

UNIVERSITE ASSANE SECK DE ZIGUINCHOR



L'excellence, ma référence

UFR DES SCIENCES ECONOMIQUES ET SOCIALES

Département Economie – Gestion

Mémoire de fin d'études Pour l'obtention du diplôme de Master
Management des Systèmes d'Information Automatisés (MSIA)

Conception et Développement d'une Plateforme de Gestion des évènements scientifiques avec Diffusion en Ligne organisés à l'Université Assane Seck de Ziguinchor.

Présenté par :

M. Libasse THIAM

Le 06 janvier 2024 à la salle Visio-conférence de l'UASZ

Sous la direction de

Dr Elhadj Malick NDOYE

Maitre-Assistant Associé à l'UASZ

M. Malaw NDIAYE

Assistant associé à l'UASZ

Sous supervision de

Pr Ousmane DIALLO

Maitre de conférences à l'UASZ

Membres du jury

Pr. Melyan MENDY

Maitre de conférences agrégé à l'UASZ

Président

Pr Ousmane DIALLO

Maitre de conférences à l'UASZ

Examineur

Dr Thierno Ahmadou DIALLO

Maitre-Assistant Associé à l'UASZ

Rapporteur

Dr Elhadj Malick NDOYE

Maitre-Assistant Associé à l'UASZ

Encadrant

M. Malaw NDIAYE

Assistant associé à l'UASZ

Co-encadrant

Année universitaire 2022 – 2023

Master MSIA

Résumé

Les journées scientifiques, colloques, et événements organisés au sein des écoles doctorales de l'Université Assane Seck jouent un rôle essentiel pour ces structures. En d'autres termes, ils contribuent significativement à la diffusion de la recherche en offrant aux doctorants l'opportunité de présenter leurs travaux, qu'ils soient en cours ou achevés. De plus, ces activités représentent des éléments cruciaux pour le bon fonctionnement des écoles doctorales, préparant ainsi les doctorants à leur future carrière, leur offrant une visibilité, leur permettant de s'évaluer, et leur donnant l'occasion de recevoir des retours constructifs.

La gestion et l'organisation de ces événements nécessitent une allocation substantielle de ressources et de temps. Le processus débute par un appel à communication, suivi de la réception des propositions de communication. Ensuite, intervient l'étude et la validation des intervenants retenus, suivies par les invitations aux participants. Ainsi la participation effective peut alors nécessiter une diffusion en ligne, cela faisant recours à des outils de visio-conférence.

Ce mémoire se propose d'analyser les limitations inhérentes à la gestion de ces événements, et de proposer une solution visant à améliorer, automatiser et centraliser le processus d'organisation, tout en offrant une meilleure efficacité. Cette étude couvre la conception et la mise en place d'une application centralisée pour la gestion et l'organisation des événements au sein des écoles doctorales de l'UASZ. Elle consiste à établir un système permettant de planifier, partager les événements, diffuser les appels à communication, permettre aux utilisateurs de soumettre leurs propositions de communication, partager leurs articles scientifiques, ainsi que participer à distance à ces événements grâce à des solutions de visioconférence.

La réalisation de ce travail fit recourue à des outils technologiques permettant de répondre efficacement aux besoins et exigences de la solution proposé. Ainsi trois technologies en générale ont été utilisé, à savoir :

- Laravel, pour le développement des fonctionnalités de gestion des événements et conférences ainsi que des utilisations et d'articles scientifiques.
- WebRTC, pour le développement des fonctionnalités de visio-conférence et la communication à temps réel avec les utilisateurs.
- Et MySQL, pour la base de données de notre application.

Mots clés : conférence, évènements, technologie, WebRTC, Laravel.

Abstract

The scientific days, symposiums, and events organized within the doctoral schools of Assane Seck University play a crucial role for these institutions. In other words, they significantly contribute to the dissemination of research by providing doctoral students with the opportunity to present their ongoing or completed work. Moreover, these activities represent crucial elements for the proper functioning of doctoral schools, preparing students for their future careers, offering visibility, enabling self-assessment, and providing an opportunity to receive constructive feedback.

The management and organization of these events require a substantial allocation of resources and time. The process begins with a call for papers, followed by the reception of communication proposals. Next comes the study and validation of selected participants, followed by invitations to attendees. Thus, effective participation may require online dissemination, utilizing video conferencing tools.

This thesis aims to analyze the inherent limitations in the management of these events and proposes a solution to enhance, automate, and centralize the organizational process, ensuring improved efficiency. This study encompasses the design and implementation of a centralized application for the management and organization of events within Assane Seck University's doctoral schools. It involves establishing a system for planning, sharing events, broadcasting calls for papers, allowing users to submit communication proposals, share scientific articles, and participate remotely in these events through video conferencing solutions.

The completion of this work involved the use of technological tools to effectively meet the needs and requirements of the proposed solution. Three technologies were employed in general:

- Laravel, for the development of event and conference management functionalities, as well as user and scientific article management.
- WebRTC, for the development of video conferencing functionalities and real-time communication with users.
- MySQL, for the database of our application.

Keywords: conference, events, technology, WebRTC, Laravel.

Dédicace

À ma très chère maman, Alima THIAM, ma source de motivation, mon amour éternel.

À mon très chère papa Madiagne THIAM, mon idole, mon guide, celui qui a éclairé mon chemin et me guide dans tous ce que j'entreprends.

Je dédie ce mémoire.

Remerciements

Je tiens tout d'abord à exprimer ma gratitude envers le Tout-Puissant pour avoir fait de moi ce que je suis aujourd'hui.

J'aimerais remercier chaleureusement les membres du jury pour leur disponibilité et le temps qu'ils ont consacré à l'évaluation de ce travail. Mes remerciements les plus sincères vont à mes encadreurs, Monsieur Malick NDOYE et Monsieur Malaw NDIAYE, qui m'ont accompagné jusqu'au bout et soutenu dans l'achèvement de ce travail.

Je souhaite également exprimer ma reconnaissance envers tous mes professeurs, depuis la licence jusqu'au master. Je tiens à adresser des remerciements particuliers à notre coordinateur, Monsieur Cheick Oumar BALDE, qui n'a ménagé aucun effort pour la réussite des étudiants de la MSIA.

Un grand merci à tous mes camarades de classe, depuis la licence, avec qui j'ai travaillé, révisé et surmonté les défis. Je ne saurais oublier de remercier mon tuteur à Ziguinchor, Monsieur Adama NDIAYE, ainsi que toutes les personnes qui m'ont accueilli à Ziguinchor, en particulier mon ami Alassane MANSALI et sa mère, chez qui j'ai résidé pendant mes premiers jours à Ziguinchor.

Enfin, je tiens à exprimer ma reconnaissance infinie à mon très cher papa, Madiagne THIAM, ainsi qu'à tous les membres de ma famille qui m'ont apporté leur soutien, et à mes amis qui ont été là pour moi.

SOMMAIRE

Résumé	i
Abstract	ii
Dédicace	iii
Remerciements	iv
SOMMAIRE	v
Liste des tableaux.....	viii
Liste des images et figures	ix
Liste des sigles et abréviations	xi
Introduction générale	1
1. Contexte	2
2. Problématique	2
3. Objectifs	3
4. Présentation du plan	3
Chapitre 1 : Etat de l'art et présentation de notre proposition	5
I. Introduction	6
II. Concepts fondamentaux	6
1. Notion de la gestion d'évènements	6
2. Evolution des méthodes de gestion d'évènements	6
III. Étude des applications de gestion d'évènements	7
1. EDAS	7
2. EasyChair	8
3. COMS	9
4. OpenConf	10
5. WikiCfp	11
6. Tableau récapitulatif	12
IV. Présentation de notre solution	13
V. Analyse des technologies utilisées dans la conception et le développement de l'application 14	
1. Outils de gestion de projet informatique	15
2. Technologies pour le développement backend	19
3. Technologies pour le développement frontend	19
VI. Conclusion	20
Chapitre 2 : Méthodologie et choix technologiques.....	21
I. Introduction	22

II.	Description de la méthodologie utilisée	22
1.	La méthodologie traditionnelle (cascade)	22
2.	La méthodologie Agile	25
3.	Méthodologie choisie	28
III.	Choix technologique	29
1.	Architecture	29
2.	Backend	32
3.	Frontend	34
4.	Choix	35
IV.	Conclusion	36
Chapitre 3 : Analyse et conception de l'application.....		37
I.	Introduction	38
II.	Cahier des charges	38
1.	Objectifs	38
III.	Analyse des besoins et des fonctionnalités	39
1.	Besoins fonctionnels	39
2.	Besoins non fonctionnels	40
IV.	Conception	41
1.	Diagramme de cas d'utilisation	41
2.	Diagramme de séquence	42
3.	Diagramme des classes	45
V.	Conclusion	47
Chapitre 4 : Développement et présentation de l'application		48
I.	Introduction	49
II.	Développement de l'application	49
1.	Développement avec Laravel	49
2.	Système d'authentification avec Laravel Fortify	50
3.	Les Policy dans Laravel	51
4.	Moteur de templating blade	52
5.	Webrtc	52
III.	Présentation de l'application	56
1.	Présentation des interfaces	56
2.	Limites de l'application	65
IV.	Conclusion	65
Chapitre 5 : Tests et déploiement de l'application.....		67

I. Introduction	68
II. Description des tests effectués	68
1. Teste de base dans Laravel	68
2. Test de publication d'un appel à communication par l'administrateur	69
3. Test de l'envoi d'une proposition de communication	70
4. Test gestion d'article	71
III. Déploiement de l'application	71
1. Le déploiement manuel	71
2. Déploiement automatisé	73
IV. Conclusion	73
Conclusion générale	75
Bibliographie et webographie.....	a
Table des matières	d

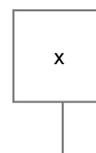
Liste des tableaux

Table 1 Récapitulatif des solutions similaires.....	13
Table 2 Récapitulatif langage et outils backend	19
Table 3 Récapitulatif langage et outils frontend	19
Table 4 Avantage et inconvénients méthodologie séquentielle	25

Liste des images et figures

Figure 1 : illustration de TeamGant.....	15
Figure 2 : Interface Asana	16
Figure 3 : Interface Trello.....	17
Figure 4 : Interface Notion	18
Figure 5 : illustration gestion de projet [12].	22
Figure 6 : illustration cycle en V	23
Figure 7 : Méthode scrum.....	27
Figure 8 Méthode kanban	28
Figure 9 : architecture micro services	30
Figure 10 architecture SOA	31
Figure 11 : architecture Monolithique.....	32
Figure 12 : logo PHP	33
Figure 13 : logo java	33
Figure 14 : logo python	34
Figure 15 Stack technologique	36
Figure 16 Tableau kanban.....	40
Figure 17 : use case utilisateur simple.....	41
Figure 18 use case Administrateur	42
Figure 19 diagramme de séquence authentification	43
Figure 20 diagramme de séquence cas publication d'appelle à communication.....	43
Figure 21 diagramme de séquence cas gestion propositions de communication	44
Figure 22 diagramme de séquence cas inscription événement	44
Figure 23 diagramme de séquence envoyer une proposition de communication.....	45
Figure 24 diagramme de classe.....	46
Figure 25 logo Laravel.....	50
Figure 26 Webrtc communication[23]	55
Figure 27 page inscription	56
Figure 28 pages connexion	57
Figure 29 pages publication appelle à communication	57
Figure 30 pages liste appelle à communication	58
Figure 31 pages liste des soumissions pour un appel	58
Figure 32 pages étudier une proposition.....	59
Figure 33 pages création salon de conférence	59
Figure 34 pages paramètres de l'appelle en ligne	60
Figure 35 pages visioconférence	61
Figure 36 pages gestion des utilisateurs.....	61
Figure 37 pages liste des événements	62
Figure 38 pages détails d'un événement	62
Figure 39 pages soumettre une proposition de communication.....	63
Figure 40 pages publication d'article	63
Figure 41 pages articles publié	64
Figure 42 pages tous les articles	64
Figure 43 pages articles	65
Figure 44 résultat premier test.....	69
Figure 45 résultat test publication appelle.....	70

Figure 46 résultat test envoie d'une proposition de communication.....	70
Figure 47 résultat test gestion d'articles.....	71
Figure 48 Filezilla clients	73



Liste des sigles et abréviations

COMS	Conférence Management System en anglais
UASZ	Université assane Seck de Ziguinchor
API	Application programming interface
PHP	PHP HyperText Préprocesseur
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
Sass	Syntactically awesome stylesheets
WIP	Work In Progress
SOA	Service-Oriented Architecture
WORA	Write once, run anywhere
UI	User Interface
UX	User Experience
POO	Programmation Orientée Objet
MVC	Model vue controller
UML	Unified Modeling Language
WebRTC	Web Real-Time Communication
ORM	Object Relational Mapping
AJAX	Asynchronous JavaScript + XML
SDP	Session Description Protocol
ICE	Interactive Connectivity Establishment
NAT	Network Address Translation
FTP	File Transfer Protocol
AWS	Amazon Web Services
GCP	Google Cloud Platform
SFTP	Secure File Transfer Protocol
EDAS	EDitor's Assistant

Introduction générale

1. Contexte

L'organisation d'événements tels que les conférences, les ateliers et les colloques occupe une place centrale au sein des écoles doctorales de l'Université Assane Seck de Ziguinchor. Ces institutions, à savoir l'école doctorale Espaces, Sociétés et Humanités, et l'école doctorale Sciences, Technologies et Ingénierie, ..., jouent un rôle crucial dans la promotion des activités scientifiques et dans la préparation des doctorants à travers des échanges, le partage d'expériences et la recherche scientifique.

Dans le cadre de ces structures, une variété d'activités et d'événements scientifiques sont régulièrement planifiés, visant à garantir le bon déroulement des programmes académiques et à faciliter le développement des compétences des doctorants. Ces événements sont conçus pour encourager l'échange d'idées, la collaboration et l'approfondissement des connaissances.

Pour piloter l'organisation de ces événements, l'école utilise très souvent des outils tels que les courriels pour les appels à communications et les soumissions des potentiels communicants, et parfois la plateforme de gestion de conférences en ligne EasyChair. Le canal privilégié demeure toutefois l'envoi de courriels, tant pour les appels à communication que pour la collecte des propositions.

2. Problématique

Dans le contexte où l'orchestration de ces activités fait appel à une communication électronique via l'utilisation de courriels et de plateformes telles qu'EasyChair, EDAS, ..., certaines limites ont été identifiées. Tout d'abord, on constate l'absence d'un environnement centralisé pour la gestion des événements. En effet, jusqu'à présent, la gestion des appels et des soumissions de communications, ainsi que la diffusion, se fait dans des systèmes séparés. De plus, il existe une absence de traçabilité des données ou informations liées aux événements, car il n'y a pas de système dédié aux écoles doctorales permettant non seulement la gestion de leurs activités, mais aussi la collecte et le partage des communications (abstract), articles et revues scientifiques liés à chaque activité, ainsi que les différentes présentations des intervenants durant ces événements.

Face à ces problèmes soulevés, nous nous sommes posé les questions suivantes afin de réfléchir à une solution tout-en-un qui saura assurer une meilleure gestion des activités des structures :

- Comment remédier à l'absence d'un environnement centralisé pour la gestion des événements ?
- Comment assurer la traçabilité des données liées aux événements et mettre en place un système dédié aux écoles doctorales ?

3. Objectifs

La réponse aux différentes questions évoquées dans notre problématique facilitera les appels à communication, la soumission des propositions de communications, la gestion des inscriptions et des paiements, la diffusion des événements ainsi que le partage d'articles et de revues scientifiques. Ainsi, l'objectif général de ce projet est de développer une application de gestion de conférences avec diffusion en ligne, permettant aux écoles doctorales de mieux gérer leurs événements grâce à une solution centralisée. Cette application offrira un système d'information destiné à améliorer le processus de gestion événementielle ; elle assurera également la diffusion en ligne de ces activités. De plus, elle offrira la possibilité de stocker et de partager les présentations ainsi que la collecte d'informations liées aux différentes activités, tout en facilitant l'accès aux articles scientifiques pour les participants.

4. Présentation du plan

Ce mémoire présente l'étude de cas portant sur la conception et le développement d'une application destinée à l'École Doctorale de l'Université Assane Seck de Ziguinchor. Il se divise en cinq (5) chapitres, à savoir :

- État de l'art :

Dans cette première partie, nous effectuerons une analyse approfondie des applications et solutions similaires, tout en présentant notre solution. Cette section inclura également une analyse des technologies utilisées.

- Description de la méthodologie et des choix technologiques :

La deuxième partie expose la méthodologie de gestion de notre projet ainsi que les choix technologiques effectués.

- Analyse et conception de l'application :

La troisième partie se concentre sur l'analyse des besoins et des fonctionnalités de l'application, ainsi que sur la conception de notre solution.

- Développement de l'application :

La quatrième partie détaille le processus de développement de l'application.

- Tests et déploiement :

Enfin, la cinquième partie est consacrée aux tests effectués et au déploiement de l'application.

Chapitre 1 : Etat de l'art et présentation de notre proposition

I. Introduction

Ce présent chapitre est consacré à l'étude de l'état de l'art dans le domaine de la gestion de conférences et de la diffusion en ligne. L'objectif principal de ce chapitre est de fournir une vue d'ensemble des applications similaires existantes et d'analyser en détail les technologies qui ont été utilisées pour concevoir et développer ces applications. Cette analyse approfondie nous permettra de mieux appréhender le contexte dans lequel s'inscrit notre projet de développement d'une application de gestion de conférences centralisée avec diffusion en ligne pour les écoles doctorales de l'Université Assane Seck de Ziguinchor.

II. Concepts fondamentaux

1. Notion de la gestion d'évènements

La gestion d'évènement englobe tout le processus de planification, d'organisation et de coordination d'une activité prévue. Elle regroupe plusieurs aspects y compris, la communication, la logistique la budgétisation, la coordination des intervenant, le suivi et l'évaluation. C'est un parcours complexe qui demande une approche stratégique dans sa globalité. En d'autres termes, elle consiste d'abord à définir clairement les objectifs de l'évènement, les détails de l'évènement, ceci est l'étape de la planification préalable. Ensuite, intervient la coordination incluant la communication marketing et la gestion des invitations et la coordination des orateurs. En fin, l'étape d'évaluation post-évènement complète permettant d'identifier les succès et les domaines d'amélioration.

2. Evolution des méthodes de gestion d'évènements

D'un point de vue globale, les évènements tels que nous les connaissons sont en pleine expansion. En effet, pour toute entité, l'organisation d'évènement devient une nécessité et cette nécessité peut s'observé soit dans le besoin de communication et de se faire connaître ou bien soit dans le besoin d'améliorer l'entité en interne ou en externe. Ainsi, pour assouvir ces besoins par le biais des évènements, les organisations ou structures ont tendance à mettre en place des stratégies pour une meilleure gestion de ces derniers.

Par ailleurs, l'émergence de la technologie a modifié la manière dont les événements sont gérés. Les organisateurs ont intégré des solutions numériques afin de simplifier la planification, la promotion, et la coordination des événements. Des plateformes d'inscription en ligne, des

applications mobiles, et des logiciels de gestion intégrée facilitent une gestion plus efficace, tout en favorisant une communication en temps réel avec les participants [1].

De ce fait, dans le cadre de l'organisation des activités tels que les colloques, journées scientifiques etc., organisé par les écoles doctorales de l'université Assane Seck de Ziguinchor, des solutions numériques ont été utilisées jusque-là pour gérer ces derniers. Il est à noter, comme mentionné du contexte de ce présent mémoire que pour la plupart du temps, le courriel électronique de l'université ou les plateformes de gestion de conférences sont très souvent utilisées pour la gestion de ces activités. Cependant, En examinant les besoins compte tenu des notions de la gestion d'évènement, nous avons notés quelques limites dans l'utilisation de ses outils susmentionnées. Ces limites ainsi que l'évolution des méthodes de gestion d'évènement sont d'ailleurs les éléments qui ont favorisé notre réflexion sur le fait de proposer une plateforme de gestion des événements avec diffusion en Ligne.

Ainsi, nous allons donc faire une étude sur les solutions similaires en termes de besoins pour enfin montrer les limites par rapport aux besoins fonctionnels.

III. Étude des applications de gestion d'évènements

Dans cette partie, nous allons étudier les applications existantes utilisées dans le domaine de la gestion de conférences et de la diffusion en ligne. Nous examinerons les fonctionnalités offertes par ces applications, les avantages qu'elles offrent ainsi que les éventuelles limites qu'elles présentent par rapport à notre contexte d'étude. Cette analyse comparative nous permettra de déterminer la pertinence de notre application pour l'école doctorale de l'université Assane Seck de Ziguinchor.

1. EDAS

EDAS signifie **EDitor's Assistant**, c'est une solution qui prend en charge le cycle de vie complète de la gestion de conférence soumission et révision, discussion et décision, inscription, droits d'auteur et subventions de voyage, CD-ROM, programme et badges [2].

Elle est aussi utilisée pour la gestion des revues mais de base elle est conçue pour les conférences.

a. Fonctionnalités

- Processus de soumission
- Révision et d'inscription des articles pour les conférences

- Ateliers et revues

b. Avantages

EDAS présente des avantages en tant que service hébergé, ce qui signifie qu'aucun logiciel n'a besoin d'être installé. De plus, le personnel d'assistance est disponible pour aider les auteurs, les réviseurs et les présidents en cas de problème. Les utilisateurs peuvent interagir avec EDAS en utilisant des navigateurs Web standards.

a. Limites

Les frais associés à l'utilisation d'EDAS comprennent des coûts variables en fonction du nombre d'articles soumis. Les cents (100) premiers articles sont tarifés à six dollars (6\$) chacun, tandis que les articles suivants sont facturés à cinq dollars (5\$) chacun (avec un minimum de soixante dollars). Des frais supplémentaires de dix dollars (10\$) s'appliquent pour chaque correction PDF demandée afin de garantir la conformité IEEE/ACM, à condition que plus de deux pour cent (2%) des documents nécessitent des ajustements tels que l'intégration de polices PDF ou la suppression de liens.

De plus, des frais de retard de 1,5 % par mois sont imposés à partir de 30 jours calendaires après le début de la conférence. Des coûts supplémentaires peuvent survenir pour des services spécifiques, tels que des tutoriels EDAS et des formations aux fonctionnalités spéciales, facturés à cinquante (50\$) par heure au-delà de 2 heures. L'enregistrement du nom de domaine pour la conférence, par exemple, `confuasz.zig.sn`, entraîne des frais de vingt-cinq dollars (25\$) pour un an, avec une charge supplémentaire de vingt dollars (20\$) pour chaque année suivante. Ces conditions définissent les limites et les coûts associés à l'utilisation d'EDAS pour la gestion de conférences [3].

2. EasyChair

EasyChair est une plateforme en ligne largement adoptée par la communauté académique et scientifique pour la gestion des conférences. Elle propose un ensemble complet de fonctionnalités qui simplifient l'organisation, la soumission, l'évaluation et la publication d'articles scientifiques dans le contexte des conférences [4].

b. Fonctionnalités

EasyChair offre les fonctionnalités clés suivantes :

- Gestion de la conférence
- Soumission d'articles

- Processus de révision
- Communication
- Programmes de conférence
- Archivage et publication

c. Avantages

EasyChair est une plateforme conviviale qui offre une centralisation des données liées à la conférence. Les articles soumis, les commentaires des réviseurs et les programmes de conférence sont tous regroupés sur la plateforme, simplifiant ainsi la gestion et l'accès aux informations. Elle facilite également la collaboration en ligne entre les participants à la conférence et permet aux organisateurs de suivre facilement l'avancement de la révision des articles.

d. Limites

Malgré ces avantages qu'offre EasyChair, il est important de noter certaines limites. L'une de ces limites est la dépendance à la plateforme pour la gestion de la conférence et la publication des articles. Cette dépendance peut restreindre la flexibilité des organisateurs qui préféreraient utiliser d'autres méthodes ou plateformes pour l'organisation de leur conférence. De plus, l'utilisation de EasyChair peut nécessiter un temps d'apprentissage pour les utilisateurs qui ne sont pas familiers avec la plateforme. Bien qu'elle soit conviviale, il peut y avoir une courbe d'apprentissage pour maîtriser pleinement toutes les fonctionnalités offertes par EasyChair. Il est donc important pour les organisateurs de conférences de prendre en compte cette dépendance et de considérer si EasyChair correspond à leurs besoins spécifiques. Une évaluation approfondie des fonctionnalités, des avantages et des limites de la plateforme est recommandée avant de décider de l'utiliser pour la gestion de leur conférence.

