

# **DOSSIER PROJET**

**PROJET : Exp@tmedia**  
FRANCE

**Rédacteur : KABA SIDIKI**

**Date : 25/10/2023**

## **TABLE DES MATIÈRES**

### **LISTE DES COMPÉTENCES DU RÉFÉRENTIEL COUVERTES PAR LE PROJET**

#### **Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité.**

- Maquetter une application
- Développer une interface utilisateur desktop
- Développer des composants d'accès aux données
- Développer la partie front-end d'une interface utilisateur web
- Développer la partie back-end d'une interface utilisateur web

#### **Concevoir et développer la persistance des données en intégrant les recommandations de sécurité.**

- Concevoir une base de données
- Mettre en place une base de données
- Développer des composants dans le langage d'une base de données

#### **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité.**

Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

- Concevoir une application
- Développer des composants métier
- Construire une application organisée en couches
- Développer une application mobile
- Préparer et exécuter les plans de tests d'une application
- Préparer et exécuter le déploiement d'une application

### **RÉSUMÉS DU PROJET**

- Résumé en français
- Résumé en anglais

# **CAHIER DES CHARGES**

## **Présentation d'ensemble du projet**

- Présentation de l'entreprise / du client
- Description de l'existant
- Objectifs du projet
- Intervenants sur le projet
- Cible adressée par le projet
- Principes de référencement
- Exigences de performances et de volumétrie
- Multilinguisme & adaptations pour un public spécifique
- Reprise de l'existant

## **Description graphique et ergonomique**

- Composants de la charte graphique.
- Design, Responsive design et autres exigences liées au design
- Maquettes

## **Besoins fonctionnels « métier »**

- Utilisateurs du projet
- Processus utilisateur impacté
- Informations relatives aux contenus
- Inventaire des besoins fonctionnels

## **Prestations attendues**

- Livrables et prestations
- Budget
- Planning et principaux jalons
- Méthodologie de suivi projet

## **GESTION DE PROJET**

### **Méthodologie**

### **Outil**

### **Planning et suivi**

### **Environnement technique et humain**

# **SPÉCIFICATIONS FONCTIONNELLES**

## **Description de la solution**

- Caractéristiques de la solution
- Historique de la solution
- Acteurs du projet
- Processus utilisateurs impactés
- Applications connexes
- Documentation référence
- Lexique/glossaire
- Aspect métier
  - Criticité de l'application
  - Cartographie (Arborescence)

## **Généralités fonctionnelles**

### **Utilisateurs**

### **Fonctionnalités**

- Fonctionnalité n°1 : Connexion au front-office
- Fonctionnalité n°2 : Créer et enregistrer un évènement (Front-office)
- Fonctionnalité n°3 : Voir la liste des évènements créés (Front-office)
- Fonctionnalité n°4 : Voir la liste des évènements non créés (Front-office)
- Fonctionnalité n°5 : Voir son calendrier d'évènements (Front-office)
- Fonctionnalité n°6 : Rechercher un évènement (Front-office)
- Fonctionnalité n°7 : Voir le contenu d'un évènement (Front-office)
- Fonctionnalité n°8 : Participer à un évènement (Front-office)
- Fonctionnalité n°9 : S'intéresser à un évènement (Front-office)
- Fonctionnalité n°10 : Signaler un évènement (Front-office)
- Fonctionnalité n°11 : Modifier un évènement (Front-office)
- Fonctionnalité n°12 : Supprimer un évènement (Front-office)
- Fonctionnalité n°13 : Connexion au back-office
- Fonctionnalité n°14: Voir la liste de tous les évènements non signalés (Back-office)
- Fonctionnalité n°15: Voir la liste de tous les évènements signalés (Back-office)
- Fonctionnalité n°16: Rechercher un évènement (Back-office)
- Fonctionnalité n°17: Voir le contenu d'un évènement (Back-office)
- Fonctionnalité n°18: Masquer le contenu d'un évènement (Back-office)
- Fonctionnalité n°19: Modifier un évènement (Back-office)
- Fonctionnalité n°20 : Supprimer un évènement (Back-office)

### **Règle métier**

### **Autorisations**

**Envoi de mail**  
**Impression**  
**Aide en ligne**  
**Gestion des langues**  
**Workflow**  
**Accessibilité**  
**Traitements planifiés**

## **Diagramme d'activité**

### **Fonctionnalités détaillées**

Fonctionnalité n° 1 : Connexion au front-office

Description

Données

Écran/Affichage

Fonctionnalité n° 2 : Créer et enregistrer un évènement (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 3 : Voir la liste des évènements créés (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 4 : Voir la liste des évènements non créés (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 5 : Voir son calendrier d'évènements (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 6 : Rechercher un évènement (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 7 : Voir le contenu d'un évènement (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 8 : Participer à un évènement (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 9 : S'intéresser à un évènement (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 10 : Signaler un évènement (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 11 : Modifier un évènement (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 12 : Supprimer un évènement (front-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 13 : Connexion au back-office

Description

Données

Écran/Affichage

Fonctionnalité n° 14 : Voir la liste de tous les évènements non signalés (back-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 15 : Voir la liste de tous les évènements signalés (back-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 16 : Rechercher un évènement (back-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 17 : Voir le contenu d'un évènement (back-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 18 : Masquer le contenu d'un évènement (back-office)

Description

Données

Écran/Affichage

Fonctionnalité n° 19 : Modifier un évènement (back-office)

Description

Données

Écran/Affichage

Règles métier

Fonctionnalité n° 20 : Supprimer un évènement (back-office)

Description

Données

Écran/Affichage

## **SPÉCIFICATIONS TECHNIQUES**

**Référencement**

**Performances et Volumétrie**

**Navigation**

**Mockup**

**Accessibilité**

**Services tiers**

**Sécurité**

**Gestion du contenu, des données**

Modèle Conceptuel de Données

Description de la base de données

Primary keys & foreign keys

Index

## **RÉALISATIONS**

### **Exemple 1 : Import des magasins et création des événements associés**

Extrait(s) de code

Argumentation

### **Exemple 2 : Exemple de requêtes du modèle “Event”**

Extrait(s) de code

Argumentation

### **Exemple 3 : méthode handle() de la commande sendReminderEventEmail**

Extrait(s) de code<sup>65</sup>

Argumentation<sup>65</sup>

## **PRÉSENTATION DU JEU D’ESSAI DE LA FONCTIONNALITÉ LA PLUS REPRÉSENTATIVE**

**Fonctionnalité testée**

**Description et résultats des tests**

**Résultats des tests**

**Conclusion**

## **DESCRIPTION DE LA VEILLE SUR LES VULNÉRABILISÉS DE SÉCURITÉ**

## **DESCRIPTION D’UNE SITUATION DE TRAVAIL AYANT NÉCESSITÉ UNE RECHERCHE SUR SITE ANGLOPHONE ou FRANCOPHONE**

**Description du besoin**

**Description de la recherche**

Moteur de recherche

Mots clés

Résultat de la recherche (liste Sites)

Critères de sélection et choix du(es) site(s)

Extrait du(es) site(s)

Traduction de l’extrait

**Exploitation du résultat**

Extrait(s) de code

Argumentation



# LISTE DES COMPÉTENCES DU RÉFÉRENTIEL COUVERTES PAR LE PROJET

## Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité.

### Maquetter une application

En utilisant le cahier des charges, j'ai réalisé la première version des maquettes à l'aide du logiciel Dia, Ce qui m'a permis de faire des modifications rapides selon les remarques des autres membres de l'association. La méthode agile m'a permis de faire évoluer le design des applications pour avoir le rendu que j'ai aujourd'hui.

### Développer une interface utilisateur desktop

Pour la réalisation de cette application, nous avons développer deux interfaces utilisateur de type desktop : celle du front-office destinée aux grand public et celle du back-office pour l'administrateur.

