

Rapport de stage d'été

Présenté à

**L'Institut Supérieur d'Informatique et
de Multimédia de Sfax**

Spécialité

Licence en Big Data et Analyse de Données

Plateforme de Gestion d'Événements

Elaborée Par

Ayoub Ghoula

Réalisé au sein de

Tradrly

Encadré par

Mme. Aya Abdessamad

Année Universitaire 2024-2025

Remerciements

À l'issue de ce projet, j'aimerais exprimer ma gratitude envers toutes les personnes qui ont participé à son élaboration et à son succès.

Je souhaite adresser un remerciement particulier à Mme Aya Abdessamad, mon encadrante chez Tradrly. Son soutien indéfectible, ses conseils judicieux et son accompagnement tout au long de ce travail ont été déterminants. Sa disponibilité et ses retours constructifs ont constitué une véritable aide pour faire progresser et réussir ce projet.

Je tiens aussi à exprimer ma reconnaissance à l'ensemble de l'équipe de Tradrly, qui m'a offert un accueil chaleureux, partagé son expertise et apporté une assistance précieuse. Leur collaboration et leurs retours professionnels ont enrichi ce projet, me permettant d'acquérir des compétences pratiques et pertinentes.

J'aimerais également remercier le corps professoral et administratif de l'Institut Supérieur d'Informatique et de Multimédia de Sfax, pour les connaissances et compétences qu'ils ont transmises tout au long de ma formation.

Je tiens aussi à remercier les membres du jury qui ont examiné et évalué mon travail.

Pour conclure, je souhaite remercier toutes les personnes, de près ou de loin, qui ont contribué à la réalisation de ce projet, pour leur aide précieuse et leur motivatio

Table des matières

Introduction générale.....	1
Chapitre 1 : Présentation générale du projet	
1.Introduction.....	3
2.Présentation de la société.....	3
2.1 Présentation de Tradrly	3
2.2 Activités et services.....	3
3. Problématique.....	3
4.Solution proposée.....	4
5. Contexte du projet	4
6.Objectifs à atteindre.....	4
7.Conclusion.....	5
Chapitre 2 : Analyse des besoins	
1.Introduction.....	7
2.Choix de la méthodologie de conception.....	7
2.1 Présentation du langage UML.....	7
2.2 Présentation d'outil de modélisation	7
3. Analyse des besoins.....	7
3.1 Besoins fonctionnels.....	7
3.2 Besoins non fonctionnels.....	8
4. Identification des Acteurs.....	8
5. Diagrammes de cas d'utilisations.....	9
5.1 Diagramme de cas d'utilisation globale.....	9
5.2 Diagramme de cas d'utilisation détaillé.....	10
5.2.1 Cas d'utilisation « inscrire au plateforme » - « vérifier email »	10
5.2.2 Cas d'utilisation « Rechercher un événement»	11
5.2.3 Cas d'utilisation « Réserver des tickets».....	11
5.2.4 Cas d'utilisation « Créer un événement».....	12
5.2.5 Cas d'utilisation « Gérer les réservations».....	12
5.2.6 Cas d'utilisation « gérer le Profil ».....	13

6. Conclusion du chapitre.....	13
Chapitre 3 : Modélisation conceptuelle	
1.Introduction.....	15
2. conception architecturale.....	15
2.1 Architecture MVC.....	15
3. Diagramme de séquence.....	16
2.2 Diagramme de séquence « inscrire au plateforme » - « vérifier email ».....	20
2.2 Diagramme de séquence « authentification ».....	21
4. Diagramme de classe.....	17
5.conclusion.....	17
Chapitre 4 : Réalisation	
1.Introduction.....	19
2.Environnement de travail.....	19
3.Description des interfaces de l'application.....	26
4.Conclusion.....	35
Conclusion Générale.....	36
Bibliographie.....	36