3. COMS

Le système de gestion de conférences (Conférence Management System en anglais), ou COMS, est une plateforme informatique conçue pour faciliter la gestion complète d'une conférence ou d'un événement similaire. Il s'agit d'un outil qui permet aux organisateurs de conférences de gérer les différentes étapes du processus, depuis l'appel à contributions jusqu'à la publication des actes de la conférence [5].

a. Fonctionnalités

COMS dispose les fonctionnalités suivantes

- Soumission d'articles
- Gestion des révisions
- Programme de la conférence
- Inscriptions
- Gestion des actes

b. Avantages

COMS automatise de nombreuses tâches liées à la gestion d'une conférence, ce qui permet de gagner du temps et d'optimiser les processus. Il simplifie et facilite la communication entre les différentes parties prenantes, offre un suivi et un statut des soumissions d'articles, des révisions et des décisions d'acceptation ou de rejet. De plus, il permet de centraliser les informations telles que les détails de la conférence, le programme, les articles et les actes, ce qui facilite l'accès et la gestion des données.

c. Limites

COMS présente certaines limites à prendre en compte. La configuration initiale peut être complexe, nécessitant du temps et des ressources techniques pour répondre aux besoins spécifiques de la conférence. Certains systèmes ont une adaptabilité limitée, ce qui signifie qu'ils sont moins flexibles en termes de personnalisation pour s'adapter aux exigences particulières d'un événement. De plus, la dépendance technologique est une limite, car le bon fonctionnement du COMS repose sur la disponibilité et la stabilité des serveurs, des réseaux et des systèmes de stockage de données. Des problèmes techniques peuvent donc affecter son utilisation.

4. OpenConf

OpenConf est une solution de gestion de conférences open source qui permet la soumission et l'évaluation des articles, la planification du programme et la gestion des inscriptions [6].

a. Fonctionnalités

OpenConf offre les fonctionnalités suivantes :

- Soumission d'articles
- Gestion des révisions
- Création et gestion du programme de la conférence
- Gestion des actes

b. Avantages

OpenConf présente plusieurs avantages. D'abord, étant un logiciel open source, il est gratuit et peut être modifié selon les besoins de la conférence. Cela offre une grande flexibilité et possibilité de personnalisation. En outre, en étant open source, OpenConf permet de réduire les coûts liés à l'achat de licences ou à l'utilisation de plateformes propriétaires, ce qui est bénéfique pour les organisateurs de conférences ayant des budgets limités. Il offre également une grande souplesse pour s'adapter aux exigences spécifiques de chaque événement.

c. Limites

L'utilisation de OpenConf peut présenter certaines limites. Tout d'abord, sa configuration initiale et sa mise en place peuvent être complexes, nécessitant des compétences techniques pour une installation et une configuration correcte du logiciel. De plus, en tant que logiciel open source, le niveau de support technique peut varier et dépendre de la disponibilité de la communauté des utilisateurs et des développeurs pour répondre aux problèmes et aux questions éventuelles. Enfin, par rapport à certaines solutions commerciales, OpenConf peut avoir des fonctionnalités moins avancées ou moins développées, en fonction des besoins spécifiques de la conférence.

5. WikiCfp

Wickicfp est un outil pour les appels à communication dans les domaines scientifiques et technologique, très utilisé par les chercheurs. En effet, il y a plus de cent milles appels à communication publié sur la plateforme et plus de cent milles chercheurs l'utilise par mois [7]. Wickicfp à comme fonctionnalité principale la gestion des appels à communication.

a. Fonctionnalité

- Auteurs
 - Consulter, rechercher et parcourir des appels à communications
 - S'abonné aux listes des différents catégories
 - Recevoir des rappels par e-mail
- Organisateurs
 - Présenter des appels à communication à des milliers de chercheurs.
 - Diffuser des appels à communication sur twitter et Facebook.
 - Mettre à jour les informations en temps opportun.

- Vérifier le nombre de vue et nombre d'utilisateurs suivant les appels à communication.

b. Avantages

Wickicfp est tout d'abord une plateforme avec une large communauté, elle permet aussi de diffuser les appels à communication sur twitter et Facebook, ce qui favorise une meilleure communication externe, permettant d'accroître la visibilité de l'évènement

c. Limites

Wickicfp se concentre uniquement sur les appels à communication, elle n'offre pas la possibilité de partage d'évènements et d'articles scientifique. La qualité des appels dépend de la fiabilité de l'organisateur elle-même, c'est-à-dire les informations fournis par les organisateurs.

6. Tableau récapitulatif

Outils	Avantages	Limites
EDAS	<ul style="list-style-type: none"> - Disponibilité du personnel de support - Accessibilité et prise en charge par les navigateurs Web standards 	<ul style="list-style-type: none"> - Frais supplémentaires par rapport au nombre d'articles soumis - Frais sur le choix d'un nom de domaine pour un évènement
EasyChair	<ul style="list-style-type: none"> - Centralisation des données liées à la conférence - Facilite la collaboration en ligne entre les participants à la conférence. - Permet aux organisateurs de suivre facilement l'avancement de la révision des articles. 	<ul style="list-style-type: none"> - Dépendance à la plateforme pour la gestion de la conférence et la publication des articles. - Demande une documentation préalable sur ses fonctionnalités
COMS	<ul style="list-style-type: none"> - Tâches liées à la gestion d'une conférence 	<ul style="list-style-type: none"> - Configuration initiale complexe - Adaptabilité limitée - Dépendance technologique

	<ul style="list-style-type: none"> - Simplifie et facilite la communication entre les différentes parties prenantes 	
OpenConf	<ul style="list-style-type: none"> - Open source, gratuit et peut être modifié selon les besoins de la conférence - Réduction des coûts liés à l'achat de licences ou à l'utilisation de plateformes propriétaires 	<ul style="list-style-type: none"> - Sa mise et sa configuration en place peuvent être complexes - Le niveau de support technique peut varier et dépendre de la disponibilité de la communauté des utilisateurs et des développeurs
Wickicfp	<ul style="list-style-type: none"> - Une large communauté - Diffusion et publication des appels sur twitter et Facebook 	<ul style="list-style-type: none"> - Uniquement réservé aux appels à communication - Qualité des informations fortement dépendant des organisateurs.

Table 1 : Récapitulatif des solutions similaires

Ce tableau récapitulatif permet d'obtenir une vision simplifiée des différents outils précédemment présentés. Il illustre avec plus de précision les avantages et les limites de chaque solution, ce qui peut nous aider à positionner notre application par rapport à ces solutions déjà existantes. Ainsi, notre solution, axée sur la mise en place d'une plateforme de gestion d'événements scientifiques organisés au sein de l'Université Assane de Ziguinchor, présente des similitudes avec ces solutions mais s'inscrit dans un contexte différent, offrant en outre une approche distincte en termes d'utilisation et de fonctionnalité. Dans la section suivante, nous allons présenter de manière plus détaillée notre solution et ses particularités.

IV. Présentation de notre solution

Les différents outils présentés jusqu'à présent ont tous en commun la gestion des conférences et de la diffusion en ligne, et ils offrent de nombreux avantages. En effet, pour la gestion des journées scientifiques ou des colloques, l'UASZ utilise souvent EasyChair ou des procédures séparées consistant à envoyer des courriels via la messagerie de l'université, les

réseaux sociaux ou les sites web de l'université pour publier l'événement et l'appel à communications. Ensuite, les soumissions sont reçues par courrier électronique afin de passer à la gestion des révisions et d'approuver ou non les soumissions. Après cela, vient l'étape de partage des réponses avec les différents soumissionnaires. Enfin, s'il y a lieu de diffuser en ligne, l'outil utilisé est Google Meet, avec le serveur de l'université qui offre la possibilité d'accueillir un nombre illimité de participants.

Dans ce dernier cas, il est à noter quelques inconvénients qui peuvent entraîner une communication inefficace, une difficulté de suivi, un manque de transparence et une complexité administrative. Dans le cas de l'utilisation de solutions telles qu'EasyChair, il est possible qu'il y ait des coûts supplémentaires associés, tels que l'achat de licences, l'accès à des fonctionnalités avancées ou des frais d'abonnement. Cependant, un autre aspect important est que l'utilisation d'EasyChair ne permettrait pas à l'UASZ de gérer ses conférences de manière autonome dans un système centralisé qui appartient exclusivement à l'université. En conservant la gestion en interne, l'UASZ peut maintenir un contrôle total sur ses données, personnaliser, les procédures selon ses besoins spécifiques et avoir une plus grande autonomie dans la gestion de ses événements scientifiques.

Ainsi, c'est dans ce contexte que notre solution de gestion d'événements destinée à l'UASZ trouve sa raison d'être. Cette plateforme permettra de centraliser la gestion de tous les événements scientifiques des écoles doctorales de l'UASZ, depuis la publication jusqu'aux soumissions, inscriptions, validations et diffusions. Grâce à ce système centralisé, l'université pourra bénéficier d'une gestion complète et efficace de ses événements, simplifiant les procédures et offrant une expérience optimale aux participants.

V. Analyse des technologies utilisées dans la conception et le développement de l'application

Le développement d'une application web implique l'utilisation d'une variété de langages de programmation, d'outils et de technologies. Ces différents éléments sont indispensables à chaque étape, depuis la conception jusqu'à la réalisation du codage, des tests et du déploiement de nos applications. Cette section présentera les technologies les plus couramment utilisées dans la conception et la gestion de projets, ainsi que les outils nécessaires pour le codage, les tests et le déploiement d'une application web.

1. Outils de gestion de projet informatique

L'utilisation d'un logiciel de gestion de projet nous permettra de mieux gérer les contraintes de temps, de budget et de portée, afin d'organiser de manière optimale le processus de réalisation de notre application web. Il existe de nombreuses solutions numériques pour la gestion de projets, parmi lesquelles on peut citer :

a. TeamGant

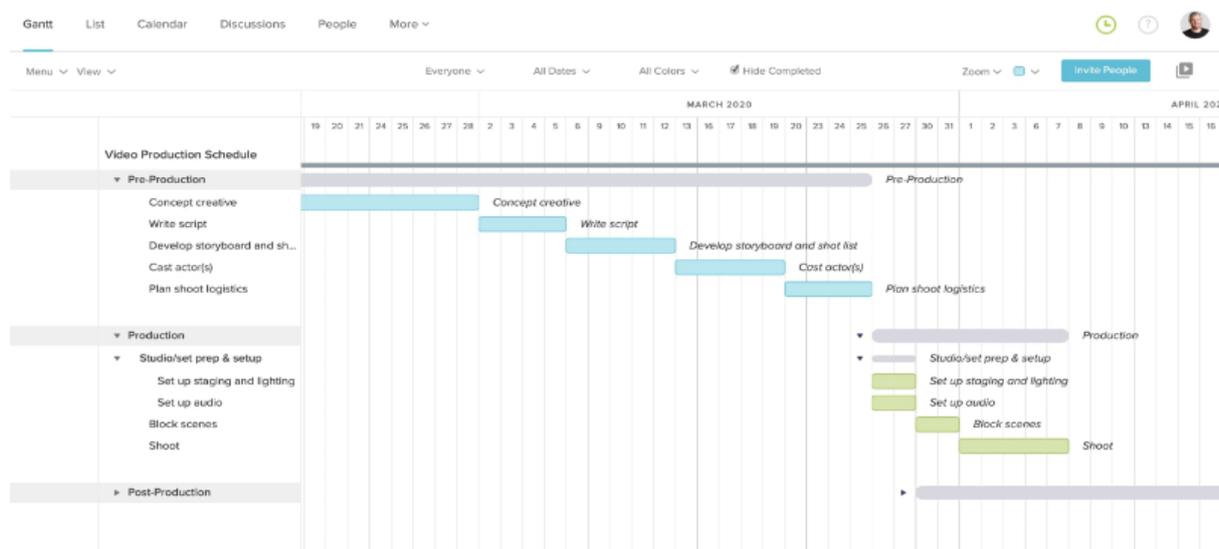


Figure 1 : illustration de TeamGant

TeamGant est un outil de gestion de projet qui permet de créer des diagrammes de Gantt, de collaborer en équipe, d'assigner des tâches et de gérer les ressources. C'est un outil convivial et populaire utilisé par des équipes de différentes tailles et dans différents domaines. Il facilite la planification, la coordination et le suivi des projets, assurant ainsi une bonne organisation et le respect des délais [8].

b. Asana

The screenshot shows the Asana interface for a project named 'Customer Stories - Q4'. The interface includes a navigation bar with options like 'List', 'Board', 'Timeline', 'Calendar', 'Progress', 'Forms', and 'More...'. Below the navigation bar, there are filters for 'Incomplete tasks', 'Filter', 'Sort', 'Rules', and 'Fields'. The main content area displays a list of tasks with columns for 'Task name', 'Assignee', 'Due date', 'Audience', 'Tags', 'Estimated hours', 'Channel', and 'Effort'. The tasks are grouped into two sections: 'Recurrent pieces' and 'New formats'. Each task row includes a task name, an assignee's profile picture, a due date, an audience tag (e.g., Premium, Business), a priority tag (e.g., Low priority, High Priority, Med priority), estimated hours, a channel tag (e.g., Press, Social, Blog), and an effort tag (e.g., Low, Medium, High). Summary rows at the end of each group show 'SUM' and a total value (34 for Recurrent pieces, 16 for New formats).

Task name	Assignee	Due date	Audience	Tags	Estimated hours	Channel	Effort
Recurrent pieces							
Press release on acquisition 4	Blake Pham	23 Apr	Premium	Low priority	2	Press	Low
Create new infographic	Blake Pham	30 Apr			5	Social	Low
Editorial calendar	Blake Pham	21 May	Premium		4	Social	Medium
Customer spotlight #1 2	Kat Mooney	19 Jun	Business	High Priority	3	Blog	Medium
Customer spotlight #2 interview 1	Nikki Henderson ...	22 May	Premium	Low priority	3	Blog	Medium
New feature roundup	Kat Mooney	24 Apr	Premium	Med priority	5	Blog	High
Christmas campaign 6	Avery Lomax	28 Aug	Business	High Priority	3	Social	High
Create campaign	Blake Pham	1 Oct	Premium	Low priority		Social	Medium
SUM						34	
New formats							
Work-life balance newsletter	Avery Lomax	20 Mar	Premium	High Priority	4	Email	Low
Roadmap	Jennifer Lu	25 Mar	Business		3	Web	Medium
Apollo for nonprofits 2	Kat Mooney	19 Mar	Business	Low priority	9	Press	Medium
SUM						16	

Figure 2 : Interface Asana

Asana est une plateforme de gestion de projet qui offre plusieurs fonctionnalités pour aider les équipes à travailler ensemble de manière organisée [9]. On peut créer des tâches, attribuer des échéances et les regrouper dans des projets. On peut aussi utiliser un calendrier pour visualiser les délais. On peut assigner des tâches à des membres de l'équipe et discuter des travaux en ajoutant des commentaires. Asana permet également de créer des tableaux de bord pour suivre les progrès et offre des automatisations pour simplifier les tâches répétitives. Il s'intègre avec d'autres outils courants tels que Slack ou Google Drive. Asana propose différents plans d'abonnement, y compris une version gratuite avec des fonctionnalités de base.

c. Trello

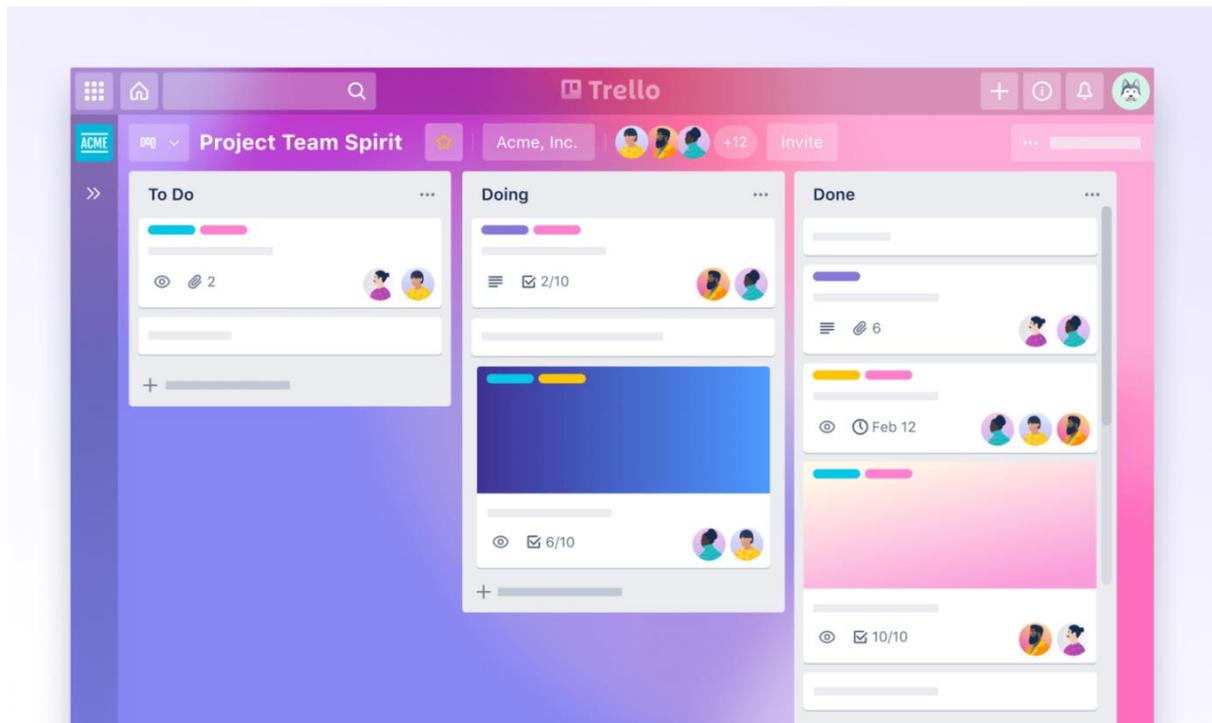


Figure 3 : Interface Trello

Trello est une plateforme de gestion de projet basée sur des tableaux Kanban. Elle offre une approche visuelle et intuitive pour organiser les tâches et collaborer en équipe [10]. Les utilisateurs peuvent créer des tableaux pour représenter leurs projets, des listes pour regrouper les différentes étapes, et des cartes pour représenter les tâches individuelles. Les cartes peuvent être déplacées facilement entre les listes pour refléter l'avancement. On peut ajouter des descriptions, des échéances, des pièces jointes, et assigner des membres à chaque carte. Trello facilite la collaboration en permettant aux membres de l'équipe de discuter et de commenter les cartes. Il propose également des fonctionnalités avancées telles que des rappels, des intégrations avec d'autres outils, et des options de personnalisation. Trello est disponible en version gratuite avec des fonctionnalités de base, ainsi qu'en versions payantes offrant des fonctionnalités supplémentaires et un niveau d'intégration plus avancé.

d. Notion

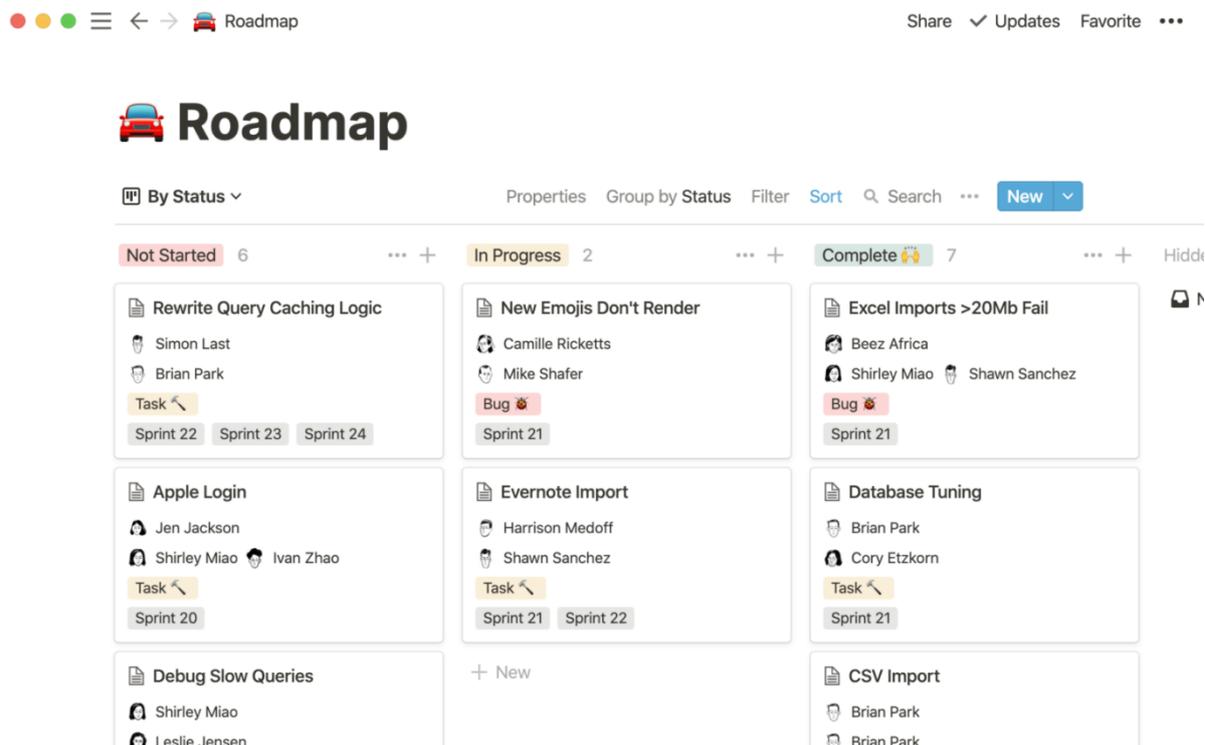


Figure 4 : Interface Notion

Notion est une plateforme pratique qui permet de gérer des projets, prendre des notes, travailler en équipe et organiser des connaissances [11]. Elle a une interface souple et personnalisable qui permet aux utilisateurs de créer des pages et des bases de connaissances selon leurs besoins. On peut y faire des listes, des tableaux, des calendriers et bien d'autres choses. Notion permet de bien organiser les informations en les classant de façon hiérarchique et en ajoutant des liens entre les différentes pages. Les utilisateurs peuvent collaborer en temps réel, laisser des commentaires et donner des tâches à faire. Notion offre aussi des fonctions avancées comme l'intégration avec d'autres outils, des automatisations et une API pour personnaliser l'expérience. Il y a des versions gratuites et payantes de Notion avec différentes options et limites d'utilisation.

2. Technologies pour le développement backend

Le développement backend se concentre sur la mise en place de la logique des applications, ce qui implique la gestion des données, la communication avec les serveurs, la robustesse et les performances des applications, entre autres. Pour construire notre backend, nous aurons besoin de langages, d'outils et de Framework spécialement conçus pour cette tâche. Le tableaux ci-dessous les technologies les plus couramment utilisées dans le développement backend.

Domaine	Langages/Outils	Spécificités/Explications
Base de données	MySQL, PostgreSQL, MongoDB	Systèmes de gestion de bases de données relationnelles (MySQL, PostgreSQL) et non relationnelles (MongoDB) utilisés pour stocker et organiser les données de manière efficace.
Serveurs	Node.js, Apache, Nginx	Environnements serveur permettant d'héberger les applications web, de gérer les requêtes des utilisateurs et de renvoyer les réponses appropriées.
Langages	PHP, Python, Ruby, Java	Langages de programmation utilisés pour créer la logique et les fonctionnalités des applications backend. Chaque langage a ses propres caractéristiques et avantages.
Framework	Laravel, Django, Ruby on Rails	Framework facilitant le développement backend en fournissant des outils, des bibliothèques et des structures prédéfinies pour créer des applications robustes et efficaces.