### Développer des composants d'accès aux données

Contrairement à un **ORM** (Object-Relational Mapping), qui est utilisé pour les bases de données relationnelles, **Mongoose** est une **bibliothèque pour Node.js** spécifiquement conçu pour simplifier l'interaction avec des bases de données **NoSQL** comme **MongoDB** en fournissant un **ODM**.

Mongoose vous permet de définir des **schémas** pour vos documents MongoDB et de créer des **modèles** (ou **classes**) basés sur ces schémas. Vous pouvez ensuite utiliser ces modèles pour **gérer les opérations CRUD** (create, read, update, delete) sur la base de données MongoDB.

### Développer la partie front-end d'une interface utilisateur web

Après avoir réalisé les maquettes, j'ai commencé le développement des interfaces **front-end** de l'application, le front-office et back-office, tout deux ont été développées à l'aide des bibliothèques **React.js** permettant de créer des interfaces utilisateur interactives et **Redux** pour gérer l'état de l'application de manière prévisible et centralisée. Le style a été conçu à l'aide de **css** et **scss**.

J'ai aussi utilisé **React-Bootstrap** et **Material-UI** pour accélérer le développement des interfaces utilisateurs.

### Développer la partie back-end d'une interface utilisateur web

La partie **back-end** de l'application a été développée à l'aide du framework **Express.js** (**Node.js**) avec une architecture MVC (**Model View Controller**), la base de données étant géré avec **MongoDB** (langage **NoSQL**).

## Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

### Concevoir une base de données

Après avoir collaborer étroitement avec les parties prenantes du projet pour identifiez les besoins de l'application et les objectifs de la base de données, La base de données a ici été conçue à partir de la méthode Merise élaboration d'un MCD (**modèle conceptuel de données**), puis d'un MLD (**modèle logique de données**).

### Mettre en place une base de données

La base de données mise en place est en **NoSQL** et gérée via **MongoDB Compass**.

### Développer des composants dans le langage d'une base de données

Pour toutes manipulations relatives à la base de données, j'ai utilisé l'ODM **Mongoose**, qui est une bibliothèque pour Node.js qui facilite l'utilisation de MongoDB.

## Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

### Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

Pour la réalisation de ce projet, j'ai pu collaborer avec les autres membres de l'association via une des méthodes « **agile** » : la méthode **scrum**.

Durant le scrum, on n'essaie de produire tous ensemble le **Backlog** c'est à dire **la liste des tâches, nouvelles fonctionnalités** qui doivent être priorisées et notées selon leurs difficultés. Ensuite, nous avons la période du **sprint** qui dépend des besoins mais généralement nous avons un interval d'une semaine.

Durant chaque Sprint, chaque membre de l'équipe partage leur retour d'expérience et discute des améliorations à effectuer au prochain sprint.

### Concevoir une application

Afin de concevoir l'application, j'ai réalisé différents diagrammes **UML** (*diagramme de classe, de cas d'utilisation, d'activité, de séquence*) ainsi qu'un **modèle conception** de données , un **modèle logique** de données puis un **modèle physique** de données.

## Développer des composants métier

Afin de développer les différents composants métier de l'application, j'ai utilisé l'**architecture MVC (Modèle - Vue – Contrôleur)** lors du développement de mon application qui utilise principalement **Node.js** pour le **back-end** et **React.js** pour le **front-end** .

## Construire une application organisée en couches

Pour construire l'application, j'ai utilisé l'architecture **MVC** (Modèle - Vue – Contrôleur). Le **pattern MVC** permet de bien organiser son code source. Le but du **MVC** est de séparer la logique du code en trois parties. La partie **Back-end** met également à disposition **des routes API** utilisées par l'application **React.js du front-office et du back-office**.

## Développer une application mobile

Non concerné.

## Préparer et exécuter les plans de tests d'une application

J'ai réalisé des **tests unitaires** sur certaines des routes de l'API via **Postman** pour la partie Back-end de l'application. J'ai aussi effectué des **test unitaires** via **Jest** pour la partie Front-end.

## Préparer et exécuter le déploiement d'une application

Non concerné pour ce projet.

# RÉSUMÉS DU PROJET

## Résumé en français :

**ExpatSansFrontière**, est une association composée de **10 membres de pays différents** dont le but est de faciliter *l'intégration des expatriés (Travailleurs et Étudiants) en mobilité en France*.

Pour résoudre cette problématique, l'association a opté pour **la création d'un réseau social**, qui met l'accent sur le partage d'information destiné aux expatriés.

Cela afin de faciliter leur intégration dans leur nouveau pays d'accueil la France et les aider à interagir.

Les **utilisateurs** de ce site auront accès à plusieurs fonctionnalités selon leur rôle et leur statut. Parmi les fonctionnalités qui seront accessibles sur le site , nous avons entre autres :

***La publication et la gestion des informations / L'organisation et la gestion d'un évènement / La proposition et la gestion des offres d'emploi étudiant / Rédaction et gestion d'article / Déclaration et la gestion d'une urgence /Accès a un espace e-commerce***

De plus, les utilisateurs ont le libre arbitre de gérer leurs réseaux. Ils décident avec qui ils souhaitent interagir (***suivre , discuter, ou créer des groupes de discussions avec d'autre utilisateurs***) selon leur centres d'intérêts.

La partie Front destinée aux grand public et le back-office ont été développés à l'aide du framework **React**. Les données sont alimentées grâce à une API **REST** commune développé à l'aide de **NodeJs et de Express**.

Le système de gestion des données se fera avec **MongoDB** qui est une base de données **NoSQL orientée documents**.

## Résumé en Anglais:

**ExpatSansFrontière** is an association made up of **10 members from different countries** whose aim is to facilitate **the integration of expatriates** (*Workers and Students*) on mobility in France.

To resolve this problem, the association opted for **the creation of a social network**, which emphasizes the sharing of information intended for expatriates.

This is to facilitate their integration into their new host country France and help them interact.

**Users** of this site will have access to several features depending on their role and status. Among the features that will be accessible on the site, we have among others:

***Publication and management of information / Organization and management of an event / Proposal and management of student job offers / Writing and management of articles / Declaring and managing an emergency / Access to an e-commerce space***

Additionally, users have free will to manage their networks. They decide with whom they want to interact (***follow, chat, or create discussion groups with other users***) according to their interests.

The Front part intended for the general public and the back office were developed using the **React** framework. Data is fed through a common **REST API** developed using **NodeJs and Express**.

The data management system will be done with **MongoDB** which is a **document-oriented NoSQL** database.

# CAHIER DES CHARGES

## Présentation d'ensemble du projet

### Présentation de l'entreprise / du client

En tant que nouveau venu dans un pays, vous allez potentiellement avoir des habitudes et des comportements qui peuvent au début freiner votre immersion.

C'est normal, il est difficile de quitter le mode de vie et de pensée auquel on a été habitué. Mais il est fortement conseillé de le corriger dans le cas d'une installation à long terme dans le pays d'accueil .

C'est pour résoudre cette problématique liée aux expatriés, qu'est né l'association **ExpatsansFrontiere**, une organisation à but non lucratif formée par des **expatriés**, c'est-à-dire des personnes vivant à l'étranger, généralement dans un pays différent de leur pays d'origine.

Cette association a pour mission de créer une communauté solidaire et d'apporter un soutien essentiel aux expatriés qui vivent loin de leur pays d'origine.

Parmi les solutions proposées par cette association, nous avons la création d'un réseau social pour les expatriés appelé : **Exp@tmedia**

FRANCE

### Description de l'existant

Pas d'existant à la base du projet

### Objectif du projet

**ExpatsansFrontiere** propose la création d'une plateforme en ligne, a fonctionnalité multiple, qui vise à créer un sentiment de communauté et à fournir un soutien mutuel aux expatriés, en les aidant à s'adapter à leur nouvelle vie à l'étranger, qu'elles soient temporairement ou définitivement expatriées.

