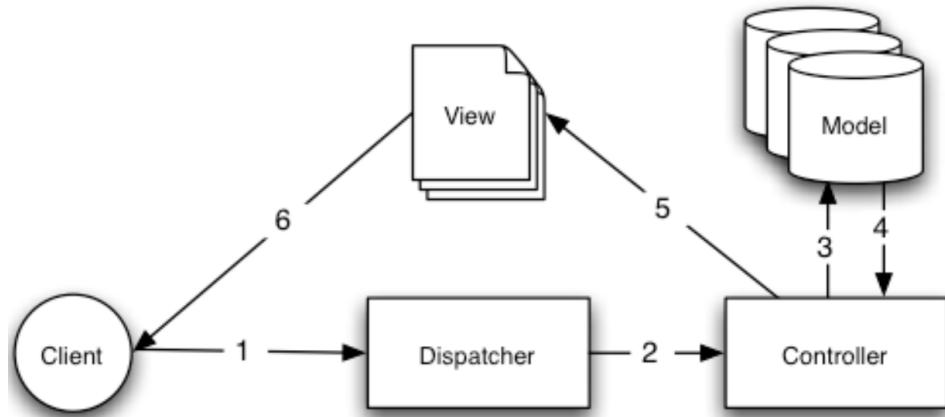


Figure 4-13 : Architecture de base de Symfony

4.2.3.2 PHPUnit 6.1

Afin d'améliorer la qualité du projet on a adopté une méthode qui consiste à faire des tests unitaires et fonctionnels pour chaque fonctionnalité pour cela on utiliser PHPUnit et le principe Un test unitaire doit être véritablement unitaire. Le TU doit s'exécuter le plus rapidement possible afin d'avoir un retour quasi immédiat. Un TU faisant partie intégrante du code applicatif, les pratiques suivantes sont recommandées :

- ◆ Il doit respecter les conventions de code .
- ◆ Il doit être simple et lisible.
- ◆ Il ne doit tester qu'un seul comportement à la fois.
- ◆ Il doit faire le moins d'appel possible aux dépendances.



Figure 4-12: Logo Symfony



Figure 4-14: Logo phpunit

L'approche TDD (Test Driven Development) reste la meilleure solution pour éviter que les tests soient écrits à la fin du développement de l'application.

4.2.3.3 React Js 15.4

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.



Figure 4-15 : Logo React

React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou framework. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.

L'annexe III illustre les raisons pour lesquelles on a abordées ce choix.

4.2.3.4 Sass

Sass est un langage de génération dynamique de feuilles de style initialement développé par Hampton Catlin et Nathan Weizenbaum. Sass un métalangage de feuilles de style en cascade. C'est un langage de script qui est interprété en CSS. SassScript est le langage de script lui-même.

Deux syntaxes existent. La syntaxe originale, nommée « syntaxe indentée », est proche de Haml. La nouvelle syntaxe se nomme « SCSS ».



Figure 4-16 : Logo Sass

Elle a un formalisme proche de CSS. Sass peut être étayé avec Compass (pratique pour les préfixes des différents navigateurs).

4.2.3.5 DraftJs

Draft.js est un framework pour la construction de rédacteurs de texte riches dans React, alimenté par un modèle immuable et abstraction des différences entre les navigateurs.



Figure 4-17 : Logo DraftJs

Draft.js facilite la création de n'importe quel type d'entrée de texte enrichi, que vous cherchez simplement à supporter quelques styles de texte en ligne ou à créer un éditeur de texte complexe pour composer des articles longs.

Dans Draft.js, tout est personnalisable, nous fournissons les blocs de construction afin d'avoir un contrôle total sur l'interface utilisateur.

4.2.3.6. Foundation 6

Foundation est responsive open source front-end framework, La fondation est un cadre frontal prépondérant. Foundation fournit une grille réactive et des composants, des modèles et des extraits de code HTML et CSS UI, y compris la typographie, les formulaires, les boutons, la navigation et d'autres éléments d'interface, ainsi que les fonctionnalités facultatives fournies par les extensions JavaScript. La base est maintenue par ZURB.

Foundation a été conçu et testé sur de nombreux navigateurs et périphériques.



Figure 4-18 : Logo Foundation

C'est un premier référentiel mobile conçu avec Sass / SCSS qui donne aux concepteurs des pratiques exemplaires pour un développement rapide. Le cadre comprend les modèles les plus communs nécessaires pour prototyper rapidement un site réactif. Grâce à l'utilisation de Sass Mixins, les composants de la Fondation sont faciles à concevoir et simples à étendre

4.2.3.7. PhantomJS

PhantomJS est un navigateur web navigateur sans tête (sans interface graphique) scriptable utilisé pour automatiser des interactions avec des pages web. PhantomJS offre une interface de programmation JavaScript permettant la navigation automatisée, la capture d'écran, les comportements utilisateurs, et l'utilisation d'assertions.



Figure 4-19 : Logo phantomJS

Ces possibilités en font un outil de choix pour exécuter des tests fonctionnels au sein d'un environnement d'intégration continue. PhantomJS est basé sur Webkit, ce qui en fait un environnement de navigation similaire à Safari ou à Google Chrome (bien que ce dernier utilise désormais le moteur de rendu Blink, un fork de Webkit). PhantomJS est un programme open source distribué sous licence BSD.

4.2.3.8 Swiftmailer

PHPMailer a été pendant longtemps une classe très utile pour réaliser l'envoi d'emails. Mais délaissée (même si une nouvelle équipe semble avoir repris le projet en main) et victime de failles de sécurité, elle ne s'avérait plus fiable et pertinente.

Swift Mailer se pose en alternative plus que concluante à PHPMailer. Comme PHPMailer, il s'agit d'une librairie PHP, et comme PHPMailer, elle permet de configurer et de gérer l'envoi d'emails.

Basiquement Swift Mailer propose de personnaliser son message email, notamment l'objet, le(s) destinataire(s), l'expéditeur, le contenu HTML et l'alternative texte, des pièces jointes, ...



figure 4-20 : Logo SwiftMailer

Mais Swift Mailer dispose également de différentes fonctionnalités avancées qui permettent de gérer de manière extrêmement précise la construction et l'envoi d'un email. Il est ainsi possible de personnaliser entièrement le header de son message en rajoutant de nouvelles lignes. Et à l'instar de PHPMailer, différentes classes de transport sont disponibles, pour préciser le protocole (SMTP, SMTP sécurisé, Sendmail, ...) et le serveur (local ou distant) à utiliser pour envoyer son message. Bien sûr, Swift Mailer bénéficie d'un réglage par défaut qui permet d'occulter tous ces paramétrages si ils ne sont pas utiles.

4.2.3.9 RestApi

REST décrit un style d'architecture logicielle permettant de construire une application devant fonctionner sur des systèmes distribués, typiquement internet.

REST est un style d'architecture, pas un standard. Il n'existe donc pas de spécifications de REST. Il faut comprendre le style REST et ensuite concevoir des applications ou des services Web selon ce style.

Bien que REST ne soit pas un standard, il utilise des standards. En particulier :



figure 4-21 : Logo RestApi

- ◆ URI comme syntaxe universelle pour adresser les ressources,
- ◆ HTTP un protocole sans état (stateless) avec un nombre très limité d'opérations,
- ◆ Des liens hypermédia dans des documents (X)HTML et XML pour représenter à la fois le contenu des informations et la transition entre états de l'application,
- ◆ Les types MIME comme text/xml, text/html, image/jpeg, application/pdf, video/mpeg
- ◆ pour la représentation des ressources.

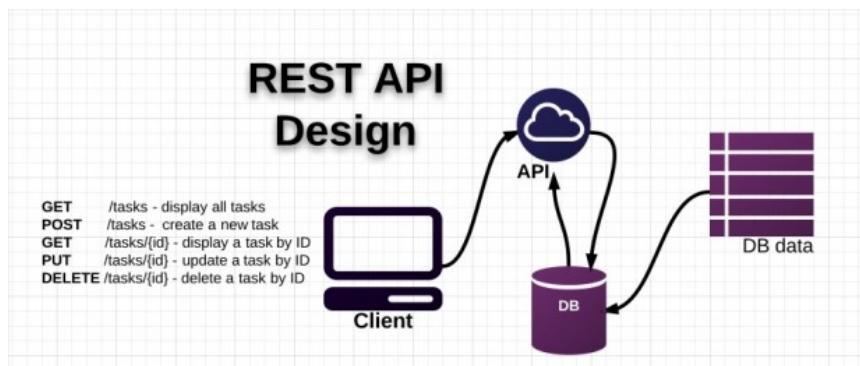


Figure 4-22 : architecture de projet avec le design «Api Rest»

4.2.3.10 Doctrine

Doctrine est un ORM (couche d'abstraction à la base de données) pour PHP. Il s'agit d'un logiciel libre sous licence GNU LGPL . Doctrine est l'ORM par défaut du framework Symfony (depuis la version 1.3 de ce framework). Cependant son utilisation dans le cadre d'un projet développé avec Symfony est optionnelle.

Doctrine "ORM" (Mapping Objet-Relationnel ou Object Relational Mapping) se base sur Doctrine "DBAL" (Couche d'abstraction de base de données ou DataBase Abstraction Layer) qui également s'appuie sur PDO (PHP Data Objects). Notez que PDO est activé par défaut depuis la version PHP 5.1 et ne fait pas partie de Doctrine.



Figure 4-23 : Logo doctrine

Le Doctrine Query Langage (DQL) est un langage de requête orienté objet propre à Doctrine. Il peut être utilisé à la place du langage SQL pour créer les requêtes d'accès et de manipulation des données de la base de données. Sa syntaxe s'inspire du Hibernate Query Langage, le langage de requête d'Hibernate

4.2.4 Architecture en couches

Le développement de l'application est basé sur le Framework SYMFONY3 MVC est un Framework PHP pour les applications 3-tiers.

L'architecture 3-tiers est une extension du modèle client/serveur, elle sépare trois couches logicielles au sein d'une même application : la couche présentation, la couche métier et la couche accès aux données.

L'applicatif réalisé est bâti sur une architecture multicouches telle que la suivante:

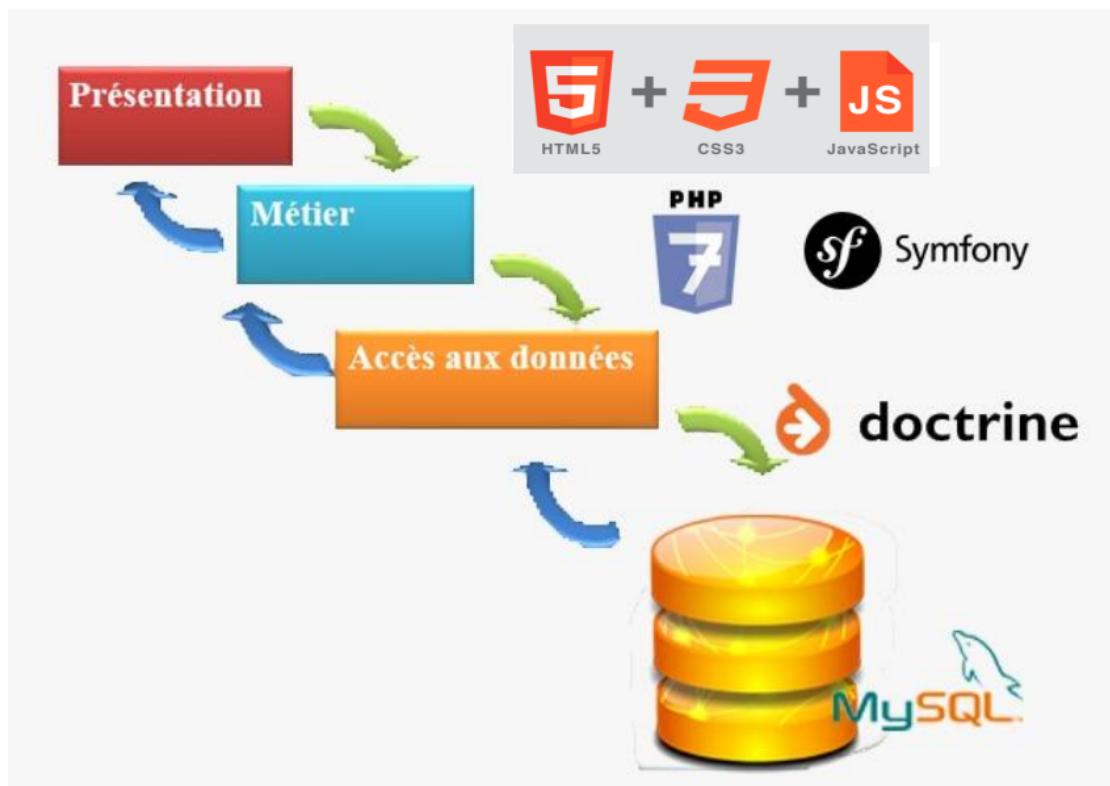


Figure 4-24 :Architecture et outils de la solution

4.2.4.1 La couche Repository

Dans une architecture en couche, il est important de prévoir une couche dédiée aux accès aux données, qui englobe les opérations CRUD.

Cette couche est responsable de la persistance de données gérée par l'ORM Doctrine qui offre en plus de son mapping relationnel et objet, l'intégration avec Symfony.

Afin de tester la couche Repository, les tests unitaires ont été réalisés par les Framework PHPUnit et SQLite.

4.2.4.2 La couche Service

La couche Service implémente les règles de gestion, elle s'appuie sur la couche Repository pour effectuer des opérations CRUD sur des objets persistés et leurs appliquer ensuite des traitements métiers.

4.2.4.3 La couche Web

La couche Web est la couche en contact avec l'utilisateur de l'application. Celui-ci interagit avec l'application au travers de pages Web.

L'utilisateur envoie une requête qui sera interprétée par Kernel ce dernier contacte un système de routage afin de faire appel au bon contrôleur, le contrôleur à son tour construit et envoie le Model, qui va remplir les parties dynamiques de la vue. Une fois le Controller reçoit le Model, il fait appel à la vue à rendre à l'utilisateur comme réponse.

4.2.5 Designs pattern

Les choix techniques qui ont été faits ont permis la mise en œuvre d'un certain nombre de patrons de conception.

Nous présentons dans ce qui suit les principes de chacun d'eux en mettant l'accent sur le plus qu'ils apporteront à notre projet.

4.2.5.1 Model View Controller 2 (MVC2)

Le MVC est un modèle de conception qui repose sur la volonté de séparer les données, les traitements et la présentation. Ainsi l'application se retrouve segmentée en trois composants essentiels :

Le modèle : représente le modèle de données.

La vue : correspond à l'IHM. Elle présente les données et interagit avec l'utilisateur.

Le contrôleur : se charge d'intercepter les requêtes de l'utilisateur, d'appeler le modèle puis de rediriger vers la vue adéquate

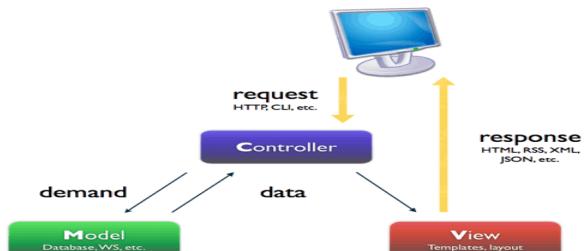


Figure 4-26 : diagramme MVC

Le MVC très pratique, peut se révéler lourd à mettre en place. Ceci à cause de la multitude de contrôleurs à implémenter. Afin de simplifier sa réalisation, une nouvelle version a été introduite : **le MVC2**.

C'est exactement le même modèle de conception à la différence qu'il n'y a plus qu'un seul contrôleur qui se charge de rediriger la requête vers le bon traitement.

4.2.5.3 Redux

Redux est un nouveau paradigme de programmation pour les interfaces Web. Il propose de centraliser l'intégralité du contexte de l'application en un seul endroit organisant toutes les transformations de façon séquentielles et reproductible.



Figure 4-27: Logo Redux

Action: Les actions sont des payloads d'informations qui envoient des données de notre application à notre Store. Ils sont la seule source d'information pour le store

Store: un grand java script objet, il contient les propriétés de l'application et il se comporte comme multiple store. Il ne change jamais durant le cycle mais on crée des nouvelles versions de ce store (selon les propriétés).

Le changement de ces propriétés implique le changement des composants dans l'interface graphique.

Components : Les composants nous permettent de diviser l'interface utilisateur en pièces indépendantes et réutilisables et de penser à chaque pièce isolée.

Conceptuellement, les composants sont comme des fonctions JavaScript.

Ils acceptent des entrées arbitraires (appelées «props») et renvoient les éléments React qui décrivent ce qui devrait apparaître sur l'écran. Dans le workflow de Redux, les composants lancent (dispatch) des actions.

Reducers: il reçoit une action à modifier de «Action», et il met à jour la propriété dans le store au lieu d'avoir multiple store (comme dans le Flux) on a ici multiples Reducers qui modifient les pieces de data, chaque reducer interagissant avec une des propriétés du Store.