Table 2 : Récapitulatif langage et outils backend

3. Technologies pour le développement frontend

La partie frontend représente l'interface utilisateur de notre application, c'est-à-dire la partie visible du site web avec laquelle les utilisateurs interagissent. Le tableau suivant récapitule les langages et outils couramment utilisés dans le développement frontend :

Table 3 : Récapitulatif langage et outils frontend

Domaine	Langages/Outils	Spécificités/Explications
Langages	HTML, CSS, JavaScript	Langages de base utilisés pour créer la structure, le style et les interactions des interfaces utilisateur des applications web.
Framework	React, Angular, Vue.js	Framework JavaScript populaires permettant de construire des interfaces utilisateur réactives et interactives, facilitant le développement frontend.
Styles	Bootstrap, Tailwind CSS, Sass	Bibliothèques et préprocesseurs de CSS qui fournissent des composants prédéfinis, des classes utilitaires et des fonctionnalités avancées pour styliser les interfaces.
Gestion des états	Redux, Vue, React Context	Outils de gestion des états des applications frontend, permettant de gérer efficacement les données et leur flux dans l'interface utilisateur.
Bibliothèques UI	Material-UI, Ant Design, Bulma	Collections de composants réutilisables et esthétiquement attrayants qui facilitent la création d'interfaces cohérentes et attrayantes.

VI. Conclusion

En conclusion, ce chapitre a examiné l'état de l'art dans le domaine de la gestion de conférences et de la diffusion en ligne. Nous avons passé en revue différentes applications similaires, telles que EasyChair, COMS, OpenConf, EDAS, Wickicfp, ..., en analysant leurs fonctionnalités, leurs avantages et leurs limites. Nous avons aussi vu les différentes technologies pour la gestion de projet informatique, le développement des applications avec la partie backend et frontend.

Chapitre 2 : Méthodologie et choix technologiques

I. Introduction

La réussite d'un projet repose sur l'application d'une méthodologie de pilotage adaptée et du choix judicieux sur les technologies à utilisés. Pour ce faire, il est essentiel de déterminer préalablement la méthode à utiliser en fonction des besoins spécifiques du projet et de déterminer les technologies à utiliser en fonction de l'architecture et des besoins du projet. Dans cette section, nous procéderons tout d'abord à une étude comparative des méthodologies de gestion de projet les plus couramment utilisées. Ensuite faire une étude sur les architectures et technologies les plus utilisés pour le développement d'une application afin de sélectionner celle qui convient le mieux.

II. Description de la méthodologie utilisée

Un projet est un ensemble d'activités coordonnées et maîtrisées pour atteindre un objectif en respectant des exigences de qualité, de coûts et de délais. Un projet à une date de début et une date de fin.

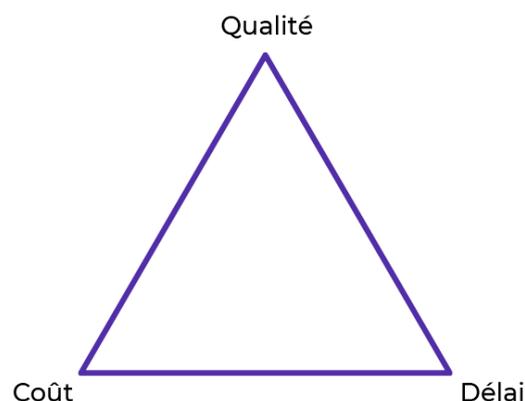


Figure 5 : illustration gestion de projet [12].

Les projets informatiques varient selon leur nature et leurs complexités, ainsi la gestion de ce dernier fait appel à des approches ou méthodologies différentes. Parmi ces méthodologies, nous avons la méthodologie traditionnelle ou en cascade, en spirale et la méthodologie agile.

1. La méthodologie traditionnelle (cascade)

La gestion de projet en cascade, également connue sous le nom de méthode Waterfall, est une approche séquentielle de gestion de projet qui se divise en plusieurs phases. Cette approche est représentée sous la forme d'un cycle en V, ce qui implique une méthode

séquentielle englobant la conception, la réalisation et la validation du projet. Le cycle en V est modélisé par une phase ascendante suivie d'une phase descendante, formant ainsi la structure en V. Les différentes tâches et leurs délais dans la méthodologie en cascade sont souvent représentés à l'aide d'un diagramme de Gantt [13].

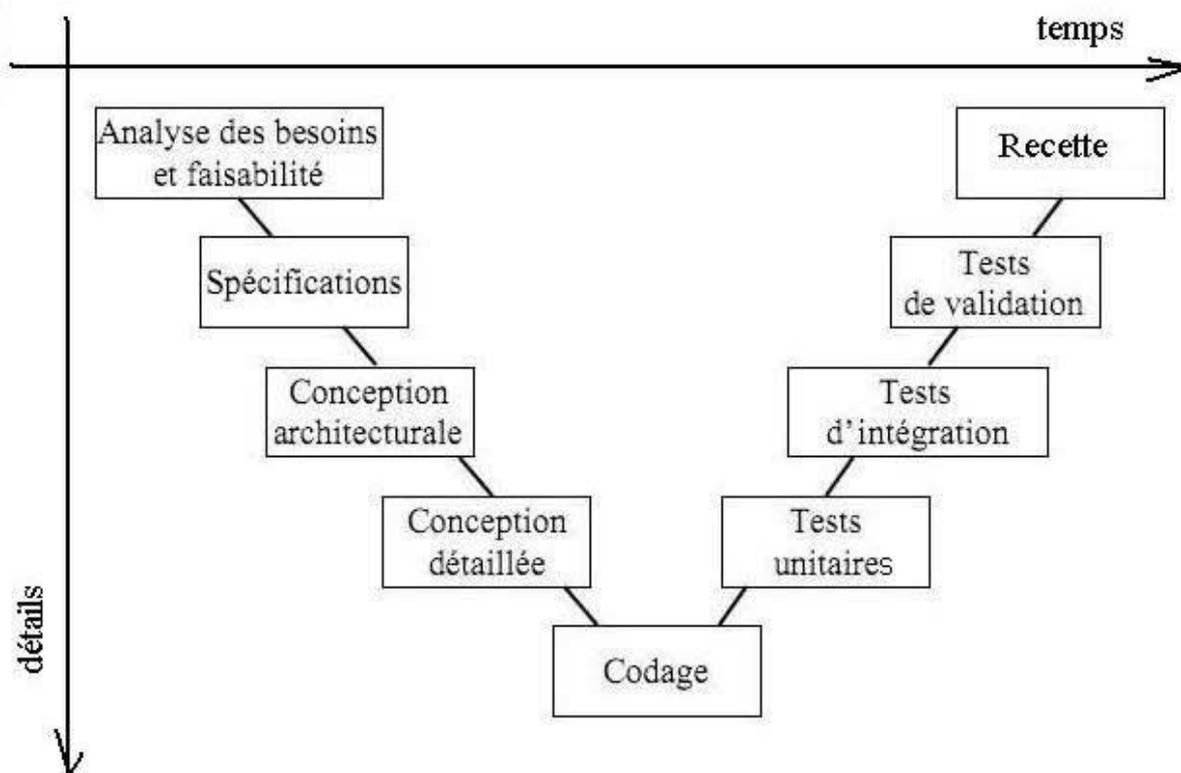


Figure 6 : illustration cycle en V

a. Les phases de la méthodologie de gestion de projet en cascade

Comme nous l'avons vu précédemment, les phases de la méthodologie de gestion de projet en cascade sont représentées sous forme de cycle en V, ce qui fait que les différentes phases du projet sont réalisées de manière linéaire, les unes après les autres.

- La phase de définition des besoins :

Cette étape consiste à définir les objectifs du projet, d'analyser les besoins fonctionnels et non fonctionnels des parties prenantes, et aussi d'établir la faisabilité du projet.

- La phase de planification :

Cette phase implique l'identification des tâches spécifiques à réaliser, l'estimation des ressources nécessaires, l'établissement d'un calendrier et la définition des critères de réussite.

- La phase de conception :

La phase de conception consiste à définir les détails techniques du projet. Cela inclut le choix du matériel, des langages de programmation et de l'interface utilisateur.

- **La phase des tests et validation**

Cette étape concerne la réalisation de tests pour s'assurer que le produit ou le système répond aux spécifications et fonctionne correctement. Différents types de tests peuvent être effectués, tels que les tests d'intégration, les tests de validation et les tests de performance.

- **La phase mise en production**

Dans cette étape, la tâche principale consiste au déploiement de l'application dans l'environnement de production. Cette phase peut impliquer la formation des utilisateurs, la préparation de la documentation et la gestion de la transition.

- **La phase de maintenance et support**

Cette dernière étape portera sur les activités de maintenance et de support pour résoudre les problèmes, appliquer des correctifs, effectuer des mises à jour et répondre aux demandes des utilisateurs.

b. Cas d'utilisation de la méthodologie en cascade

Cette méthodologie est couramment utilisée dans les projets avec des exigences stables, c'est-à-dire lorsque les besoins fonctionnels et non fonctionnels sont clairement définis et ne devraient pas connaître de changements significatifs tout au long du projet. Elle est également adaptée aux projets de petite envergure et de complexité réduite, ainsi qu'aux projets soumis à des contraintes de temps strictes et à des risques maîtrisés.

c. Avantages et inconvénients de la méthodologie en cascade

AVANTAGES	INCONVÉNIENTS
Structure linéaire : La compréhension et la planification sont facilitées par la linéarité des différentes phases.	Rigidité : Les changements après avoir terminé une tâche peuvent être coûteux et compliqués à gérer.
Documentation complète : La traçabilité et la gestion rigoureuse est assurée par les documentations faites pour chaque phase du projet.	Faible adaptabilité : La méthodologie en cascade n'est pas adéquate avec les projets qui ont des besoins évolutifs et flexibles.
Prédictibilité : Il est plus facile de prévoir	Communication limitée : Chaque phase

les délais et les coûts du projet.	peut être gérée par des équipes différentes, ce qui peut limiter la communication.
Facilité de gestion : Il est plus facile de gérer les ressources et les dépendances entre les différentes étapes. Vu que chaque phase du projet est clairement délimitée.	Validation tardive : Les tests et les validations sont effectués à la fin du projet, cela peut causer des problèmes de détection tardive des défauts.

Table 4 : *Avantage et inconvénients méthodologie séquentielle*

2. La méthodologie Agile

La gestion de projet agile permet de s'adapter à tous les projets et est extrêmement polyvalente. Elle est flexible et permet de gérer les imprévus. Elle s'appuie sur le retour d'expérience entre les différentes phases itératives et sur le cahier des charges initial. Elle laisse une liberté de modification au client, mais requiert une forte implication de sa part.

La méthodologie Agile existe bien évidemment avant l'apparition d'Internet pour le grand public. En effet, en 2001, des représentants de différentes méthodologies de développement de logiciels se sont réunis à Snowbird pour créer le Manifeste Agile. Ce manifeste définit des valeurs et des principes partagés par tous les adeptes de la méthodologie Agile [14].

a. Les valeurs de la méthodologie agile

Les valeurs établis dans le manifeste agile sont :

- « Les individus et leurs interactions plus que les processus et les outils »
- « Des logiciels opérationnels plus qu'une documentation exhaustive »
- « La collaboration avec les clients plus que la négociation contractuelle »
- « L'adaptation au changement plus que le suivi d'un plan »

Cf : manifest agile

b. Les principes de la méthodologie agile

Les douze (12) principes définis dans le manifeste :

- Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.

- Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
- Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
- Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
- Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
- La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
- Un logiciel opérationnel est la principale mesure d'avancement.
- Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
- Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.
- La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
- Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.
- À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

c. Les différentes méthodes agiles

Il existe différents types de méthodologies agiles dont les plus fréquents sont :

- **Scrum**

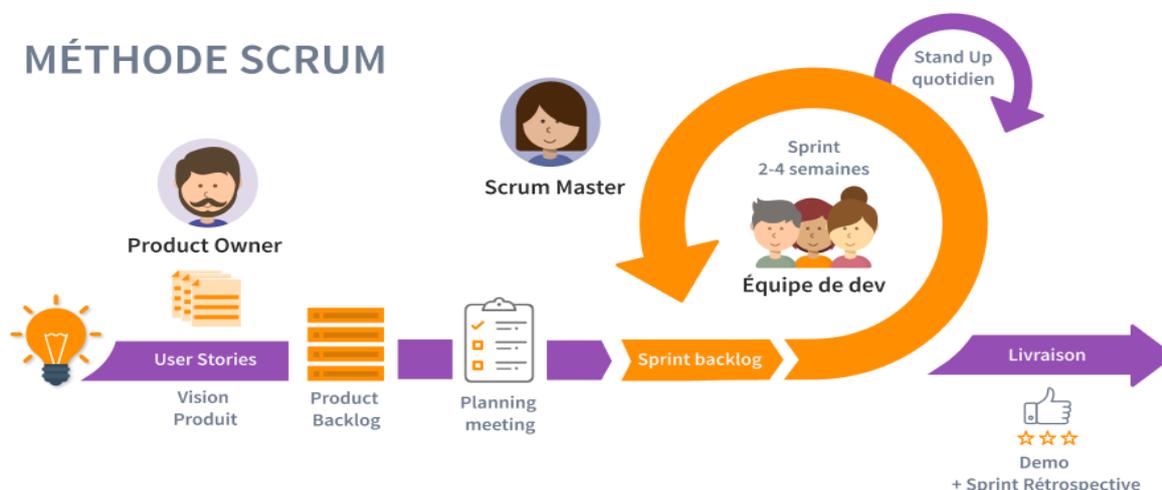


Figure 7 : Méthode scrum

Scrum est un cadre de travail qui aide à gérer et organiser les projets informatiques. C'est une méthode agile qui se base sur des cycles courts et répétitifs appelés "sprints", d'une durée généralement de deux à quatre semaines. Scrum met l'accent sur la collaboration, la transparence et la capacité à s'adapter aux changements [15].

Dans Scrum, le travail est organisé dans un "backlog", une liste des fonctionnalités à développer, classées par priorité. Au début de chaque sprint, une réunion de planification a lieu pour sélectionner les éléments du backlog qui seront réalisés pendant le sprint. L'équipe de développement s'engage ensuite à les accomplir.

Pendant le sprint, l'équipe se réunit quotidiennement lors d'une brève réunion appelée "stand-up" pour partager les progrès réalisés, discuter des éventuels obstacles et ajuster les plans si nécessaire. À la fin de chaque sprint, il y a une démonstration du travail accompli devant les parties prenantes, suivie d'une réunion de rétrospective pour évaluer le processus et identifier les améliorations possibles.

- **Kanban**

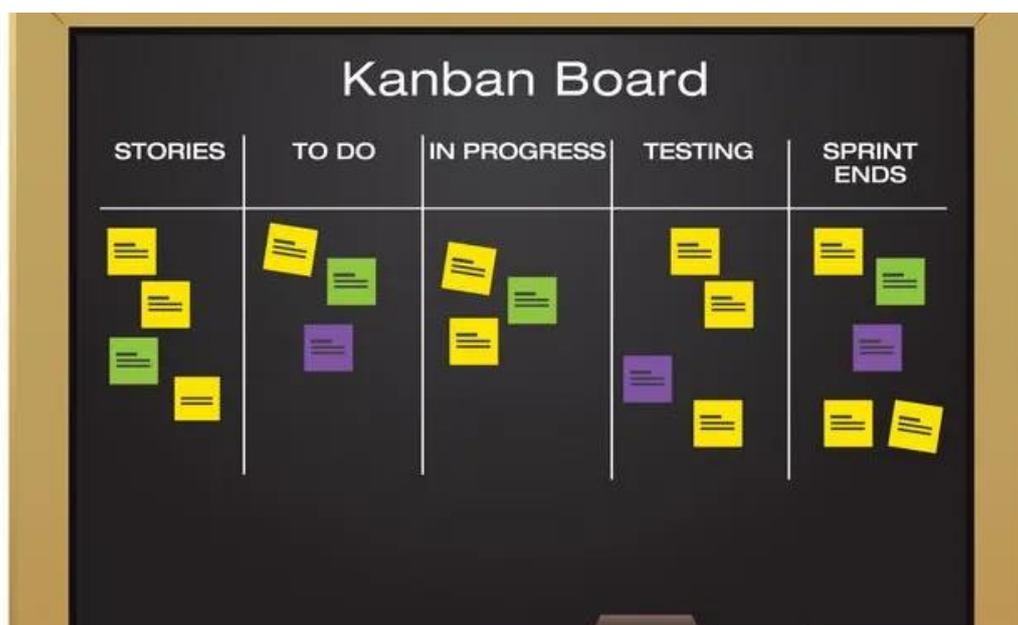


Figure 8 Méthode kanban

Kanban est une méthode de gestion de projet souvent utilisée avec la méthodologie Agile, mais qui peut aussi être utilisée seule.

Le système Kanban repose sur le principe de visualisation du flux de travail à l'aide d'un tableau Kanban. Ce tableau est divisé en colonnes représentant les différentes étapes du travail, comme "À faire", "En cours" et "Terminé". Chaque tâche ou élément de travail est représenté par une carte Kanban et déplacé à travers les colonnes au fur et à mesure de son avancement [16].

L'objectif principal de Kanban est de limiter le nombre de tâches en cours (WIP - Work In Progress) pour améliorer le flux de travail et réduire les blocages. Cela permet de voir facilement les tâches en attente, de mieux établir les priorités et de favoriser la collaboration au sein de l'équipe.

Kanban met l'accent sur l'amélioration continue en identifiant et en éliminant les obstacles et les inefficacités qui peuvent ralentir le processus de travail. Il encourage également la transparence et la responsabilité de l'équipe en fournissant une vue claire de l'état du projet et des tâches individuelles.

3. Méthodologie choisie

Étant donné que notre projet consiste à créer une application de gestion de conférences avec diffusion en ligne, il est important d'avoir plus de flexibilité dans le processus de développement. Les besoins spécifiques peuvent varier en fonction des outils ou des technologies choisis pour sa réalisation. C'est pourquoi nous avons décidé d'adopter une méthodologie agile. Cela nous permettra d'avoir une meilleure communication avec nos

encadrants pour la réalisation du projet et également d'avoir plus de souplesse pour répondre aux exigences du projet, notamment en effectuant une veille technologique sur les solutions de diffusion en ligne.

Ainsi, pour une bonne organisation et visualisation de l'état des différentes tâches et l'organisation du développement des fonctionnalités de l'application nous avons décidé de choisir Kanban comme méthode. L'outil de gestion de projet choisi est Notion, en effet, il offre un espace de travail convivial et la possibilité d'organiser nos différentes user story et de gérer leurs avancements.

III. Choix technologique

1. Architecture

L'architecture est un aspect important du développement de logiciel car elle définit la structure et l'organisation d'une application. De nos jours, plusieurs approches sont utilisées, telles que l'architecture micro services, l'architecture SOA et l'architecture monolithique. Chacune de ces approches présente des caractéristiques distinctes et peut être utilisée dans des contextes spécifiques [17].

a. Architecture micro services

L'architecture micro services est une approche qui consiste à diviser une application en plusieurs services indépendants, chacun ayant sa propre logique métier, et qui communiquent entre eux via des interfaces bien définies. Cette architecture présente plusieurs avantages : elle permet une meilleure scalabilité, car il est plus facile de faire évoluer des parties spécifiques de l'application sans impacter l'ensemble du système ; elle offre une plus grande flexibilité, car les services peuvent être développés, déployés et mis à l'échelle indépendamment les uns des autres ; enfin, elle facilite la maintenance, car les services peuvent être facilement mis à jour ou remplacés sans affecter le fonctionnement global de l'application. Cependant, elle présente également quelques inconvénients : la complexité de la gestion de la communication entre les services peut augmenter en raison de la nature distribuée des micro services, et le coût de développement initial peut être plus élevé, car il peut nécessiter plus de temps et d'efforts pour mettre en place une architecture basée sur des micro services. La figure ci-dessous offre une illustration plus détaillée de l'architecture micro service.

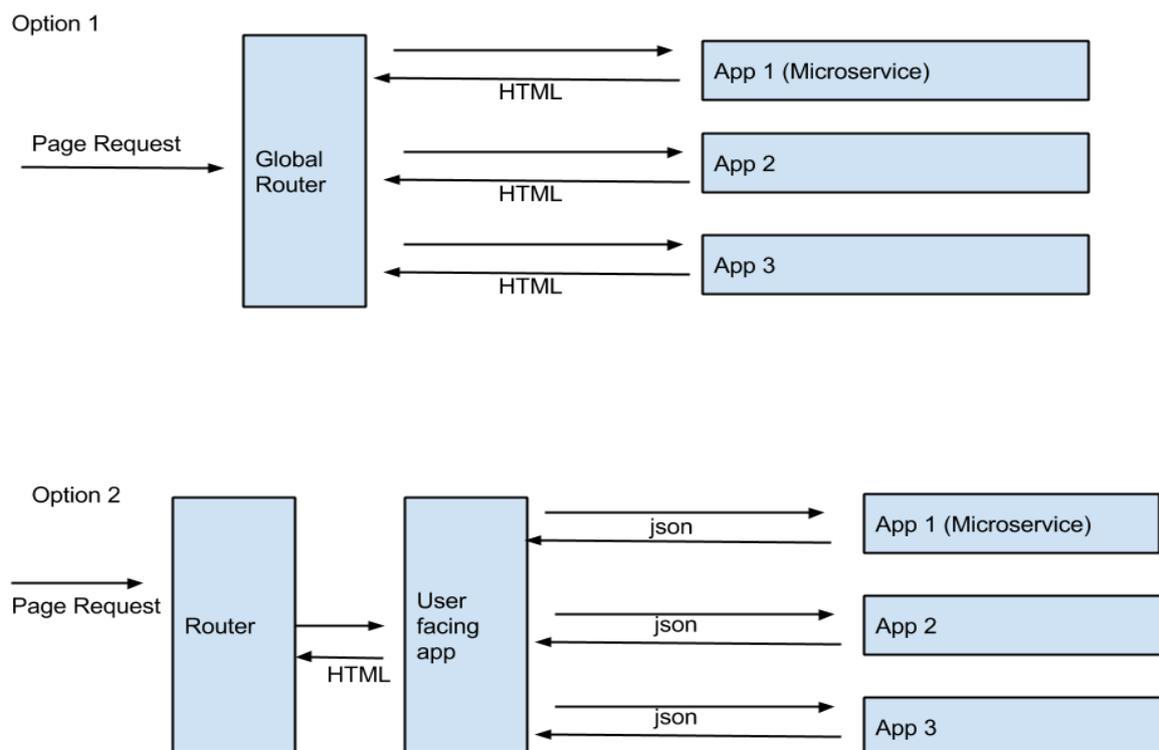


Figure 9 : architecture micro services

b. Architecture SOA

L'architecture SOA (Service-Oriented Architecture) est une approche visant à concevoir des systèmes logiciels en mettant l'accent sur des services interconnectés. Dans cette architecture, les fonctionnalités de l'application sont découpées en services distincts et autonomes, qui sont ensuite exposés et peuvent être utilisés par d'autres services ou applications. Les avantages de cette architecture résident dans sa réutilisabilité, permettant aux services d'être conçus de manière à être réutilisables dans différentes applications. De plus, elle offre une flexibilité et une facilité d'évolution, car les services peuvent être programmés dans différents langages de programmation et s'exécuter sur différentes plateformes, tout en étant facilement modifiables, remplaçables ou ajustables sans perturber l'ensemble du système. Cependant, cette architecture présente également des inconvénients, tels que sa complexité, en raison de la mise en place et de la gestion nécessitant une coordination et une communication entre les services. Elle peut également entraîner une surcharge de communication entre les différents composants du système. De plus, la conception et la mise en œuvre d'une architecture SOA peuvent nécessiter un investissement initial plus important en termes de temps, de ressources et de compétences. L'image ci-dessous représente une illustration de cette architecture.