Ce projet sera constitué de deux parties :

- D'une part on aura, la partie front-office, qui sera accessible aux utilisateurs du site uniquement après que ceux-ci aient terminé leurs inscription sur le site.

- D'autre part on aura, la partie back-office, destinée à l'administrateur du site qui lui permet d'assurer le bon fonctionnement du site, la sécurité des données et la gestion des utilisateurs, des contenus et des fonctionnalités.

## Intervenants sur le projet

L'équipe est composée de 10 personnes uniquement que des étudiants dans les spécialités différentes. Du coup, chacun essaye d'apporter sa contribution dans la mesure du possible le temps que une organisation précise soit établie.

## Cible adressée par le projet

les principales catégories de personnes qui peuvent être les cibles de notre réseau social sont :

- **Expatriés** : Les expatriés (*actuels, futurs et anciens*) eux-mêmes sont la cible principale de notre projet
- **Amis et familles des expatriés** : Les proches des expatriés, tels que les amis, la famille, les conjoints et les partenaires, qui souhaitent rester en contact avec leurs proches vivant à l'étranger.
- **Touristes**: Les étrangers qui visitent la France et qui souhaitent entrer en contact avec des expatriés locaux pour obtenir des recommandations, des conseils de voyage, etc.
- **Entreprises et employeurs** : Les entreprises et les recruteurs qui cherchent à embaucher des talents internationaux et à se connecter avec des professionnels expatriés.
- **Communautés locales (Non expatriés)**: Les habitants du pays d'accueil qui souhaitent se connecter avec la communauté d'expatriés, que ce soit pour des échanges culturels, des opportunités linguistiques ou des amitiés interculturelles.
- **Autres associations** : Les organismes gouvernementaux et les ONG qui souhaitent fournir des informations et des services aux expatriés ou collaborer avec des expatriés pour des projets spécifiques. Ou encore,

## Principes de référencement

Actuellement , le projet n'est pas concerné par le référencement.

## Exigences de performance et de volumétrie

À l'issue de la première version de l'application, des tests de charge et de performance seront réalisés pour évaluer si le site est en mesure de gérer la volumétrie et l'activité attendues. En fonction des résultats de ces tests, des ajustements peuvent être nécessaires pour optimiser la performance.

## Multilinguisme & adaptation pour un public spécifique

Le projet ne concernant que les expatriés de la France, aucun système d'adaptation pour un public anglophone ou autre mais cela sera peut être disponible dans une version ultérieure du site.

## Reprise de l'existant

Pas d'existant à reprendre.

## Description graphique et ergonomique

### Composants de la charte graphique

Non concerné pour le moment

### Design, Responsive design et autres exigences liées au design

La partie front-office de l'application doit être responsive mobile, les dimensions sont indiquées ci-dessus.

Image à insérer



## Besoins fonctionnels « métier »

### Utilisateurs du projet

Les utilisateurs du projet sont :

- > Les personnes inscrites sur le site y compris les membres de l'association
- > L'administrateur du Site

### Processus utilisateur impacté

Le processus utilisateur impacte sur **l'expérience Utilisateur (Front-office)** et sur la **gestion et la maintenance du Site (Back-office)**.

### Informations relatives aux contenus

Les deux composantes, le front-office et le back-office, sont cruciales pour le fonctionnement d'un site de réseau social. Le **front-office** offre **une interface conviviale** aux utilisateurs, tandis que le **back-office** permet aux administrateurs de **gérer, surveiller et maintenir le site**.

### Inventaire des besoins fonctionnels

Les besoins fonctionnels d'une telle application sont nombreuses, raison pour laquelle dans ce tableau ci-dessous je vais uniquement présenter les besoins fonctionnels du système de gestion des événements.

Tableau des **besoins fonctionnels** (du système de gestion des événements) de **l'interface utilisateur** :

<b>Id</b>	<b>Thème</b>	<b>Qui</b>	<b>Quoi</b>	<b>Pourquoi</b>
<b>1</b>	<b>Inscription et création de profil</b>	Utilisateur	Création de comptes d'utilisateurs avec authentification sécurisée.	Accéder à l'application
<b>2</b>	<b>Création d'événements</b>	Organisateur	Les utilisateurs doivent pouvoir créer des événements, en fournissant des informations telles que le nom de l'événement, la date, l'heure, l'emplacement, la description, image, etc.	Organiser un événement

<b>3</b>	<b>liste des évènements</b>	Organisateur	Les utilisateurs doivent pouvoir consulter la liste des événements qu'ils ont créés et enregistrés	Voir la liste de ces événements
<b>4</b>	<b>liste des évènements</b>	Utilisateur	Les utilisateurs doivent pouvoir consulter la liste des événements organisés par un autre utilisateurs	Voir la liste des événements non créée
<b>5</b>	<b>liste des évènements</b>	Utilisateur	Les utilisateurs doivent voir les événements à venir et passés	Voir calendrier événement
<b>6</b>	<b>Filtrer les évènements</b>	Utilisateur	Les utilisateurs doivent pouvoir rechercher un événement spécifique en fournissant le titre de l'évènement et le nom de l'organisateur	Rechercher un événement
<b>7</b>	<b>Consulter un évènement</b>	Utilisateur	L'organisateur et le participant doivent pouvoir consulter un événement. Par contre, seul l'organisateur a accès à la liste des participants et des intéressés pendant la consultation.	Voir le contenu d'un événement créé
<b>8</b>	<b>Participation à un évènement</b>	Participant	Le participant doit pouvoir assister à un événement organisé par un autre utilisateur.	Participer à un événement
<b>9</b>	<b>Être intéressé à un évènement</b>	Participant	Le participant doit pouvoir signifier si l'évènement l'intéresse ou pas.	S'intéresser à un événement

<b>10</b>	<b>Signalement d'évènement</b>	Participant	Le participant doit pouvoir signaler l'abus, le spam ou le contenu inapproprié.	Signaler un évènement
<b>11</b>	<b>Modification un évènement</b>	Organisateur	L'organisateur doit pouvoir modifier son évènement	Modifier un évènement
<b>12</b>	<b>Supprimer un évènement</b>	Organisateur	L'organisateur doit pouvoir supprimer son évènement	Supprimer un évènement

Tableau des **besoins fonctionnels** de l'**interface d'administration** :

<b>Id</b>	<b>Thème</b>	<b>Qui</b>	<b>Quoi</b>	<b>Pourquoi</b>
<b>1</b>	<b>Authentification</b>	Administrateur	S'authentifier	Accéder au back-office
<b>2</b>	<b>Liste des évènements</b>	Administrateur	L'administrateur doit pouvoir consulter la liste des événements non signalés	Voir la liste de tous les évènements non signalés
<b>3</b>	<b>Liste des évènements</b>	Administrateur	L'administrateur doit pouvoir consulter la liste des événements signalés	Voir la liste de tous les évènements signalés
<b>4</b>	<b>Filtrer la liste des évènements</b>	Administrateur	Il doit pouvoir rechercher un événement spécifique via son id	Rechercher un évènement
<b>5</b>	<b>Consulter un évènement</b>	Administrateur	L'administrateur doit pouvoir consulter un événement au même titre que l'organisateur.	Voir le contenu d'un évènement
<b>6</b>	<b>Modération de contenu</b>	Administrateur	L'administrateur doit pouvoir consulter un	Masquer le contenu d'un évènement

			évènement au même titre que l'organisateur.	
7	<b>Modification d'un évènement</b>	Administrateur	L'administrateur doit pouvoir modifier un évènement au même titre que l'organisateur.	Modifier un évènement
8	<b>Suppression d'un évènement</b>	Administrateur	L'administrateur doit pouvoir supprimer un évènement au même titre que l'organisateur.	Supprimer un évènement

## Prestations attendues

### Livrables et prestations

Liste des prestations à livrer :

--Plan de projet

- Rédaction du cahier des charges
- Rédaction des spécifications fonctionnelles & techniques
- Maquettage & design du site
- Développement du site
- Intégration
- Documentation
- Maintenance et mises à jour
- Hébergement sur un serveur
- Déploiement
- Assistance technique

### Budget

Le financement du budget pour la mise en place de ce projet provient des cotisations des membres, de dons, de subventions ou d'autres sources de financement.