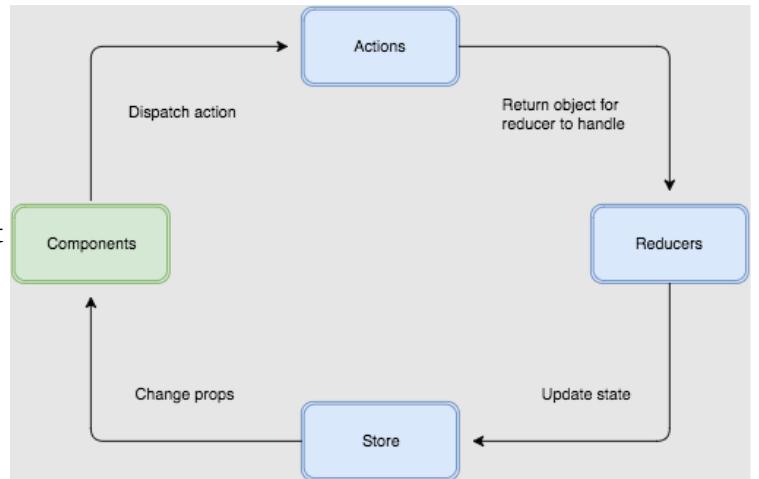


Figure 4-28 : workflow of redux

Voici un schéma qui illustre l'utilisation du Redux avec l'application et l'API en backend

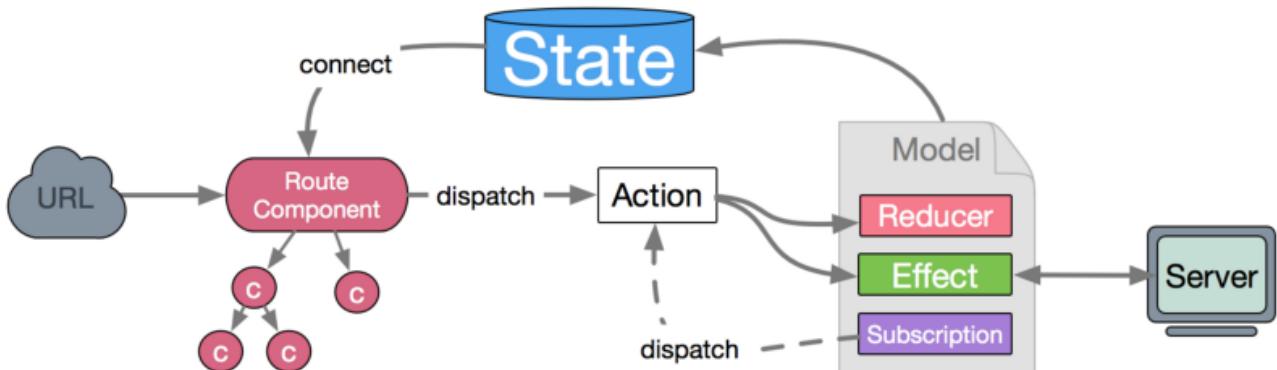


Figure 4-29 : architecture de Redux liant le côté serveur au côté client

4.2.5.4 Adapter

Le design pattern adaptateur fonctionne comme un pont entre deux interfaces incompatibles. Ce type de modèle de conception est soumis à un modèle structurel car ce modèle combine la capacité de deux interfaces indépendantes.

Ce pattern implique une classe unique qui est responsable d'associer des fonctionnalités d'interfaces indépendantes ou incompatibles. Un exemple de vie réelle pourrait être un cas de lecteur de carte qui agit comme un adaptateur entre une carte mémoire et un ordinateur portable. Vous connectez la carte mémoire au lecteur de carte et au lecteur de carte dans l'ordinateur portable afin que la carte mémoire puisse être lue via un ordinateur portable.

Un pattern adaptateur peut être utilisé lorsque l'emballage «wrapper» doit respecter une interface particulière et doit supporter un comportement polymorphe. Alternativement, un décorateur permet d'ajouter ou de modifier le comportement d'une interface au moment de l'exécution, et une façade est utilisée lorsqu'une interface plus facile ou plus simple à un objet sous-jacent est souhaitée.

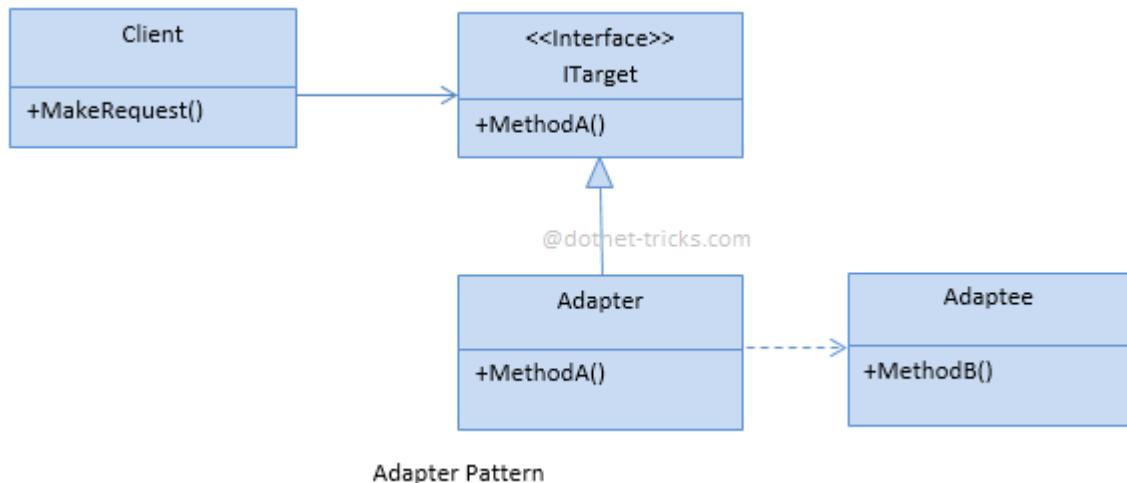


Figure 4-30 : diagramme de classe de design pattern adaptateur

Amazon Web Services

Amazon Web Services (AWS) est une division du groupe américain de commerce électronique Amazon.com, dédiée aux services de cloud computing pour les entreprises et particuliers. En 2015, AWS génère 7% du chiffre d'affaires d'Amazon et représente la première source de profit de l'entreprise. Amazon Web Services occupe le premier rang des acteurs du cloud computing, avec 31% de parts de marché, distançant largement des acteurs informatiques traditionnels comme Microsoft (11%) ou IBM (7%).



Figure 4-31 : logo amazon web service

Parmi ses grands clients figurent la NASA, Netflix et le service de renseignement extérieur américain (CIA).

Les offres Amazon Web Services sont accessibles en HTTP, sur architecture REST et par le protocole SOAP. Tout est facturé en fonction de l'utilisation, avec la valeur exacte variant de service à service, ou selon la zone géographique d'appel.

4.3.1 Amazon web service SES

Amazon Simple Email Service (Amazon SES) est un service de messagerie économique qui repose sur l'infrastructure fiable et évolutive développée par Amazon.com pour sa propre clientèle. Avec Amazon SES, vous pouvez envoyer et recevoir des e-mails sans aucun engagement à long terme.

Amazon web service SES gère bien la dérivation: c'est la probabilité qu'un e-mail arrive dans la boîte de réception du destinataire sans être identifié en tant que courrier indésirable ou directement bloqué:

Pour maximiser la délivrabilité de l'ensemble de ses expéditeurs, Amazon SES :



Figure 4-32 : logo amazon SES

- Aide à conformer facilement aux protocoles standard d'authentification des e-mails.
- Fournit des adresses IP dédiées pour donner un contrôle total sur la réputation des adresses IP qui envoient nos messages.
- Permet de suivre les métriques d'envoi d'e-mails pour qu'on puisse détecter et résoudre les problèmes immédiatement.
- Filtre le contenu intégrant des virus ou des programmes malveillants et bloque ces messages avant qu'ils soient envoyés.

Afin d'affiner notre stratégie de messagerie, Amazon SES propose différents moyens de surveiller les envois. On surveille nos envois d'e-mail afin de vérifier les événements tels que les retours à l'expéditeur, les plaintes, les messages délivrés, les e-mails envoyés et les e-mails rejetés.

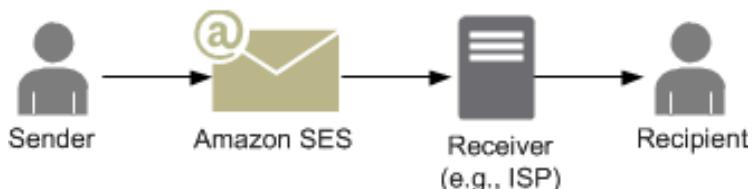


Figure 4-33 :envoi de mail par SES

4.3.2 Amazon SNS Pub/Sub Messaging

La solution Amazon Simple Notification Service (SNS) est un service de messagerie pub/sub et de notifications pour applications mobiles flexibles et entièrement gérées qui permet de coordonner l'envoi de messages à des points de terminaison et des clients abonnés. SNS permet de déployer des messages à un grand nombre d'abonnés, notamment des systèmes et des services décentralisés, et à des appareils mobiles. Il est facile de configurer, d'exploiter et d'envoyer de manière fiable des notifications vers tous les points de terminaison, quelle que soit l'échelle. En utilisant AWS Management Console, l'interface de ligne de commande AWS ou bien SDK AWS avec seulement trois simples API .

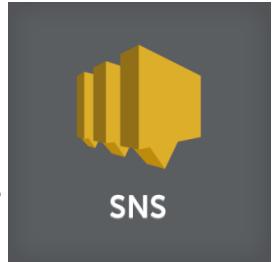


Figure 4-34 : logo amazon SNS

L'utilisation de SNS ne prend que quelques minutes. SNS élimine la complexité et les frais généraux associés à la gestion et à l'exploitation de logiciels et d'infrastructures de messagerie dédiés.

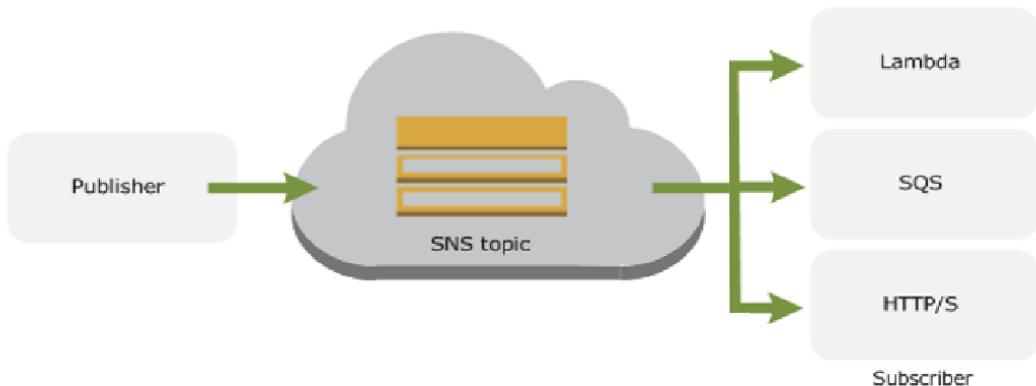


Figure 4-35 : Diagramme SNS

Grâce à SNS, on peut utiliser des rubriques pour découpler les éditeurs de messages des abonnés, déployer des messages à plusieurs destinataires et supprimer les sondages des applications. SNS prend en charge plusieurs types d'abonnement, ce qui vous permet d'envoyer directement des messages en mode push vers des files d'attente d'Amazon Simple Queue Service (SQS), des fonctions AWS Lambda et des points d'extrémité HTTP. Les services AWS, tels qu'Amazon EC2, Amazon S3 et Amazon CloudWatch, peuvent publier des messages sur les rubriques SNS pour déclencher la programmation et les flux de travail événementiels. SNS collabore avec SQS pour fournir une puissante solution de messagerie permettant de créer des applications basées sur le cloud tolérant les défaillances et faciles à mettre à l'échelle.

4.3.3 Amazon Relational Database Service (RDS)

Avec Amazon Relational Database Service (Amazon RDS), on peut configurer, gérer et dimensionner facilement une base de données relationnelle dans le cloud. Ce service offre une capacité économique et ajustable ainsi qu'une automatisation des tâches administratives chronophages, telles que l'allocation de matériel, le paramétrage de bases de données, l'application de correctifs et les sauvegardes. On peut ainsi se concentrer librement sur nos applications, afin de leur donner les performances rapides,



Figure 4-36 : logo amazon RDS

la haute disponibilité, la sécurité et la compatibilité dont elles ont besoin.

Amazon RDS est disponible sur plusieurs types d'instances de base de données, optimisées pour la mémoire, les performances ou les I/O. Amazon RDS vous donne le choix entre six moteurs de base de données communs, notamment Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle et Microsoft SQL Server.

Plus de détail sur Amazon et les réponses d'un mail sont décrits dans l'**annexe IV**.

Pixel espion de tracking

Un pixel espion (aussi appelé balise web, pixel invisible) est une très petite image numérique transparente utilisée pour collecter des informations sur l'activité des utilisateurs. Ce nom provient de leur dimension réduite, généralement un seul pixel pour minimiser les temps de chargement puisque l'image en elle-même n'apporte aucune information.

Un pixel espion permet à un service de statistiques ou de mesure d'audience de pister la navigation d'utilisateurs à travers différents sites web. L'image en question est incluse dans les pages des différents sites utilisant le service mais elle est mise en ligne sur les serveurs du service.

Un pixel espion inclus dans une page ne permet pas à lui tout seul de récolter les informations voulues. Le mécanisme est en général le suivant :

Le visiteur Bob ouvre une campagne A qui inclut le pixel espion. Lorsque Bob demande l'image au service il envoie la plupart du temps un référent qui permet au service de savoir qu'il vient de A (ou bien l'adresse exacte de l'image contient un identifiant du campagne A); le service lui demande d'enregistrer un cookie associé à cette information (que ce soit sa valeur ou une donnée de la session qu'il représente).

Dans notre système, on a opté pour la démarche suivante :

1- l'utilisateur ouvre l'email

2- Le client télécharge automatiquement le lien de pixel de tracking qui est lié à l'id de la campagne envoyé depuis le serveur

3- Le serveur marque l'image du id ouvert comme lu, et retourne à l'utilisateur une image comme réponse.

4- L'image générée (qui est transparente) est renvoyée à l'utilisateur.

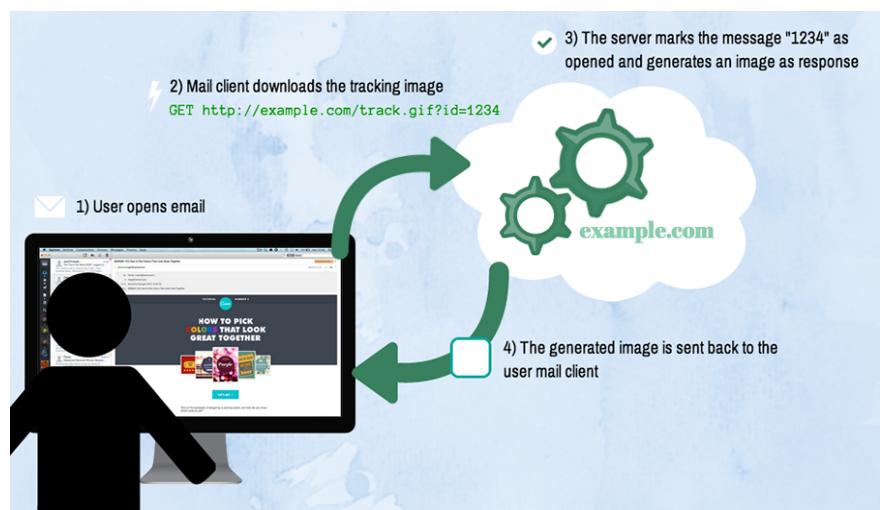


Figure 4-37 : schéma explicative de procédure de pixel de track

Conclusion

Ce chapitre a été consacré à la présentation des différentes technologies, les outils utilisés lors de la réalisation du projet. Le chapitre suivant consistera à effectuer une conception détaillée de la solution retenue dans l'optique de faciliter sa réalisation .

Chapitre V :

Conception

Introduction

5.1. Prototypage

5.1.1. Les maquettes réalisées

5.2. Conception du système

5.2.1. Diagramme de Classe Métier

5.2.2. Diagrammes état de transition

5.2.3. Diagrammes d'Activité

5.3. Conception et documentation de l'API REST

Conclusion

Introduction

Après avoir achevé les phases de spécifications et d'analyses, celle de l'étude technique, ce chapitre va élaborer en détails l'étude conceptuelle effectuée muni des diagrammes UML correspondants, et des interfaces hommes machines qui sont réalisées lors de cette phase.

Le prototypage

Le prototypage est la démarche qui consiste à réaliser un prototype. Ce prototype est un exemplaire incomplet et non définitif de ce que pourra être le produit ou l'objet final.

Deux méthodes de prototypage selon le niveau d'interactivité offert par le prototype :

Le prototype horizontal : correspond uniquement à l'interface, c'est une maquette statique.

Seul l'agencement général et les éléments de l'interface sont présentés : boutons, menus, champs de saisie, etc. Il permet :

- ◆ La confirmation des exigences de l'interface utilisateur et l'étendue du produit
- ◆ Une version de démonstration pour obtenir des retours des commanditaires du produit par rapport à leurs besoins initiaux
- ◆ Une première estimation du développement en termes de temps, de coût et de charge de travail

Le prototype vertical : met en œuvre certaines des fonctionnalités afin que l'utilisateur puisse dérouler un scénario d'utilisation typique, une tâche complète et significative du produit. Il permet :

- ◆ L'estimation du volume de données à prévoir ;
- ◆ Le dimensionnement du système de données, les besoins d'interaction de l'interface, les dimensions du réseau et le niveau de performance .
- ◆ Des précisions sur la complexité des exigences par rapport aux fonctionnalités actuelles du produit (selon le cas).

5.1.1.Les maquettes réalisées

Cette interface permet au client de créer un nouveau prospect.