Listes des figures

Figure 1 : Logo Tradrly	21
Figure 2 : Logo UML	32
Figure 3 : Logo Enterprise Architect.....	32
Figure 4 : Diagramme de cas d'utilisation globale	33
Figure 5 : Modèle MVC	47
Figure 6 : Modèle MVC	47
Figure 7 : Modèle MVC	47
Figure 8 : Modèle MVC	47
Figure 9 : Diagramme de classe	48
Figure 10 : Logo de Laravel	63
Figure 11 : Logo de React.js	66
Figure 12 : Logo de TypeScript	69
Figure 13 : Logo de Tailwind CSS	72
Figure 14 : Logo de MySQL	76
Figure 15 : Logo de Postman	79
Figure 16 : Logo de Postman	79
Figure 17 : Logo de Postman	79
Figure 18 : Logo de Postman	79
Figure 19 : Logo de Postman	79
Figure 20 : Logo de Postman	79
Figure 21 : Logo de Postman	79
Figure 22 : Logo de Postman	79
Figure 23 : Logo de Postman	79
Figure 24 : Logo de Postman	79

Introduction Générale

L'informatique a connu une évolution rapide et profonde, transformant notre façon de vivre, de travailler et d'interagir. Parmi les nombreux secteurs impactés, celui de la gestion et de l'organisation des événements a particulièrement bénéficié des avancées technologiques. La numérisation de ces processus est devenue essentielle pour répondre aux besoins des utilisateurs modernes, toujours en quête de solutions rapides, intuitives et efficaces.

Avant l'avènement des plateformes numériques, la gestion des événements se faisait principalement de manière traditionnelle, avec des limites en termes d'accessibilité et d'organisation. Aujourd'hui, grâce aux systèmes d'information et aux applications web, les organisateurs peuvent gérer leurs événements plus efficacement, et les participants peuvent accéder à des informations précises et réserver en quelques clics.

C'est dans ce contexte que s'inscrit EventMaster, une application web conçue pour simplifier et améliorer la gestion des événements. Elle offre une plateforme intuitive et moderne permettant aux utilisateurs de créer, gérer et participer à divers événements. EventMaster vise à répondre aux attentes de trois catégories principales d'utilisateurs :

- Participants : Explorent les événements, réservent des tickets, et interagissent avec les organisateurs.
- Organisateurs : Créent et gèrent leurs événements en assurant une expérience utilisateur optimale.
- Administrateurs : Supervisent l'ensemble du système pour garantir son bon fonctionnement et sa sécurité.

Ce rapport, nous présentons en 4 chapitres :

- Le premier chapitre, « Présentation générale du projet » : nous procéderons à une introduction complète de notre projet. Cela englobera une présentation générale du contexte, des objectifs et des principales parties prenantes.
- Le deuxième chapitre, « Capture des besoins » : détaillera les besoins fonctionnels et non fonctionnels, ainsi que les diagrammes de cas d'utilisation.
- Le troisième chapitre, « Modélisation Conceptuelle », se concentrera sur la création de modèles conceptuels pour définir la structure et les interactions de l'application à l'aide de diagrammes de séquence et de classes.
- Le quatrième chapitre, « Réalisation », décrit la réalisation et la mise en œuvre de l'application

Chapitre 1 : Présentation générale du projet

1. Introduction

Dans ce chapitre, nous présentons l'organisme d'accueil, Tradrly, la société au sein de laquelle nous effectuons notre projet de fin d'année. Nous abordons ensuite la mise en lumière de la problématique à laquelle notre projet cherche à répondre, ainsi que la formulation des objectifs.

2. Présentation de la société :

2.1 Présentation de Tradrly :



Figure 1 : Logo Tradrly

Créée le 16 septembre 2023, **Tradrly** est une startup innovante basée à Mahdia, en Tunisie. Elle se positionne à l'avant-garde des technologies en proposant des solutions de pointe dans les domaines du web, du mobile, et de l'intelligence artificielle.

2.2 Activités et services :

Tradrly travaille avec un éventail de clients, allant des startups aux entreprises établies, pour développer :

- Des applications web et mobiles sur mesure.
- Des solutions intégrées basées sur l'intelligence artificielle.
- Une transformation digitale complète adaptée aux besoins de chaque projet.

Avec chaque réalisation, Tradrly cherche à dépasser les attentes et à redéfinir l'avenir des technologies numériques.

3. Problématique :

La gestion des événements est une tâche complexe nécessitant une organisation précise et efficace des différentes activités liées à leur planification et leur réalisation. Actuellement, les organisateurs et participants d'événements font face à des défis tels que :

- La difficulté pour les organisateurs de gérer les inscriptions, d'informer les participants, et de suivre les modifications.
- La complexité pour les participants de trouver des événements correspondant à leurs intérêts et de réserver des tickets de manière fluide.
- Le manque de supervision centralisée pour assurer la qualité et la conformité des événements.