### Planning et principaux jalons

14/09/23 : la premier scrum pour bien cerner le projet  
Du 15/09/23 au 28/09/23 : Création des graphiques  
Du 29/09/23 au 20/10/23 : Développement  
21/10/23 : Phase de tests

### Méthodologie de suivi projet

Le suivi du projet se fait sur les principes de la **méthodologie Scrum avec des sprints** d'une semaines. La gestion des tâches se fait sur **Trello et GitHub**.

# GESTION DE PROJET

## Méthodologie

Inspiré de la méthode Agile, toutes les fin semaines il y a une réunion pour savoir ce qui a été fait et ce qu'il reste à faire.

Chaque membre de l'association prend la parole afin que chacun exprime sa vision en fonction du travail qu'il a accompli ou qu'il a à accomplir.

On consulte la liste des tâches sur **Trello**, chaque membre fait son récapitulatif en fonction de ses avancées..

Lorsqu'une tâche est terminée, une autre est attribuée jusqu'à la finalité du projet.

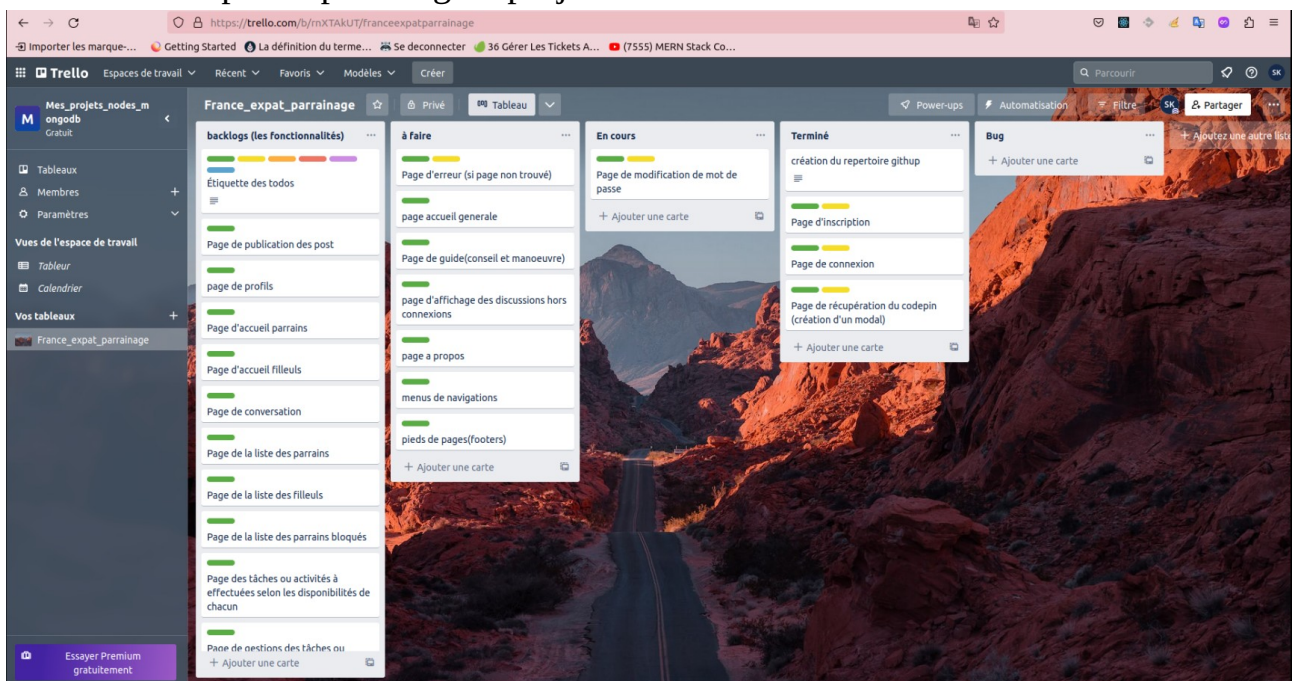
## Outil

Pour la gestion du projet, j'utilise **Trello**, les maquettes ont été réalisées avec le logiciel **DIA** et Pour ce qui est de la **communication entre membre**, elle se fait soit par **Zoom** ou soit lors des **réunions les week-ends**.

Tout le code est poussé sur **GitHub** dans la branche principale.

## Planning et suivi

Voici un exemple de planning du projet sur Trello :



## Environnement technique et humain

Je travaille avec une distribution « **Linux** », « **Ubuntu** ». Pour des raisons de simplicité et de maîtrise j'ai choisi cette OS et cette distribution.

J'utilise « **Visual Studio code** » comme éditeur de texte et le logiciel « **Postman** » qui me permet de tester les requêtes envoyées à l'**api**,

J'interagis régulièrement avec mon tuteur , pour m'épauler sur le projet en cas de soucis.

# SPÉCIFICATIONS FONCTIONNELLES

## Description de la solution

### Caractéristiques de la solution

Notre application Web est une solution accessible via navigateur, mettant à disposition les fonctionnalités suivantes :

-La partie **Front-office** qui offre une interface conviviale aux utilisateurs. En effet, cette partie vise à offrir un espace virtuel où les expatriés peuvent interagir, partager des expériences, échanger des informations et établir des liens sociaux avec d'autres individus qui se trouvent dans des situations similaires.

-La partie **Back-office** : est la partie du site qui n'est pas visible pour les utilisateurs finaux, mais qui joue un rôle crucial dans la gestion, le fonctionnement et la maintenance de la plateforme. Elle permet aux administrateurs de s'assurer du bon fonctionnement du site, la sécurité des données et la gestion des utilisateurs, des contenus et des fonctionnalités.

### Historique de la solution

Non concerné, le projet n'existait pas auparavant.

### Acteurs du projet

Tous les dix membres de l'association ont contribué à la réalisation du projet.

### Processus utilisateurs impactés

Les utilisateurs impactés sont les personnes inscrites sur le site ainsi que l'administrateur.

### Applications connexes

Il n'y a pas d'applications connexes concernées.

### Documentation référence

Non concerné, ce projet n'a pas de documentation de référence.

### Lexique/glossaire

**Front-office** : Le front office est l'ensemble de tout ce que les utilisateurs du site voient et avec lequel ils interagissent directement. Il fournit une interface conviviale



aux utilisateurs et de permettre leur interaction avec d'autres utilisateurs et le contenu.

**Back-office** : Sur internet, la notion de back-office correspond à la partie invisible au grand public. Il est utilisé par l'administrateur et permet de veiller au bon fonctionnement de l'ensemble du site, à la gestion des données, à la sécurité et à la performance.