The screenshot shows a web page titled "A Web Page" with the URL <http://www.dev2.tamtam.com/emailing/contact/create>. The main content is a form titled "Nouveau Prospect". The form consists of several input fields and dropdown menus:

- Two text input fields for "Nom" and "Prenom".
- Two text input fields for "Email" and "Société".
- Two text input fields for "Fonction" and "Sexe". A dropdown menu under "Sexe" shows "Homme" as the selected option.
- Two text input fields for "Langue" and "Groupe". A dropdown menu under "Langue" shows "Français" as the selected option. A blue "X" icon is positioned next to the "Groupe" field.
- A large text input field at the bottom containing the word "Save".

Figure 5-1 :création d'un nouveau prospect

Cette interface permet au client de modifier les données d'un prospect.

A Web Page

[http://www.dev2.tamtam.com/emailing/contact/edit](#)

Edit Prospect

| | | | |
|-------------------------------------|------------------------|---------|----------------------------|
| Nom | Bomboir | Prenom | Aude |
| Email | Bomboir_aude@gmail.com | Société | Fibaco sprl |
| Fonction | Comptable | Sexe | Homme |
| Langue | Anglais | Groupe | Generale X |
| <input type="button" value="Save"/> | | | |

Figure 5-2 : édition d'un prospect existant

Cette interface permet au client d'affecter un contact à un groupe donné, le contact peut appartenir à plusieurs groupes. L'interface permet aussi de retirer un contact d'un groupe donné.

A Web Page

[http://www.dev2.tamtam.com/emailing/contact/edit](#)

Edit Contact

| | | | |
|--|------------------------|------------------|--|
| Nom | Bomboir | Prenom | Aude |
| Email | Bomboir_aude@gmail.com | Société | Fibaco sprl |
| Fonction | Comptable | Sexe | Homme |
| Langue | Anglais | Groupe | Generale X |
| group1 X Comptable X age < 40 ans X | | Add a new groupe | <input type="button" value="Choix du groupe"/> Groupe2 Groupe3 Groupe4 |
| <input type="button" value="Save"/> | | | |

Figure 5-3 : Affecter contact à un groupe

Cette interface permet le client de lister les campagnes existants ainsi que leur etats, chercher une campagne par nom, filtrer les campagne ou enregister la liste des campagnes.

Figure 5-4 : Lister les campagnes

Cette interface permet le client de créer une nouvelle campagne markdown en spécifiant les groupes d'envoi, et le contenu du message.

Figure 5-5 : création d'un campagne markdown

Cette interface permet le client de créer une nouvelle campagne web en spécifiant les groupes d'envoi, et le lien de la page web à envoyer (ou choisir une parmi les modèles).

A Web Page
http://www.dev2.tamtam.com/emailings/campaings

From Email []
From Name []
Sujet []
Langue: Français Néerlandais Anglais
Page Web Message personnalisé
upload file [chemin_vers_le_fichier_html]

Groupe
Groupe inclus Generale [X](#)
Groupe exclus Hommes [X](#)
20 Destinataires

Tests
Emails [] [X](#)
Tester

Modeles
Modeles Modele 1 Modele 2

Envoyer

Figure 5-6 : création d'une campagne web

Cette interface permet le client de visualiser les statistiques d'une campagne envoyée(vues, retour, spam, cliques...), ainsi des détails sur les destinataires de la campagne.

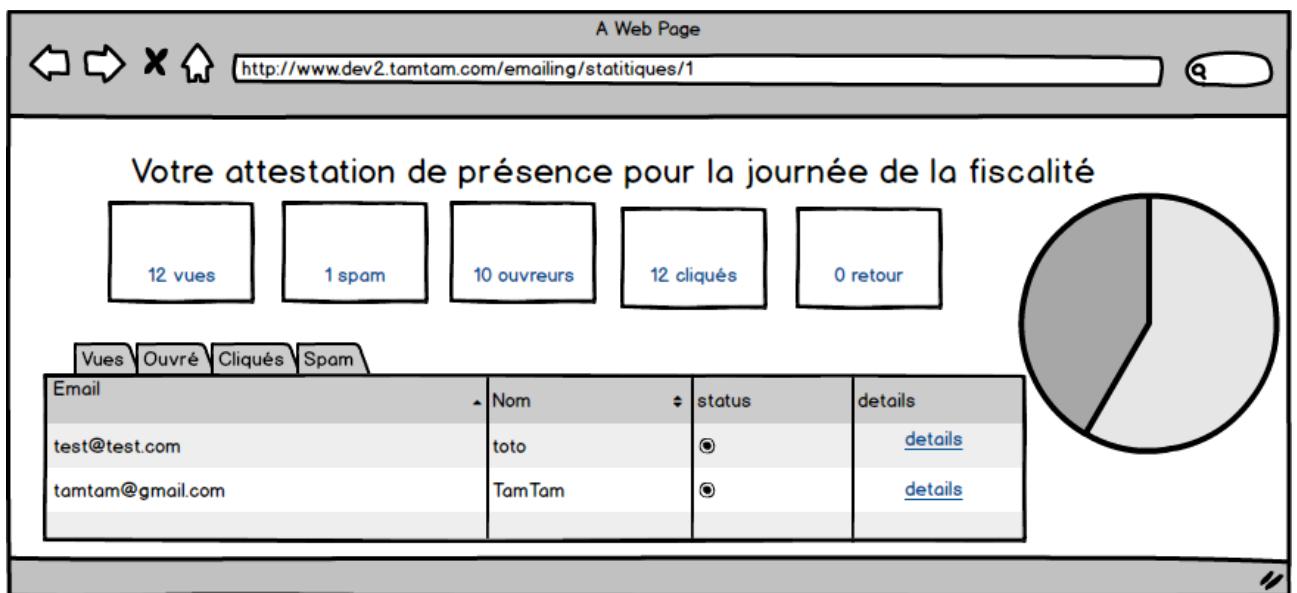


Figure 5-7 : statistique sur un emailing

Conception de du système

5.2.1. Diagramme de Classe Métier

On va présenter le diagramme de classe de l'application pour avoir une idée sur les relations qu'on a au sein de l'application .



Figure 5-8 : diagramme de classe métier

5.2.1.1. Description des packages de diagramme de classe

| Package | Description |
|----------------|---|
| EmailingBundle | C'est le package de notre projet, il contient toutes les classes reliées à notre conception du proje emailing, il est lié au plusieurs autres projets tels que ApiBundle, EzineBundle et profil. |
| EzineBundle | Ce package lié au projet Blog-Ezine, ce projet est conçu pour créer des newsletter manuellement suite au besoin du client. Chaque newsletters génère plusieurs tempalates, ces templates sont la base d'une campagne web dans EmailingBundle. |
| Profil | Ce package est dédié à tout ce qui est en relations avec le client de notre système, notre système est un système centralisé sur le client tous les projet tel que l'emailingBunde doivent impérativement être liés au client. |
| ApiBundle | Ce package englobe les classes reliées aux utilisateurs de notre application, dans notre cas ces utilisateurs sont les destinataires de nos campagnes d'envoi. |

Tableau 5-1 : description des packages de diagramme classe

5.2.1.2. Description des classe de diagramme de classe

| Classe | Description |
|-------------------|---|
| Campagne | C'est la classe principale de notre projet, cette classe inclus toutes les données principales à envoyer en plus de «from»(e-mail de l'émetteur) et «fromName»(nom de l'émetteur), elle inclu le type de campagne «type» (html ou markdown) le message à envoyer «markdown» , le type de destinataire «recipientType» (prospect ou user), ses différents statutess (test, envoyé, brouillon, créer), les liens inclus dans la campagne «links» ainsi la date de création «createdAt» et de modification «updatedAt» cette classe est reliée à toutes les autres classes de notre système. |
| CampagneRecipient | Cette classe est reliée directement à la classe campagne, elle présente les destinataires, le contenu envoyé relié à chaque destinataire et les statistiques de chaque destinataire d'une campagne, ces statistiques sont la date de livraison «deliverAt», la date de retour «bounceAt», la date de complaint (spam) «complaintAt» la date d'ouverture du mail «openAt», la date du clic sur les liens du mail «clickAt» l'état de livraison «sendStatus» (Sucess, Failed, Refused) et le délai maximal «delaiMax» au cas d'une erreur transitoire. |
| Link | Cette classe représente les liens inclus dans une campagne web est le nombre de clic «clickCount» de chaque lien. |
| Markdown Model | Cette classe représente les models markdown déjà crés. Afin de choisir une pour l'envoi. |
| Groupe | Cette classe représente les groupes d'envoi d'une campagne, une campagne est envoyée à un groupe de users, chaque groupe contient plusieurs users, le groupe peut être inclu dans l'envoi ou exclu de l'envoi d'une campagne «status». |
| Prospect | Cette classe représente les utilisateurs non-potentiels de l'application au contraire des users, les prospects ne sont pas inscrits dans la plate-forme. Le client peut les ajouter est les affecter au groupe d'envoi. |
| Document | Classe de gestion des fichiers. |
| Attachmenet | Classe des pieces-jointes d'une campagne, elle hérite du classe principal Document. |
| BlackList | Classe présente les users qui sont inclus dans les groupes d'envoi, mais qu'on désire les |

| | |
|-----------------|---|
| | exclure de la réception de cette campagne. |
| NewsLetters | La classe inclus dans le projet Ezine, cette classe génère un ou plusieurs template. |
| Template | La classe newsletters ou le client génère différents template, ces templates sont présentées pour choisir une dans la campagne web afin d'être envoyer. |
| TemplatePreview | Classe hérite du Document, cette classe sert à telecharger l'image du template choisi afin d'être visualiser au client. |
| Client | l'utilisateur de notre application. |
| User | Le destinataire des campagnes envoyées. |

Tableau 5-2 : description des classes de diagramme classe

5.2.2. Diagrammes état de transition

Les diagrammes d'états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie.

5.2.2.1. Diagramme état de transition d'une campagne

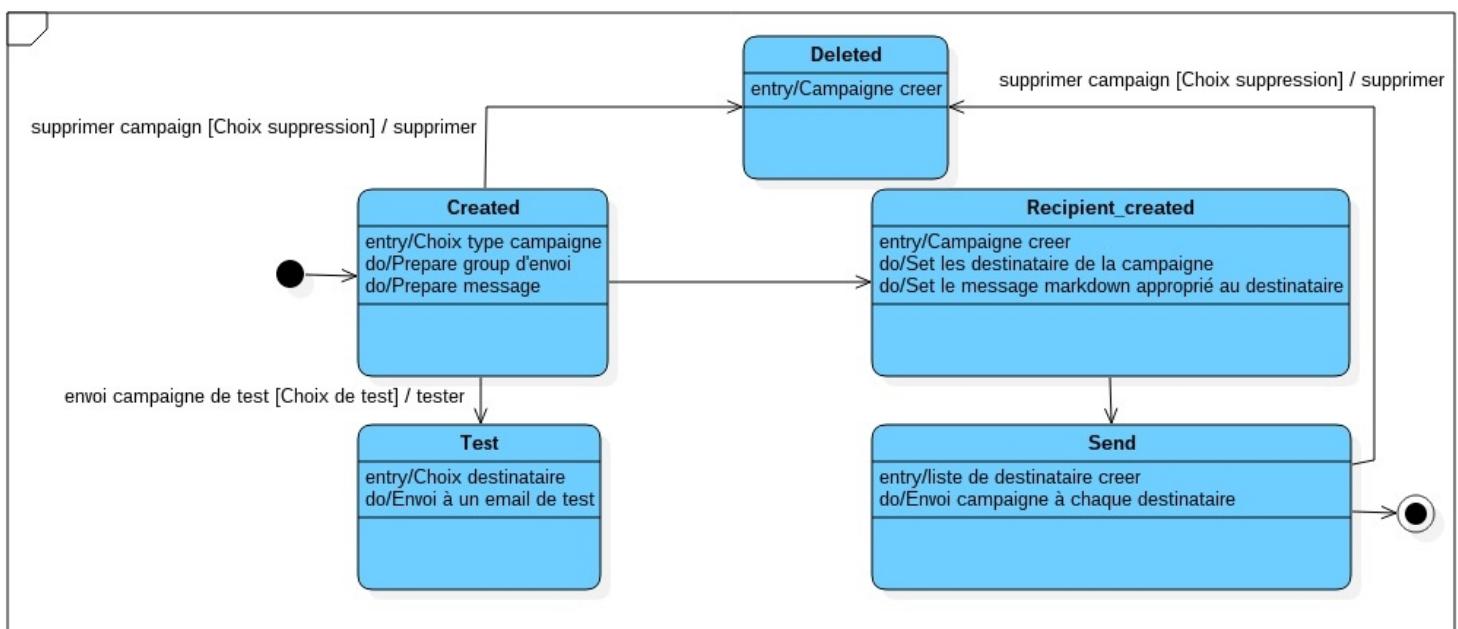


Figure 5-9 : diagramme état transition campagne

Ce diagramme explique les états d'une campagne depuis la création jusqu'au l'envoi de la campagne, la campagne peut avoir 5 états expliqués dans le tableau suivant :

| État | Description |
|-------------------|--|
| Created | Le premier état d'une campagne dans le processus d'envoi, après que le client choisit ses groupes d'envoi, son contenu de campagne quel que soit son type, l'envoi de campagne n'implique pas un envoi direct de la campagne, mais un enregistrement dans un état créé, à la fin de cet état les groupes sont bien préparés, ainsi que le contenu général (si on markdown). |
| Recipient_created | Cet état vient jusqu'à après l'état de création, on récupère notre campagne créée, on extrait les destinataires depuis les groupes en prenant compte des groupes inclus et exclus, d'autre part, on coordonne chaque message markdown avec son destinataire afin d'avoir chaque message propre à son destinataire et on mappe ces données dans une table statistique «campaignRecipient» et on procède avec l'envoi de message |
| Send | Une fois les données sont bien mappées sur la nouvelle table, la campagne passe de l'état «Recipient_created» vers l'état send. |
| Test | Un état optionnel d'une campagne, le client envoie une campagne de test un seul destinataire, la campagne est envoyée dans ce cas automatiquement sans passer par les 3 états précédents. La campagne dans ce cas passe à l'état test. |
| Deleted | Un état optionnel d'une campagne, le client choisit de supprimer une campagne créée ou envoyée, la suppression d'une campagne est une suppression logique et non pas physique. |

Tableau 5-3: description de l'état de transition d'une campagne

5.2.2.2. Diagramme état de transition de la classe campagneRecipient

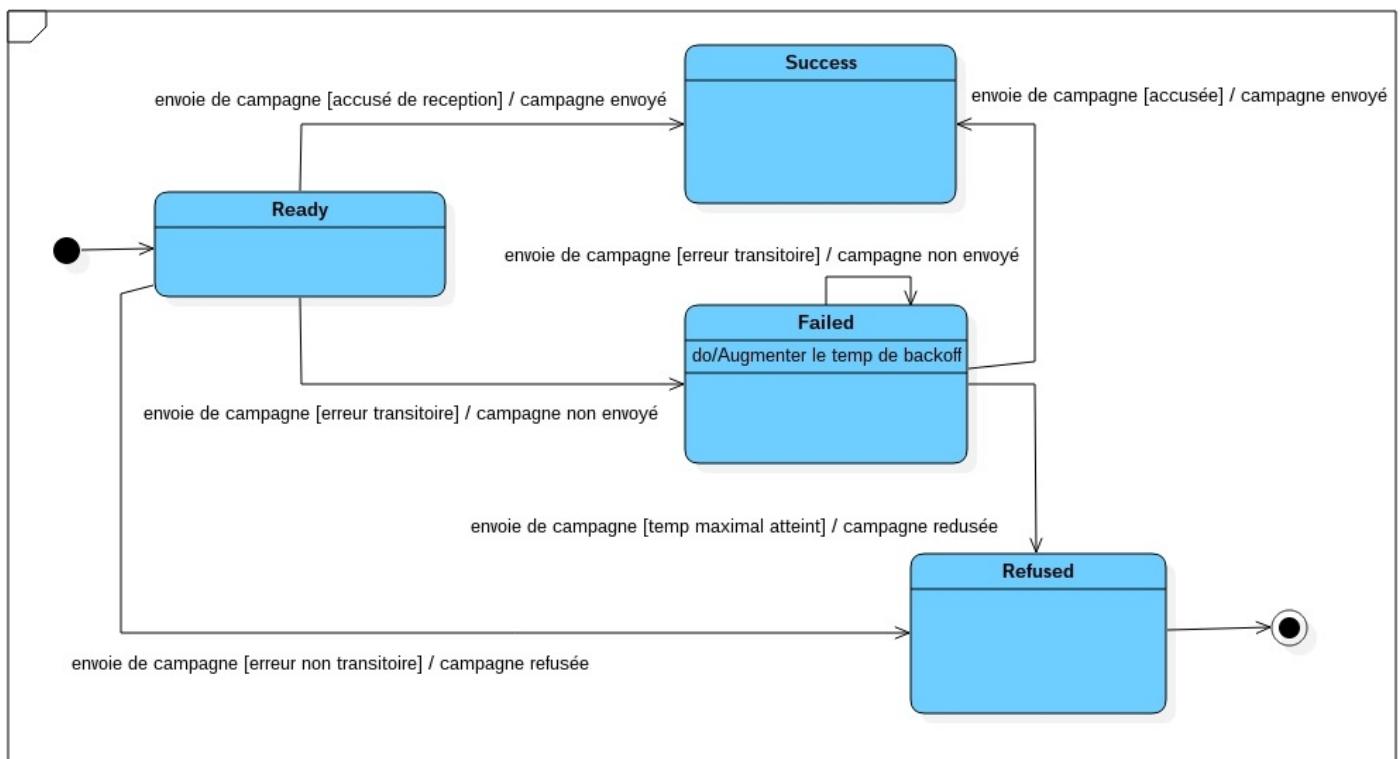


Figure 5-10 : diagramme état transition de campagneRecipient

Ce diagramme explique les états d'une campagneRecipient, après que les données sont mappées lors de l'état «send» de la classe campagne, un autre processus d'envoi pour chaque destinataire est déclenché, le tableau suivant décrit les états d'un message envoyé à une destinataire :

| Etat | Description |
|---------|---|
| Ready | Une fois que la campagne passe à l'état «send» chaque ligne des données est mappé dans la table «campagneRecipient» avec un état Ready pour qu'il soit envoyé à son destinataire. |
| Success | Si la campagne est délivrée avec succès au destinataire, la campagneRecipient passe à l'état success pour ce destinataire. |
| Failed | Si le message n'est pas envoyé suite à une erreur transitoire, on crée un délai maximal d'envoi à partir de l'erreur détectée puis, on relance l'envoi de campagne en augmentant le délai d'attente entre chaque tentative selon l'algorithme « exponentielle backoff », si on atteint le délai maximal d'attente, le destinataire est considéré inactif et la campagneRecipient passe à l'état «refused» pour ce destinataire, si on reçoit un accusé de réception avant d'atteindre le délai maximal, la campagneRecipient passe à l'état «success» pour ce destinataire. |
| Refused | Si la campagne n'est pas délivrée à un destinataire suite à une erreur non transitoire (hard bounce, spam ...), ou suite à une erreur transitoire qui a dépassé le délai maximal d'attente, la campagneRecipient passe à l'état «refused» pour ce destinataire. |

Tableau 5-4 : description de l'état de transition d'une campagneRecipient

5.2.3. Diagrammes d'activité

Le diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution. Les actions sont les unités de base du comportement du système.