Ces défis engendrent souvent des pertes de temps, des malentendus, et une expérience utilisateur insatisfaisante, qu'il s'agisse d'organiser ou de participer à un événement.

Ainsi, la problématique centrale de ce projet réside dans la recherche d'une solution innovante permettant de concevoir une plateforme de gestion d'événements. Cette plateforme devra simplifier les interactions entre participants, organisateurs, et administrateurs tout en optimisant l'organisation des activités pour améliorer l'expérience utilisateur globale.

4. Solution proposée :

Suite à cette étude, nous présentons brièvement la solution que nous proposons pour répondre à la problématique identifiée. Notre approche consiste à développer une plateforme web de gestion d'événements, permettant aux organisateurs de créer et gérer des événements de manière simple et efficace, tout en offrant aux participants un accès facile aux événements, avec la possibilité de réserver des tickets et de recevoir des notifications .

L'objectif principal de cette plateforme est de faciliter la gestion des événements tout en améliorant l'expérience utilisateur, tant pour les organisateurs que pour les participants.

5. Contexte du projet :

Dans le cadre de notre projet de fin d'année, nous avons été sollicités pour la conception et le développement d'une **plateforme web de gestion d'événements**, visant à simplifier l'organisation et la participation aux événements.

Cette plateforme a pour objectif principal de répondre aux besoins des organisateurs, des participants, et des administrateurs en proposant une solution intuitive et efficace. Les fonctionnalités clés incluent :

- La création, la modification et la suppression d'événements par les organisateurs.
- La réservation de tickets et la réception de notifications par les participants.
- La supervision et la gestion centralisée des activités par les administrateurs.

Ce projet s'inscrit dans un contexte où la digitalisation des processus est essentielle pour améliorer l'efficacité, l'accessibilité et l'expérience utilisateur dans le domaine de la gestion d'événements.

6. Objectifs à atteindre :

Les objectifs visés pour notre plateforme web de gestion d'événements sont les suivants :

- Concevoir une interface utilisateur intuitive pour simplifier l'expérience des participants et des organisateurs.

- Mettre en œuvre des mesures de sécurité robustes pour protéger les données des utilisateurs et assurer la fiabilité du système.
- Permettre aux participants de rechercher facilement des événements, réserver des tickets et recevoir des notifications en temps réel.
- Offrir aux organisateurs des outils efficaces pour créer, modifier et gérer leurs événements.
- Fournir aux administrateurs un système centralisé pour superviser les activités et traiter les plaintes.

7. Conclusion :

En conclusion de ce chapitre, nous avons posé les bases de notre projet en présentant la société **Tradrly**, ses activités innovantes dans le domaine des technologies web, mobiles et d'intelligence artificielle, ainsi que les motivations qui sous-tendent le développement de notre plateforme web de gestion d'événements.

Dans le chapitre suivant, nous entreprendrons une étude de conception afin de modéliser les différentes parties de notre projet.

Chapitre 2 : Analyse des besoins

1. Introduction

Dans ce chapitre, nous analysons les besoins liés à la plateforme web de gestion d'événements afin d'identifier les fonctionnalités et les exigences requises pour répondre efficacement à la problématique. Nous décrivons les besoins fonctionnels et non fonctionnels, ainsi que les principaux acteurs et leurs interactions avec le système.

2.Choix de la méthodologie de Conception :

2.1 Présentation du langage UML :

UML (Unified Modeling Language) est un langage standard de modélisation orientée objet utilisé pour représenter visuellement les objets, états et processus d'un logiciel ou système. Il facilite la conception structurée, sert de modèle pour les projets, et aide les développeurs à communiquer clairement avec les spécialistes externes.



Figure 2 : Logo UML

2.2 Présentation d'outil de modélisation :

L'outil de modélisation que nous avons utilisé est "Enterprise Architect", un logiciel puissant et polyvalent qui facilite la création, la gestion et la documentation des modèles UML pour notre projet de conception d'une plateforme web de gestion d'événements.



Figure 2 : Logo Enterprise Architect

3. Analyse des besoins :

3.1. Besoins fonctionnels :

Les besoins fonctionnels décrivent les fonctionnalités spécifiques que la plateforme doit fournir :

Pour les participants :

- Rechercher des événements par titre, type, ou date.
- Consulter les détails des événements (description, date, lieu, image).
- Réserver des tickets en ligne et payer.
- Télécharger des tickets .
- Recevoir des notifications pour les événements réservés.