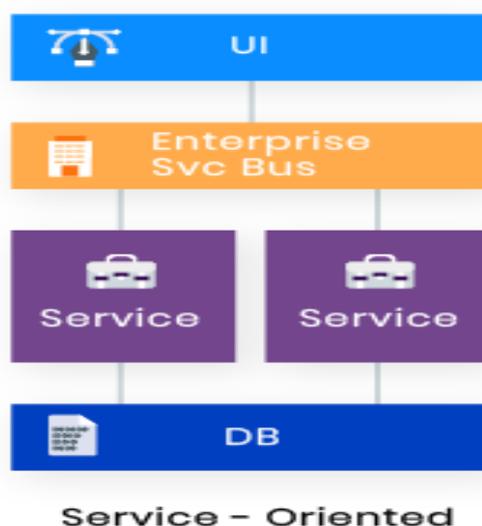


Figure 10 architecture SOA

c. Architecture monolithique

L'architecture monolithique est une approche de conception de systèmes logiciels où toutes les fonctionnalités sont regroupées en un seul bloc, formant une application unique et indivisible. Dans cette architecture, il n'y a pas de découpage en services distincts. Les avantages de cette approche résident dans sa simplicité et sa facilité de déploiement, car l'ensemble de l'application est géré comme une entité unique. Cela peut également réduire la complexité liée à la coordination et à la communication entre différents services. Cependant, l'architecture monolithique présente des inconvénients en termes de scalabilité et de flexibilité. Lorsque l'application devient plus complexe et nécessite une mise à l'échelle ou des modifications spécifiques, il peut être difficile d'isoler et de modifier uniquement certaines parties sans impacter l'ensemble du système. De plus, les mises à jour et les déploiements peuvent être plus lourds, car toute l'application doit être redéployée. En conclusion, l'architecture monolithique offre une simplicité et une facilité de déploiement, mais elle peut être moins flexible et évolutive que d'autres approches architecturales telles que les architectures micro services ou SOA. L'image suivante illustre l'architecture monolithique.

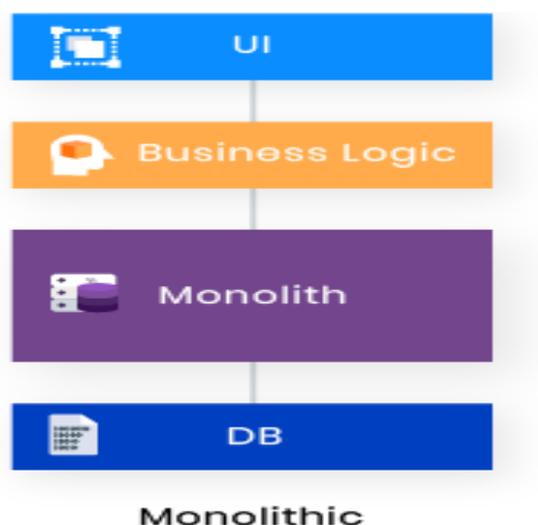


Figure 11 : architecture Monolithique

2. Backend

Dans le développement web, la partie backend, également appelée "côté serveur", fait référence à la partie d'une application ou d'un site web qui traite les requêtes des utilisateurs, gère les données, effectue les calculs et assure la logique métier. En d'autres termes, le backend est responsable de la gestion des opérations qui se produisent en arrière-plan et qui ne sont pas directement visibles par les utilisateurs. Les langages de programmation couramment utilisés pour le développement backend incluent PHP, Java, Python et JavaScript (avec des Framework tels que Node.js). Ces langages permettent aux développeurs de mettre en place la logique métier, de manipuler les données, d'interagir avec la base de données et de fournir des réponses appropriées aux requêtes des utilisateurs.

a. PHP

« Un langage de script populaire à usage général, particulièrement adapté au développement Web. Rapide, flexible et pragmatique, PHP gère tout, de votre blog aux sites Web les plus populaires au monde » [18].

Comme décrit au niveau de sa documentation officielle, PHP (HyperText Préprocesseur) est un langage de programmation populaire utilisé principalement pour le développement web. Il est principalement utilisé côté serveur pour générer du contenu dynamique et interagir avec les bases de données. C'est un langage de script open source, ce qui signifie qu'il est gratuit et largement accessible. Il est conçu pour être intégré directement dans le code HTML, permettant ainsi aux développeurs de mélanger facilement du code PHP

avec des éléments HTML pour générer des pages web dynamiques. Il est également utilisé par de nombreux CMS (Content Management System) populaires tels que WordPress, Drupal et Joomla.



Figure 12 : logo PHP

b. Java

Java est un langage de programmation polyvalent et largement utilisé, apprécié pour sa portabilité, sa simplicité, sa sécurité et sa robustesse. Il a été développé par Sun Microsystems (maintenant acquis par Oracle) et est basé sur le concept de "write once, run anywhere" (WORA), ce qui signifie que le code Java peut être exécuté sur différentes plateformes sans nécessiter de modifications majeures [19]. Java est souvent utilisé pour le développement d'applications logicielles, de sites web, d'applications mobiles, d'applications de bureau, de systèmes embarqués et de nombreuses autres applications. Il est également utilisé comme langage de programmation pour le développement d'applications d'entreprise et de solutions backends.



Figure 13 : logo java

c. Python

« Python est puissant... et rapide, joue bien avec les autres, court partout, est convivial et facile à apprendre ; est ouvert [20]. »

Python est un langage de programmation interprété, polyvalent et largement utilisé. Il a été créé par Guido van Rossum et est caractérisé par une syntaxe simple et lisible qui favorise la lisibilité du code. Python est utilisé dans de nombreux domaines, tels que le développement web, l'analyse de données, l'apprentissage automatique, l'automatisation de tâches, la bio-informatique et bien d'autres. Sa facilité d'apprentissage et sa polyvalence en font un choix populaire parmi les développeurs débutants et expérimentés.



Figure 14 : logo python

3. Frontend

Le frontend, également connu sous le nom de développement côté client, désigne la partie visible et interactive d'une application ou d'un site web avec laquelle les utilisateurs interagissent directement. Il comprend la conception, le développement et la mise en œuvre de l'interface utilisateur (UI) ainsi que la gestion de l'expérience utilisateur (UX). Les langages clés utilisés dans le développement frontend sont HTML (HyperText Markup Language), CSS (Cascading Style Sheets) et JavaScript. HTML est utilisé pour structurer le contenu de la page, CSS est utilisé pour la mise en forme et la présentation visuelle, tandis que JavaScript est utilisé pour ajouter des fonctionnalités interactives et dynamiques. Dans le frontend, une attention particulière est accordée à la conception de l'interface utilisateur en utilisant des techniques de design, de mise en page, de typographie, de couleurs et d'icônes. Les développeurs frontend font souvent appel à des Framework et des bibliothèques tels que React, Angular, Vue.js et jQuery, qui facilitent le développement d'applications web interactives et réactives. Ces outils offrent des fonctionnalités préconstruites et aident à organiser et à gérer le code de manière efficace.

4. Choix

Pour notre application de gestion de conférences avec diffusion en ligne, nous avons adopté une architecture monolithique. Ce choix est justifié par des raisons de fiabilité et d'efficacité sur sa réalisation selon le contexte où nous sommes. En effet cette application sera composée de deux modules en général, notamment la gestion de conférence et la diffusion en ligne, ce qui ne nous impose pas une orientation vers une approche micro service ou orienté service.

Concernant le choix de la technologie, nous avons choisi d'utiliser PHP en combinaison avec le Framework Laravel en raison de plusieurs avantages clés qu'ils offrent.

Tout d'abord, PHP est un langage de programmation largement utilisé dans le développement web. Il est reconnu pour sa simplicité et sa flexibilité, ce qui facilite la création de sites web dynamiques et interactifs. PHP bénéficie également d'une large communauté de développeurs, ce qui signifie qu'il est facile de trouver des ressources, des tutoriels et un support en cas de besoin.

En ce qui concerne Laravel, c'est l'un des Framework PHP les plus populaires et les plus puissants. Il offre une structure de développement solide et facilite la création d'applications web robustes et évolutives. Laravel intègre de nombreuses fonctionnalités prêtes à l'emploi, telles que l'authentification, la gestion des bases de données, la création de routes, la manipulation des sessions, etc. Cela nous permet de gagner du temps et d'améliorer notre productivité en tant que développeurs.

Un autre avantage majeur de Laravel est son système de gestion des migrations et des bases de données. Il nous permet de définir et de gérer facilement la structure de notre base de données, de créer des migrations pour mettre à jour notre schéma de base de données de manière transparente, et d'interagir avec la base de données à l'aide d'une syntaxe élégante et conviviale. Laravel suit également le principe de la programmation orientée objet (POO), ce qui rend notre code plus modulaire, réutilisable et facile à maintenir. Nous pouvons organiser notre code en utilisant des modèles (model), des vues (view) et des contrôleurs (controllers), ce qui facilite la séparation des préoccupations et garantit un code propre et bien structuré.

De plus, Laravel dispose d'une vaste bibliothèque de packages tiers, appelée Composer, qui offre une multitude de fonctionnalités supplémentaires prêtes à être intégrées à notre application. Cela nous permet d'étendre les fonctionnalités de base de Laravel et de bénéficier d'une plus grande flexibilité dans le développement de notre application.

Le tableau suivant présente notre stack technologique :

Domaine	Technologie
Langage de programmation	PHP
Framework	Laravel
Base de données	MySQL
Frontend	HTML, CSS, JavaScript
Serveur web	Apache
Contrôle de gestion	Git, GitHub
Outils de développement	Visual studio code, git bash, xampp, chrome

Figure 15 Stack technologique

IV. Conclusion

Dans ce chapitre nous avons examiné la méthodologie et les choix technologiques qui sous-tendent notre projet. Nous avons présenté deux approches pour la gestion du projet, à savoir la méthodologie traditionnelle (cascade) et la méthodologie Agile. Après avoir analysé les avantages et les inconvénients de chaque méthode, nous avons pris la décision de choisir une approche spécifique pour notre projet, qui est la méthodologie Agile. Cette méthode nous permettra de travailler de manière itérative, en favorisant la collaboration, la flexibilité et l'adaptabilité aux changements.

En ce qui concerne les choix technologiques, nous avons pris en compte plusieurs aspects tels que l'architecture, le backend et le frontend de notre application. Pour l'architecture, nous avons opté pour une approche qui facilite le développement rapide de notre application. Ce choix architectural est parfaitement adapté aux exigences de Laravel, que nous avons choisi pour le développement de notre application.

Chapitre 3 : Analyse et conception de l'application

I. Introduction

Après avoir choisi la méthodologie et les technologies à utiliser, nous allons passer à la conception en tant que telle de notre application. Celle-ci est une étape importante pour sa réalisation, dans cette phase, nous allons faire le cadrage en ressortissant les besoins fonctionnels et non fonctionnels de l'application, en décrivant d'une manière concise les besoins ou fonctionnalités pour chaque type d'utilisateur de notre système (user story) et enfin représenter les différents diagrammes de notre application, tout en se basant de notre cahier de charge.

II. Cahier des charges

La gestion de conférences et d'événements scientifiques nécessite une communication interne et externe. En interne, il s'avère crucial d'adopter une approche méthodologique dans la gestion afin de mieux orchestrer l'organisation et la réussite de l'événement. En externe, l'idée est d'avoir une communication visant à accroître la visibilité des informations liées à l'événement. Ainsi, face à ces exigences, il serait judicieux d'avoir une plateforme tout-en-un pour mieux gérer et diffuser les événements scientifiques organisés au sein de l'Université Assane Seck de Ziguinchor.

1. Objectifs

L'objectif principal de cette application est de faciliter la gestion des événements scientifiques organisés au sein de l'Université Assane Seck de Ziguinchor. L'application doit offrir une plateforme complète pour la planification, la publication, l'organisation, la promotion et le suivi des événements scientifiques. L'application doit aussi offrir la possibilité de publier et de gérer des appels à communication dans toutes ces étapes. C'est-à-dire, permettre aux responsables des évènements de publier des appels à communication, de recevoir et de traiter des propositions de communications pour les appels. Elle doit aussi leur permettre de gérer et de valider ou non les soumissions faites par les préposés communicants, mais aussi de diffuser en ligne l'évènement afin de permettre une participation à distance des événements et de soumettre des articles scientifiques.

III. Analyse des besoins et des fonctionnalités

1. Besoins fonctionnels

Un besoin fonctionnel est une exigence spécifique qui définit une fonctionnalité ou un comportement attendu d'un système logiciel. Il décrit ce que le logiciel doit être capable de faire du point de vue de l'utilisateur, en se concentrant sur les actions et les résultats attendus. Dans la méthodologie agile, les besoins fonctionnels sont couramment ressortis sous forme de description concise d'une fonctionnalité ou d'un besoin du point de vue de l'utilisateur. Ceci est appelé User Story. Concernant notre application, nous avons en général deux types d'utilisateurs, à savoir l'administrateur et l'utilisateur simple.

Les User Stories sont rédigées de manière simple et compréhensible par tous les membres de l'équipe, ce qui facilite la communication et la collaboration. Elles permettent de focaliser sur la valeur ajoutée pour l'utilisateur et de garder une vision centrée sur les besoins réels.

- User stories pour notre application

Admin : (responsables école doctorale)

- En tant qu'admin, je dois pouvoir publier un appel à communication et/ ou un événement
- En tant qu'admin je dois pouvoir consulter les propositions de communication
- En tant qu'admin je dois pouvoir valider ou non une proposition de communication
- En tant qu'admin je dois pouvoir notifier les utilisateurs auquel leurs proposition de communication est validée
- En tant qu'admin je dois pouvoir démarrer ou programmer une diffusion en ligne
- En tant qu'admin je dois pouvoir partager le code d'un appel aux participants

Utilisateur

- En tant qu'utilisateur je dois pouvoir consulter la liste des événements et m'y inscrire
- En tant qu'utilisateur je dois pouvoir envoyer une proposition de communication pour une appelle à communication
- En tant qu'utilisateur je dois pouvoir partager, modifier ou supprimer un article
- En tant qu'utilisateur je dois pouvoir participer à une visioconférence à travers le code fourni par l'administrateur.

Dans le cas de notre application, nous avons utilisé notion pour visualiser facilement l'état d'avancement des fonctionnalités, de prioriser les tâches et de gérer le flux de travail de

manière efficace. Dans notre tableau Kanban, les colonnes représentent généralement les différents états des tâches ou des User Stories, tels que "À faire", "En cours" et "Terminé". Les User Stories sont placées dans la colonne "À faire" au début, puis sont déplacées d'une colonne à l'autre au fur et à mesure de leur progression.

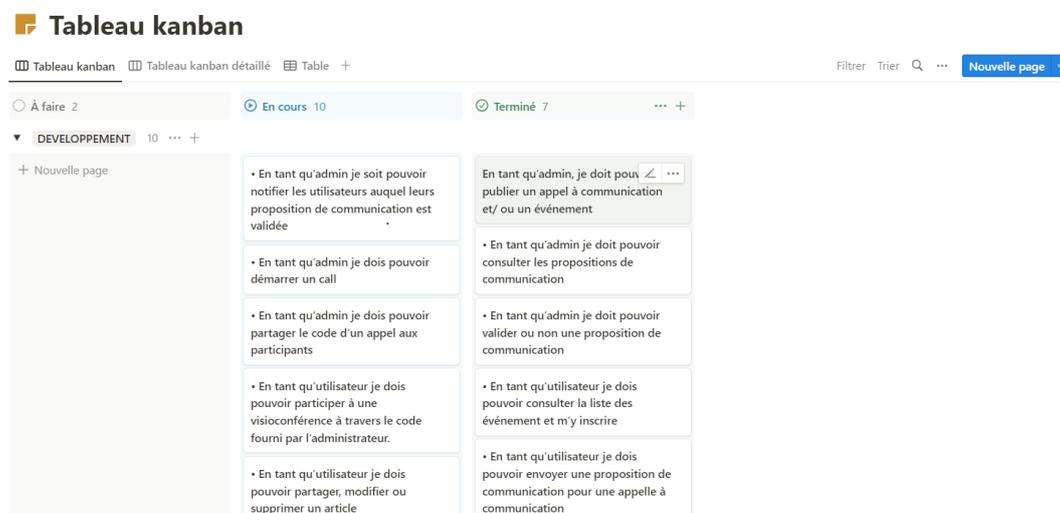


Figure 16 Tableau kanban

2. Besoins non fonctionnels

Un besoin non fonctionnel est une spécification qui concerne les caractéristiques et les contraintes d'une application, plutôt que ses fonctionnalités directes. Contrairement aux besoins fonctionnels, qui se concentrent sur ce que le logiciel doit faire, les besoins non fonctionnels se concentrent sur la manière dont le logiciel doit être.

Notre application de gestion de conférence a les exigences suivantes :

- **Sécurité** : le logiciel doit protéger les données sensibles et prévenir les accès non autorisés.
- **Convivialité** : le logiciel doit être intuitif et facile à utiliser, avec une interface utilisateur claire et cohérente.
- **Fiabilité** : le logiciel doit fonctionner de manière stable et prévisible, en évitant les pannes ou les comportements inattendus.
- **Maintenabilité** : le logiciel doit être conçu de manière à être facilement modifiable, extensible et testable.
- **Performance** : le logiciel doit répondre rapidement aux interactions de l'utilisateur ou traiter un grand volume de données dans des délais spécifiés.

IV. Conception

1. Diagramme de cas d'utilisation

Dans UML, un diagramme de cas d'utilisation communément appelé « use case » est une représentation graphique des informations des utilisateurs de notre application et leurs interactions avec cette dernière. Sa principale utilité c'est d'identifier les différentes façons dont les utilisateurs interagissent avec les besoins fonctionnels de notre application.

Comme mentionné précédemment, notre application à généralement deux utilisateurs principaux, notamment l'administrateur et l'utilisateur simple. *cf. Besoins fonctionnels*

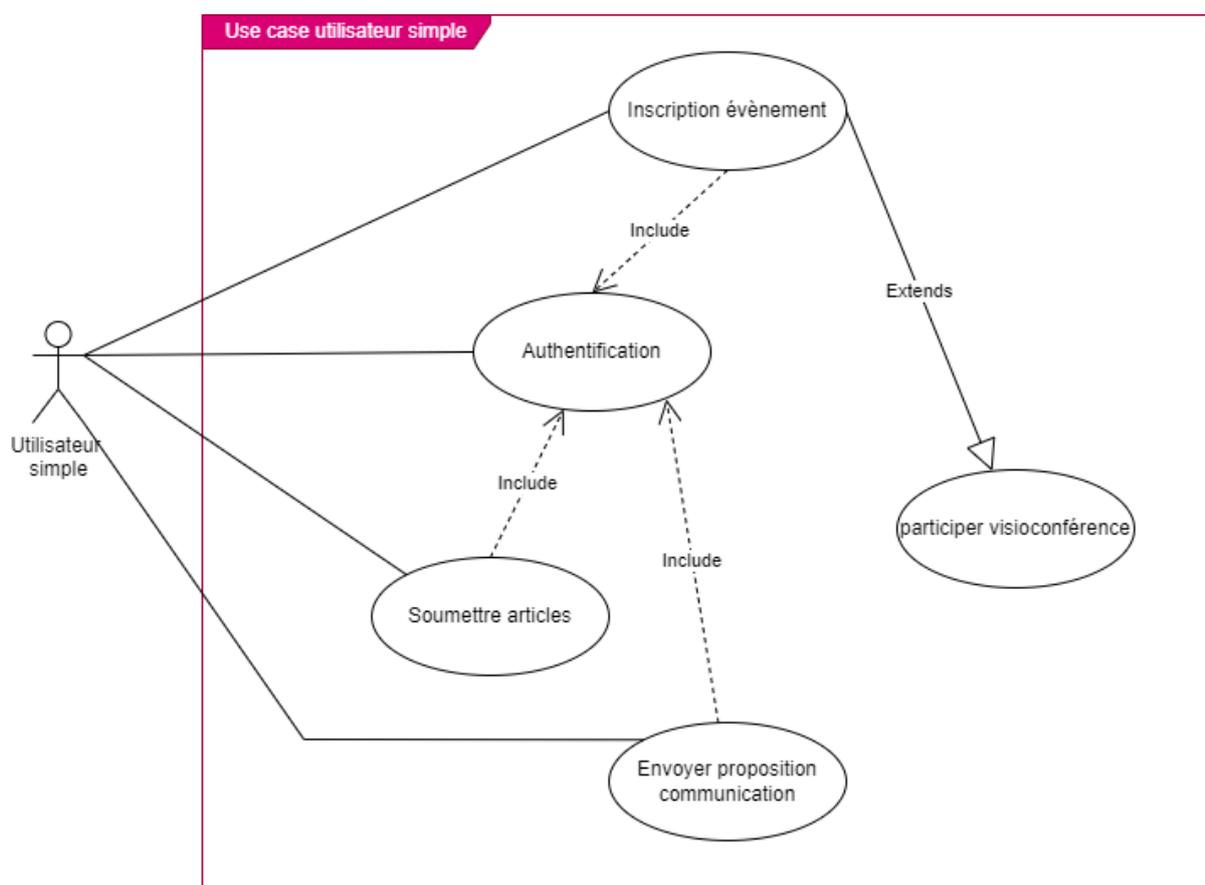


Figure 17 : use case utilisateur simple

Ce diagramme de cas d'utilisation pour l'utilisateur simple indique que dans notre système, l'utilisateur peut s'inscrire à un événement, publier un article et envoyer une proposition de communication. Cependant, pour effectuer ces tâches, il devra s'authentifier d'abord, ce qui est représenté par leur relation d'inclusion avec l'authentification. Il peut

également participer à une visioconférence, cette action se fait après s'être inscrite à un événement.

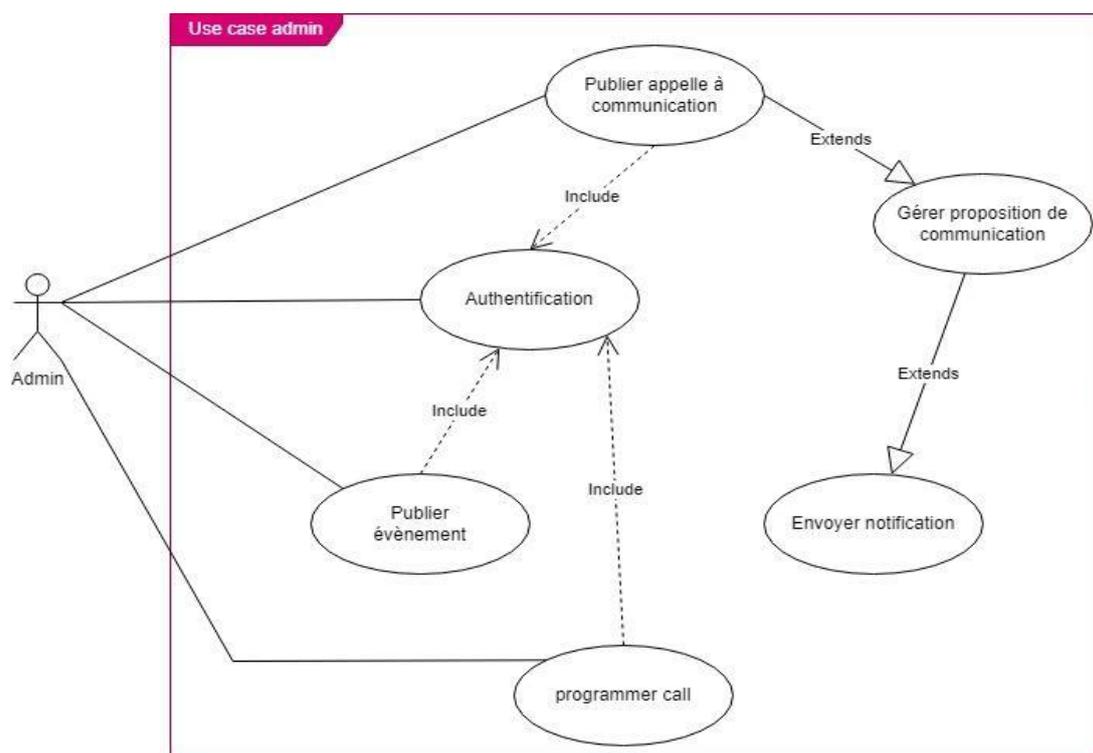


Figure 18 use case Administrateur

Le diagramme de cas d'utilisation pour l'administrateur décrit que l'administrateur peut publier un appel à communication, publier un événement et programmer un appel après s'être authentifié. Autrement dit, ces actions ne peuvent être exécutées par l'administrateur qu'après l'authentification. Il peut également gérer les propositions de communication et envoyer des notifications aux communicateurs après avoir validé ou non leurs propositions de communication.