Le diagramme d'activité est utilisé dans cette phase pour documenter le fonctionnement d'un cas d'utilisation et mettre en évidence les flots d'information contrairement aux diagrammes de séquence qui focalisent plus sur la chronologie des interactions entre le système et ses composants.

5.2.3.1. Algorithme de gestion des tentatives

Dans ce diagramme, on procède comme suit :

1- on appelle le service d'envoi lors d'envoi d'une campagne.

2- si on détecte une erreur transitoire, on définit un délai maximal d'attente, ce délai change selon le type d'erreur et le temps de latence.

3- on augmente le délai d'attente en utilisant l'algorithme d'exponentielle backoff, puis on reappelle le service.

4- si l'erreur persiste, on ré-augmente le délai d'attente et on renvoie la campagne, jusqu'au l'envoi de la campagne ou l'atteint de délai maximal d'attente.

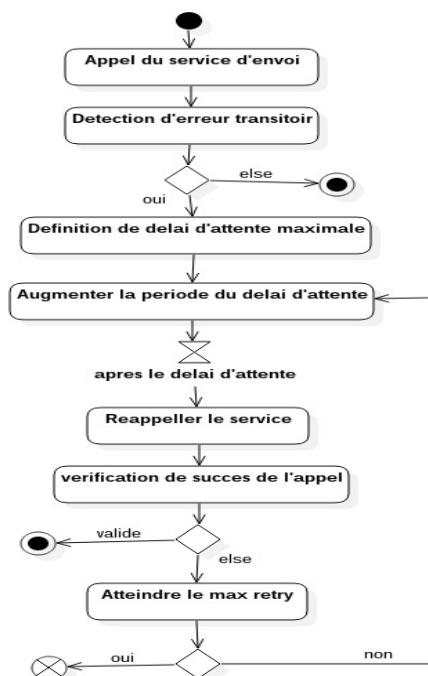


Figure 5-11 : diagramme d'activité de gestion des tentatives

5.2.3.2. Envoi de campagne

Dans ce diagramme, on procède comme suit :

1- on choisit une campagne d'envoi.

2- si c'est une campagne markdown on édite le contenu markdown, si non on choisit une template, le preview est généré automatiquement.

3- on choisit les groupes d'envoi, les groupes inclus ainsi que les groupes exclus.

4- on choisit la possibilité d'envoyer une campagne de test avant d'envoyer la campagne.

5-envoi de la campagne après vérification des données.

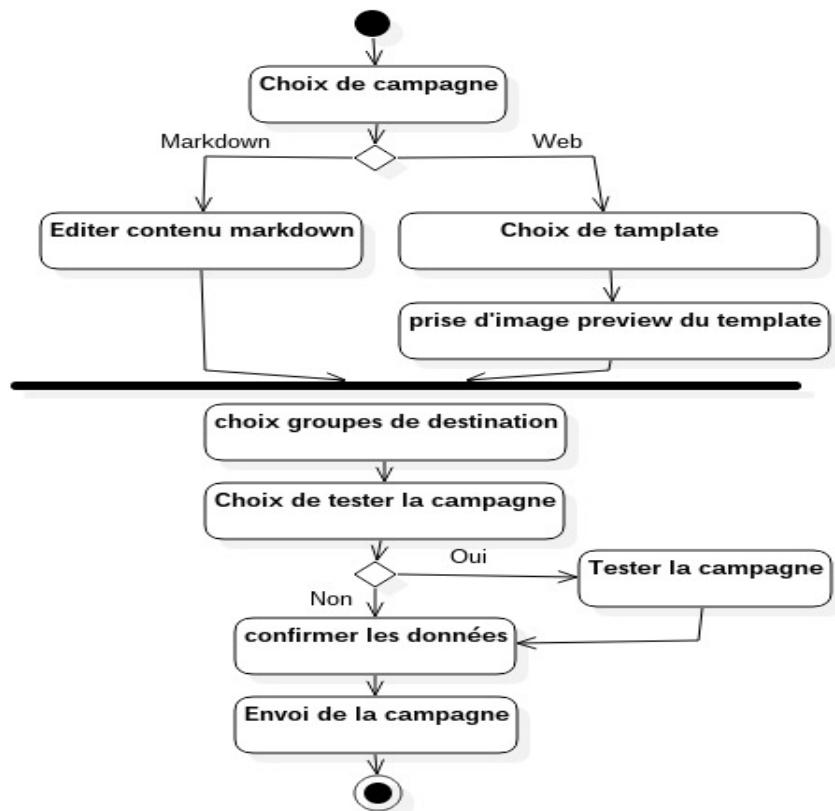


Figure 5-12 : diagramme d'activité d'envoi de campagne

Conception et documentation de l'API REST

La conception de l'API consiste à écrire les URIs qui seront utilisés pour les appels http.

La figure ci-dessous représente la route qui permet de récupérer tous les messages de la campagne.

```
export const urls = {
  index: '/',
  autoLogin: 'autologin',
  campaign_list: 'campaigns',
  campaign_form: 'campaign(/:campaignId)',
  campaign_stats: 'stats/:campaignId',
  notFound: '*'
};
```

Figure 5-13 : création des urls

```
return axios.get(requestUrl, {
  params: {
    access_token: token,
    fields: fields.join(','),
    filter: JSON.stringify(filter),
    nolimit: 1,
    sort: JSON.stringify(sort)
  }
});
```

Figure 5-14: appel en utilisant restApi

Pour notre application on a mis un fichier qui est consacré à gérer les URI dont on a besoin, pour faciliter la modification dans le cas des changements, la dernière illustre un appel en utilisant un filtre et trie sur les résultats.

Pour bien organiser les tâches et que la réponse soit compatible avec la demande on a mis une place un tableau dans lequel on spécifie la structure de notre demande ainsi que celle souhaitée de l'envoi pour le cas d'appel au groupe.

| | |
|-------------------|--|
| Name: | Appel aux groupes. |
| Type: | JSON |
| Request Type: | GET |
| Input parameters: | Clienid: int |
| Response: | // avec succès: { 'erreur':Null, 'groupe': {'fetched':true, 'fetching':false, 'items': array, 'nbResult':int } } |

```

// echec
{
  'erreur':Not Null,
  'groupe': {'fetched':false,
              'fetching':true,
              'items': null,
              'nbResult':0
            }
}

```

Tableau 5-5 : tableau de spécification pour API REST

```

next state ▼ Object ⓘ
  ► auth: Object
  ▼ campaigns: Object
    ► filter: Object
    ► list: Object
    ► recipients: Object
    ► sort: Object
    ► stats: Object
    ► __proto__: Object
  ▼ groups: Object
    error: null
    fetched: true
    fetching: false
  ▼ items: Array(13)
    ▼ 0: Object
      category: "CONTACT"
      id: 4
      name: "Age < 30"
      type: "NATIVE"
      ► __proto__: Object
    ► 1: Object
    ► 2: Object
    ► 3: Object
    ► 4: Object
    ► 5: Object
    ► 6: Object
    ► 7: Object
    ► 8: Object
    ► 9: Object
    ► 10: Object
    ► 11: Object
    ► 12: Object
    length: 13
    ► __proto__: Array(0)
    nbResult: 13
  ► proto : Object

```

Figure 5-15 : reponse json de l'api

Dans la requête de la réponse en remarque qu'elle correspond à ce qui décrit dans le tableau des spécifications, de la même façon le reste des appels est effectués.

Conclusion

Ce chapitre a décrit l'étude conceptuelle du système. Durant cette phase on s'est surtout focalisé sur les diagrammes d'activités et d'état de transition ainsi que les maquettes réalisées, après on a dessiné le diagramme de classe pour comprendre la logique de la base de données vers la fin on a invoqué l'API REST et comment on l'a utilisé. On détaillera la phase de réalisation de notre projet dans le chapitre suivant.

Chapitre VI :

Réalisation

Introduction

- 6.1. Organisation du Travail
- 6.2. Environnement du travail
- 6.3. Structure du projet
 - 6.3.1. coté serveurs
 - 6.3.2. coté client
- 6.4. Les interfaces graphiques
 - 6.4.1. les interfaces de configuration d'Amazon SNS
 - 6.4.2. les interfaces de réalisation

Conclusion

Introduction

Après la conception, ce chapitre sera dédié à la réalisation de notre système. Nous y présenterons les outils d'implémentation de notre solution, les Frameworks créés pour la réalisation du projet avant de finir avec une présentation de quelques scénarios d'utilisation du système via quelques interfaces graphiques.

Organisation du travail

L'équipe de TamTam se compose d'un chef de projet qui se charge d'affecter et encadrer l'équipe qui sont les développeurs faisant l'étude fonctionnelle, technique, la conception et la réalisation .

Le chef de projet se chargeait d'organiser une réunion au début de chaque sprint, discuter le cahier de charge, estimer le temps consacré à chaque tâches, affecter des tâches sur Jira, cependant on déplace les tâches dans les différentes colonnes:

ToDo : les tâche à faire.

In Progress : les taches en cours de développement.

Pull Request : les tache sont finies et ils sont mise en github en «pull request» afin d'avoir des remarques ou correction d'équipe.

Staging : la tache est corrigée et prête à être émergée par le chef de projet.

Done : la tache est émergée dans la production.

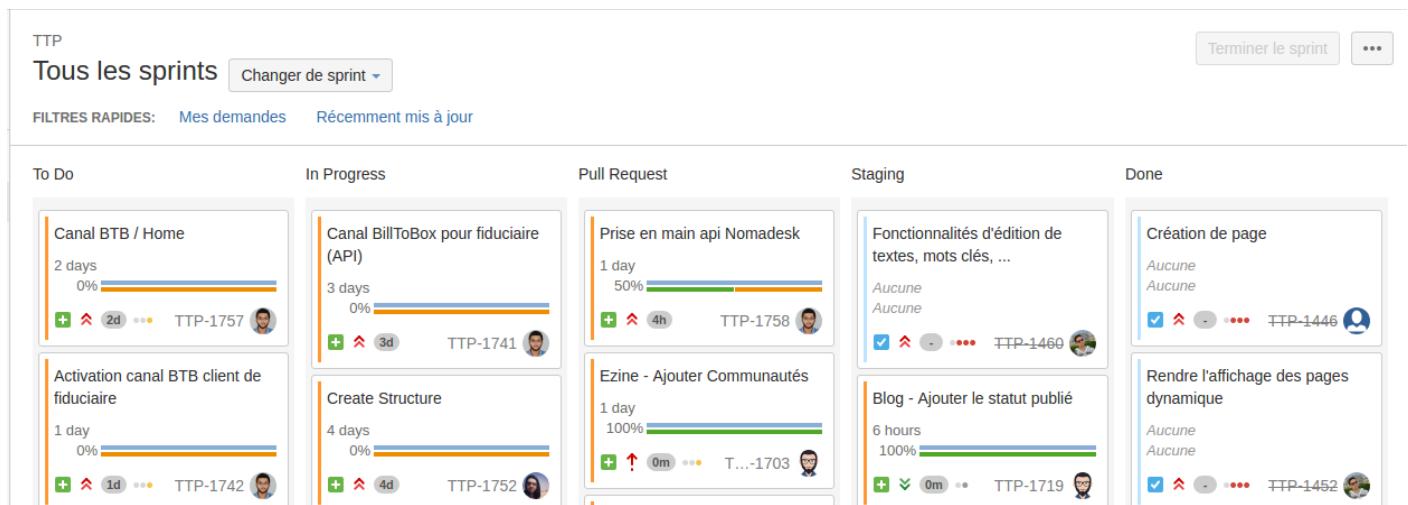


Figure 6-1 : workflow on Jira

Environnement du travail

Dans cette partie nous allons parler de l'environnement et des outils de développement.

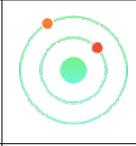
| | | |
|------------------------------|---|--|
| Linux (Ubuntu 16.04) |  | est un système d'exploitation open source basé sur la distribution Linux Debian. |
| PhpStrom (2016.3) |  | IDE compatible avec php, permet de faciliter le travail avec le Framework SYMFONY |
| WebStrom (2017.1) |  | IDE compatible avec JavaScript, permet de faciliter le travail avec React |
| Git |  | gestionnaires des versions , pour enregistrer les versions et les visualiser dans le serveur |
| GitKraken |  | Est un client graphique pour le logiciel de virsionaisation Git |
| Mysql |  | Base de donnée relationnelle et libre, c'est la base de donnée de l'application |
| SQLite |  | Base de donnée relationnelle utilisée pour les tests de l'application. |
| ReduxDevTools |  | Extension qui Facilite à débugger l'application front sur le navigateur |
| NodeJs |  | pour exécuter des commandes sur notre projet, télécharger les dépendances et exécuter le projet soit sur le navigateur. |
| Webpack |  | c'est un module bundler, il prend des modules avec des dépendances et génère des assets statiques qui peuvent ensuite être utilisés sur le navigateur. |
| Composer |  | c'est un gestionnaire de paquets libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin |

Tableau 6-1 : outil de développement utilisée

Structure du projet

Nous avons conçu l'architecture de notre projet en respectant l'architecture logique que nous avons établie lors de l'étude technique. Les figures suivantes représentent les différents dossiers/fichiers de notre application :

5.3.1 Coté serveur

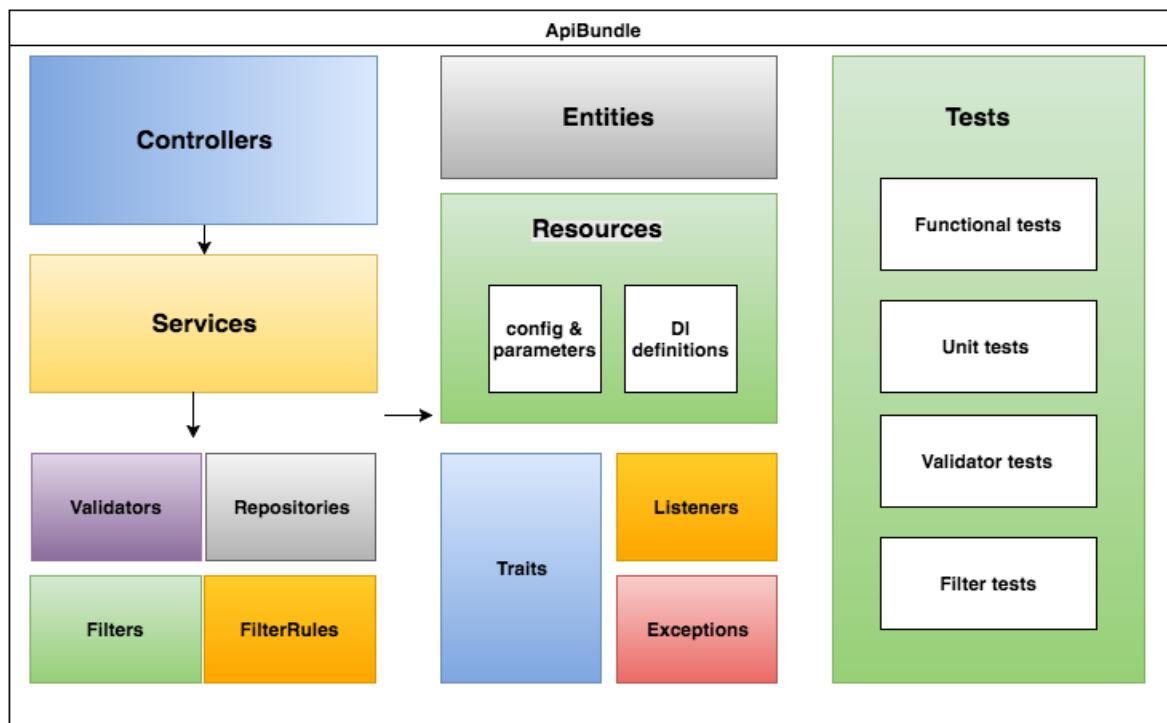


Figure 6-2 : structure de base du backend

voila une description de quelque dossier de l'application :

Entity : contient les entités de notre application.

Repository : La préparation des requêtes DQL.

Ressource : fichier de configuration de l'application (des fichier en format YML compatible avec les standard de symfony).