- Déclarer des problèmes ou plaintes sur un événement.

Pour les organisateurs :

- Créer de nouveaux événements.
- Modifier ou supprimer des événements, avec notification aux participants concernés
- Consulter et gérer la liste des participants.
- Envoyer des notifications aux participants.

Pour les administrateurs :

- Valider les comptes des organisateurs.
- Gérer les utilisateurs.
- Traiter les plaintes des participants.

3.2 Besoins non fonctionnels :

Les besoins non fonctionnels décrivent les caractéristiques qualitatives de la plateforme :

- **Sécurité** : Assurer la confidentialité des données utilisateurs et la fiabilité des paiements.
- **Performance** : Garantir un temps de réponse rapide et une navigation fluide.
- **Évolutivité** : Permettre l'ajout de nouvelles fonctionnalités à l'avenir.
- **Accessibilité** : Rendre la plateforme compatible avec différents appareils.

4. Identification des acteurs

Les principaux acteurs impliqués dans la plateforme sont :

- Participants : Utilisateurs qui recherchent et participent à des événements.
- Organisateurs : Utilisateurs qui créent et gèrent des événements.
- Administrateurs : Supervisent les utilisateurs et la conformité des activités.

5. Diagrammes de cas d'utilisations :

5.1 Diagramme de cas d'utilisation globale :

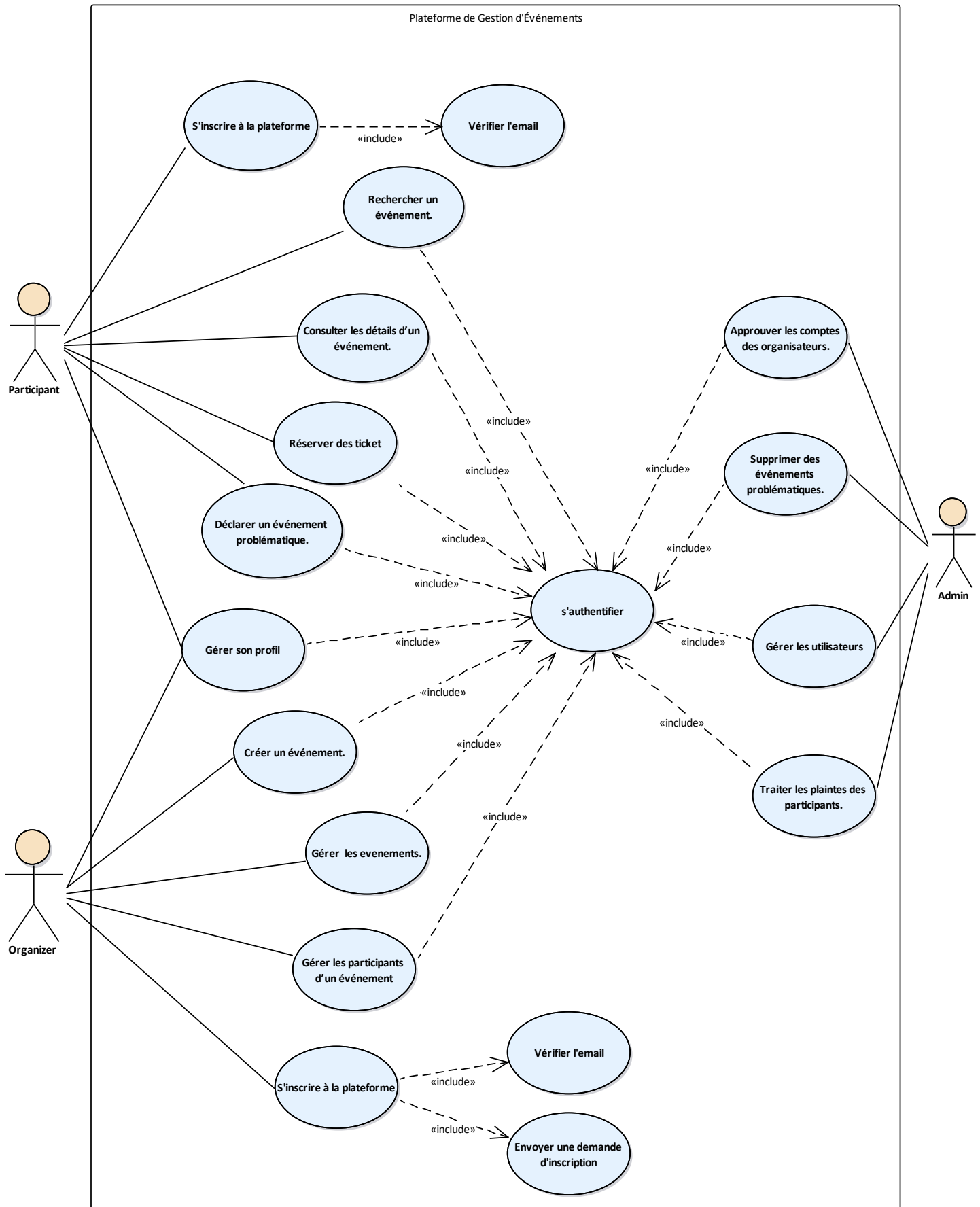


Figure 3 : Diagramme de cas d'utilisation globale

5.2 Diagramme de cas d'utilisation détaillé :

5.2.1 Description textuelle du cas d'utilisation « inscrire au plateforme » - « vérifier email » :

<u>Titre :</u> S'inscrire à la plateforme
<u>Acteur :</u> Participant
<u>Précondition :</u> Le participant a accédé à la page d'inscription de la plateforme.
<u>Postcondition :</u> Un compte participant est créé avec succès après la vérification de l'email.
<u>Scénario Nominal :</u> <ol style="list-style-type: none">1. Le participant accède à la page d'inscription de la plateforme.2. Le participant remplit les informations nécessaires.3. Le système vérifie la validité des informations fournies.4. Si les informations sont valides, le système génère un email de confirmation et envoie un code de vérification unique5. Le participant consulte son