2. Diagramme de séquence

Pour une représentation plus explicite et détaillée, nous allons construire nos diagrammes de séquences à partir des différents cas d'utilisations pour chaque utilisateur. En effet, le diagramme de séquence représente la séquence temporelle des interactions entre les objets ou les acteurs d'un système pendant l'exécution d'un scénario. Il offre une meilleure compréhension du système et permet d'analyser les différents scénarios associés à un cas d'utilisation spécifique en clarifiant sur comment les fonctionnalités du système sont mise en œuvre.

- Authentification

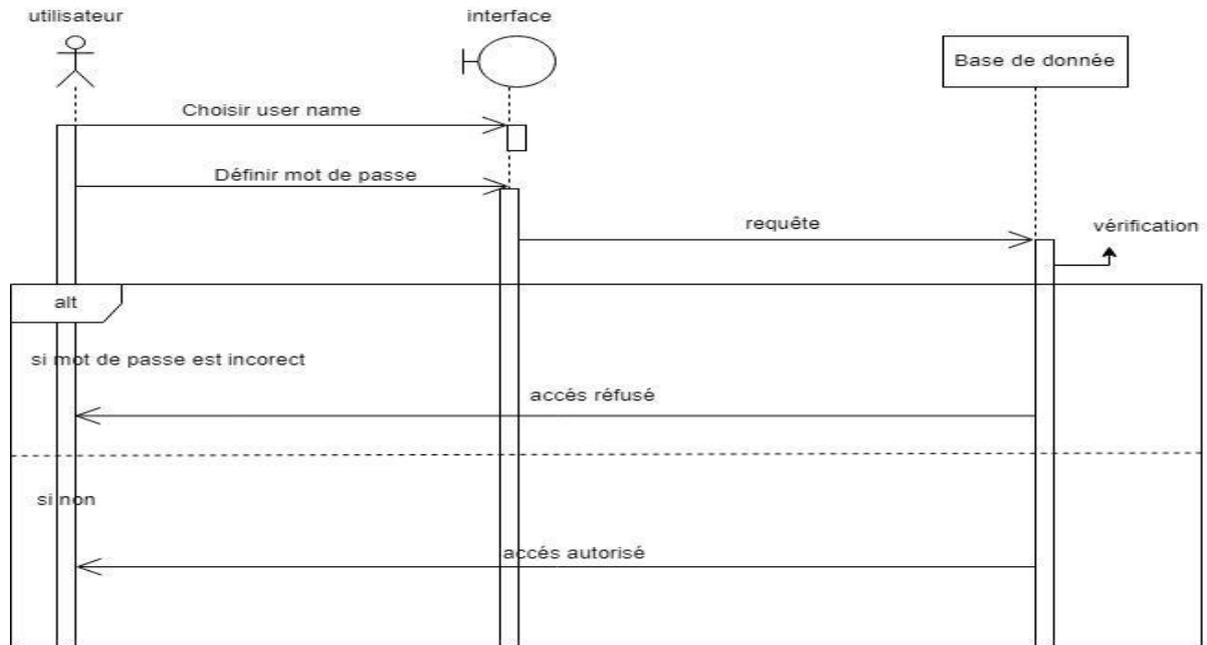


Figure 19 diagramme de séquence authentification

- Cas d'une publication d'appelle à communication

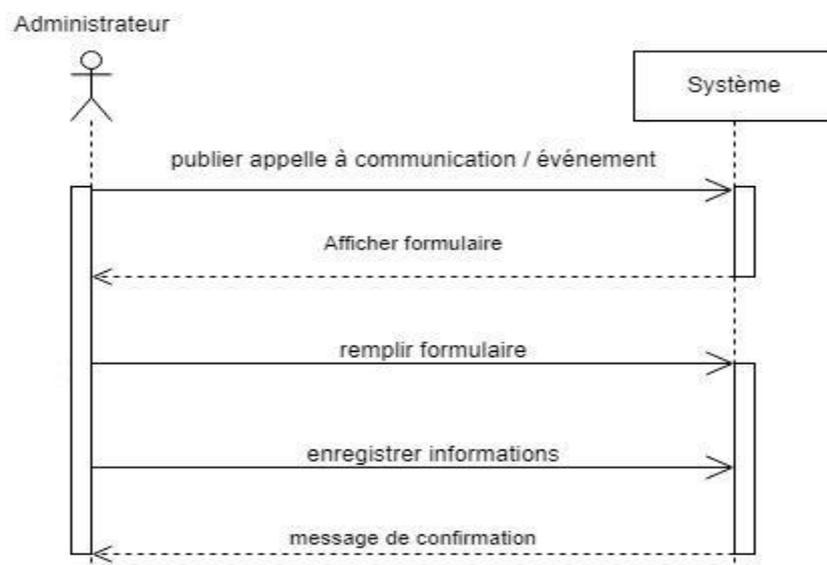


Figure 20 diagramme de séquence cas publication d'appelle à communication

- Cas gérer les propositions de communication

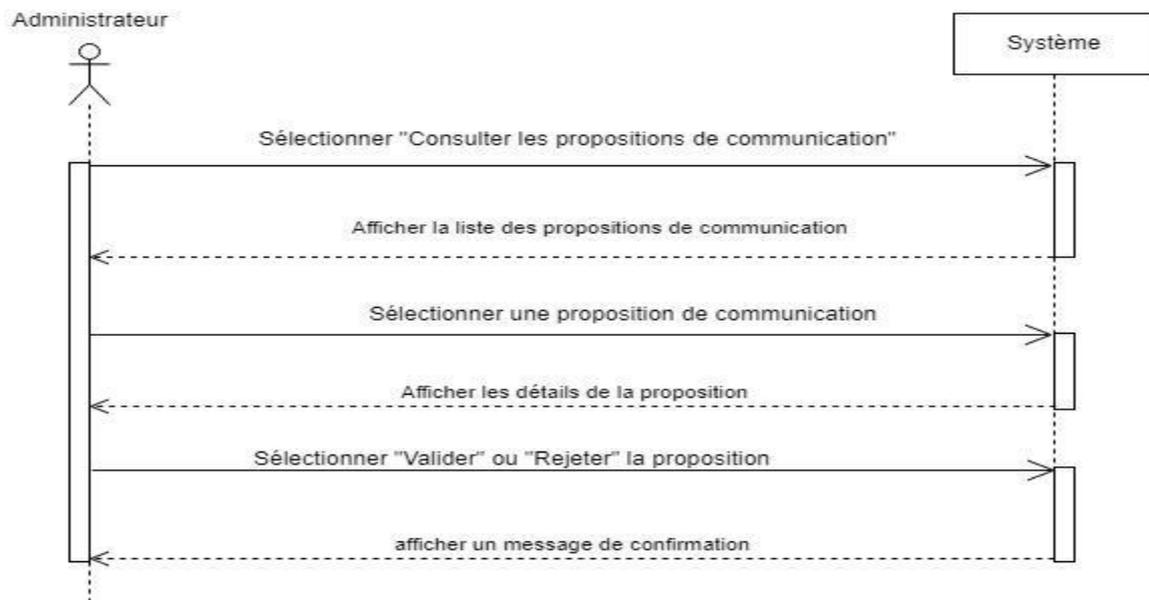


Figure 21 diagramme de séquence cas gestion propositions de communication

- Cas Inscription à un événement

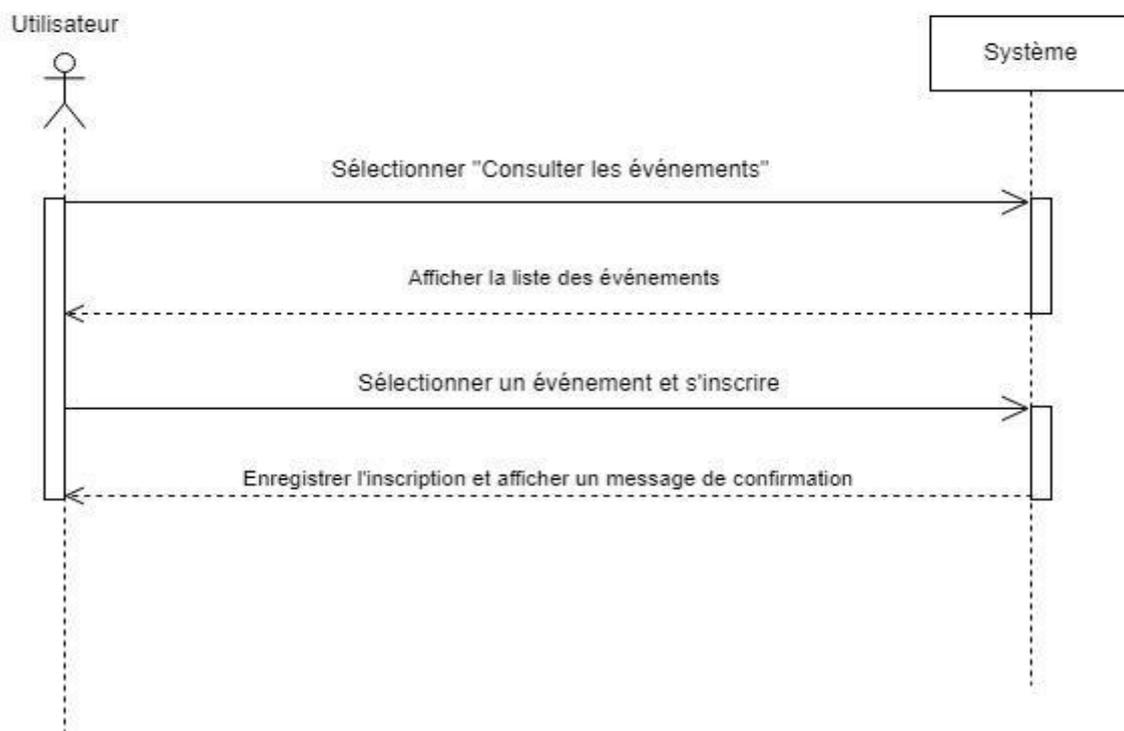


Figure 22 diagramme de séquence cas inscription événement

- Cas envoyer une proposition de communication pour un appel à communication

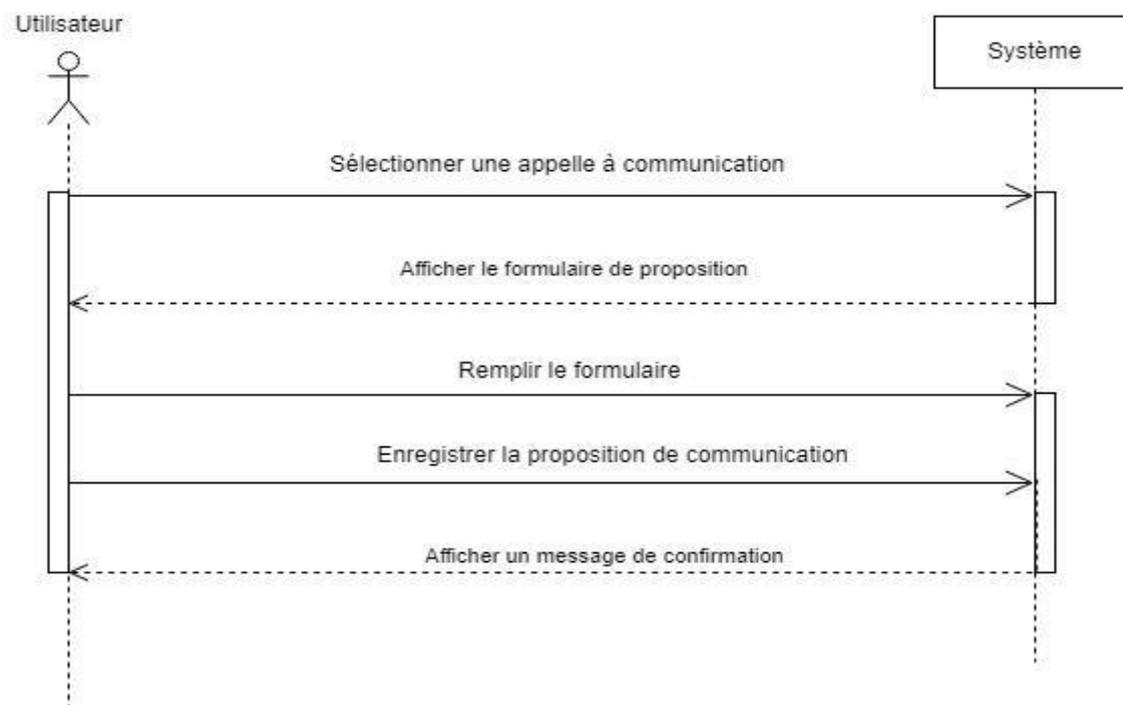


Figure 23 diagramme de séquence envoyer une proposition de communication

3. Diagramme des classes

Le diagramme de classe est la représentation qui va nous permettre de modéliser les classes de notre système en spécifiant généralement leurs attributs et les relations entre elles. En effet le diagramme de classe représente un élément important dans la modélisation et la conception orienté objet, car il permet d'identifier et d'avoir une vue d'ensemble sur les entités de notre système ainsi que de leurs relations. Etant un document visuel, le diagramme de classe sert aussi de documentation pour les développeurs, les testeurs et d'autres parties prenantes, facilitant la compréhension du système. Le diagramme ci-dessous représente comment est organisé les différentes entités de notre application ainsi que les relations qu'elles ont. En effet, dans UML (Unified Modeling Language), les diagrammes de classe montrent comment les différentes parties d'un programme informatique sont connectées. On utilise des flèches pour représenter ces relations. Par exemple, une "association" indique une connexion entre deux classes, pouvant aller dans une direction ou dans les deux sens. Les relations d'agrégation et de "composition" montrent comment des objets sont liés, avec des nuances sur leur durée de vie respective. Enfin, l'"héritage" montre qu'une classe hérite des caractéristiques d'une autre. Ces relations aident les développeurs à comprendre comment les morceaux d'un logiciel interagissent, facilitant ainsi la conception et la communication dans le processus de développement.

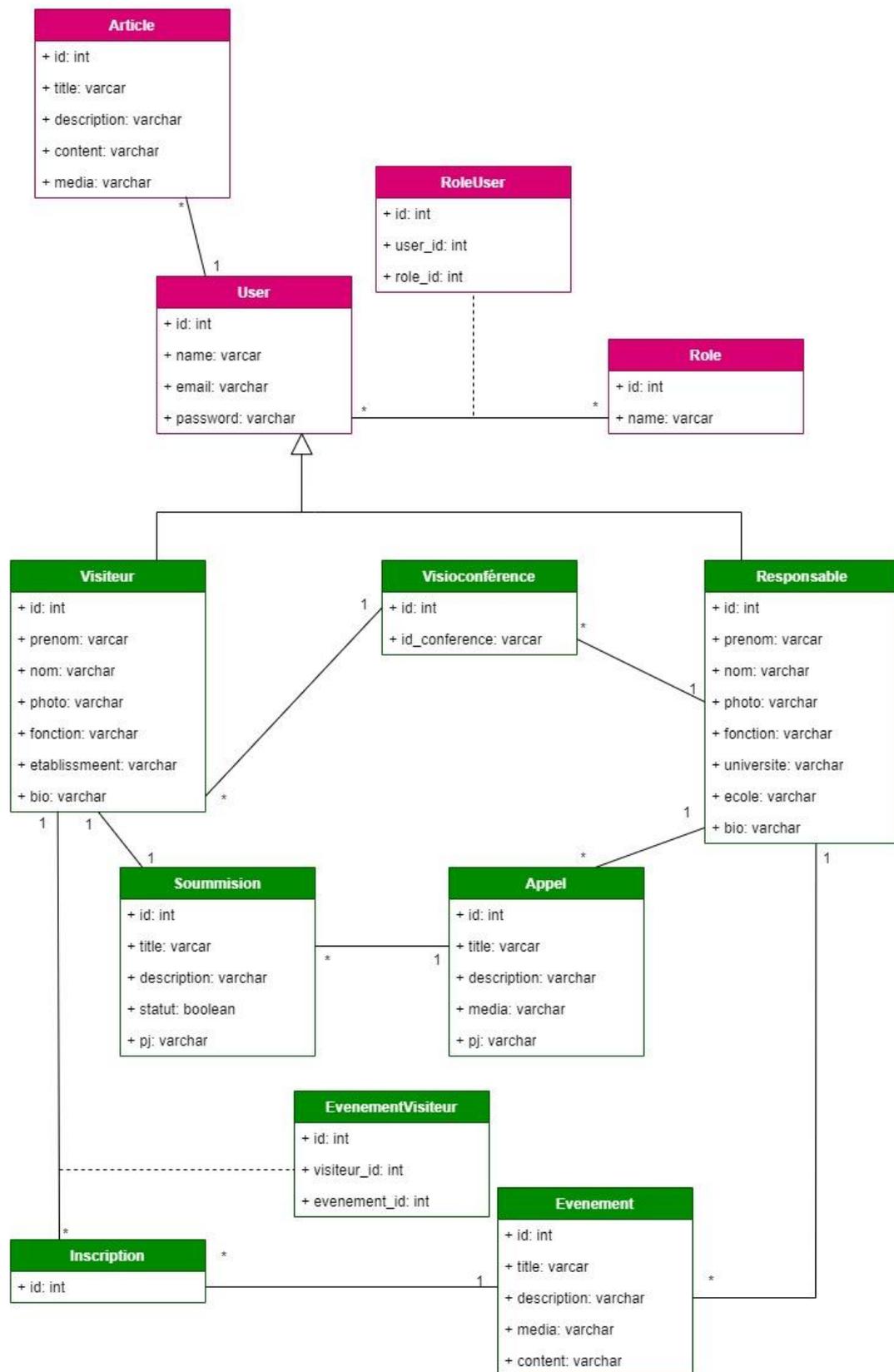


Figure 24 diagramme de classe

V. Conclusion

Dans ce chapitre portant sur l'analyse et la conception de notre application, nous avons mis en évidence les besoins fonctionnels et non fonctionnels de notre solution, répertorié les user stories, qui décrivent les actions requises pour chaque utilisateur. Nous avons également réalisé la conception en créant des diagrammes de classes, de séquences et de cas d'utilisation. Cette étape nous permettra de commencer la réalisation effective de notre application.

Chapitre 4 : Développement et présentation de l'application

I. Introduction

Après avoir terminé l'analyse et la conception de notre solution, nous pouvons alors entamer le développement proprement dit de l'application. Cette étape marque la transition entre la théorie et la pratique. En effet, dans ce chapitre, nous présenterons le processus de développement de notre application, ainsi que ses différentes fonctionnalités et interfaces.

II. Développement de l'application

Notre application est composée de trois (3) modules principaux : le module de gestion des événements, le module de gestion des conférences et de diffusion en ligne, et le module de gestion des articles. Pour leur développement, comme mentionné précédemment, nous avons choisi d'utiliser Laravel, un Framework qui offre une solution complète pour la logique de l'application ainsi que la création des différentes interfaces grâce au moteur de templating Blade.

L'authentification de l'application est gérée par Fortify, tandis que le système de gestion des rôles et des permissions est implémenté à l'aide de la fonctionnalité d'autorisation (Policy) proposé par Laravel.

Pour la diffusion en ligne, nous avons utilisé WebRTC (Web Real-Time Communication), une technologie qui permet d'intégrer un système de communication instantanée directement dans le navigateur. Ces choix de développement garantissent une architecture solide et une expérience utilisateur fluide et sécurisée pour notre application.

1. Développement avec Laravel

Laravel est un Framework PHP convivial qui facilite le développement d'applications web. Il offre de nombreuses fonctionnalités, notamment l'injection de dépendances, une couche d'abstraction de base de données expressive grâce à son ORM (Object Relational Mapping) appelé Éloquent, la gestion des files d'attente et des tâches planifiées, ainsi que des outils pour les tests unitaires et d'intégration etc. [21].

Laravel est un Framework progressif et évolutif, bénéficiant d'une large communauté de développeurs. Il met à leur disposition une variété de bibliothèques préprogrammées, telles que l'authentification, le routage et la création de modèles HTML, ce qui simplifie grandement le processus de création rapide d'applications web robustes.

Il suit l'architecture Modèle-Vue-Contrôleur (MVC), qui favorise la séparation des préoccupations et la maintenabilité du code. Dans le modèle MVC, le modèle représente les données et leur logique métier, la vue est responsable de l'affichage des données et l'interface utilisateur, tandis que le contrôleur gère les interactions entre le modèle et la vue, ainsi que le flux de l'application. Cette approche facilite la structuration du code et permet une meilleure réutilisation et testabilité des différentes composantes de l'application.

Laravel est un Framework PHP puissant, doté d'une riche gamme de fonctionnalités, d'une communauté active et d'une architecture MVC qui favorise le développement d'applications web robustes et maintenables.

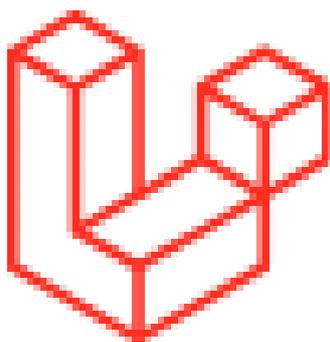


Figure 25 logo Laravel

2. Système d'authentification avec Laravel Fortify

L'authentification joue un rôle essentiel dans la sécurité d'une application, mais sa mise en œuvre reste complexe et nécessite une approche prudente. Heureusement, Laravel propose des kits de démarrage (starter kits) tels que Laravel Breeze, Laravel Jetstream et Laravel Fortify, qui offrent des solutions d'authentification faciles à mettre en place et sécurisées.

Laravel Breeze est un kit de démarrage léger pour l'authentification. Il propose une configuration simple et minimaliste pour les fonctionnalités d'authentification de base, telles que l'inscription, la connexion, la réinitialisation de mot de passe et la vérification par e-mail. Breeze utilise le moteur de templates Blade intégré à Laravel, ce qui facilite la personnalisation et l'extension des vues et des contrôleurs en fonction des besoins spécifiques de l'application.

Laravel Jetstream, quant à lui, est une bibliothèque d'infrastructure d'authentification plus complète. Elle constitue une base solide pour développer des applications modernes et riches en fonctionnalités, avec une prise en charge de la multi-tenue. Jetstream propose deux options de pile : Livewire et Inertia. La pile Livewire utilise Laravel Livewire pour les composants rendus côté serveur, tandis que la pile Inertia s'intègre aux Framework côté client tels que Vue.js ou React, permettant ainsi une approche d'application monopage. Jetstream offre des

fonctionnalités avancées, telles que l'authentification à deux facteurs, la gestion d'équipe, la génération de jetons d'API, et bien plus encore.

Enfin, Laravel Fortify est une infrastructure d'authentification flexible pour les applications Laravel. Elle propose un ensemble de composants et de fonctionnalités d'authentification personnalisables. Fortify permet aux développeurs de définir leurs propres routes, contrôleurs et vues d'authentification, ce qui leur donne un contrôle total sur le processus d'authentification. En s'intégrant parfaitement aux systèmes d'authentification et de gestion de session de Laravel, Fortify facilite l'intégration de besoins et de flux d'authentification personnalisés dans une application.