Controller : les fonctions CRUD de l'application.

Validator : les validation des attributs des entités soit par filtrage ou des Roles .

Test/controller : les tests fonctionnels de l'application.

Test/validation : les tests unitaires de l'application.

Test/dataFixtures : les fakes Data générer pour les tests (utilisation de plugin Fzaninotto) .

Command : les deamon utilisé dans l'application

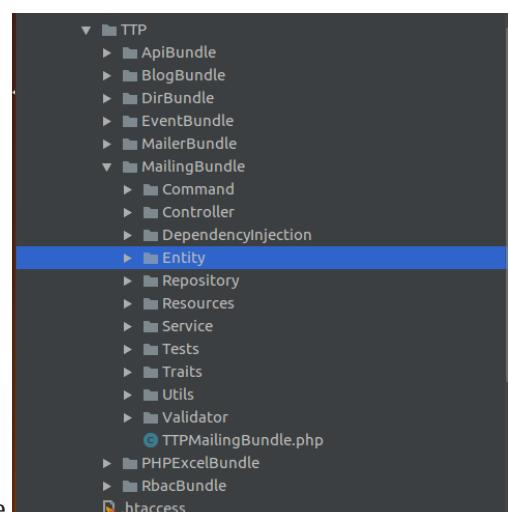


Figure 6-3: structure des dossier dans PhpStorm

5.3.2 Coté Client

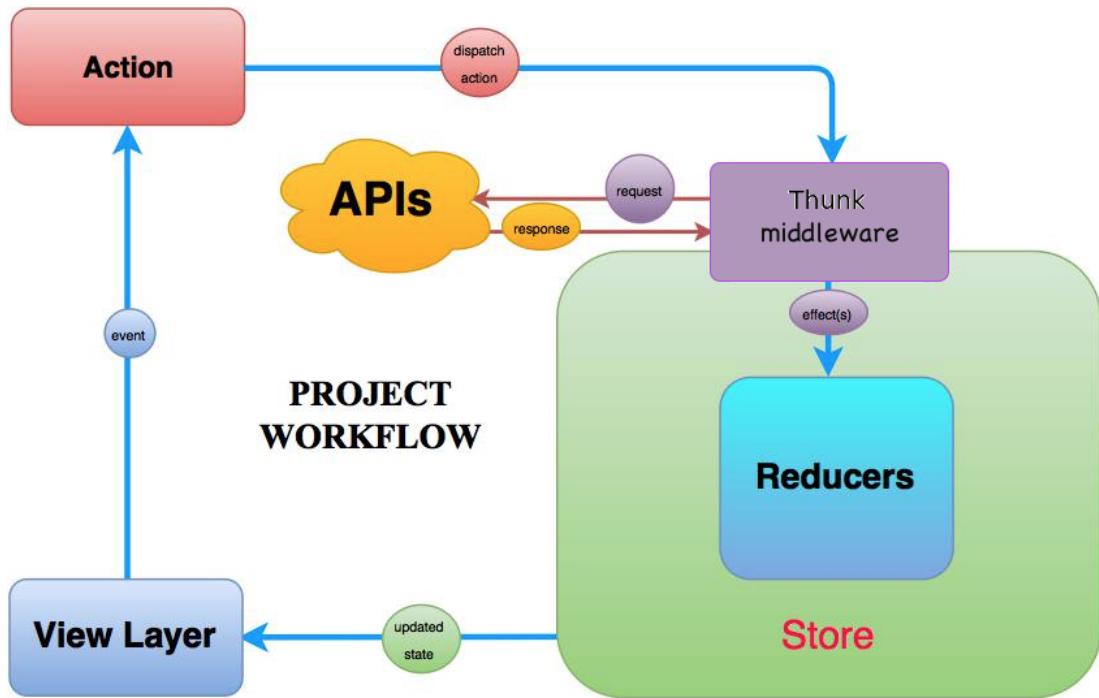


Figure 6-4 : workflow de l'architecture en front en liaison avec l'API

THUNK : Le middleware Redux, Thunk nous permet d'écrire des créateurs d'actions qui renvoient une fonction au lieu d'une action. Le thunk peut être utilisé pour retarder l'envoi d'une action, ou pour expédier uniquement si une certaine condition est remplie. La fonction interne reçoit les méthodes de Store, envoi et getState comme paramètres.

voilà une description de quelques dossiers de l'application :

Router : les routes de l'application.

Api : liaison de l'application de front avec l'api (coté serveur).

Style : les fichiers de style (fichier sass de l'extension .scss).

Public : tout ce qui doit apparaître dans le site (image, logo ...)

Action : les actions de la structure REDUX

Action/thunks : les middlewares de l'application

Components: les composants react de l'application

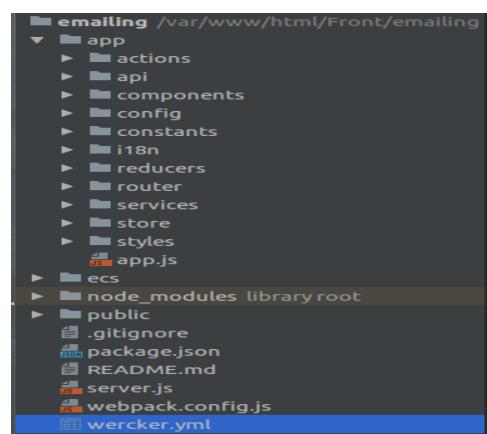


Figure 6-5 : structure des dossiers dans Webstorm

Les interfaces graphiques

6.4.1. les interfaces de configuration d'Amazon SNS

Dans cette section on présente la configuration d'amazon SNS pour les statistiques, on prendra par exemple le cas des retours (bounces).

Dans cette interface on choisit notre topic parmi les topics d'amazon SNS dans ce cas le bounce

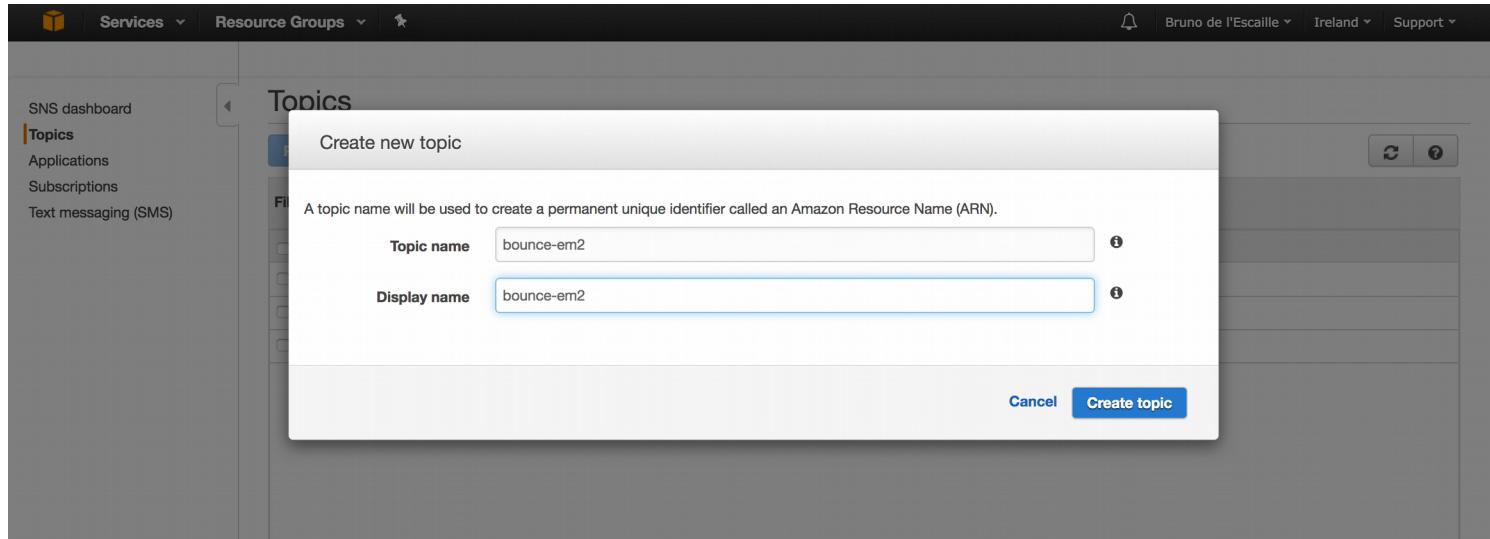


figure 6-6 : spécification du topic

Dans cette interface en spécifiant ARN (id) du topic choisi, on sélectionne le type de l'end-point de retour, dans notre cas bien sûr c'est du Http, en suite on introduit notre end-point de l'application

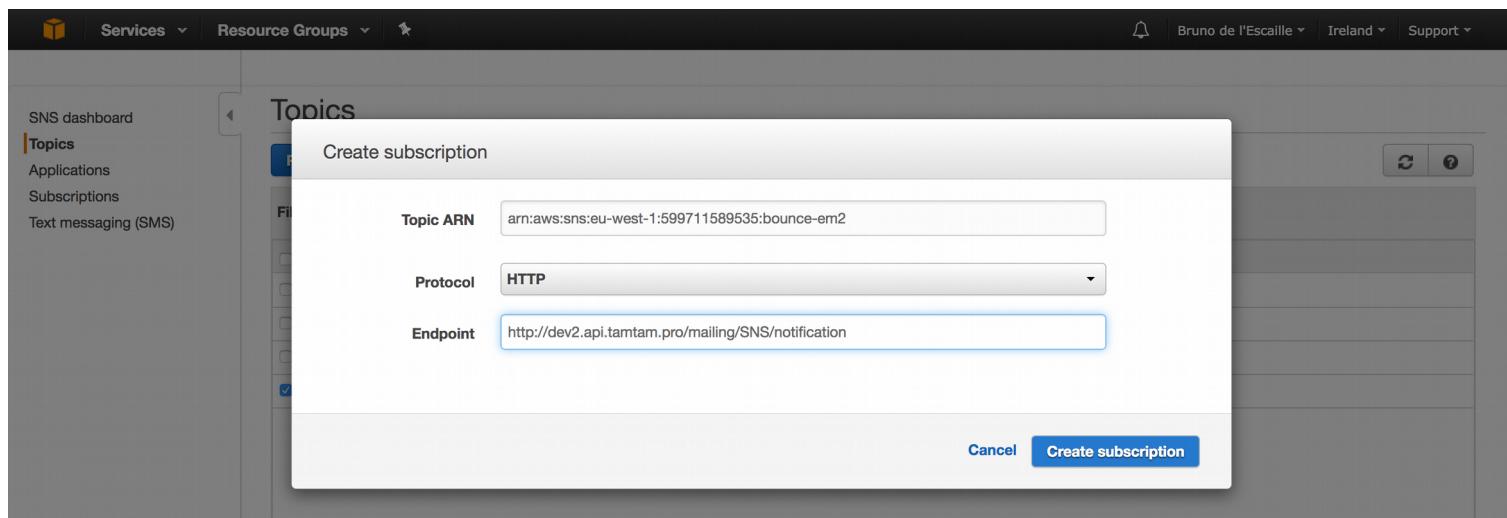


figure 6-7 : spécification de l'end-pointe

Après une confirmation de l'end-point l'état de notre topic passe à l'activation,

Subscriptions

| Subscriptions | | | | |
|--------------------------|--|----------------------|---------------------------------|--------------|
| Actions | | Subscription Details | | |
| | | Protocol | Endpoint | Subscriber |
| <input type="checkbox"/> | Subscription ID | | | |
| <input type="checkbox"/> | arn:aws:sns:eu-west-1:599711589535:bounce-em2:50589952-972b-4463-9be8-c15e377... | http | http://dev2.api.tamtam.pro/... | 599711589535 |
| <input type="checkbox"/> | PendingConfirmation | http | http://uat2.api.tamtam.pro/... | 599711589535 |
| <input type="checkbox"/> | PendingConfirmation | http | http://dev2.api.tamtam.pro | 599711589535 |
| <input type="checkbox"/> | PendingConfirmation | http | http://api.tamtam.pro/mailin... | 599711589535 |

Figure 6-8 : validation de l'end-point

6.4.2 Les interfaces de réalisation

La page d'accueil (index) de l'application .

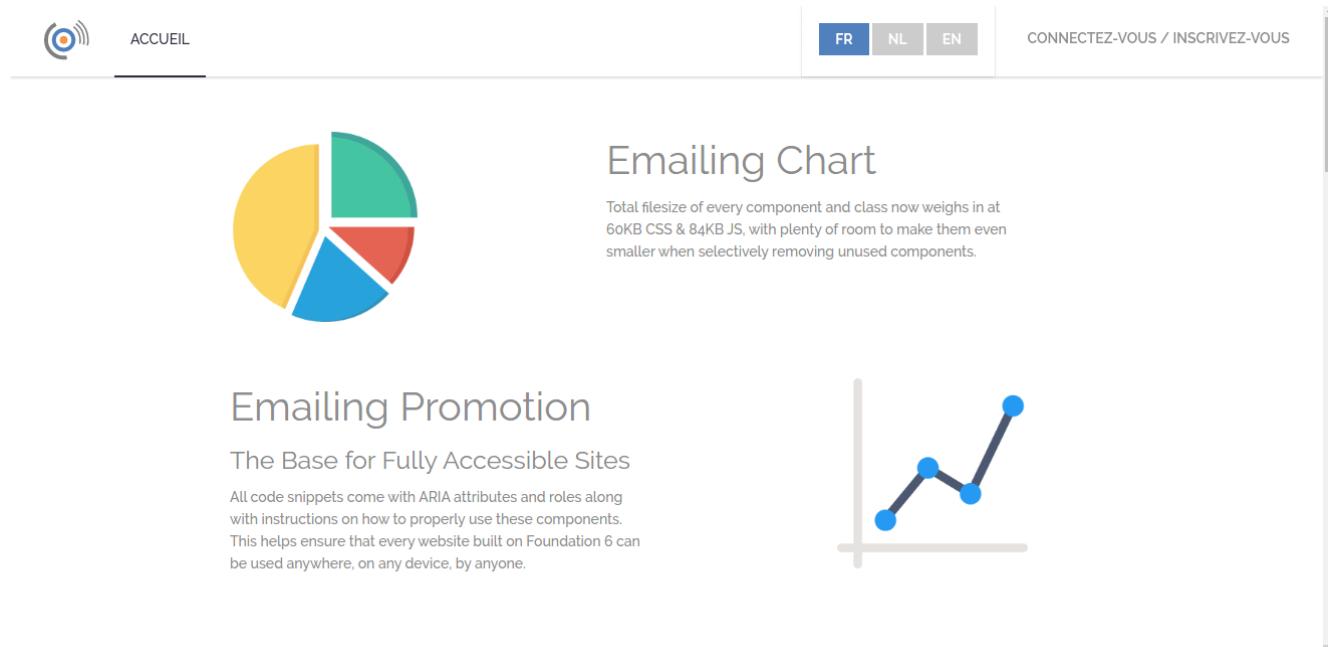


Figure 6-9 : l'accueil de l'application

Avant d'accéder aux fonctionnalités de l'application l'étape d'authentification est nécessaire, où l'utilisateur saisie son email et son mot de passe afin de prouver son identité.



Figure 6-10: l'authentification de l'application

Après son authentification le client est redirigé vers l'écran d'accueil de début mais cette fois il est autorisé d'utiliser les fonctionnalités de notre application :

- Lister les campagnes existants.
- Créer une nouvelle campagne.

Figure 6-11 : l'accueil après authentification de l'application

La vue lister les campagnes existants permet de visualiser les campagne créées et faire des actions (suppression/edit) sur chaque campagne, en choisissant le nombre de campagne par page.

The screenshot shows the Chanaa software interface for managing campaigns. At the top, there are navigation links: ACCUEIL, CAMPAGNES (which is underlined), CREATE CAMPAIGN, and CHANAA. A user icon with 'AC' is on the right. Below the header, the word 'Campaigns' is displayed, followed by a 'Create Campaign' button. There are filters for '10' items, a 'Filter' button, and a search bar. The main area shows a table with columns: 'Created At', 'Langue' (Lng), 'Subject', 'Groups', 'Status', and 'Actions'. The table contains six rows of campaign data:

| Created At | Langue (Lng) | Subject | Groups | Status | Actions |
|------------|--------------|-----------|--|--------|---------|
| 17/05/2017 | fr | test | | DRAFT | |
| 17/05/2017 | en | test | | DRAFT | |
| 16/05/2017 | fr | campaign1 | group1 | DRAFT | |
| 12/05/2017 | en | test2 | Age 30-39, Antwerpen, Expert-comptable Age < 30, Conseil fiscal | DRAFT | |
| 12/05/2017 | fr | test | | DRAFT | |

Figure 6-12 : lister les campagnes

On peut ainsi filtrer les campagnes selon le statut et langue. Dans cette exemple on choisit les campagnes d'une langue qui sont créées .