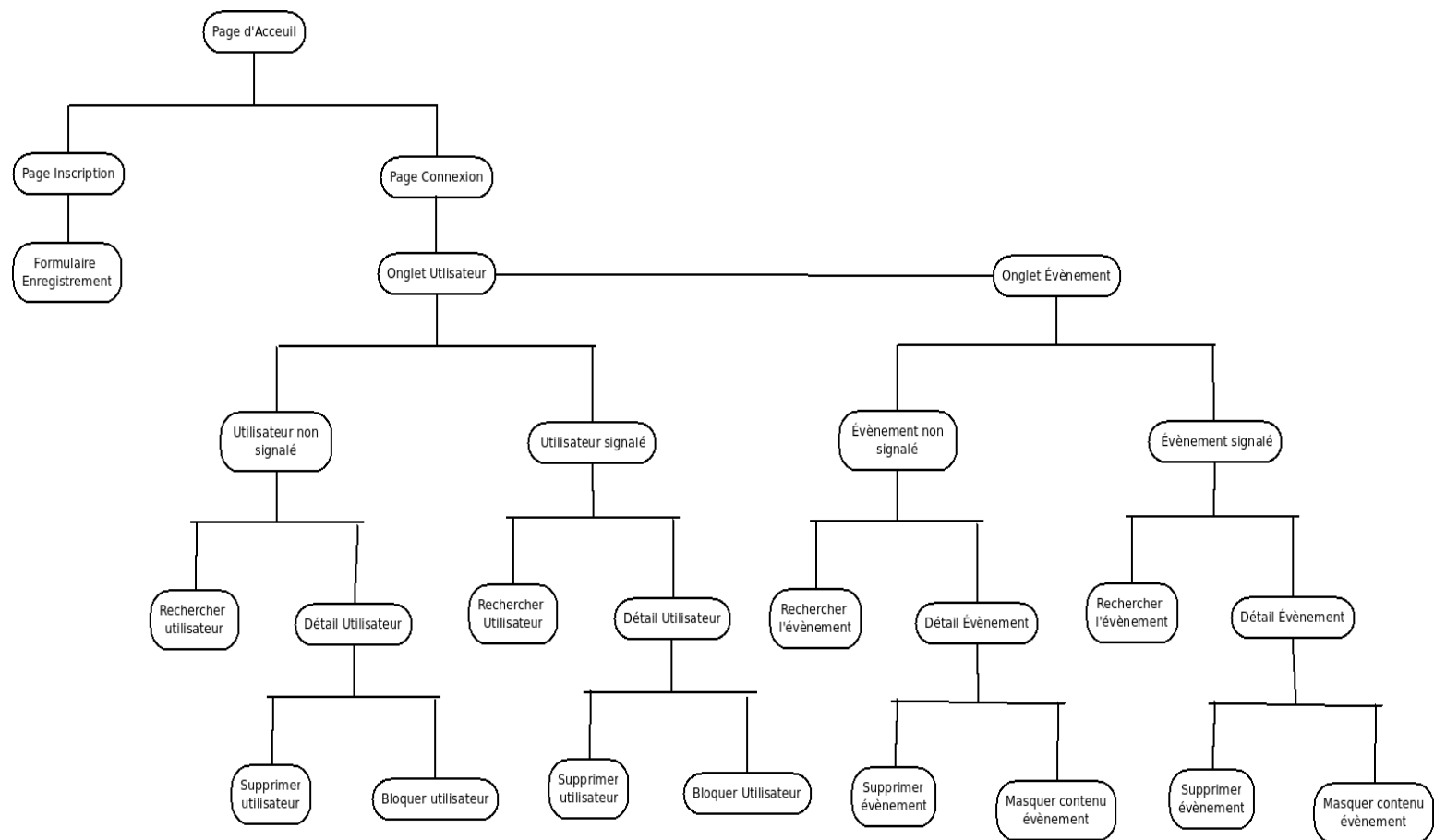
### Aspect métier

→ Criticité de l'application

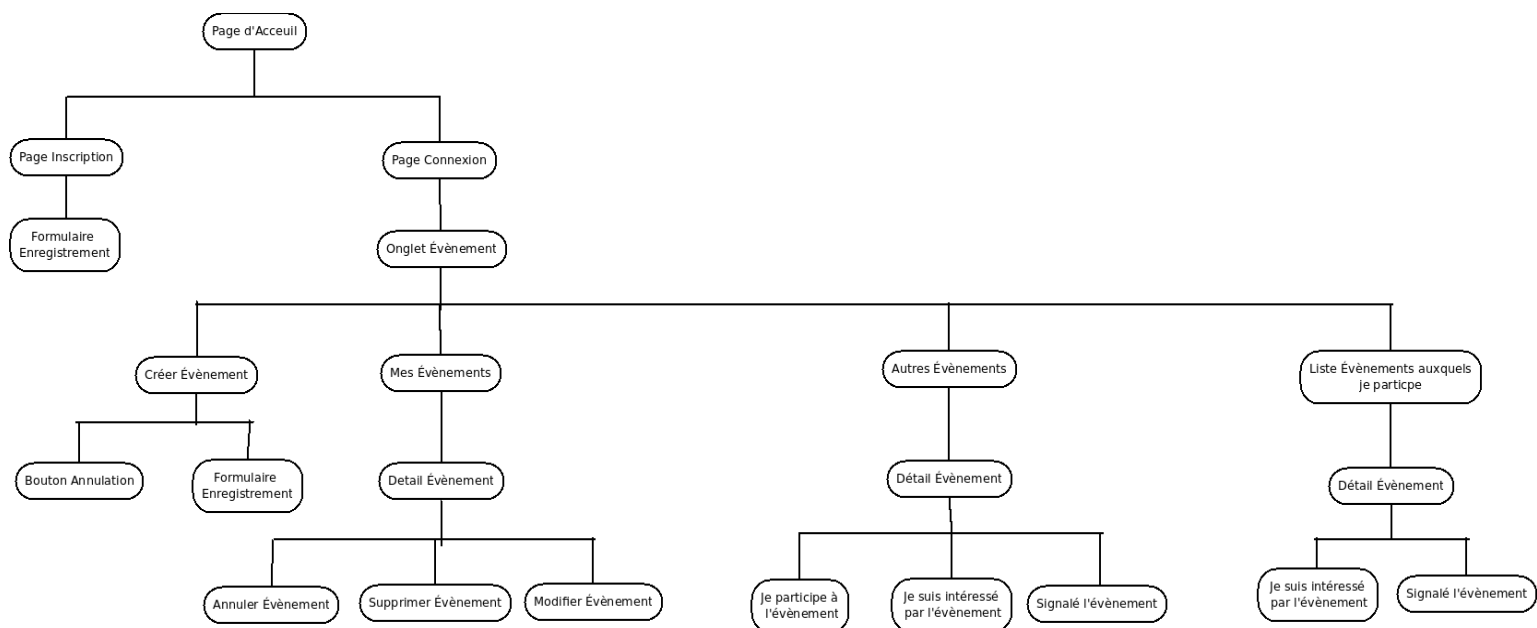
Étant donné que l'application réponds aux besoins et aux priorités spécifiques des personnes vivant à l'étranger, Il est donc important d'avoir des plans de secours et des moyens alternatifs de communication en cas de problème sur le site, tels que des contacts d'urgence, des applications de messagerie sécurisées 24/7 et une équipe essayera de résoudre les problèmes.

→ Cartographie (Arborescence)

### **Application administrateur**



## Gestion évènement :



## Généralités fonctionnelles

### Utilisateurs

Dans le tableau ci-dessous, on n'as la généralité fonctionnelle du système de gestion des évènements dans notre application.