Pour notre projet, nous avons choisi d'utiliser Laravel Fortify afin de bénéficier d'une plus grande liberté dans l'organisation des routes et des interfaces, en accord avec nos besoins spécifiques.

3. Les Policy dans Laravel

Pour garantir la sécurité de nos applications, Laravel propose un système d'autorisation qui nous permet d'implémenter des contrôles d'accès aux actions des utilisateurs dans notre application, que ce soit sur une ressource spécifique ou une fonctionnalité particulière. Par exemple, seuls les administrateurs auront accès à la gestion des utilisateurs, ou bien seul l'administrateur est autorisé à modifier les rôles d'un utilisateur, etc. Les fonctionnalités d'autorisation de Laravel offrent une approche simple et organisée pour gérer ce type de vérification d'autorisation. Laravel propose deux principales méthodes pour gérer l'autorisation des actions : les **gates** et les **policies**.

Les gates sont des fermetures (closures) qui déterminent si un utilisateur est autorisé à effectuer une action spécifique. Ils permettent de définir des règles personnalisées pour contrôler l'accès en fonction de différents critères, tels que le rôle de l'utilisateur, les permissions, ou toute autre condition spécifique. Les gates offrent une flexibilité et une granularité élevées dans la définition des autorisations.

Quant aux policies, ce sont des classes qui organisent la logique d'autorisation autour d'un modèle ou d'une ressource spécifique. Elles fournissent des méthodes pour déterminer si un utilisateur est autorisé à effectuer différentes actions sur le modèle ou la ressource concernée. Les policies permettent une encapsulation claire de la logique d'autorisation et facilitent la réutilisation des règles d'autorisation dans différentes parties de l'application.

4. Moteur de templating blade

Laravel propose le moteur de templating Blade intégré, qui offre une syntaxe simple et expressive pour la création de vues dans nos applications. Blade permet d'insérer du code PHP directement dans les fichiers de vues, tout en facilitant la séparation claire entre le code PHP et le HTML. De plus, Blade offre une excellente intégration avec les autres fonctionnalités de Laravel, telles que l'injection de dépendances, l'internationalisation, la gestion des erreurs, etc. Cela nous permet de créer des vues dynamiques et interactives, tout en bénéficiant de la puissance et de la simplicité de Blade.

Dans Laravel, les vues et les composants peuvent bénéficier des fonctionnalités d'héritage, ce qui permet d'organiser et de réutiliser efficacement le code de présentation. L'héritage des vues dans Laravel est réalisé à l'aide de la fonctionnalité de "layout" ou de "master layout". Un layout est un modèle de base qui définit la structure commune de toutes les pages de notre application. Il peut contenir des éléments tels que l'en-tête, le pied de page, les styles CSS, les scripts JavaScript, etc. Les vues spécifiques à chaque page peuvent ensuite étendre ce layout en utilisant la directive **@extends** pour définir le layout parent. Ainsi, les vues enfant héritent de la structure et des éléments du layout parent, ce qui permet d'éviter la répétition de code et de maintenir une cohérence visuelle dans toute l'application.

On peut aussi modulariser nos vues pour faciliter leur réutilisation grâce aux composants. Les composants peuvent être considérés comme des mini-vues autonomes avec leur propre logique et leur propre structure. Ils peuvent être inclus dans les vues à l'aide de la directive **@component** ou en utilisant les raccourcis Blade tels que **@include** ou **@component** pour les inclure dynamiquement.

5. WebRTC

WebRTC (Web Real-Time Communication) est un ensemble de protocoles et de technologies qui permet de mettre en place un système de communication instantanée sur le web [20]. Il offre un moyen d'avoir une communication audio et vidéo en temps réel sans plugin, avec une latence très faible, directement depuis le navigateur. Toutefois, la réalisation d'une solution de visioconférence, comme dans notre cas, nécessite l'utilisation de services ou de modules spécifiques qui peuvent être payants, ce qui n'est pas toujours intéressant. Ainsi, WebRTC rend la communication instantanée sur le web beaucoup plus simple. WebRTC est généralement composé de trois API : `MediaStream`, `PeerConnection` et `DataChannel`.

L'API `MediaStream` permet d'accéder aux flux audio et vidéo à partir de périphériques tels que les webcams et les microphones. Elle fournit des méthodes pour capturer et manipuler les flux média. Les objets `MediaStream` peuvent être utilisés pour envoyer des données audio et vidéo entre les pairs lors d'une session WebRTC.

L'API `PeerConnection` est utilisée pour établir et gérer une connexion entre deux ou plusieurs pairs. Elle permet une communication sécurisée et chiffrée en établissant une connexion directe de pair à pair. `PeerConnection` gère la négociation et l'établissement de la connexion, ainsi que la transmission et la réception des flux audio, vidéo et de données.

L'API `DataChannel` fournit un canal bidirectionnel fiable pour l'envoi de données arbitraires entre les pairs. Elle permet aux applications d'échanger des données autres que des médias, telles que des messages texte ou des données binaires, via la connexion `PeerConnection` établie. `DataChannel` offre des fonctionnalités telles que la livraison des messages dans l'ordre ou non, la fiabilité des messages et le contrôle de congestion.

La mise en place d'une communication WebRTC entre deux utilisateurs A et B se déroule de la manière suivante : Tout d'abord, A et Binstancient un `RTCPeerConnection`, qui est le point central de la communication WebRTC. Ensuite, A et B génèrent chacun une offre au format SDP (Session Description Protocol), qui contient les informations nécessaires pour établir la connexion. Pour échanger ces offres, A envoie son offre à B et B envoie la sienne à A. Cette étape est appelée le signaling et peut être réalisée via divers moyens tels que des websockets, des requêtes AJAX ou même par e-mail. Une fois qu'une offre est reçue, elle est ajoutée à l'instance de `RTCPeerConnection` correspondante à l'aide de la méthode `setRemoteDescription`. Cela permet à chaque utilisateur de connaître les détails de l'autre participant à la communication. Cependant, si la communication doit se faire entre plus de deux utilisateurs, l'envoi et la réception se feront entre tous les utilisateurs, ce qui rendra le système complexe et pourrait causer une communication extrêmement lente. Donc dans ce cas, un serveur centralisé est nécessaire pour permettre aux utilisateurs de se découvrir sans que chacun se charge d'envoyer une offre à chaque utilisateur du système.

a. Serveur de signalisation WebRTC (WebRTC Signaling Server)

Un serveur de signalisation permet aux paires de se découvrir et facilite l'échange d'informations nécessaires à l'établissement d'une connexion, telles que les adresses IP, les offres et les réponses SDP (Session Description Protocol), ainsi que les candidats ICE (Interactive Connectivity Establishment) [22]. Cependant, le serveur de signalisation n'est pas responsable du transfert réel des flux audio, vidéo ou de données entre les utilisateurs. Une fois

que la connexion est établie grâce à la signalisation, les données sont transmises directement entre les pairs via les connexions PeerConnection, sans passer par le serveur de signalisation. Dans la plupart des cas, nos machines ne sont pas directement connectées à Internet, mais sont plutôt situées derrière plusieurs couches de réseau. Nous rencontrons souvent un NAT (Network Address Translation) qui permet de router le trafic de plusieurs machines à travers une seule adresse IP externe, généralement celle du routeur auquel nous sommes connectés, plutôt que l'adresse IP de votre machine individuelle. En outre, nous faisons face à des pare-feux qui filtrent certains protocoles et ports, ainsi qu'à des proxys qui masquent l'adresse IP d'origine de notre machine. Ces configurations réseau peuvent entraîner des difficultés lorsqu'il s'agit d'établir une connexion directe entre les utilisateurs. C'est à ce moment qu'un serveur STUN/TURN peut intervenir pour surmonter ces obstacles et faciliter la communication.

b. Serveur STUN/TURN

Pour surmonter les obstacles mentionnés précédemment, le WebRTC utilise la technique Interactive Connectivity Establishment (ICE). Cette technique vise à trouver le chemin le plus direct possible pour permettre la communication entre deux machines sur le réseau, contournant ainsi les problématiques évoquées.

Lorsqu'il s'agit d'établir une connexion, un serveur STUN entre en jeu. Ce serveur permet d'obtenir l'adresse externe d'un utilisateur, contournant ainsi les configurations de NAT et révélant l'adresse IP publique nécessaire pour établir la communication.

Cependant, dans certains cas, une connexion directe n'est pas possible malgré l'utilisation de STUN. C'est là qu'intervient un serveur TURN. En réalité, un serveur TURN est un serveur STUN qui inclut également une fonctionnalité de relais. Il agit comme un intermédiaire en relayant le trafic entre les utilisateurs lorsque la connexion directe n'est pas réalisable, par exemple en raison de restrictions réseau ou de pare-feux.

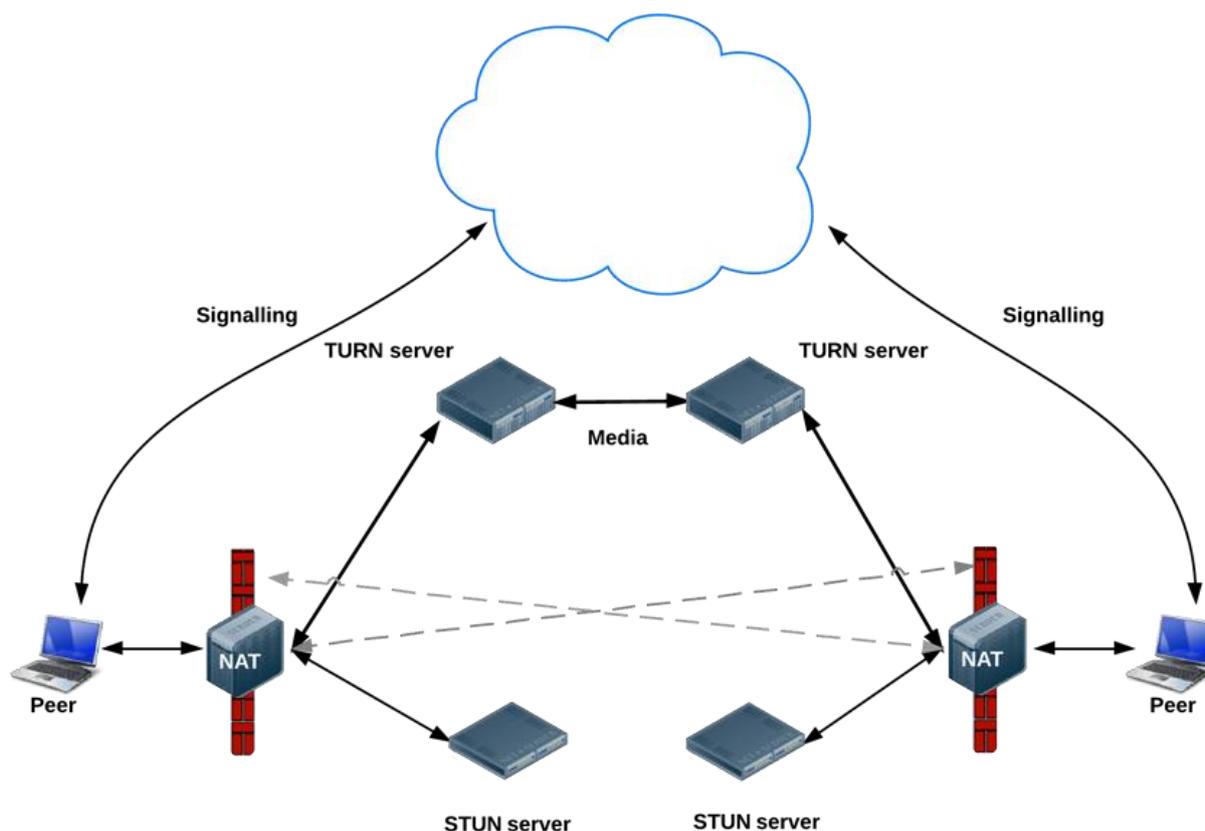


Figure 26 Webrtc communication[23]

c. Serveur WebRTC Metered | Video

Pour notre application de visioconférence avec plusieurs participants. Chaque participant souhaite partager sa vidéo en temps réel avec les autres participants.

Sans l'utilisation d'un serveur de médias, chaque participant devrait envoyer sa vidéo à chaque autre participant de l'appel. Cela entraînerait une multiplication du trafic vidéo, augmentant ainsi la charge de travail pour chaque appareil participant. De plus, avec un grand nombre de participants, la bande passante disponible pourrait rapidement devenir insuffisante, entraînant des problèmes de latence et de qualité vidéo. En utilisant un serveur de médias, la transmission des flux vidéo est optimisée. Lorsqu'un participant envoie sa vidéo, celle-ci est envoyée au serveur de médias. Le serveur de médias traite alors le flux vidéo et le redistribue aux autres participants. Au lieu de chaque participant ayant à gérer plusieurs flux vidéo en même temps, ils reçoivent simplement le flux vidéo traité par le serveur. Cela permet de réduire la charge de travail sur les appareils des participants et d'améliorer la qualité et la stabilité globale de la diffusion vidéo.

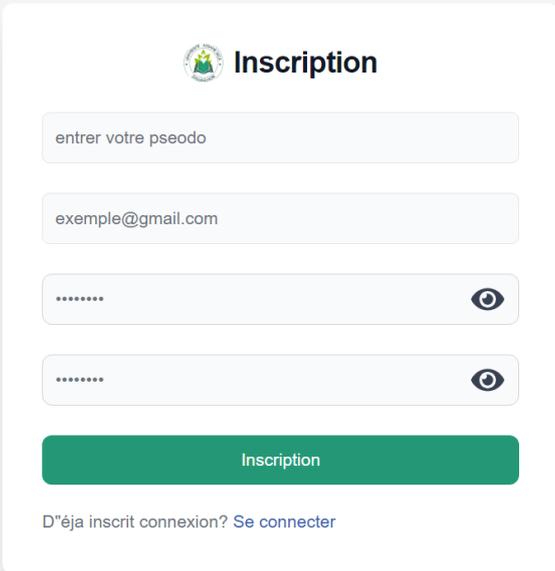
Ainsi dans le cadre de notre application, nous avons utilisé le SDK metered vidéo pour gérer ses limitations. Metered vidéo est un kit de développement logiciel permettant de [24]:

- Intégrez les appels vidéo dans une application ou un site Web existant en quelques minutes.
- Créez des réunions pouvant accueillir des milliers d'utilisateurs.
- Enregistrez la réunion.
- Créez une vidéo composée de tous les participants à la réunion.
- Diffusez la réunion en direct en utilisant HLS.
- Diffusez la réunion sur des services tiers tels que YouTube Live, Twitch Live, Vimeo Live, etc. via RTMP Out.

III. Présentation de l'application

1. Présentation des interfaces

- **Pages d'inscription et de connexion**



The image shows a registration form titled "Inscription" with a logo on the left. The form contains the following elements:

- A text input field with the placeholder "entrer votre pseudo".
- A text input field with the placeholder "exemple@gmail.com".
- A password input field with a masked password "....." and an eye icon to toggle visibility.
- A second password input field with a masked password "....." and an eye icon to toggle visibility.
- A prominent green button labeled "Inscription".
- A link below the button that reads "D' déjà inscrit connexion? [Se connecter](#)".

Figure 27 page inscription

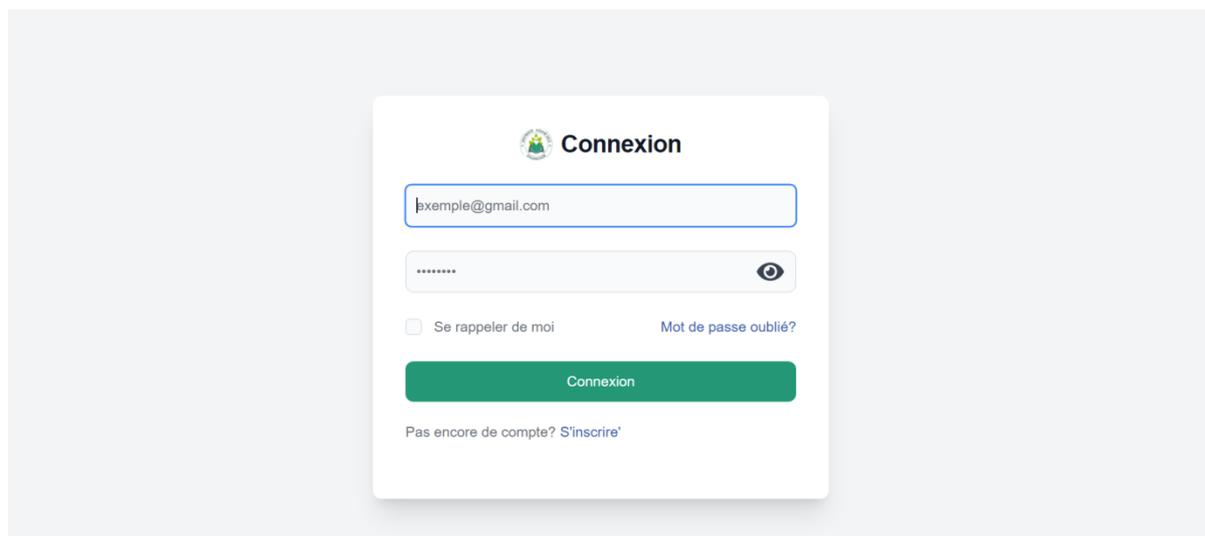


Figure 28 pages connexion

- Publication d'une appelle à communication

Cette interface réservée à l'administrateur ou aux responsables des écoles doctorales leur servira à publier un appel à communication.

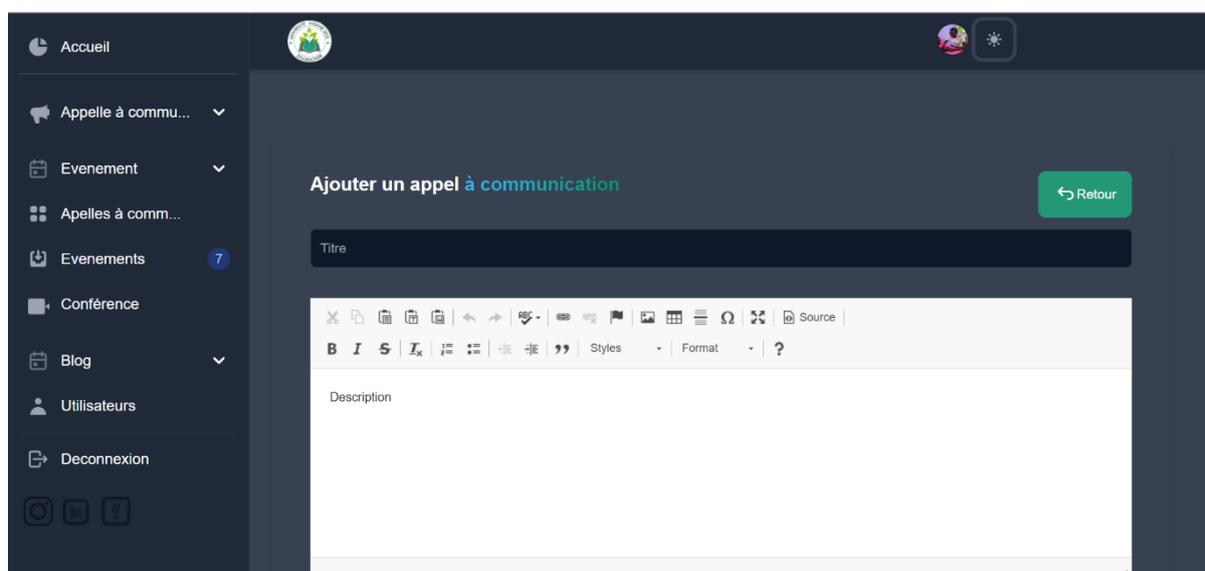


Figure 29 pages publication appelle à communication

- Liste des appels à communication publiées

Dans cette interface, le responsable pourra voir la liste des appels à communication qu'il a publiés. Chaque appel est représenté par une carte sur laquelle on trouve le titre de l'appel, la description, ainsi que deux options : une pour ouvrir ou télécharger la pièce jointe, et l'autre pour voir les détails de l'appel, afin de pouvoir le modifier, le supprimer ou consulter les propositions de communication pour cet appel.

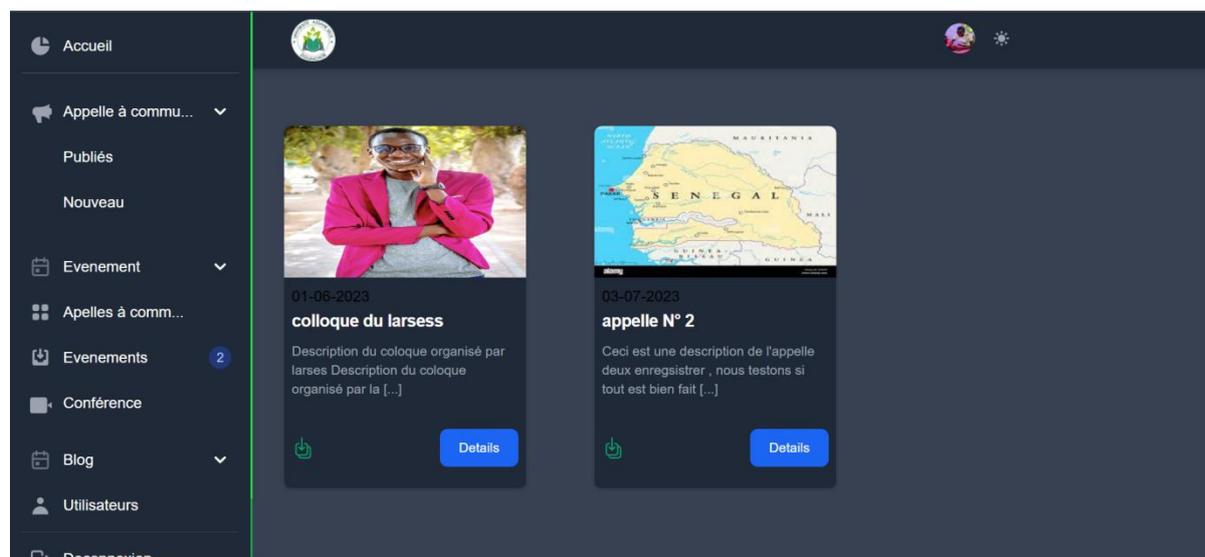


Figure 30 pages liste appelle à communication

- Liste des soumissions pour un appel

Dans cette interface, nous avons un tableau regroupant les propositions de communication. L'administrateur peut cliquer sur "Étudier" pour voir les détails de la soumission et décider de valider ou de refuser la proposition. Dans ce tableau, les appels déjà étudiés et acceptés sont marqués comme "Accepté" dans la colonne "Statut", tandis que ceux non approuvés sont marqués comme "Refusé".

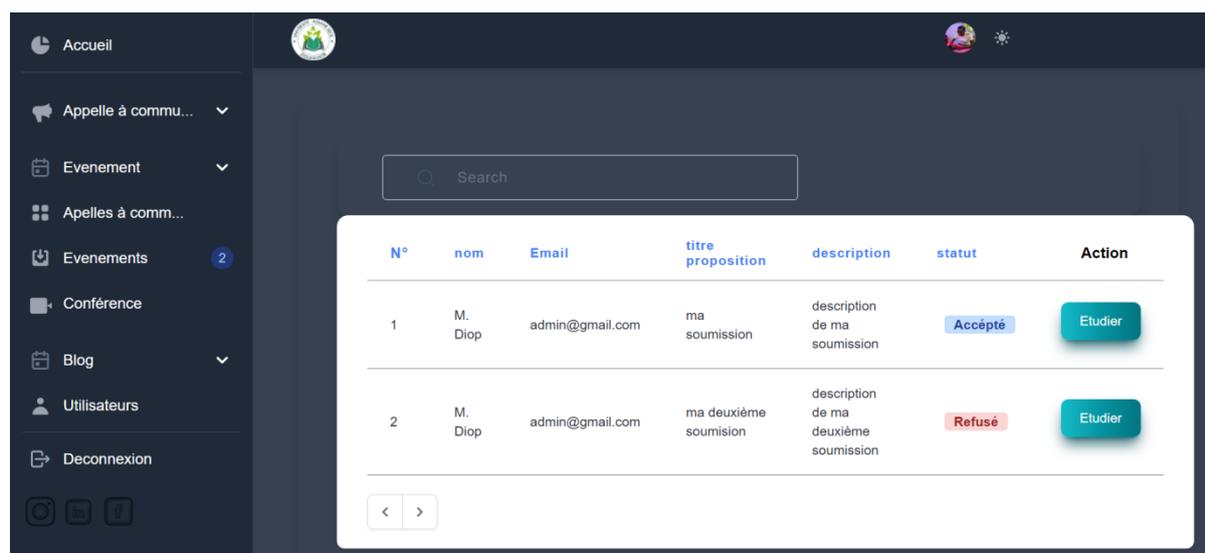


Figure 31 pages liste des soumissions pour un appel

- Étudier une proposition

Sur cette page, on peut voir les détails d'une proposition de communication, accompagnés des boutons "Accepter" et "Rejeter". On peut également visualiser le fichier joint soumis par le demandeur. Une fois la proposition acceptée ou refusée, le demandeur sera notifié de la décision par e-mail.