email et obtient le code de vérification.6. Une fenêtre modale s'affiche sur la page de l'application, demandant au participant de saisir le code de vérification reçu par email.7. Le participant saisit le code.8. Le système vérifie l'e-mail et finalise le processus d'inscription.9. Un message de confirmation s'affiche, indiquant que l'inscription est réussie.
<u>Scénario Alternatif :</u> Alt-1) Les informations fournies par le participant sont invalides : <ul style="list-style-type: none">• Le système affiche un message d'erreur demandant de corriger les champs manquants ou invalides. Alt-2) Le code saisi est invalide : <ul style="list-style-type: none">• Le système affiche un message d'erreur : "Code incorrect, veuillez réessayer." Alt-3) Le participant ne reçoit pas le code : <ul style="list-style-type: none">• Le participant peut demander un renvoi du code. Le système génère un nouveau code et l'envoie à l'adresse email du participant. Alt-4) Le compte existe déjà pour l'email fourni : <ul style="list-style-type: none">• Le système affiche un message d'erreur indiquant que l'adresse email est déjà utilisée.

5.2.2 Description textuelle du cas d'utilisation « Rechercher un événement » :

<u>Titre</u> : Rechercher un événement
<u>Acteur</u> : Participant
<u>Précondition</u> : Le participant est connecté à la plateforme.
<u>Postcondition</u> : Le participant visualise une liste d'événements correspondant à ses critères de recherche.
<u>Scénario Nominal</u> : <ol style="list-style-type: none">1. Le participant accède à la barre de recherche d'événements.2. Le participant entre des critères de recherche (titre, date, type d'événement).3. Le participant lance la recherche.4. Le système récupère les événements correspondant aux critères et affiche les résultats sous forme de liste.
<u>Scénario Alternatif</u> : Alt-1) Aucun événement ne correspond aux critères : <ul style="list-style-type: none">• Le système affiche un message : "Aucun événement trouvé pour ces critères." Alt-2) L'utilisateur entre des critères invalides ou incomplets : <ul style="list-style-type: none">• Le système affiche un message : "Veuillez entrer au moins un critère valide pour la recherche."