<i>Rôle</i>	<b>Admin</b>	<b>Utilisateur (Organisateur)</b>	<b>Utilisateur (Participant)</b>
<i>Fonctionnalité</i>			
<i>Front-Office</i>			
Connexion aux front-office		X	X
Créer et enregistrer un évènement		X	
Voir la liste des évènements créés		X	
Voir la liste des évènements non créés		X	X
Voir son calendrier d'évènement		X	X
Rechercher un évènement		X	X
Voir le contenu d'un évènement		X	X
Participer à un évènement			X
S'intéresser à un évènement			X
Partager un évènement		X	X
Signaler un évènement			X
Modifier un évènement		X	
Supprimer un évènement		X	
<i>Back-Office</i>			
Connexion aux back-office	X		
Voir la liste de tous les évènements non signalés	X		
Voir la liste de tous les évènements signalés	X		
Voir le contenu d'un évènement	X		
Rechercher un évènement	X		
Modifier un évènement	X		
Supprimer un évènement	X		
Masquer le contenu d'un évènement	X		

## **Fonctionnalités**

DIAGRAMME CAS D'UTILISATION EVENEMENT

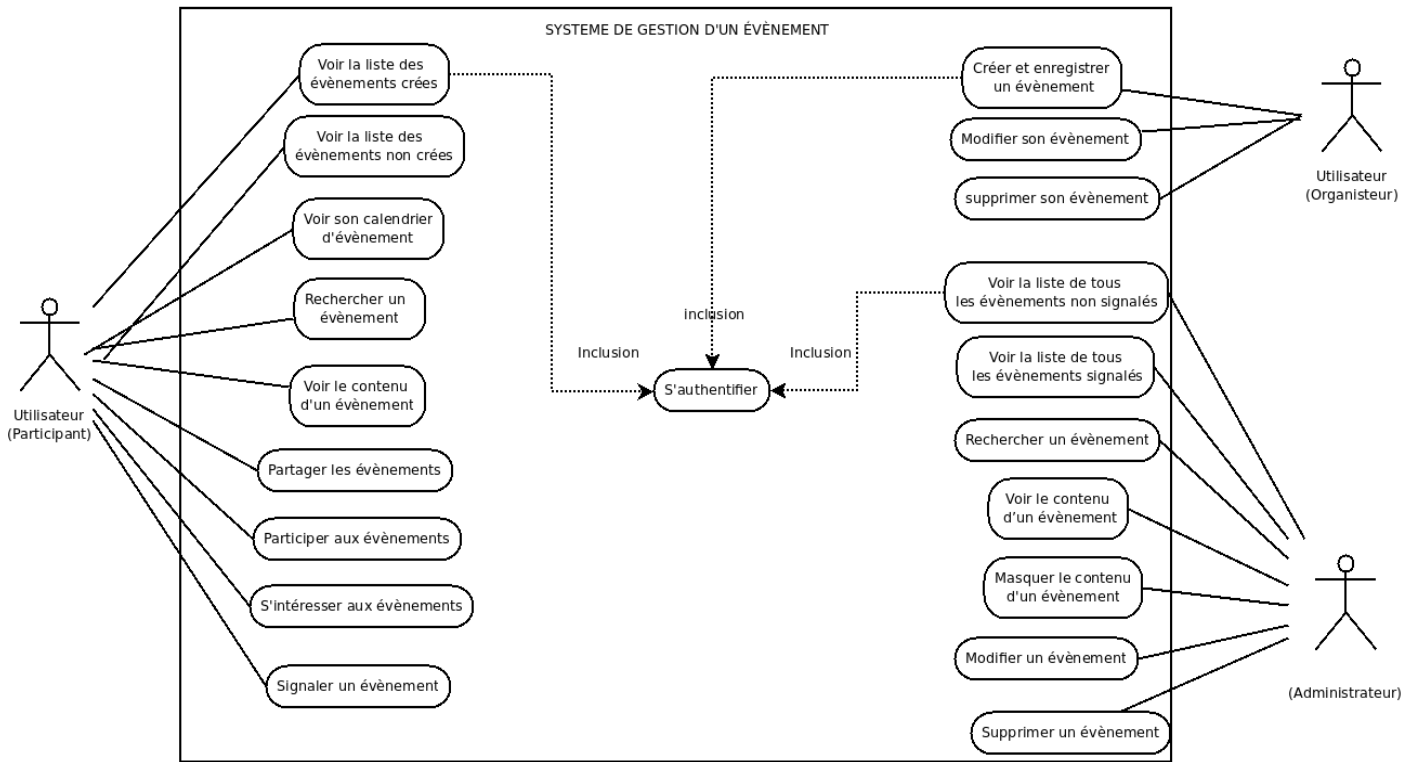
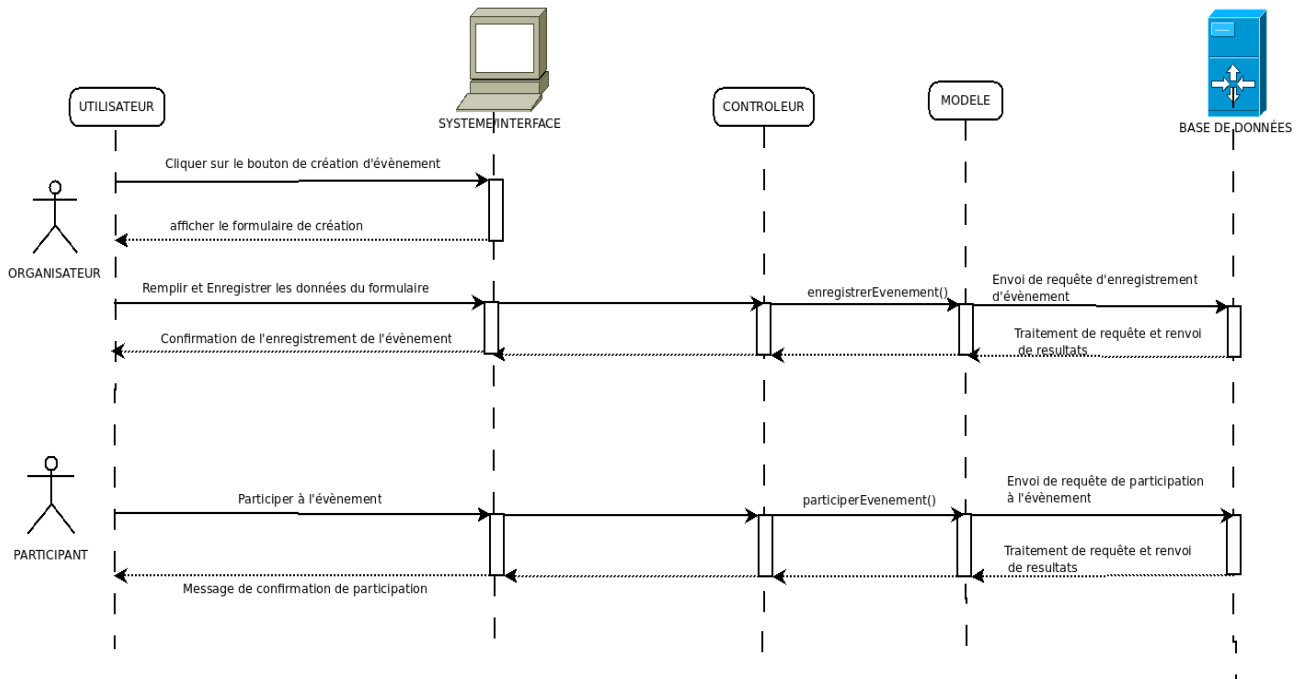


DIAGRAMME DE SEQUENCE DE PARTICIPIATION À UN EVENEMENT



### Règle métier

Non concerné.

### Autorisations

Notre application est constitué de deux interface distinctes qui sont :

-> Celle du **front-office** : Il s'agit de l'interface destinée au grand public.

-> Celle du **back-office** : seuls les utilisateurs ayant pour rôle « **admin** » associé en base de données peuvent s'y authentifier.

### Envoi de mail

Non concerné pour le moment

### Impression

Non concerné

### Gestion des langues

Non concerné pour le moment, seul le français est utilisé actuellement.