Figure 32 pages étudier une proposition

- Création salon de conférence

Le responsable pourra à partir de cette page créer une nouvelle réunion en ligne. Une fois que le bouton "Nouvelle réunion" est cliqué, le système lui enverra une clé unique qu'il pourra partager avec les participants. Ensuite, il sera redirigé vers la page de la conférence.

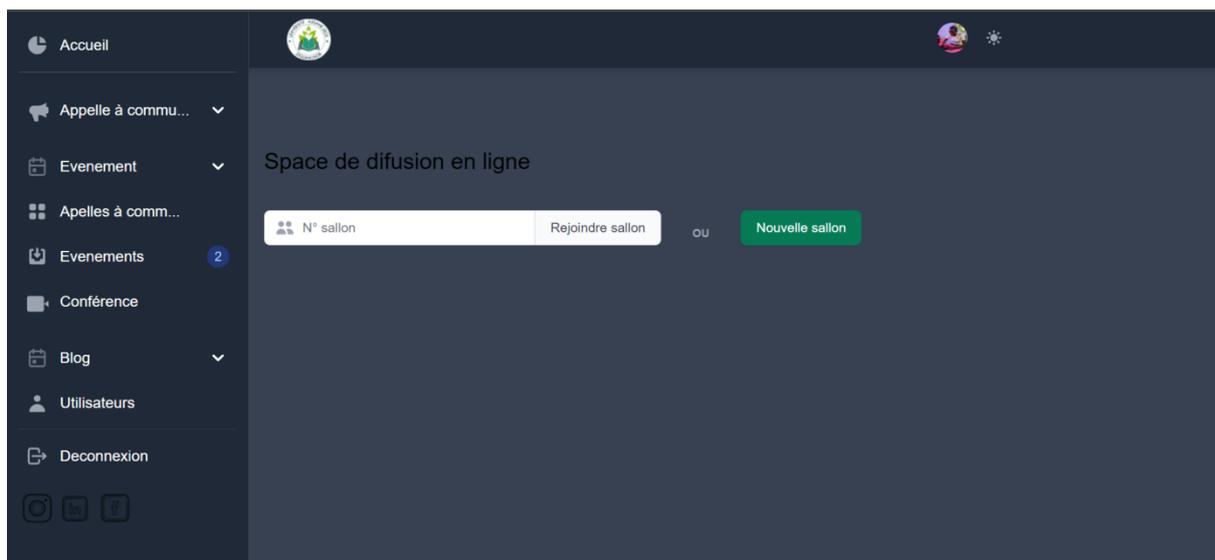


Figure 33 pages création salon de conférence

- Paramètres de l'appelle en ligne

À cette étape, le participant à la visioconférence pourra vérifier l'état de ses périphériques, notamment son microphone et sa caméra, et renseigner le nom qui s'affiche pour les autres participants. Il pourra choisir les périphériques qu'il utilisera lors de l'appel.

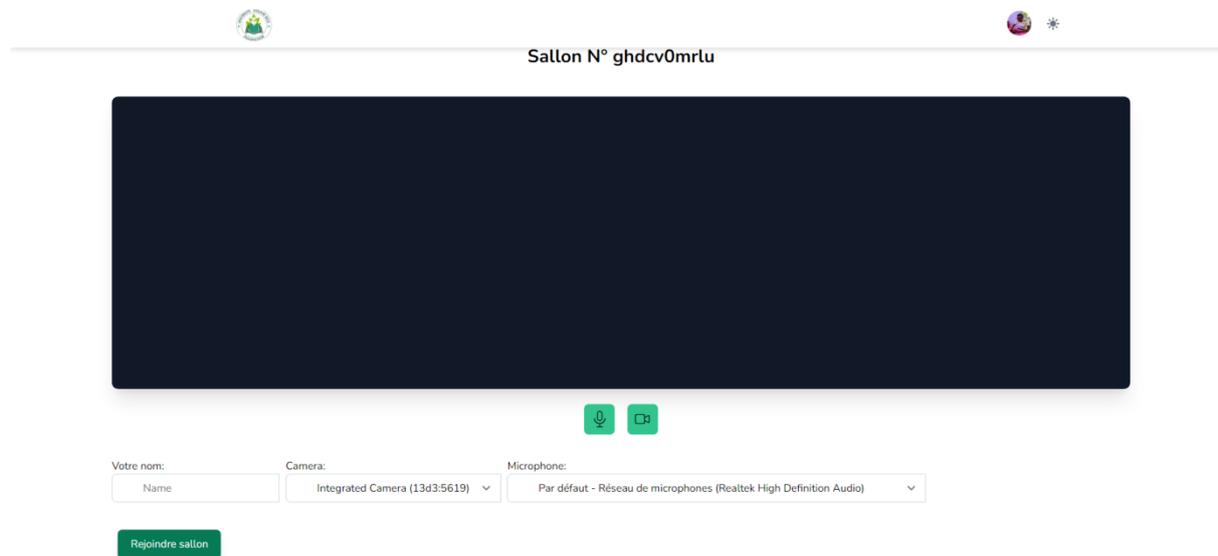


Figure 34 pages paramètres de l'appelle en ligne

- Visio-conférence

La page de visioconférence permet d'organiser des diffusions en ligne. Les participants pourront partager leur écran, activer ou désactiver leur caméra ou leur microphone. Ils auront également la possibilité de quitter l'appel à tout moment et de revenir ultérieurement grâce à la clé unique partagée par l'organisateur.

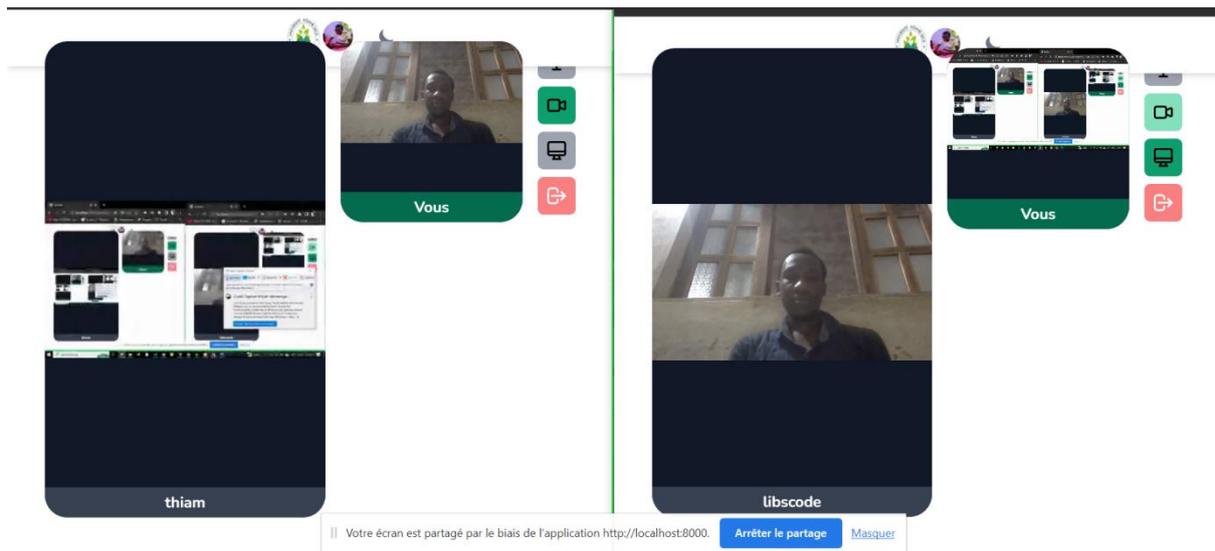


Figure 35 pages visioconférence

- Gestion des utilisateurs

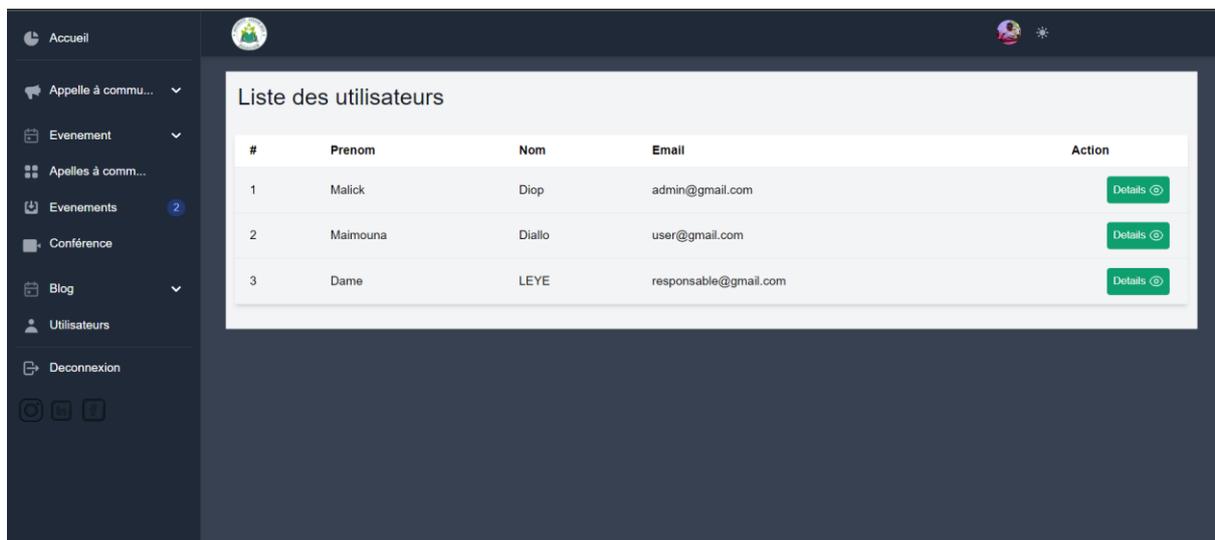


Figure 36 pages gestion des utilisateurs

- Liste des événements

Les utilisateurs peuvent consulter la liste de tous les événements publiés et s'inscrire à ceux qui les intéressent.

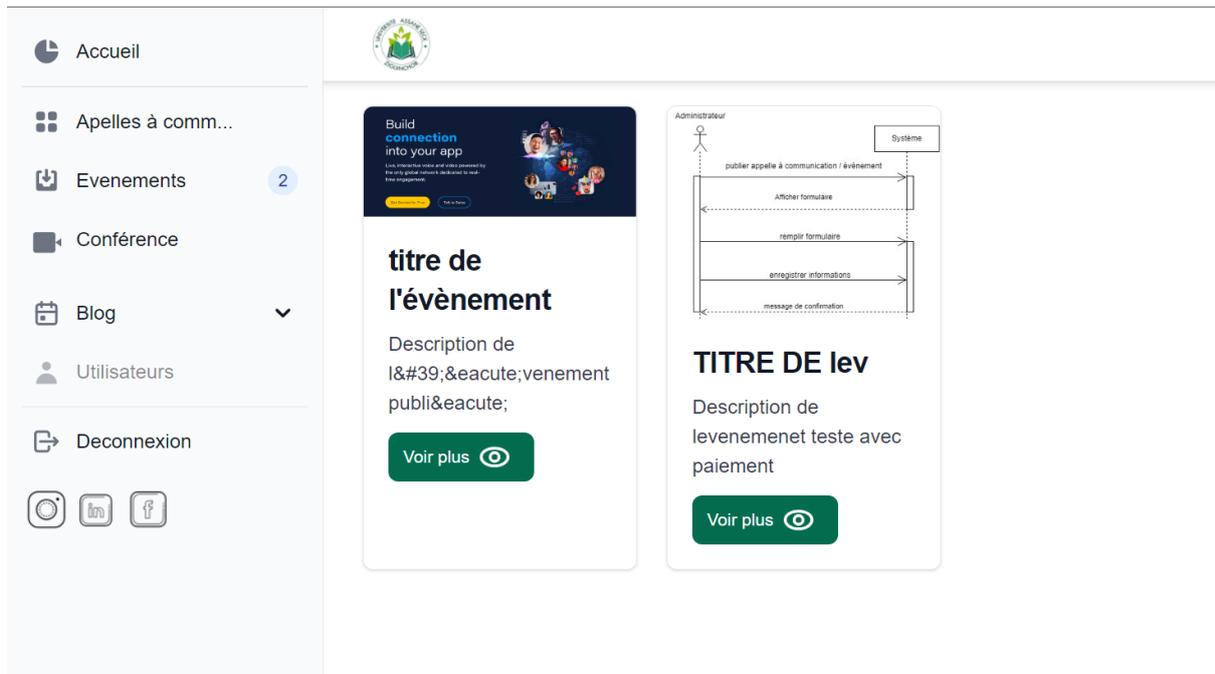


Figure 37 pages liste des événements

- Détails d'un événement

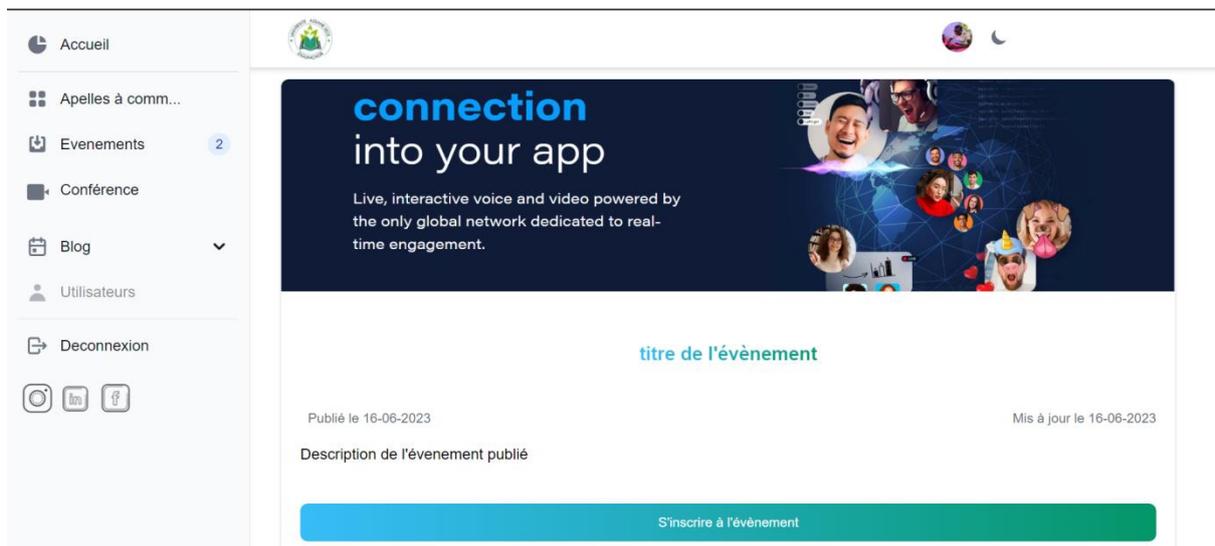


Figure 38 pages détails d'un événement

- Soumettre une proposition de communication

Afin d'améliorer l'expérience utilisateur, nous avons simplifié le processus de soumission d'une proposition de communication en créant un formulaire sous forme de pop-up. Cela permet aux communicateurs intéressés, à partir des détails de l'appel, d'envoyer leur proposition en saisissant le titre, la description et en joignant éventuellement un fichier pour fournir plus d'informations.

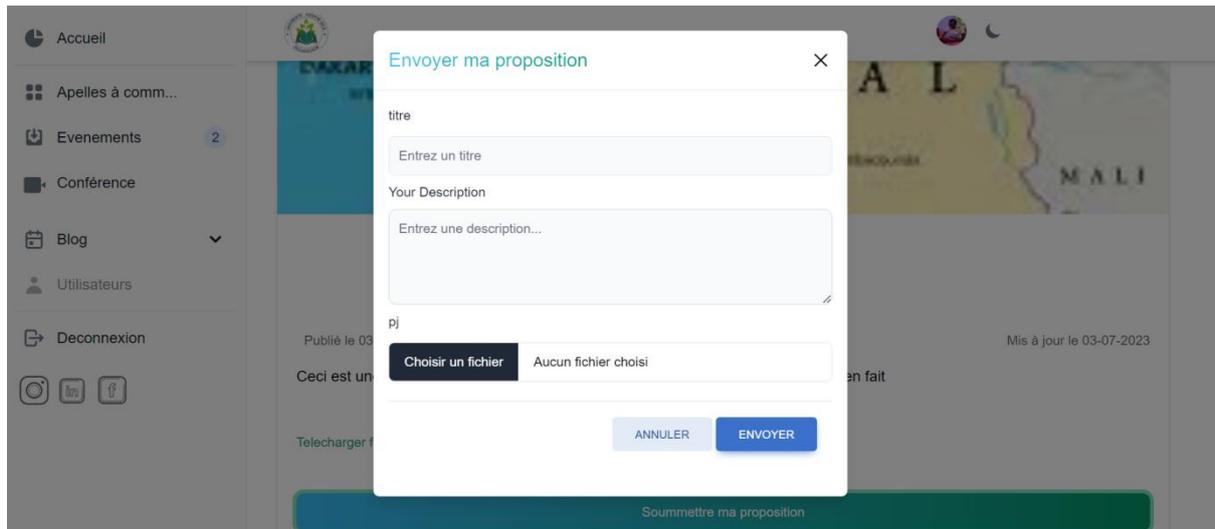


Figure 39 pages soumettre une proposition de communication

- Publication d'article

La partie blog de notre application permet aux étudiants et chercheurs de partager des articles entre eux. Dans cette interface, nous avons intégré un formulaire avec un éditeur de texte afin de faciliter la rédaction et la publication d'un article.

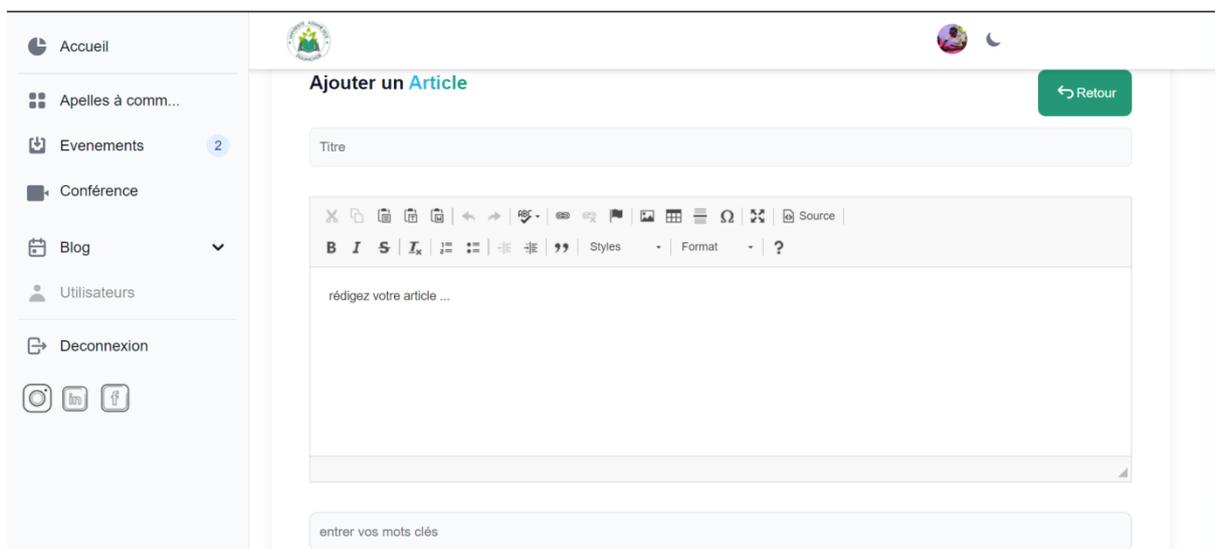


Figure 40 pages publication d'article

- Articles publiés

Chaque utilisateur peut voir la liste des articles qu'il à partager, les modifier ou les supprimer.



Figure 41 pages articles publié

- Tous les articles

Sur cette page, nous pouvons voir tous les articles qui ont été publiés. Les articles sont représentés par des cartes qui affichent la date de publication, le titre, la description et les mots clés. Nous avons également la possibilité de cliquer sur un article pour lire l'intégralité du contenu.

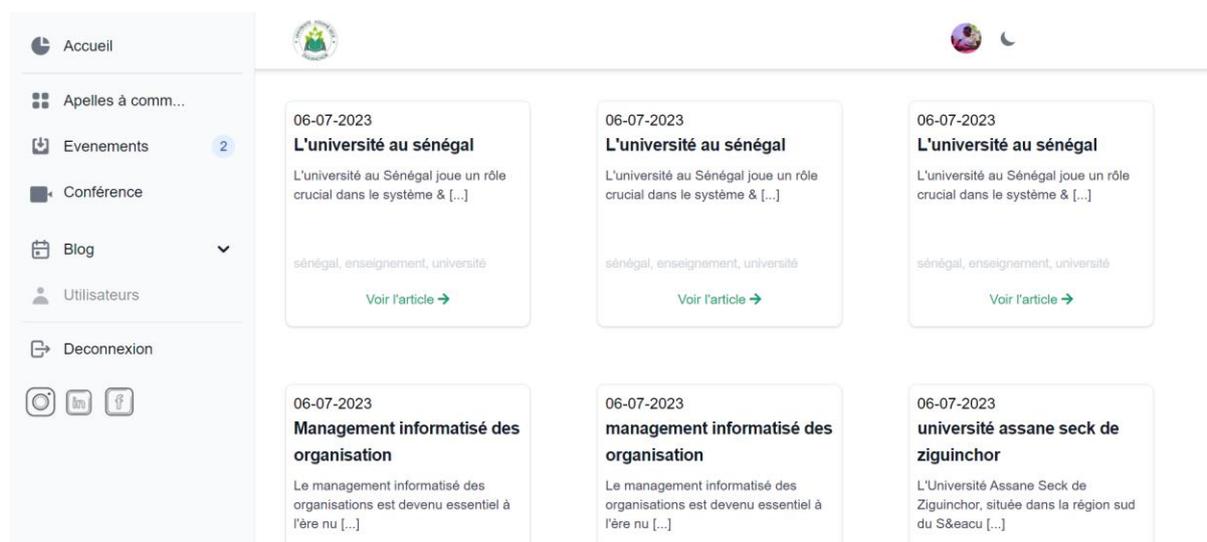


Figure 42 pages tous les articles

- Articles



Figure 43 pages articles

2. Limites de l'application

Bien que notre application de gestion de conférences avec diffusion en ligne présente de nombreux avantages, surtout dans le cas d'utilisation pour les écoles doctorales, il est important de reconnaître quelques limites potentielles.

Étant clairement conçue pour gérer un grand nombre d'utilisateurs simultanés, des limitations matérielles et de bande passante peuvent influencer la capacité réelle. Des tests approfondis seront nécessaires pour déterminer la charge maximale supportée par notre solution. Ces tests approfondis sont également valables pour le cas d'utilisation de WebRTC face à certains navigateurs en ce qui concerne la compatibilité.