5.2.3 Description textuelle du cas d'utilisation « Réserver des tickets » :

<u>Titre</u> : Réserver des tickets
<u>Acteur</u> : Participant
<u>Précondition</u> : Le participant est connecté à la plateforme et a sélectionné un événement.
<u>Postcondition</u> : Le participant réserve un ou plusieurs tickets pour un événement avec succès.
<u>Scénario Nominal</u> : <ol style="list-style-type: none">1. Le participant accède à la page de réservation pour un événement donné.2. Il sélectionne le nombre de tickets qu'il souhaite réserver.3. Il remplit les informations nécessaires pour chaque ticket (nom, email, numéro de téléphone).4. Il procède au paiement via une méthode sécurisée.5. Le système génère un ticket unique pour chaque réservation et les envoie par email au participant.6. Le participant visualise une confirmation de la réservation.

Scénario Alternatif :

Alt-1) Le participant dépasse le nombre maximal de tickets autorisé :

- Le système affiche un message : "Vous ne pouvez pas réserver plus de X tickets."

Alt-2) Le paiement échoue :

- Le système affiche un message : "Erreur de paiement. Veuillez réessayer."

5.2.4 Description textuelle du cas d'utilisation « Créer un événement » :

Titre : Créer un événement

Acteur : Organisateur

Précondition : L'organisateur est connecté à la plateforme et a un compte validé par l'administrateur.

Postcondition : Un nouvel événement est créé et visible pour les participants.

Scénario Nominal :

1. L'organisateur accède à la page de création d'événements.
2. Il remplit les informations nécessaires (titre, description, date, lieu, type, image, prix des tickets, etc.).
3. L'organisateur soumet le formulaire.
4. Le système valide les données et enregistre l'événement.
5. Le système envoie une notification aux participants pour les informer du nouvel événement.

Scénario Alternatif :

Alt-1) Les informations sont incomplètes ou invalides :

- Le système affiche un message d'erreur pour les champs manquants ou incorrects.

Alt-2) L'image uploadée dépasse la taille autorisée :

- Le système affiche un message d'erreur : "L'image dépasse la taille autorisée (4 Mo)."

5.2.5 Description textuelle du cas d'utilisation « Gérer les réservations » :

Titre : Gérer les réservations

Acteur : Organisateur

Précondition : L'organisateur a créé un événement et des participants ont réservé des tickets.

Postcondition : L'organisateur consulte et télécharge les informations des participants inscrits.

Scénario Nominal :

1. L'organisateur accède à la page de gestion des réservations.

2. Il sélectionne un événement parmi ceux qu'il a créés.
3. Le système affiche la liste des participants ayant réservé des tickets pour cet événement.
4. L'organisateur télécharge la liste en format PDF contenant les détails (nom, email, numéro de téléphone, code unique des tickets).

Scénario Alternatif :

Alt-1) Aucun participant n'a réservé de tickets :

- Le système affiche un message d'erreur pour les champs manquants ou incorrects.

5.2.6 Description textuelle du cas d'utilisation « gérer le Profil »:

Titre : Gérer le Profil

Acteur : Participant

Précondition : Le participant est connecté à la plateforme.

Postcondition : Les actions de consultation, de modification, et de suppression du profil sont effectuées avec succès.

Scénario Nominal :

1. Le participant accède à la section "Profil" de l'application.
2. Le système affiche les informations du profil du participant.
3. Le participant choisit l'option "Modifier" pour mettre à jour ses informations.
4. Le participant apporte les modifications nécessaires.
5. Le système enregistre les changements effectués.
6. Le participant peut éventuellement choisir l'option "Supprimer" pour effacer son profil.
7. Le système confirme la suppression du profil.
8. Le profil du participant est supprimé avec succès.

Scénario Alternatif :

Alt-1) Si le participant choisit d'annuler les modifications :

- Le système conserve les informations précédentes sans effectuer de modifications.
- Le profil du participant reste inchangé.

Alt-2) Si le participant annule la suppression du profil :

- Le système conserve les informations du profil sans suppression.
- Le profil du participant reste actif dans le système.

6. Conclusion

Ce chapitre a permis de définir les besoins fonctionnels et non fonctionnels de la plateforme, ainsi que les acteurs et leurs rôles. Ces éléments constituent une base solide pour modéliser et concevoir les différentes parties du projet, qui seront détaillées dans le prochain chapitre.

Chapitre 3 : Modélisation Conceptuel

1. Introduction :

Ce chapitre traite de la modélisation conceptuelle de notre plateforme de gestion d'événements. Nous présentons les diagrammes de séquences pour illustrer la vue dynamique des principaux cas d'utilisation, suivis du diagramme de classes, qui décrit les entités du système et leurs relations.