### Workflow

Non concerné

### Accessibilité

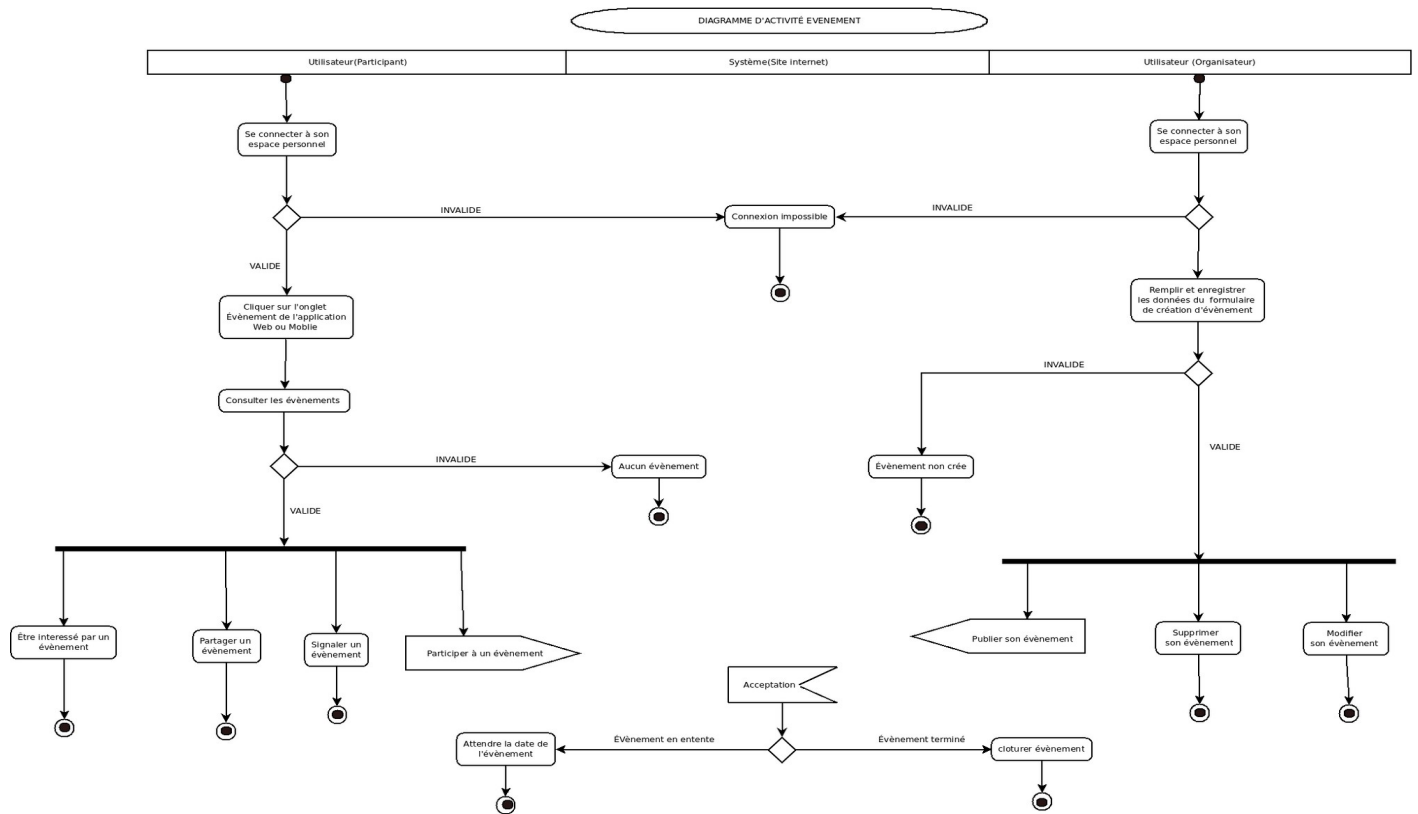
Non concerné

### Traitements planifiés

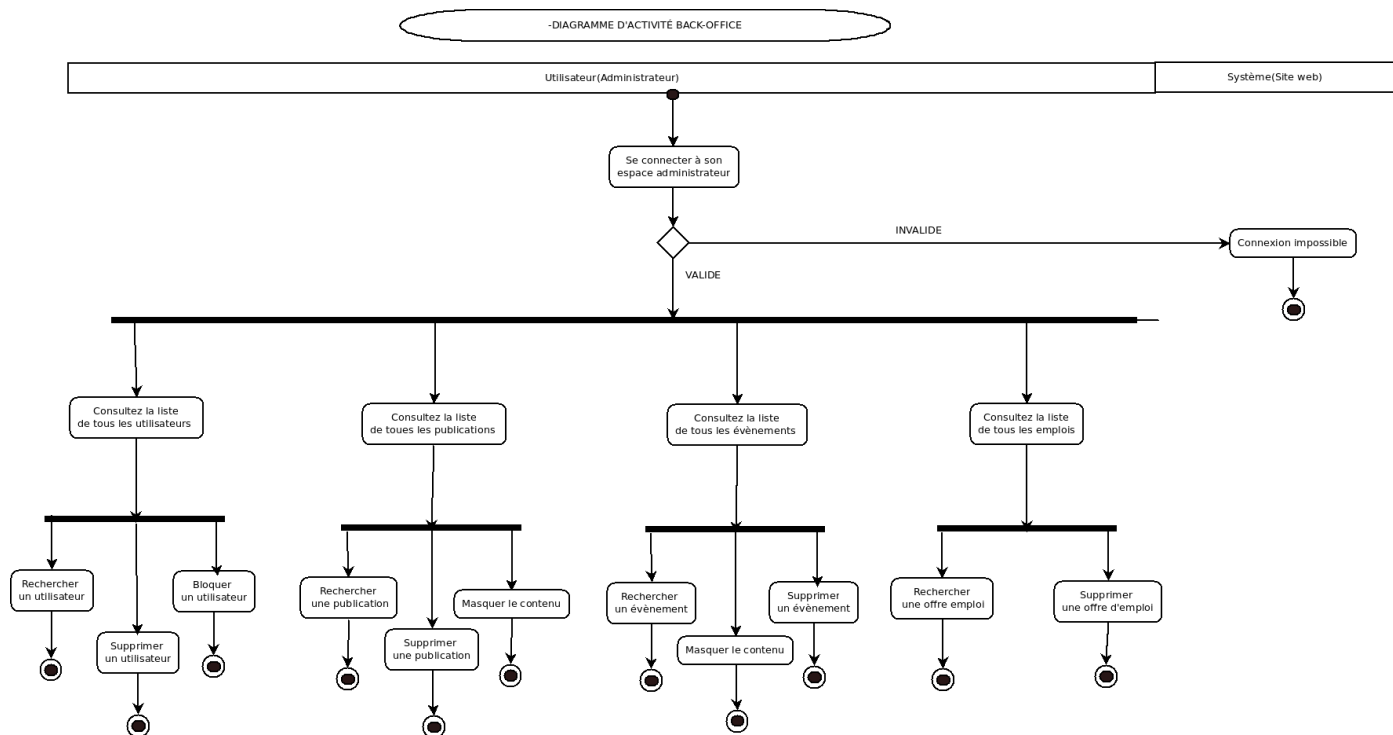
Non concerné

## Diagramme d'activité

### Diagramme d'activité du système de gestion de l'évènement (front-office) :



## Diagramme d'activité du back-office :



## Fonctionnalités détaillées

### Fonctionnalité n° 1 : Connexion au front-office

#### Description

C'est depuis cette page que le client peut s'authentifier grâce à ses identifiants (e-mails et mot de passe).

#### Données

Une requête **POST** avec deux données en entrée : **email** de type string, **password** de type string.

Trois données en sortie : **data** de type object.

#### Écran/Affichage

Il s'agit d'un formulaire de deux champs avec un **input** de type **text** et un autre de type **password**.