The screenshot shows the Chanaa software interface for managing campaigns, similar to Figure 6-12 but with filtering applied. The 'Filtre' button is highlighted. The filter sidebar on the left includes dropdowns for 'Créé à', 'Statut' (set to 'ALL'), 'Type' (set to 'ALL'), and 'Langue' (set to 'ALL'). A 'Réinitialiser' button is also present. The main table has columns: 'Groupes', 'Type', 'Status', and 'Actions'. The table contains five rows of campaign data, all of which have 'group1' listed under 'Groupes' and are marked as 'SENT' or 'TEST' under 'Status'.

| Groupes | Type | Status | Actions |
|----------------|----------|--------|---------|
| group1 | PAGE WEB | SENT | |
| group1, group1 | PAGE WEB | SENT | |
| | PAGE WEB | SENT | |
| | PAGE WEB | TEST | |
| | PAGE WEB | TEST | |

Figure 6-13 : lister les campagnes selon un filter

Dans le cas de l'envoi d'une campagne web on select une parmi les templates

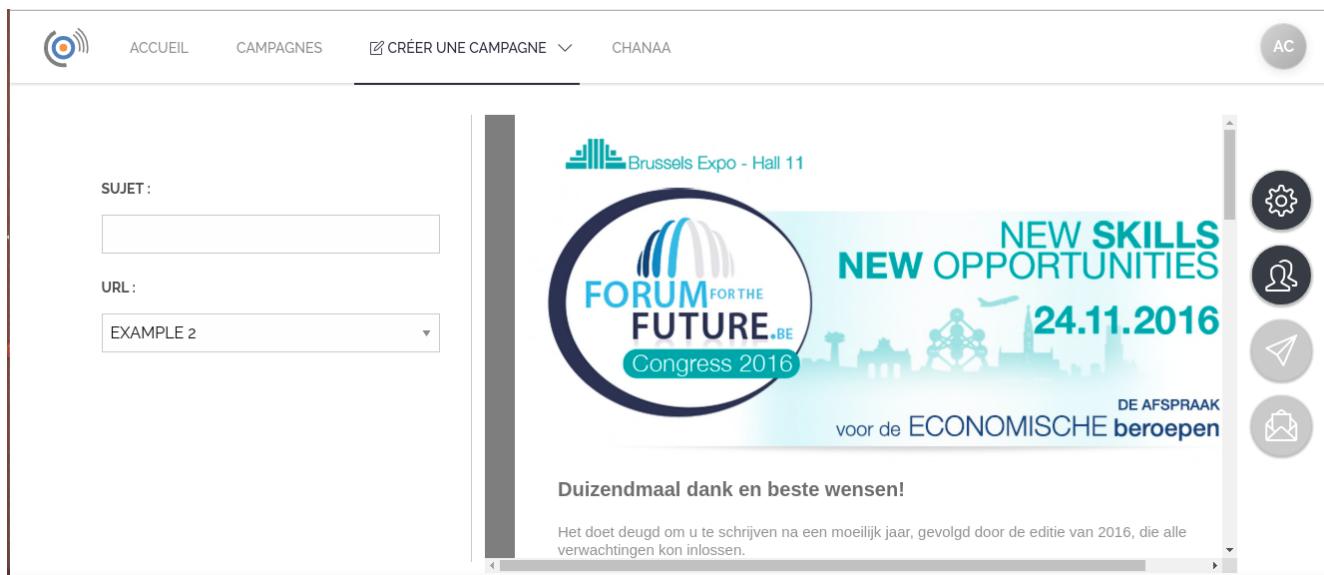


figure 6-14 : campagne web visualisé

Dans notre application l'envoi de la campagne passe par 4 étapes :

étape1 : les formations de base sur la campagne, elles incluent les données du client (émetteur)

INFORMATIONS SUR LA CAMPAGNE

| | |
|--|--|
| DE L'EMAIL | DU NOM |
| <input type="text" value="abdessamad139@gmail.com"/> | <input type="text" value="abdessamad chanaa"/> |
| RÉPONDRE À | |
| <input type="text" value="tamtam@pro.com"/> | |
| LANGUE | TYPE DE DESTINATAIRE |
| <input type="text" value="FRENCH"/> | <input type="text" value="CONTACT"/> |

Figure 6-15 : étape 1 de l'envoi

étape2 : choix du groupe de destination, dans cette étape on choisit les groupes inclus et exclus de notre campagne ainsi que les emails (users) exclus «blackList»

SÉLECTIONNEZ UN PUBLIC CIBLÉ

2

 CONTACT

 GROUPES INCLUS  GROUPES EXCLUS

| | | | |
|-------------|----------|------------------|-----|
| x Age 40-49 | x group1 | x Vlaams-Brabant | x ▾ |
| x Langue NL | x group1 | | |

| | |
|------------|-----|
| x Age < 30 | x ▾ |
|------------|-----|

10 ▾

search 

| Name | Email | Action |
|-------------------|-----------------------------|---|
| abdessamad CHANAA | abdessamad.chanaa@gmail.com |  |
| TEST AZER | lid@azer.com |  |

figure 6-16 : étape 2 de l'envoi

étape3 : étape optionnelle, sert à envoyer la campagne à un e-mail de test avant l'envoi au user généralement cet e-mail de test est l'email du client.

CAMPAGNES  CRÉER UNE CAMPAGNE ▾ CHANAA 

ENVOYER LE TEST À :

a|
abdessamad.chanaa@gmail.com



Figure 6-17 : étape 3 de l'envoi

étape4 : étape finale, sert à vérifier les données et envoyer la campagne

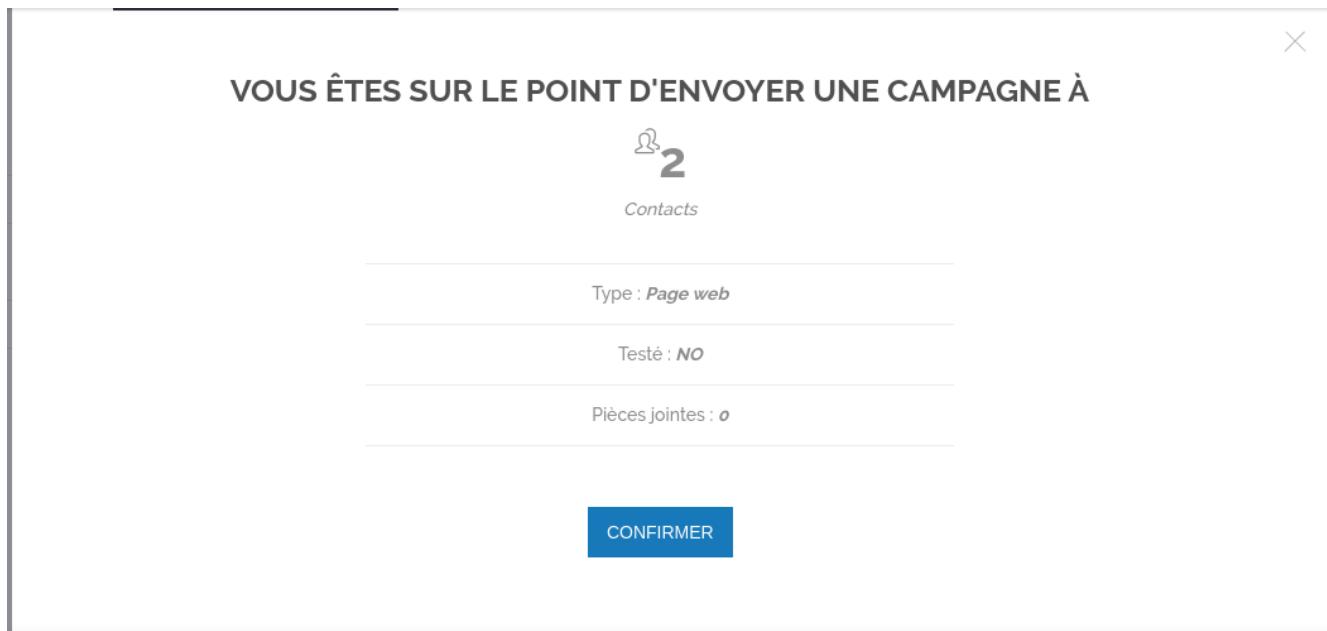


Figure 6-18 : étape 4 de l'envoi

Un message de confirmation apparaît afin de valider l'envoi de la campagne.



Figure 6-19 : validation de l'envoi

D'autre part on peut envoyer une campagne markdown au lien d'une campagne web

Cette interface permet d'envoyer une campagne de type markdown, le contenus est écrit en markdown.

From email abdessamad139@gmail.com

From name Abdessamad

Subject Test de rapport

FR EN NL

Salut {{receiver.firstName}} {{receiver.lastName}}.
c'est un message de test envoyé par {{sender.name}} de l'email {{sender.email}}
Cordialement,

CANCEL save or send immediately SAVE

Figure 6-20 : La campagne envoyées

La campagne créée est enregistrée à la liste des campagnes.

| | | | | | | |
|------------|----|-----------------|------|---------|--|--|
| 22/05/2017 | fr | Test de rapport | TEXT | CREATED | | |
|------------|----|-----------------|------|---------|--|--|

Figure 6-21 : état de la campagne

Après, le mail envoyé arrive aux destinataires avec une interprétation du contenu.

Test de rapport Boîte de réception x

noreply@tamtam.pro via amazones.com 20:58 (Il y a 5 minutes)
À moi

Salut abdessamad chanaa,
c'est un message de test envoyé par Abdessamad de l'email abdessamad139@gmail.com
Cordialement,

Figure 6-22 : campagne reçue

Cette interfaces présente les statistiques d'une campagne.

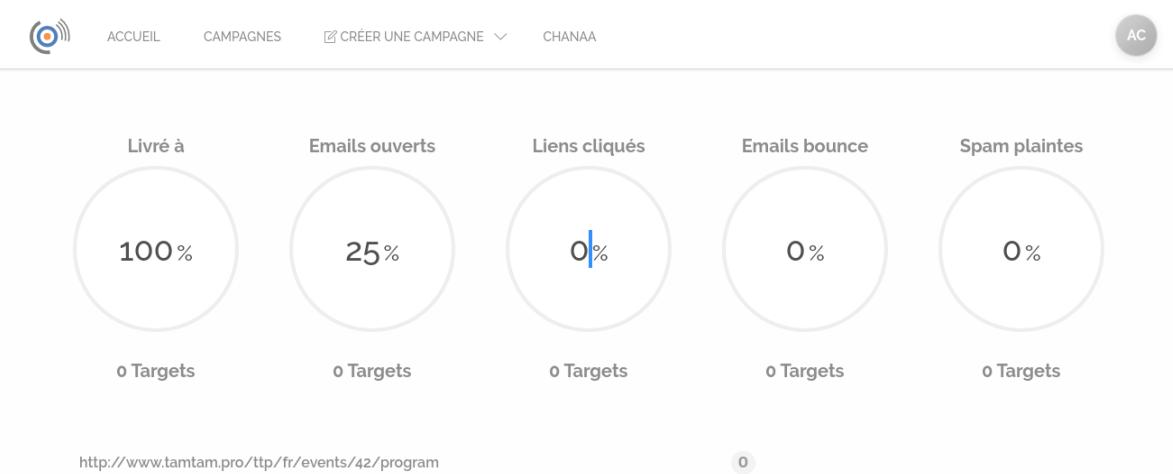


Figure 6-23 :statistique d'une campagne

D'autre part au cas d'erreur ou un lien invalide une redirection automatique est lancé vers la component InvalidPage.

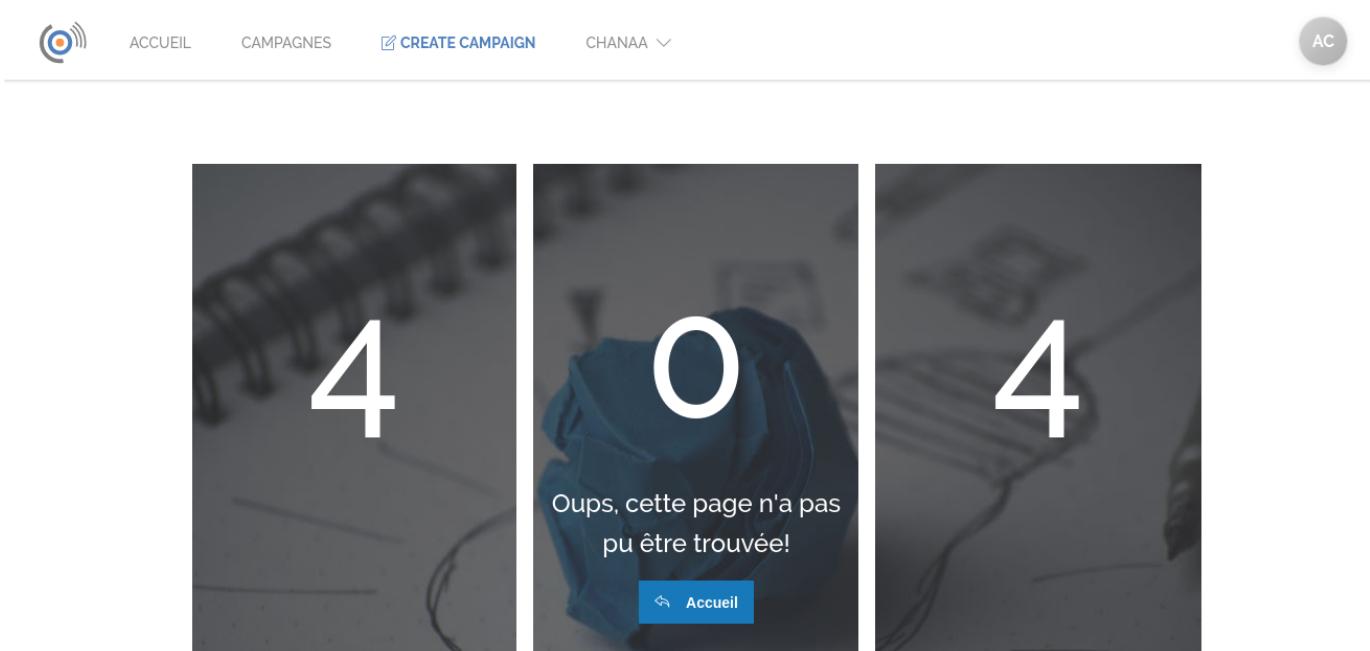


Figure 6-24 : page d'erreur

Pour une partie d'administration du système on peut créer un nouveau prospect, ainsi que l'affecter à un groupe d'envoi.

Ajouter un nouveau prospect

| | |
|--|--------------------|
| PRENOM * | NOM * |
| abdessamad | chanaa |
| EMAIL * | N° DE TELEPHONE * |
| | +32 (0)61 74 22 42 |
| FONCTION | LANGUE * |
| Comptable | Français |
| TITRE | SEXE |
| | Male |
| <input type="checkbox"/> groupe17 <input type="checkbox"/> ENSA <input type="checkbox"/> TAMAM | |
| <input type="button" value="ANNULER"/> <input type="button" value="ENREGISTRER"/> | |

Figure 6-25 : création de prospect

Conclusion

Dans ce chapitre, nous avons présenté la partie réalisation de notre projet. Nous y avons présenté et décrit le processus de l'organisation du travail, les outils de développement, ainsi que quelques interfaces de notre application.

Chapitre VII :

Conclusion Générale

Conclusion Générale

Notre stage s'est effectué dans des conditions opportunes, le soutien de mes encadrants et de l'équipe de TamTam a facilité mon intégration au sein de l'entreprise. Nous sortons très satisfaits des travaux réalisés. Les résultats obtenus sont très encourageants pour la suite du projet. L'étude que nous avons menée pendant ces quelques mois de stage, a abouti à la réalisation d'une première version du système. En outre la période de formation visant ma monté en compétences dans différentes technologies m'a permis de mieux saisir les concepts techniques, et par la suite être aussitôt opérationnel.

L'étude a démarré par une découverte de l'organisme qui nous accueillit pour notre stage et qui constitue le contexte du notre projet, après nous avons défini une méthodologie de travail pour nous conduire tout au long du projet et qui est en accord avec la structure d'accueil et son organisation. Le choix fut fait sur la méthode d'analyse et de conception basée sur UML guidée par le processus Scrum.

Une bonne étude préliminaire suivi par une étude fonctionnelle et technique au début de chaque sprint ont donné lieu à une bonne compréhension du projet. Ce qui nous a permis de bien élucider les besoins des utilisateurs et à dégager les fonctionnalités du système. Cette analyse du système a donné lieu à une phase de fusion des deux précédentes et a mené à la conception applicative et d'une solution adaptée aux besoins des utilisateurs. Elle concerne les étapes de la conception préliminaire, la conception détaillée, le codage et surtout les tests unitaires et fonctionnels puis l'étape de recette.

Nous avons rencontré au début des problèmes surtout avec l'utilisation des nouvelles technologies comme docker qui représente l'architecture de notre application et wercker qui permet d'automatiser les builds, le lancement des tests unitaires, les déploiements. Mais après Nous avons réussi à implémenter toutes les fonctionnalités assurées par API afin de servir notre application «Emailing» .

Du point de vue connaissance, ce stage nous a apporté beaucoup. Pour réaliser notre projet, nous avons découvert, dans un premier temps, toutes les technologies du domaine, nous avons effectué des choix et avons sélectionné notre boîte à outils. Dans une deuxième étape, nous avons eu l'occasion de manipuler toutes ces nouvelles technologies touchant à divers domaines comme l'aspect orienté service, l'architecture devops avec Docker, le design pattern Redux avec React... .

L'élaboration de ce travail nous a permis, d'une part, d'approfondir les connaissances et les avoir-faire acquis durant les années de ma formation à l'ENSA de Marrakech, ainsi de préparer notre intégration à la vie professionnelle et de nous se situer sur le marché de travail et découvrir la différence entre les projets professionnels et ceux à caractère pédagogique.