2. Conception Architecturale :

Pour la conception de notre plateforme, nous avons choisi l'architecture **MVC** (Modèle-Vue-Contrôleur), qui s'avère très efficace pour organiser l'interaction entre les différents composants de notre plateforme. Cette architecture favorise la modularité et facilite la maintenance ainsi que l'évolution de l'application.

2.1. Architecture MVC :

L'architecture **MVC** est un modèle en trois couches largement utilisé pour la conception des applications web, des interfaces graphiques et des systèmes client/serveur. Sa force réside dans la séparation claire entre les données (Modèle), l'affichage (Vue) et la logique de traitement (Contrôleur).

- **Modèle :**

Le Modèle représente les données et la logique métier de l'application. Il interagit directement avec la base de données pour lire, écrire ou mettre à jour les informations. Le Modèle est également chargé d'informer la Vue des modifications apportées aux données.

- **Vue :**

La Vue se concentre uniquement sur l'affichage des informations. Elle reçoit les données traitées par le Contrôleur et les présente à l'utilisateur sous forme graphique ou textuelle. La Vue constitue l'interface utilisateur de l'application.

- **Contrôleur :**

Le Contrôleur agit comme un intermédiaire entre la Vue et le Modèle. Il récupère les entrées de l'utilisateur (via la Vue), les traite et envoie les commandes appropriées au Modèle. Ensuite, il transmet les données mises à jour à la Vue pour affichage.

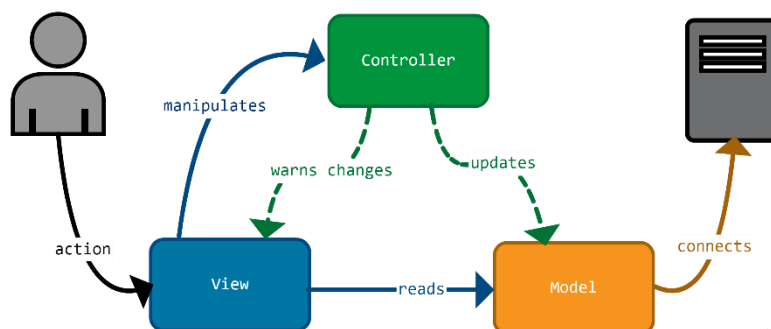
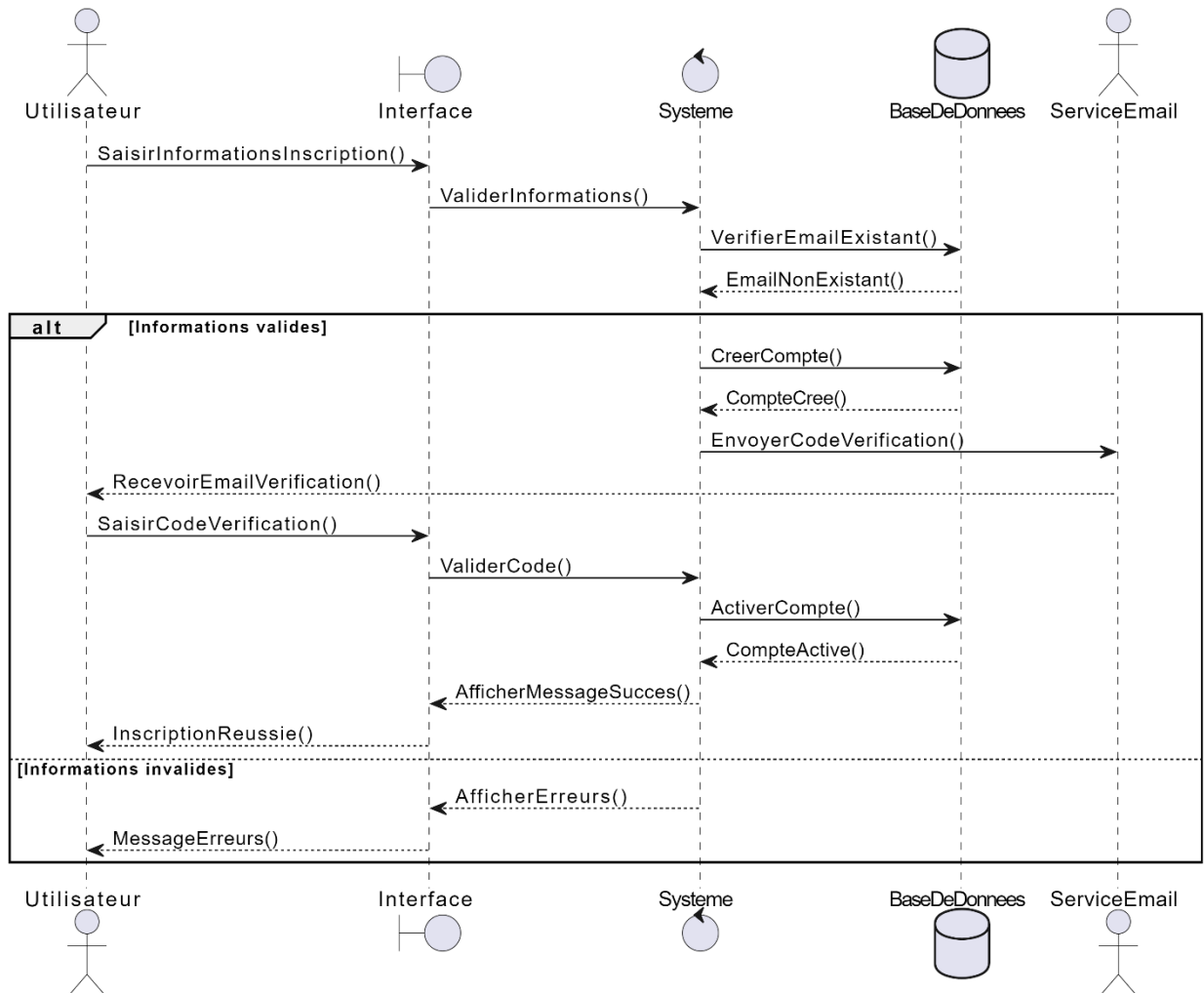