Voir la maquette à la page ???



## Fonctionnalité n° 2 : Créer et enregistrer un évènement

### ***Description***

L'utilisateur va devoir renseigner les informations obligatoires qui permettront de créer un évènement (Titre, Type d'évènement, Ville, Adresse, Date et Heure, Durée de l'évènement, la description , photo de couverture) , puis enregistrer l'évènement en cliquant sur le bouton Créer

### ***Données***

Lors de la validation finale, toutes les informations concernant l'évènement (Titre, Type d'évènement, Ville, Adresse, Date et Heure, Durée de l'évènement, la description , photo de couverture) sont transmises au serveur (requête de type post, sous forme d'objet).

On va ensuite coté serveur vérifier que la date saisi par l'utilisateur est exacte et que il n'existe pas un autre évènement organisé par ce dernier à la même date et heure. Si c'est le cas, on affichera un message de confirmation de création de l'évènement.

### ***Écran/Affichage***

Le formulaire de création est composé de 8 input tous obligatoire dont 7 de type "text" et 1 de type date:

Un bouton de soumission du formulaire (input type "submit")

Lorsque la création est terminée, un message de confirmation s'affiche.

(Maquette page 18 - 3.2.3.6)

## Fonctionnalité n° 3 : Voir la liste des évènements créés

### ***Description***

Cette page permet à l'utilisateur de voir la liste des tous les évènements qui lui sont associés.

### ***Données***

Une **requête GET** sans donnée en entrée.

Une donnée en sortie : data de type object.

### ***Écran/Affichage***

Il s'agit d'une page constituée d'un ou plusieurs composants de type carte avec de bref information(Titre, nombre de participant, Date et Heure , Ville) sur l'évènement .Ces composants sont disposée de façon verticale , maximum trois par ligne. Il est possible de voir le contenu des évènements après un clic sur l'un d'entre eux.

## Fonctionnalité n° 4 : Voir la liste des évènements non créés

### ***Description***

Cette page permet à l'utilisateur de voir la liste des tous les évènements qui ne lui sont pas associés.

### ***Données***

Une **requête GET** sans une donnée en entrée.

Une donnée en sortie : data de type object.

### ***Écran/Affichage***

Il s'agit d'une page constituée d'un ou plusieurs composants de type carte avec de bref information(Titre, nombre de participant, Date et Heure , Ville) sur l'évènement .Ces composants sont disposée de façon Horizontale , maximum un par ligne. Il est possible de voir le contenu d'un évènement après un clic sur l'un d'entre eux.

## Fonctionnalité n° 5 : Voir son calendrier d'évènements

### ***Description***

Cette page permet à l'utilisateur de voir la liste des tous les évènements dans lesquels il souhaite assister.

### ***Données***

Une requête GET sans une donnée en entrée.

Une donnée en sortie : data de type object.

### ***Écran/Affichage***

Il s'agit d'une fenêtre qui affiche le titre de tous les évènements dans lesquels l'utilisateur décide de participer. Il est possible de voir le contenu d'un évènement après un clic sur le titre.

## Fonctionnalité n° 6 : Rechercher un évènement (Front-office)

### ***Description***

L'utilisateur va ici pouvoir rechercher les évènements via le titre de l'évènement.

### ***Données***

Une **requête GET** sans données en entrée pour filtrer la liste des évènements via le nom de l'organisateur et le titre de l'évènement.

Une donnée en sortie : data de type object.

### ***Écran/Affichage***

La recherche d'un évènement se présente sous forme de deux champs qui permet de trouver un évènement associé à un organisateur.

## Fonctionnalité n° 7 : Voir le contenu d'un évènement

### ***Description***

L'utilisateur va ici pouvoir consulter les évènements. Cependant, si l'utilisateur connecté est l'organisateur de l'évènement, il pourra avoir accès à des informations supplémentaires tels que les informations sur les participants et les personnes intéressés à l'évènement.

### ***Données***

Une **requête GET** avec une données en entrée (**l'Id de l'évènement**) pour afficher les détails sur l'évènement sélectionné via le nom de l'organisateur et le titre de l'évènement.

Une donnée en sortie : data de type object.

### ***Écran/Affichage***

Il s'agit d'une page qui affiche plus d'information sur l'évènement consulté. Nous avons les information tel que : (Le nom, la catégorie, la description, le lieu, le nombre de participant, le nombre d'intéressé, la date et l'heure, la durée, le nom de l'organisateur, la photo de couverture). Tous ces informations sont disposées de façon claire et compréhensive.

## Fonctionnalité n° 8 : Participer à un évènement

### *Description*

L'utilisateur va pouvoir ici s'inscrire à un évènement après avoir cliquer sur le bouton participer

### *Données*

Une requête POST avec deux données en entrée : **eventId** de type string, **userId** de type string.

Une donnée en sortie : data de type object.

### *Écran/Affichage*

Il s'agit d'un bouton qui permettra a l'utilisateur de s'inscrire à un évènement après avoir cliqué sur le bouton. Ainsi, il sera ajouté à la liste des participant à l'évènement. Cette liste n'est visible que par l'organisateur

## Fonctionnalité n° 9 : S'intéresser à un évènement

### *Description*

L'utilisateur va pouvoir ici **s'intéresser** à un évènement après avoir cliquer sur le bouton « je suis intéressé »

### *Données*

Une requête POST avec deux données en entrée : **eventId** de type string, **userId** de type string.

### *Écran/Affichage*

Il s'agit d'un bouton qui permettra a l'utilisateur d'être intéressé à un évènement après avoir cliqué sur le bouton. Ainsi, il sera ajouté à la liste des intéressés à l'évènement. Cette liste n'est visible que par l'organisateur

## Fonctionnalité n° 10 : Signaler un évènement

### *Description*

L'utilisateur va pouvoir ici **signaler** un évènement après avoir cliquer sur le bouton « signaler l'évènement »

### *Données*

Une requête **post** avec une données en entrée : **eventId** de type string

### ***Écran/Affichage***

Il s'agit d'un bouton qui permettra a l'utilisateur de signaler un évènement après avoir cliqué sur le bouton. Ainsi, il sera ajouté à la liste des évènement signalés. Cette liste n'est visible que par l'administrateur.

## **Fonctionnalité n° 11 : Modifier un évènement**

### ***Description***

L'utilisateur va pouvoir ici **modifier** un évènement après avoir cliquer sur le bouton «modifier l'évènement» et ensuite remplir le formulaire.

### ***Données***

Une requête put avec une données en entrée : **eventId** de type string

### ***Écran/Affichage***

Il s'agit d'un bouton qui permettra a l'utilisateur d'afficher une fenêtre constituée d'un formulaire qui permettra de modifier un évènement. Un message de confirmation sera affiché après avoir modifier l'évènement.

## **Fonctionnalité n° 12 : Supprimer un évènement**

Description

Données

Écran/Affichage

## **Fonctionnalité n° 13 : Connexion au back-office**

Description

Données

Écran/Affichage

## **Fonctionnalité n° 14 : Voir la liste de tous les évènements non signalés**

Description

Données

Écran/Affichage

### **Fonctionnalité n° 15 : Voir la liste de tous les évènements signalés**

Description

Données

Écran/Affichage

### **Fonctionnalité n° 16 : Rechercher un évènement**

Description

Données

Écran/Affichage

### **Fonctionnalité n° 17 : Voir le contenu d'un évènement**

Description

Données

Écran/Affichage

### **Fonctionnalité n° 18 : Masquer le contenu d'un évènement**

Description

Données

Écran/Affichage

### **Fonctionnalité n° 19 : Modifier un évènement**

Description

Données

Écran/Affichage

### **Fonctionnalité n° 20 : Supprimer un évènement**

Description

Données

Écran/Affichage



