Comme le dit l'adage, aucun système n'est infaillible. Cela dit, même si nous avons pris les mesures de sécurité nécessaires pour notre application, des mises à jour régulières et des renforcements de sécurité seront obligatoires.

IV. Conclusion

Dans ce chapitre, nous avons examiné le processus de développement de notre application, ainsi que les différentes technologies et outils utilisés pour sa réalisation. Nous avons également présenté le résultat final en expliquant les différentes interfaces de l'application.

Une fois la conception et la réalisation de notre solution terminées, nous passons maintenant aux tests et au déploiement de celle-ci.

Chapitre 5 : Tests et déploiement de l'application

I. Introduction

Ce chapitre aborde deux aspects essentiels du développement de notre application de gestion de conférence : les tests et le déploiement. Nous commencerons par souligner l'importance des tests dans le processus de développement, en mettant en avant l'utilisation de Laravel. Ensuite, nous aborderons le déploiement de notre application, en explorant les différentes stratégies et les bonnes pratiques pour rendre notre application accessible aux utilisateurs finaux.

II. Description des tests effectués

Les tests sont essentiels pour maximiser la qualité et la fiabilité de nos applications. Ils constituent un aspect crucial du développement logiciel moderne, garantissant que votre code fonctionne comme prévu et réduisant ainsi les bugs et les erreurs potentielles. Laravel offre des fonctionnalités intégrées qui facilitent l'écriture et l'exécution de tests, qu'ils soient unitaires ou fonctionnels.

Les tests unitaires se concentrent sur la vérification du bon fonctionnement des parties individuelles de votre code. Ils permettent de détecter rapidement les erreurs et les incohérences. Laravel simplifie et rend plus efficace l'écriture des tests unitaires grâce à ses fonctionnalités intégrées. Vous pouvez facilement créer des scénarios de test et les exécuter automatiquement.

Quant aux tests fonctionnels, ils simulent le comportement réel de votre application en émulant les interactions de l'utilisateur avec l'interface utilisateur. Ces tests de bout en bout sont particulièrement utiles pour détecter les problèmes d'intégration, de navigation ou de flux de données.

Dans le cas de notre application, nous avons plusieurs fonctionnalités à tester. Ainsi, nous allons présenter quelques tests et leurs résultats.

1. Teste de base dans Laravel

Laravel est conçu en pensant aux tests. En fait, la prise en charge des tests avec PHPUnit est incluse dès le départ et un phpunit.xml fichier est déjà configuré pour votre application. Le Framework est également livré avec des méthodes d'assistance pratiques qui vous permettent de tester de manière expressive vos applications [25].

Dans la structure de notre application, nous avons deux dossiers, "Unit" et "Feature", situés à la racine. Le dossier Unit est destiné aux tests unitaires, tandis que le dossier Feature est réservé aux tests de fonctionnalités. Lorsque nous lançons la commande "PHP artisan serve" initialement, nous obtenons deux réponses pour chaque type de test, provenant des classes de test d'exemple fournies par Laravel. L'image ci-dessous présente les résultats de nos premiers tests effectués pour notre application.

```
PS C:\Users\BELLA DIALLO\Desktop\Libasse\DEV\Uasz-doctorale-scho
ol-event> php artisan test

 PASS Tests\Unit\ExampleTest
 ✓ that true is true 0.01s

 PASS Tests\Feature\ExampleTest
 ✓ the application returns a successful response 0.35s

Tests: 2 passed (2 assertions)
Duration: 0.60s

PS C:\Users\BELLA DIALLO\Desktop\Libasse\DEV\Uasz-doctorale-scho
ol-event> █
```

Figure 44 résultat premier test

2. Test de publication d'un appel à communication par l'administrateur

Grâce à la commande **artisan PHP artisan make :test AppelleTest**, nous avons créé notre classe de test fonctionnel héritant de la classe **TestCase**. Le premier test effectué dans cette classe consiste à simuler la publication d'un appel à communication par un administrateur. Ce test nous a permis de vérifier si la logique de publication d'un appel dans notre application fonctionne correctement. Vous pouvez voir le résultat du test dans l'image ci-dessous.

```
e-school-event> php artisan test

PASS Tests\Unit\ExampleTest
✓ that true is true 0.01s

PASS Tests\Feature\AppelleTest
✓ admin pulich appelle 1.75s

PASS Tests\Feature\ExampleTest
✓ the application returns a successful respons... 0.06s

Tests: 3 passed (3 assertions)
Duration: 2.07s
```

Figure 45 résultat test publication appelle

3. Test de l'envoi d'une proposition de communication

Le test de l'envoi d'une proposition de communication vise à vérifier que les utilisateurs sont en mesure de soumettre avec succès une proposition de communication dans l'application. Il s'agit d'un test fonctionnel qui simule le comportement d'un utilisateur réel.

```
-doctorale-school-event> php artisan test

PASS Tests\Unit\ExampleTest
✓ that true is true 0.01s

PASS Tests\Feature\AppelleTest
✓ admin pulich appelle 1.86s
✓ envoie proposition communication 0.29s

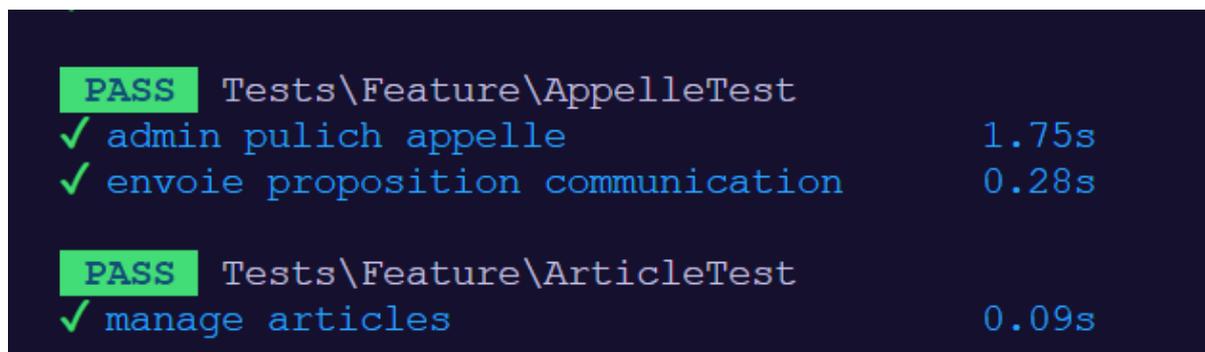
PASS Tests\Feature\ExampleTest
✓ the application returns a successful... 0.09s

Tests: 4 passed (4 assertions)
Duration: 2.54s
```

Figure 46 résultat test envoie d'une proposition de communication

4. Test gestion d'article

Le test fonctionnel pour la gestion des articles (ajout, lecture, suppression, mise à jour) nous a permis de garantir que les utilisateurs peuvent effectuer les actions essentielles pour la gestion des articles. En simulant le comportement d'un utilisateur réel, ce test permet de vérifier l'interaction fluide entre les différentes fonctionnalités de l'application. En ajoutant un nouvel article, en le mettant à jour et en le supprimant, ce test valide la capacité de l'application à gérer les opérations de base liées aux articles.



```
PASS Tests\Feature\AppelleTest
✓ admin pulich appelle 1.75s
✓ envoie proposition communication 0.28s

PASS Tests\Feature\ArticleTest
✓ manage articles 0.09s
```

Figure 47 résultat test gestion d'articles

III. Déploiement de l'application

La phase de déploiement marque en quelque sorte la dernière étape du développement de notre application, elle consiste à le mettre en ligne. En effet, le déploiement d'une application fait référence au processus de mise en production d'un logiciel afin de le rendre accessible et utilisable par les utilisateurs finaux. Cela comprend la configuration et l'installation de l'application sur des serveurs ou des environnements appropriés, la mise en place de l'infrastructure nécessaire, la gestion des dépendances, des bases de données et des ressources associées, ainsi que la vérification de la disponibilité et de la stabilité de l'application.

Nous avons plusieurs options pour déployer notre application, ainsi que différentes méthodes et outils de déploiement. En général, nous pouvons distinguer deux types de déploiement : le déploiement manuel via SFTP et le déploiement automatisé.

1. Le déploiement manuel

Cette méthode de déploiement implique le transfert de notre projet vers un serveur en ligne via FTP (File Transfer Protocol) afin de le rendre accessible aux utilisateurs. Elle est couramment utilisée pour le déploiement d'applications basées sur une architecture monolithique ou de sites web utilisant WordPress, parmi d'autres. Pour réaliser le transfert de

fichiers via FTP, nous aurons besoin d'un serveur web pour héberger notre application, ainsi que d'un client FTP tel que Filezilla, WinSCP, etc., pour se connecter au serveur distant en utilisant les informations de connexion fournies par ce dernier. Pour l'hébergement de notre application, nous avons plusieurs options disponibles qui nous permettent de choisir un serveur correspondant aux besoins spécifiques de notre application.

a. Serveurs d'hébergement

Comme mentionné précédemment, Pour héberger notre application, nous avons besoin d'un serveur approprié. Il existe plusieurs options pour choisir un serveur d'hébergement en fonction des besoins spécifiques de notre application.

- Hébergement mutualisé

Dans cette option, notre application est hébergée sur un serveur partagé avec d'autres sites web. Cela peut être une solution économique, mais cela peut entraîner des limitations en termes de ressources et de performances.

- Serveur dédié

Dans cette option, nous optons pour la location d'un serveur physique dédié à notre application. Cette solution offre une flexibilité accrue et des performances améliorées, car toutes les ressources du serveur sont exclusivement dédiées à notre application. Toutefois, cela requiert une expertise en gestion de serveur.

- Hébergement cloud

L'hébergement cloud, tel que les services proposés par des fournisseurs tels que Amazon Web Services (AWS), Microsoft Azure, ou Google Cloud Platform (GCP), permet de déployer notre application sur des serveurs virtuels dans le cloud. Cela offre une grande évolutivité, une flexibilité et une facilité de gestion.

b. Client FTP Filezilla

Filezilla est un logiciel de transfert de fichiers très utilisé par les développeurs. Il offre une interface conviviale et intuitive, permettant de se connecter facilement à des serveurs distants via les protocoles FTP, SFTP et FTPS. Avec ses fonctionnalités avancées telles que le glisser-déposer, la gestion des files d'attente et la possibilité de reprendre les transferts interrompus, Filezilla facilite le transfert de fichiers de manière efficace et fiable. Il est également open source, ce qui permet une personnalisation et une contribution communautaire [26].

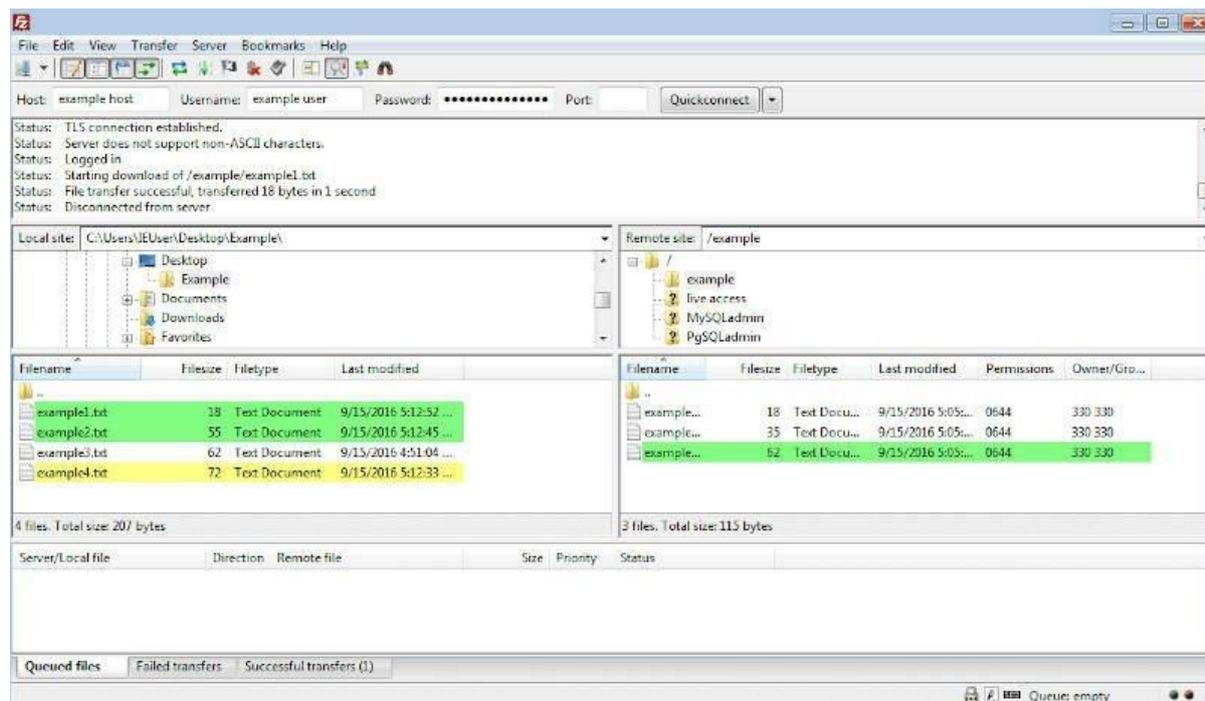


Figure 48 Filezilla clients

2. Déploiement automatisé

Ce type de déploiement est généralement effectué à l'aide d'un dépôt git et les webhooks. Elle consiste à automatiser les tâches de déploiement en faisant communiquer notre dépôt central et notre serveur de déploiement. Dans ce type de déploiement, on n'aura pas besoin d'envoyer les fichiers vers notre serveur de déploiement mais c'est au serveur de chercher les fichiers et les dernières modifications dans notre dépôt centralisé. Ici, le dépôt centralisé n'est rien d'autre qu'un système de contrôle de version qui sert à gérer le code source lors du développement de nos applications, pour ce faire, les outils les plus utilisés sont GitHub et gitlab.

Ce type de déploiement semble être la plus simple mais nécessite une certaine connaissance technique pour le mettre en place.

IV. Conclusion

En conclusion, ce chapitre a mis en lumière l'importance cruciale des tests et du déploiement dans le processus de développement de notre application de gestion de conférence. Nous avons aussi établi une étude sur les différents types de déploiement possible pour notre application, ainsi que la présentation des tests effectués. Dans le cadre de notre projet, avons

utilisé le déploiement manuel dans un premier temps pour l'évaluation et le test de notre première version.

Conclusion générale

Ce mémoire porte sur la conception et la réalisation d'un système centralisé pour la gestion et la diffusion de conférences et d'événements organisés par les écoles doctorales de l'Université Assane Seck de Ziguinchor. En effet, le processus de gestion de ces activités a toujours été complexe et n'a pas été jusqu'à présent centralisé dans un système d'information tout-en-un. Ceci dit, hormis les problèmes liés à l'organisation, la publication d'appels à communication et des soumissions de propositions de communication, nous avons aussi noté une non-traçabilité des données liées à chaque activité organisée.

Ainsi, face à ces défis, nous avons proposé la mise en place d'une solution informatique qui va tout d'abord faciliter le partage et la collecte d'informations liées à un appel à communication, ainsi qu'aux autres types d'événements, mais aussi offrir la possibilité d'avoir un recueil d'articles scientifiques rédigés par les chercheurs. Notre application offre aussi un système de diffusion en ligne qui permet à ceux qui veulent participer aux différentes activités à distance de le faire à distance.

Ainsi, la réalisation de cette application nous a tout d'abord permis de mettre en pratique l'ensemble des compétences que nous avons acquises durant tout notre cursus universitaire, y compris celles acquises dans nos différents stages effectués. Comme difficultés, nous avons eu des contraintes de temps, d'environnements de travail propice pour le développement de notre solution. Cependant, la réalisation de ce travail fut pour nous une tâche passionnante nous avons découvert et appris de nouvelles technologies lors de la réalisation, à savoir les Websockets et WebRTC pour la diffusion en ligne, ainsi que l'outil de gestion de projet que nous avons utilisé pour piloter le processus de développement de notre solution.

En termes de travail, nous avons mis en place une première version de notre application et avons noté quelques aspects qui pourraient être mis en place pour une éventuelle mise à jour. De ce fait, la prochaine étape consistera à la collecte des retours d'utilisateurs, les observations et les résultats des premières utilisations de l'application. Ces informations nous serviront de base pour planifier et mettre en œuvre des mises à jour futures visant à optimiser les fonctionnalités existantes et à intégrer de nouvelles caractéristiques qui renforcent davantage la valeur de notre solution.

Bibliographie et webographie

- [1] B. Neuhofer, D. Buhalis, et A. Ladkin, « Technology as a Catalyst of Change: Enablers and Barriers of the Tourist Experience and Their Consequences », in *Information and Communication Technologies in Tourism 2015*, I. Tussyadiah et A. Inversini, Éd., Cham: Springer International Publishing, 2015, p. 789- 802. doi: 10.1007/978-3-319-14343-9_57.
- [2] « EDAS: Editor's Assistant ». Consulté le: 16 décembre 2023. [En ligne]. Disponible sur: <https://edas.info/doc/>
- [3] « EDAS: Editor's Assistant: Features ». Consulté le: 18 décembre 2023. [En ligne]. Disponible sur: <https://edas.info/doc/features.html>
- [4] « EasyChair Home Page ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://easychair.org/>
- [5] « Conference Management Software | COMS ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://conference-service.com/>
- [6] OpenConf, « OpenConf Peer-Review, Conference and Abstract Management Software System ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.OpenConf.com/>
- [7] « WikiCFP : Appel à communications de conférences, ateliers et revues ». Consulté le: 8 janvier 2024. [En ligne]. Disponible sur: <http://www.wikicfp.com/cfp/>
- [8] « Project Planning & Management Guides | TeamGantt ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.teamgantt.com/guides>
- [9] « Home - Asana ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://app.asana.com/0/home/1206049597703717>
- [10] « Manage Your Team's Projects From Anywhere | Trello ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://trello.com/home>
- [11] « Documentation », Notion. Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.notion.so/fr-fr/help/guides/category/documentation>
- [12] « Analysez les besoins de votre client », OpenClassrooms. Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://openclassrooms.com/fr/courses/4296701-gerez-un-projet-digital-avec-une-methodologie-en-cascade/4296708-analysez-les-besoins-de-votre-client>

- [13] « Cycle en V en gestion de projet : définition et méthode ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.manager-go.com/gestion-de-projet/cycle-en-v.htm>
- [14] « Manifeste pour le développement Agile de logiciels ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://agilemanifesto.org/iso/fr/manifesto.html>
- [15] Atlassian, « Qu'est-ce que Scrum ? [+ Comment se lancer] », Atlassian. Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.atlassian.com/fr/agile/scrum>
- [16] « La méthode Agile Kaban : explication et comment l'utiliser ? - LBdD ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.leblogdudirigeant.com/methode-agile-kanban/>
- [17] « Quelle architecture logicielle pour son application ? - SoftFluent ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.softfluent.fr/blog/architecture-logicielle-pour-application/>
- [18] « PHP: Hypertext Preprocessor ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.php.net/>
- [19] « Le développement avec Java : le tutorial développons en Java et développons en Java avec Eclipse ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: https://www.jmdoudoux.fr/accueil_java.htm
- [20] « Welcome to Python.org », Python.org. Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.python.org/about/>
- [21] « Laravel - The PHP Framework For Web Artisans ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://laravel.com/>
- [22] « Premiers pas avec WebRTC ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://webrtc.org/getting-started/overview?hl=f>
- [23] « What is WebRTC Signaling? » Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.onsip.com/voip-resources/voip-fundamentals/webrtc-signaling>
- [24] « Introduction to Metered Video | Metered Video Docs ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://www.metered.ca/docs/>
- [25] « Testing: Getting Started - Laravel 10.x - The PHP Framework For Web Artisans ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://laravel.com/docs/10.x/testing>
- [26] « Documentation - FileZilla Wiki ». Consulté le: 17 décembre 2023. [En ligne]. Disponible sur: <https://wiki.filezilla-project.org/Documentation>

Table des matières

Résumé	i
Abstract	ii
Dédicace	iii
Remerciements	iv
SOMMAIRE	v
Liste des tableaux.....	viii
Liste des images et figures	ix
Liste des sigles et abréviations	xi
Introduction générale	1
1. Contexte	2
2. Problématique	2
3. Objectifs	3
4. Présentation du plan	3
Chapitre 1 : Etat de l'art et présentation de notre proposition	5
I. Introduction	6
II. Concepts fondamentaux	6
1. Notion de la gestion d'évènements	6
2. Evolution des méthodes de gestion d'évènements	6
III. Étude des applications de gestion d'évènements	7
1. EDAS	7
a. Fonctionnalités	7
b. Avantages	8
a. Limites	8
2. EasyChair	8
b. Fonctionnalités	8
c. Avantages	9
d. Limites	9
3. COMS	9
a. Fonctionnalités	9
b. Avantages	10
c. Limites	10

4. OpenConf	10
a. Fonctionnalités	10
b. Avantages	11
c. Limites	11
5. WikiCfp	11
a. Fonctionnalité	11
b. Avantages	12
c. Limites	12
6. Tableau récapitulatif	12
IV. Présentation de notre solution	13
V. Analyse des technologies utilisées dans la conception et le développement de l'application	14
1. Outils de gestion de projet informatique	15
a. TeamGant	15
b. Asana	16
c. Trello	17
d. Notion	18
2. Technologies pour le développement backend	19
3. Technologies pour le développement frontend	19
VI. Conclusion	20
Chapitre 2 : Méthodologie et choix technologiques	21
I. Introduction	22
II. Description de la méthodologie utilisée	22
1. La méthodologie traditionnelle (cascade)	22
a. Les phases de la méthodologie de gestion de projet en cascade	23
b. Cas d'utilisation de la méthodologie en cascade	24
c. Avantages et inconvénients de la méthodologie en cascade	24
2. La méthodologie Agile	25
a. Les valeurs de la méthodologie agile	25
b. Les principes de la méthodologie agile	25
c. Les différentes méthodes agiles	26
3. Méthodologie choisie	28
III. Choix technologique	29
1. Architecture	29
a. Architecture micro services	29
b. Architecture SOA	30

c. Architecture monolithique	31
2. Backend	32
a. PHP	32
b. Java	33
c. Python.....	34
3. Frontend	34
4. Choix	35
IV. Conclusion	36
Chapitre 3 : Analyse et conception de l'application.....	37
I. Introduction	38
II. Cahier des charges	38
1. Objectifs	38
III. Analyse des besoins et des fonctionnalités.....	39
1. Besoins fonctionnels	39
2. Besoins non fonctionnels.....	40
IV. Conception.....	41
1. Diagramme de cas d'utilisation.....	41
2. Diagramme de séquence	42
3. Diagramme des classes	45
V. Conclusion.....	47
Chapitre 4 : Développement et présentation de l'application	48
I. Introduction	49
II. Développement de l'application	49
1. Développement avec Laravel	49
2. Système d'authentification avec Laravel Fortify.....	50
3. Les Policy dans Laravel.....	51
4. Moteur de templating blade.....	52
5. Webrtc.....	52
a. Serveur de signalisation WebRTC (WebRTC Signaling Server).....	53
b. Serveur STUN/TURN	54
c. Serveur WebRTC Metered Video	55
III. Présentation de l'application.....	56
1. Présentation des interfaces	56
2. Limites de l'application	65
IV. Conclusion	65
Chapitre 5 : Tests et déploiement de l'application.....	67

I. Introduction	68
II. Description des tests effectués	68
1. Teste de base dans Laravel	68
2. Test de publication d'un appel à communication par l'administrateur	69
3. Test de l'envoi d'une proposition de communication	70
4. Test gestion d'article	71
III. Déploiement de l'application	71
1. Le déploiement manuel	71
a. Serveurs d'hébergement	72
b. Client FTP Filezilla	72
2. Déploiement automatisé	73
IV. Conclusion	73
Conclusion générale	75
Bibliographie et webographie	a
Table des matières	d