Webographie

codeproject :

<https://www.codeproject.com/Articles/1136943/Understanding-Retry-Pattern-with-Exponential-back>

documentation amazon :

<http://docs.aws.amazon.com/sns/latest/dg/DeliveryPolicies.html>

documentation symfony :

<http://symfony.com/doc/current/reference/constraints/Type.html>

documentation react :

<https://facebook.github.io/react/docs/lists-and-keys.html>

Loige :

<https://loige.co/transparent-pixel-response-with-symfony-how-to-track-email-opening/>

gist de github :

<https://gist.github.com/aquvelito/8596717>

stackoverflow :

<https://stackoverflow.com/questions/20903967/gmails-new-image-caching-is-breaking-image-links-in-newsletter?rq=1>

documentation doctrine :

<http://www.doctrine-project.org/api/common/2.3/class-Doctrine.Common.Collections.ArrayCollection.html>

react rocks :

<https://react.rocks/tag/DataTable>

Annexe I: Choix de Docker

1. introduction

Docker est un outil qui permet d'empaqueter une application dans un conteneur virtuel qui pourra s'exécuter sur n'importe quel serveur (Linux et Windows Server). C'est une technologie qui a pour but de faciliter les déploiements d'application, et la gestion du dimensionnement de l'infrastructure sous-jacente. Techniquement parlant, Docker utilise une approche totalement différente à l'ensemble de l'automatisation de l'infrastructure informatique. Au lieu de tirer parti de l'approche VM (Machine Virtuelle), ils prennent l'approche conteneur dans lequel vous pouvez réellement voir de très près le fonctionnement.

2. approche conteneur

Un conteneur est fondamentalement le nouveau terme de fantaisie autour du niveau du système d'exploitation de virtualisation. L'idée est de créer des instances dans un espace utilisateur isolés au lieu d'un seul. En d'autres termes, le partage du matériel est de rendre disponible de nombreux opérateurs sur le noyau lui-même plutôt que d'une autre couche comme le fait ESX, Hyper-V. Ces instances d'espace utilisateur sont souvent appelés VPS 88(serveurs privés virtuels) ou VE (Virtualisation Engine). Gardez à l'esprit que Docker n'est pas le seul à le faire. Il existe de nombreux projets à travers le même concept.

On peut dire qu'un conteneur est une sorte de boîte qui est totalement isolés du système d'exploitation dont laquelle on peut installer toutes les bibliothèques dont l'application a besoin pour fonctionner.

Si on veut parler d'un cas concret, c'est notre application qui a besoin d'une version 7 de PHP, besoin d'un serveur asynchrone nginx et d'autres dépendances, donc on se trouve devant une application qui demande beaucoup de dépendances. La solution c'est le système de conteneur qu'on met en disposition sous tamamp/php7.

Pour comparer avec la virtualisation classique, les hyperviseurs doivent créer une copie complète du système d'exploitation qui fonctionne sur un espace matériel virtuel.

L'hyperviseur est donc responsable de tous les échanges de données. Exécuter plusieurs machines virtuelles sur un même serveur demande de grosses performances et un nombre suffisant de ressources pour assurer plusieurs machines virtuelles.

3. c'est quoi l'avantage de se baser sur l'approche des conteneurs que que sur l'approche VM?

Vous pouvez penser Docker comme étant un gestionnaire de conteneurs pour les systèmes d'exploitation basés sur UNIX. Donc, fondamentalement, il vous permet de créer de nombreux environnements isolés sur un seul noyau comme le montre le schéma ci-dessous comparé au mode de virtualisation classique.

La virtualisation traditionnelle permet, via un hyperviseur, de simuler une ou plusieurs machines physiques, et les exécuter sous forme de machines virtuelles (VM) sur un serveur machine. Ces VM intègrent elles-mêmes un OS sur lequel les applications qu'elles contiennent sont exécutées. Ce n'est pas le cas du

container. Le container fait en effet directement appel à l'OS de sa machine hôte pour réaliser ses appels système et exécuter ses applications.

Historiquement, Docker repose sur le format de containers Linux, alias LXC. Il l'étend par le biais d'une API dans l'optique d'exécuter les applications dans des containers standards, qui sont donc portables d'un serveur Linux à l'autre.

Docker a 3 composants: Docker Daemon, Docker Images et Docker repositories. Docker Daemon fonctionne en tant que root et orchestre essentiellement tous les conteneurs. Docker images sont des images de système d'exploitation virtuels et techniquement ils ont de manière moins d'empreinte que les images d'OS réels. Les systèmes de fichiers sauvegardés sont empilés, ils peuvent donc être unifiés à la dernière révision à tout moment. Un peu comme des « snapshot » d'une VM, mais en mieux (comme un VCS).

En raison de la nature du système de fichiers toutes ces opérations d'écriture sont également mises en cache. À savoir que, si Docker découvre que vous ne disposez pas des changements jusqu'à une certaine révision, il joue le tout jusqu'à ce que la révision, notamment du cache en appliquant les écritures. Ceci donne un énorme avantage de performance sur le temps d'approvisionnement. En plus de tous ces avantages, les images peuvent aussi être échangées avec d'autres personnes. Dernière composante de Docker l'écosystème est le référentiel. Les images que vous détenez doivent être stockées quelque part, que ce soit privé ou public. Docker fournit alors un référentiel commun.

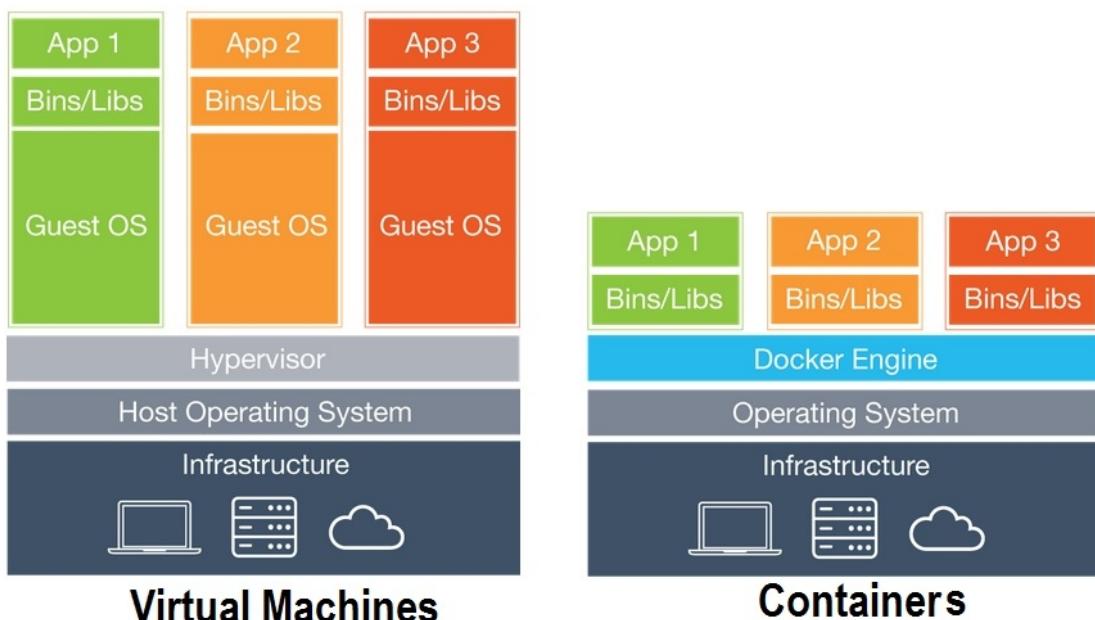


Figure annexe 1-1 : structure docker

4.conclusion

Docker est une des dernières technologies de virtualisation proposées au grand public. Il faut prendre en considération que Docker est amené à grandir encore de plus en plus même si le cas s'est déjà présenté.

Annexe II: Choix du PHP 7.0

1. Le choix du langage de programmation

Le choix du langage de programmation est un atout et n'est pas une chose à laisser au hasard. Ce choix doit être fait en se basant sur les exigences fixées auparavant. Le tableau ci-dessous présente une comparaison entre les langages de programmation (coté serveur) les plus populaires.

Le tableau ci-dessous présente une comparaison entre les langages de programmation (coté serveur) les plus populaires.

| | Java | PHP | Python | Ruby |
|--|------------------------------|-----------------------------------|-------------|---------------------------|
| <i>type</i> | Compilé Bytecode | Interprété | interprété | interprété |
| <i>Typage</i> | Statique, fort, Sûr | Dynamique | Dynamique | Dynamique |
| <i>Orienté Objet</i> | Oui | Oui | Oui | Purement oui ⁵ |
| <i>Compatibilité ascendante</i> | Oui | Pas toujours | Oui | Oui |
| <i>Multiplateforme</i> | Oui | Oui | Oui | Oui |
| <i>Gestion de la mémoire</i> | Automatique | Automatique | Automatique | Automatique |
| <i>Consommation de la mémoire</i> | Toute mémoire disponible ! | Moyenne | Moyenne | Moyenne |
| <i>Supporte UTF-8</i> | Oui | Oui | Oui | Oui |
| <i>Applications ciblés</i> | toutes | Web | Toutes | toutes |
| <i>Communauté</i> | Large | Large | Large | Moyenne |
| <i>Popularité⁶</i> | 17.822% | 2.868% | 3.999% | 1.359% |
| <i>rapidité⁷ des programmes</i> | Jusqu'à 10 fois plus Rapide | Moyen, rapide avec Zend Optimiser | moyen | moyen |
| <i>Syntaxe et grammaire</i> | Hérité de C | Hérité de C | Hérité de C | Similaire à perl |
| <i>Rapidité de programmation</i> | Moyen | Rapide | Très Rapide | Très rapide |
| <i>Frameworks et APIs</i> | Oui | Oui | Oui | Oui |
| <i>Open source</i> | Oui, mais propriété d'oracle | Oui | Oui | Oui |

Tableau annexe 2-1 : comparaison entre les langages de programmation

Pour ce qui est des langages de programmation, la différence est très nette. En effet, PHP7 est plus rapide de 243% par rapport à Ruby 2.1, 400% par rapport à Python 2.7.8 et 740% par rapport à Perl 5.18.4. Les gains en performances qu'apporte donc cette nouvelle version sont énormes, de plus, elle utilise beaucoup moins de ressources ce qui permettra d'héberger beaucoup plus de sites web sur le même serveur. Finalement, on a décidé d'utiliser PHP7 comme langage de programmation pour notre système puisque c'est le plus adaptable à nos attentes et nos exigences déjà fixées.

Le diagramme ci-dessous présent les performances d'un programme écrit dans les langages PHP, Ruby, Perl, Python.

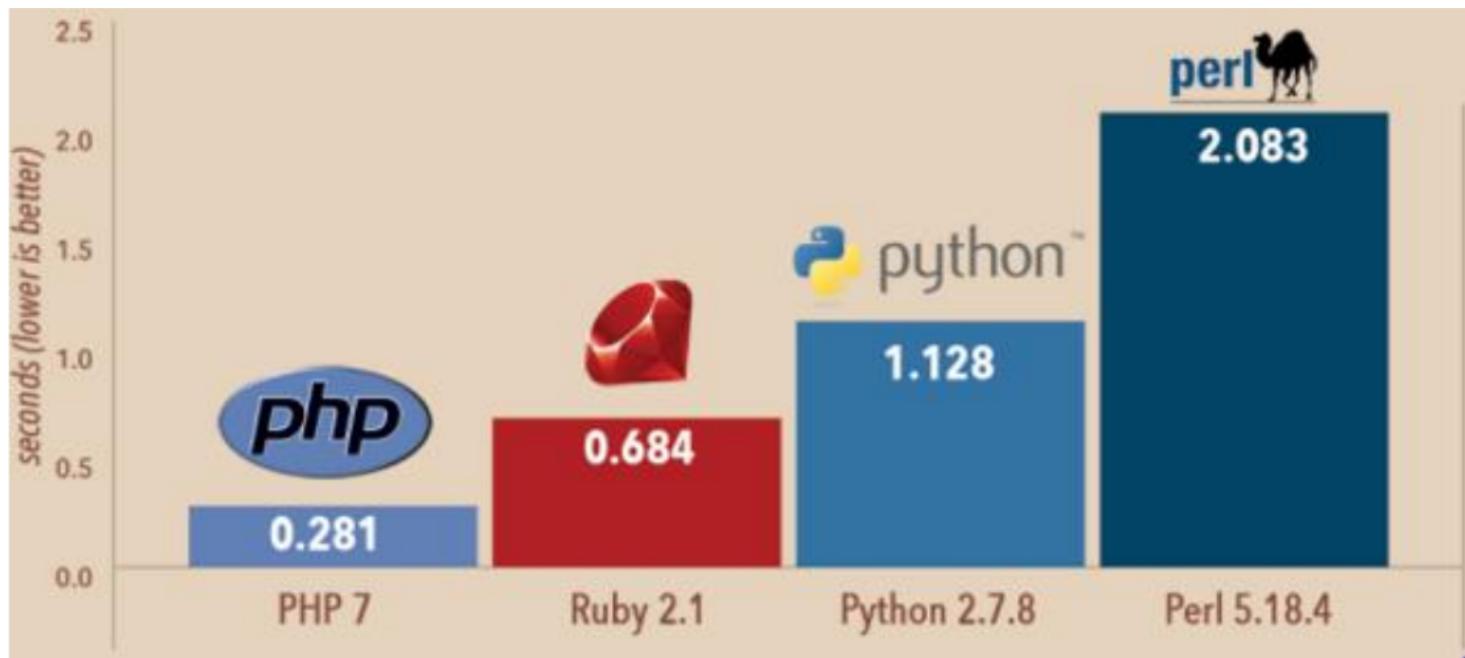


Figure annexe 2-1 : comparaison de performance entre les langages de programmation

Annexe III: Choix de React

React est une librairie pour construire des interfaces utilisateurs par composants. Il y a beaucoup d'autres outils comme Angular, Backbone, Knockout et Ember qui font des choses similaires mais en comparaison React est conçu pour résoudre des problèmes « business » plutôt que techniques. Vous trouverez ci-dessous les raisons pour lesquelles React est important et comment il peut bénéficier à toute entreprise d'ingénierie.

Réduire le risque

Stabilité—Facebook est fortement investi dans React (flux d'actualité, Instagram, Messenger, gestion des publicités, etc.) et dispose d'un grand nombre d'ingénieurs dédiés exclusivement au projet. Cet investissement massif n'est présent sur aucun autre framework concurrent. En complément des contributions de Facebook, il y a une gigantesque vague d'intérêt mondiale autour de React. Il y a actuellement 571 contributeurs au projet (en décembre 2015) ainsi qu'un calendrier de conférences et une cadence de livraison de mises à jour régulière.

React en Production (en): AirBnB, Asana, Atlassian, BBC, Chegg, CloudFlare, CNN.com, Codecademy, Coursera, Craftsy, Dailymotion, Dropbox, Expedia, Facebook, Feedly, Flipboard, HipChat, IMDb, Imgur, Instagram, Khan Academy, KISSmetrics, Mattermark, Minerva Project, Netflix, OkCupid, Rackspace, Rally Software, Ralph Lauren, Reddit, Redfin, Salesforce, Squarespace, The New York Times, Trunk Club, Twitter, Uber, University of Cincinnati, Venmo, WhatsApp, Wired, Wix, WordPress, Yahoo, Zendesk

Développement efficient

Mise en place progressive et facile—React peut être introduit dans des petites parties de pages existantes sans se soucier de l'existant ce qui permet aux développeurs de migrer à la vitesse de leur choix. Le poids de React est tellement léger (39.4kb gzipé pour React v0.14.0) qu'une migration progressive n'impactera nullement le poids de la page jusqu'à ce que les anciennes librairies soient inutiles et donc supprimées.

Performant par défaut—React utilise des modèles qui rendent très difficile d'écrire du code peu performant. De plus, dans la mesure où il élimine les interactions directes avec le DOM, il ne remplace pas seulement la manière de générer la vue (comme Angular / Backbone / Ember), mais aussi les dépendances comme jQuery, ce qui peut ainsi réduire la taille du code déployé.

Référencement —La prise en compte du SEO se fait par l'envoi d'une page complète calculée par le serveur directement au bot de recherche. React est aussi pensé pour le référencement, en calculant la page soit côté client soit côté serveur en utilisant NodeJS. D'autres outils permettant également le calcul côté serveur mais demandent un grand nombre de modifications instables et donc une contrainte forte en support et maintenance pour l'infogérance. React a le potentiel de simplifier les outils de développement et de réduire les coûts de maintenance.

Ré-utilisabilité du code accrue—React a la capacité unique de fournir des performances impressionnantes ainsi que de gérer intégralement le cycle de vie de rendu des composants. En intervenant sur la création, la composition et l'utilisation de composants isolés ré-utilisables plus simplement, les développeurs sont plus à même de gagner du temps en utilisant et créant des éléments communs. C'est aussi bien valable pour des éléments simples comme les boutons, que pour des éléments complexes comme du contenu dépliable.

Améliorer l'efficacité des développeurs

Réduit les effets de bord entre ressources concurrentes—React part du principe qu'HTML et JavaScript sont inévitablement liés et que les séparer n'a pas de logique autre que technique. Le concept peut être étendu aux CSS, ce qui retire les problèmes récurrents des espaces de nom, de portée et d'isolation des variables inhérents au développement CSS. Plus de détails : Radium (en) et React: CSS in js (en).

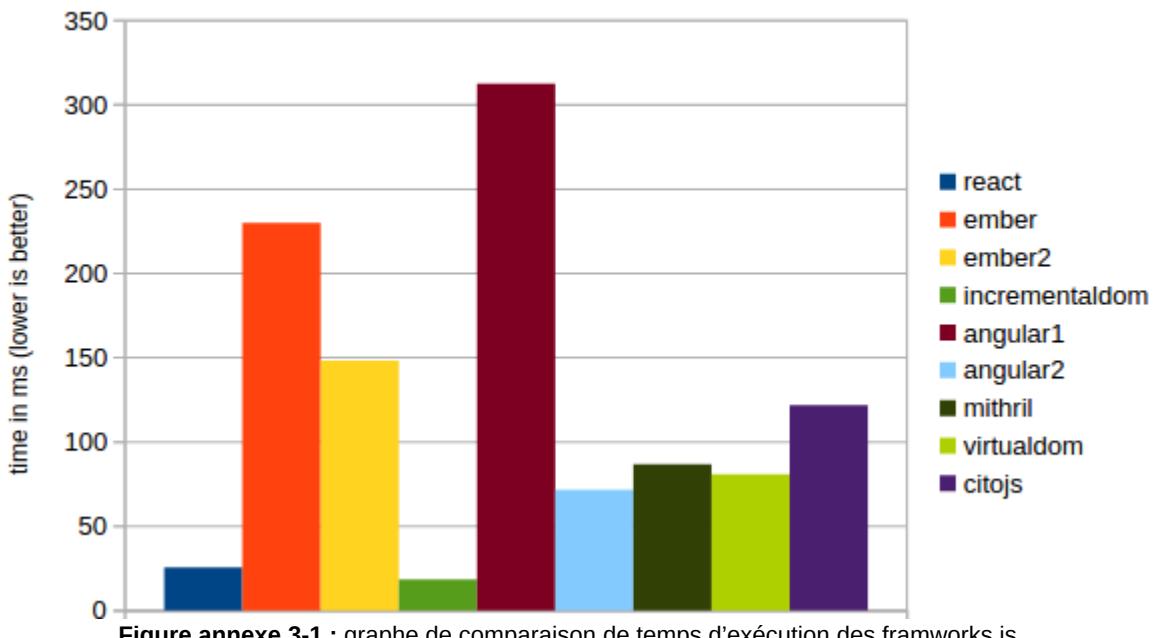
Détection plus rapide des erreurs—Facebook a fourni des outils spécifiques de développement intégrés au navigateur qui fournissent des détails sur les données et les composants utilisés pour générer un morceau d'interface. Voir cette conférence(en) pour plus d'informations.

Code facile à comprendre—Dans la plupart des outils actuels, les Modèles (qui représentent les données) et les Vues (qui affichent les données) sont liées pour produire ensemble des interfaces utilisateurs riches. Dans ce contexte une modifications sur une vue ou un modèle entraîne très souvent des modifications en cascade sur les autres vues qui dépendent des mêmes données. Dans une application d'une certaine taille, le graph de dépendances devient complexe et peut devenir difficile à modifier sans introduire des anomalies d'effets de bord. En React, les données circulent dans une seule direction, ce qui rend les vues plus compréhensibles. Ci-dessous, vous verrez des diagrammes représentant les deux types de flux.

Facilement testable—Un composant, dans le concept fondamental de React, prend en entrée des données et fourni en sortie une représentation du DOM sans effet de bord. Ces composants sont à la fois plus atomiques et testables, grâce au découplage des données et du rendu visuel.

Bénéfices organisationnels

Courbe d'apprentissage rapide—L'API de React est très petite. Combinée à sa syntaxe déclarative et sa subdivision en composants d'éléments d'interface, elle rend très rapide son apprentissage par les jeunes développeurs, spécialement ceux qui arrivent tout juste sur le marché du travail ou ceux qui ne sont pas familiers de l'environnement « front-end ».



Annexe IV: Envoi via Amazon

Cet annexe décrit ce qui se passe lorsque vous envoyez un courriel avec AmazonSES en passant par Amazon SNS et les différents résultats qui peuvent se produire après l'envoi du courrier électronique. La figure suivante est un aperçu de haut niveau du processus d'envoi :

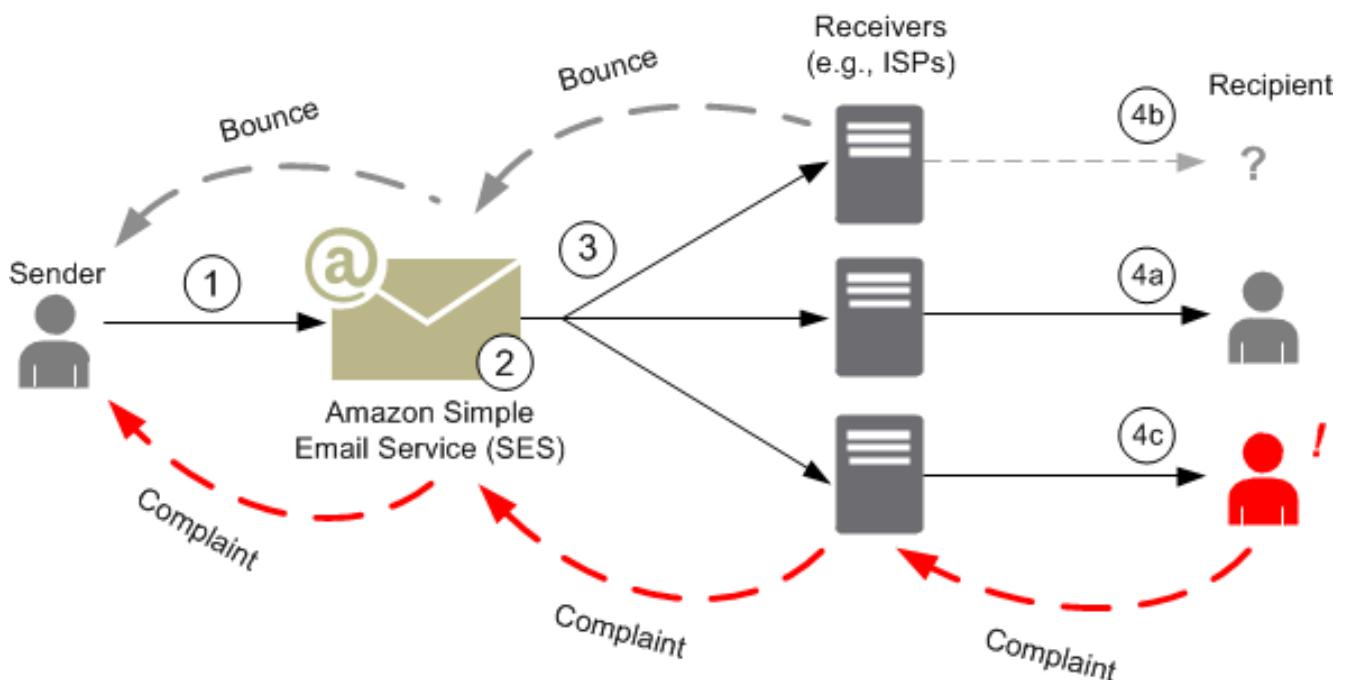


Figure annexe 4-1 : l'envoi de message via amazon SES

1- une application client, en tant qu'émetteur de message , demande à Amazon SES d'envoyer un courrier électronique à un ou plusieurs destinataires.

2- si la demande est valide, Amazon SES accepte le courrier électronique et l'envoi par Internet au destinataire. Une fois que le message est transmis à Amazon SES, il est généralement envoyé immédiatement, la première tentative de livraison se produisant normalement en millisecondes.

3- à ce stade, il existe différentes possibilités. Par exemple :

Le message délivré dans la boîte de réception du destinataire.

L'adresse email du destinataire n'existe pas, de sorte que Amazon SNS envoie une notification de rebond à Amazon SES. Amazon SES transmet la notification à l'expéditeur.

Le destinataire reçoit le message mais le considère comme du spam et enregistre une plainte auprès du Amazon SNS. Amazon SES envoie la plainte à Amazon SES, qui l'envoie ensuite à l'expéditeur.

Les sections suivantes examinent les résultats individuels possibles après qu'un expéditeur envoie une demande par courrier électronique à Amazon SES et après que Amazon SES envoie un message électronique au destinataire.

1-Après qu'un expéditeur envoie une demande de courrier électronique à Amazon SES

Lorsque l'expéditeur demande à Amazon SES d'envoyer un courrier électronique, l'appel peut réussir ou échouer. Les sections suivantes décrivent ce qui se passe dans chaque cas.

1.1. Demande d'envoi réussie

Si la demande à Amazon SES réussit, Amazon SES renvoie une réponse réussie à l'expéditeur. Ce message comprend l'ID du message, une chaîne de caractères qui identifie de manière unique la demande. Vous pouvez utiliser l'ID du message pour identifier le courrier électronique envoyé ou pour suivre les problèmes rencontrés lors de l'envoi. Amazon SES assemble ensuite un message électronique en fonction des paramètres de la demande, analyse le message pour contenir des contenus et des virus douteux, puis l'envoie par Internet à l'aide de SMTP (Simple Mail Transfer Protocol). Votre message est généralement envoyé immédiatement ; La première tentative de livraison se produit généralement en millisecondes.

1.2. Requête d'envoi échouée

Si la demande d'envoi de courrier électronique de l'expéditeur à Amazon SES échoue, Amazon SES répond à l'expéditeur avec une erreur et dépose le courrier électronique. La demande pourrait échouer pour plusieurs raisons. Par exemple, la demande peut ne pas être formatée correctement ou l'adresse email peut ne pas avoir été vérifiée par l'expéditeur.

La méthode par laquelle vous pouvez déterminer si la demande a échoué dépend de la façon dont vous appelez Amazon SES. Voici des exemples de la façon dont les erreurs et les exceptions sont renvoyées :

- ◆ Si vousappelez Amazon SES via l'API, les actions renverront une erreur. Pour plus d'informations, reportez-vous à la Référence de l'API Amazon Simple Email Service.
- ◆ Si vous utilisez un SDK AWS pour un langage de programmation utilisant des exceptions, l'appel vers Amazon SES lancera une MessageRejectedException. (Le nom de l'exception peut varier légèrement en fonction du SDK.)
- ◆ Si vous utilisez l'interface SMTP, l'expéditeur reçoit un code de réponse SMTP, mais la façon dont l'erreur est transmise dépend du client de l'expéditeur. Certains clients peuvent afficher un code d'erreur; D'autres peuvent ne pas.

Pour plus d'informations sur les erreurs qui peuvent survenir lorsque vous envoyez un courriel avec Amazon SES, consultez Amazon SES.

2. Après que Amazon SES envoie un courrier électronique

Si la demande de l'expéditeur à Amazon SES réussit, Amazon SES envoie le courrier électronique et l'un des résultats suivants se produit :

La livraison réussie et le destinataire ne s'opposent pas au courrier électronique: le courrier électronique est accepté par le SNS et transmet le courrier électronique au destinataire. Une livraison réussie est indiquée dans la figure suivante.

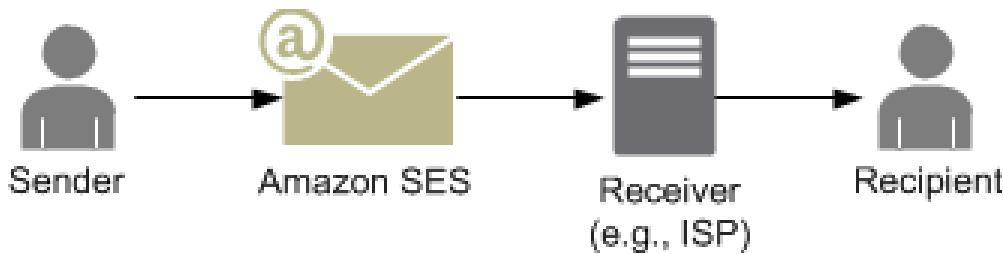


Figure annexe 4-2 : envoi réussie

Hard bounce : le courrier électronique est rejeté en raison d'une condition persistante ou rejeté par Amazon SES, car l'adresse électronique est sur la liste de suppression SES d'Amazon. Une adresse e-mail est sur la liste de suppression de SES d'Amazon si elle a récemment provoqué un rebond difficile pour tout client Amazon SES. Un rebond difficile peut se produire, car l'adresse du destinataire est invalide. Une notification de rebond difficile est envoyée à Amazon SES, qui notifie l'expéditeur par courrier électronique ou via Amazon Simple Notification Service (Amazon SNS), selon la configuration de l'expéditeur. Amazon SES avise que l'expéditeur de la liste de suppression rebondit par les mêmes moyens. Le chemin d'un rebondissement dur d'un FAI est montré dans la figure suivante.

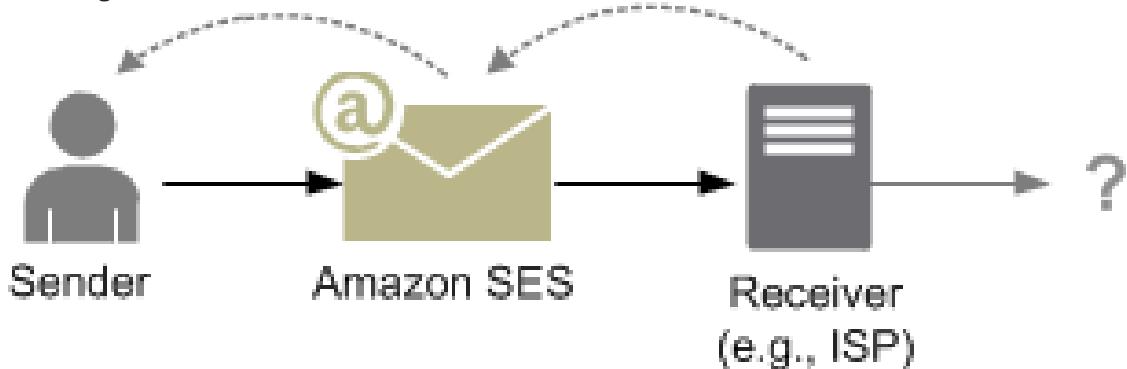


Figure annexe 4-3 : hard bounce

Soft Bounce : le fournisseur d'accès Internet ne peut pas envoyer le courrier électronique au destinataire en raison d'une condition temporaire, comme le FAI est trop occupé pour gérer la demande ou la boîte aux lettres du destinataire est pleine. Un rebond doux peut également se produire si le domaine n'existe pas. On envoie une notification de rebond doux à Amazon SES ou, dans le cas d'un domaine inexistant, Amazon SES ne peut pas trouver un serveur de messagerie pour le domaine. Dans les deux cas, Amazon SES tente le courrier électronique pendant une période prolongée. Si Amazon SES ne peut pas envoyer le courrier électronique pendant cette période, il vous envoie une notification de rebond via un courrier électronique ou via Amazon SNS. Si Amazon SES peut envoyer le courrier électronique au destinataire lors d'une nouvelle tentative, la livraison est couronnée de succès. Un rebond doux est illustré dans la figure suivante. Dans ce cas, Amazon SES tente d'envoyer le courrier électronique, et SES finalement capable de le transmettre au destinataire.

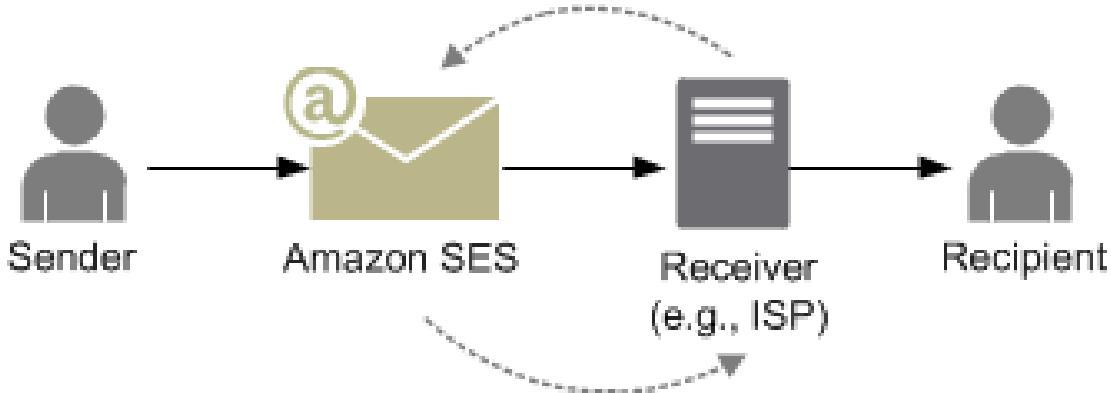


Figure annexe 4-5 : soft bounce

Complaint (spam) : le courrier électronique est accepté par le FAI et livré au destinataire, mais le destinataire considère que le courrier électronique est du spam et clique sur un bouton tel que « Marquer comme spam » dans son client de messagerie. Si Amazon SES dispose d'une boucle de retour avec le FAI, une notification de plainte est envoyée à Amazon SES, qui transmet la notification de la plainte à l'expéditeur. La plupart des FAI ne fournissent pas l'adresse électronique du destinataire qui a soumis la plainte, de sorte que la notification de plainte d'Amazon SES fournit à l'expéditeur une liste de destinataires qui ont pu envoyer la plainte, en fonction des destinataires du message original et du FAI à partir de laquelle Amazon SES a reçu la plainte. Le chemin d'une plainte est indiqué dans la figure suivante.

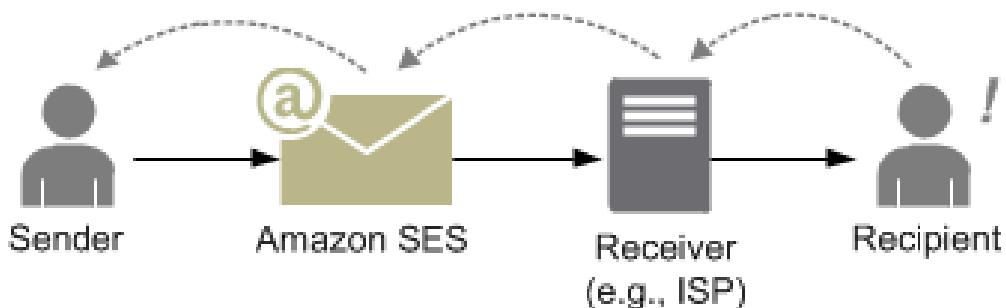


Figure annexe 4-5 : plainte