Figure 4 : Modèle MVC

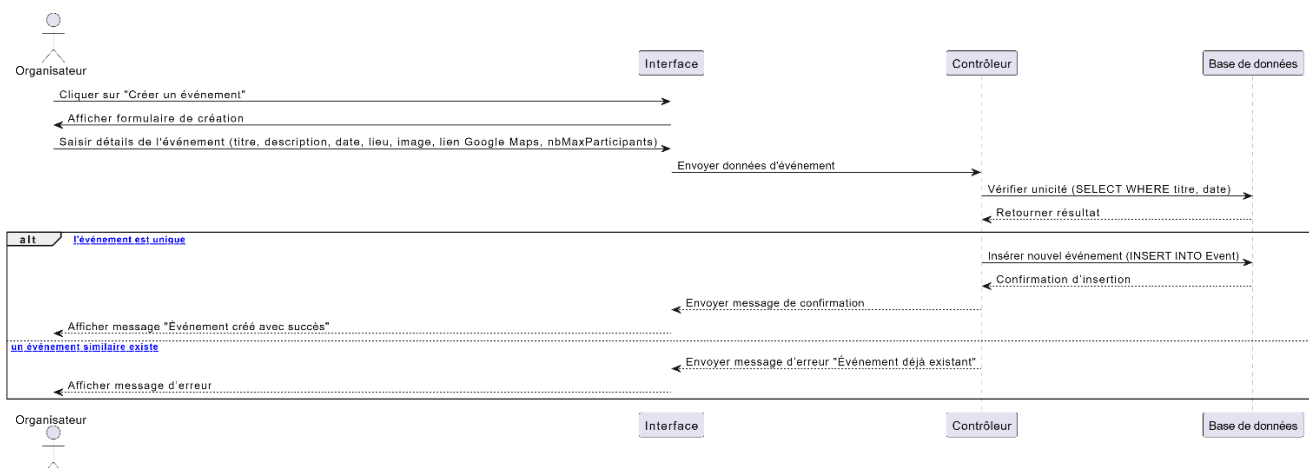
3. Diagramme de séquence :

Un diagramme de séquence décrit l'aspect dynamique du système. Il modélise les interactions entre les objets ou entre utilisateur et objet, en mettant l'accent sur la chronologie des messages échangés.

3.1 Diagramme de séquence « inscrire au plateforme » - « vérifier email »:



3.2 Diagramme de séquence « Création d'un événement »:



4. Diagramme de classe :

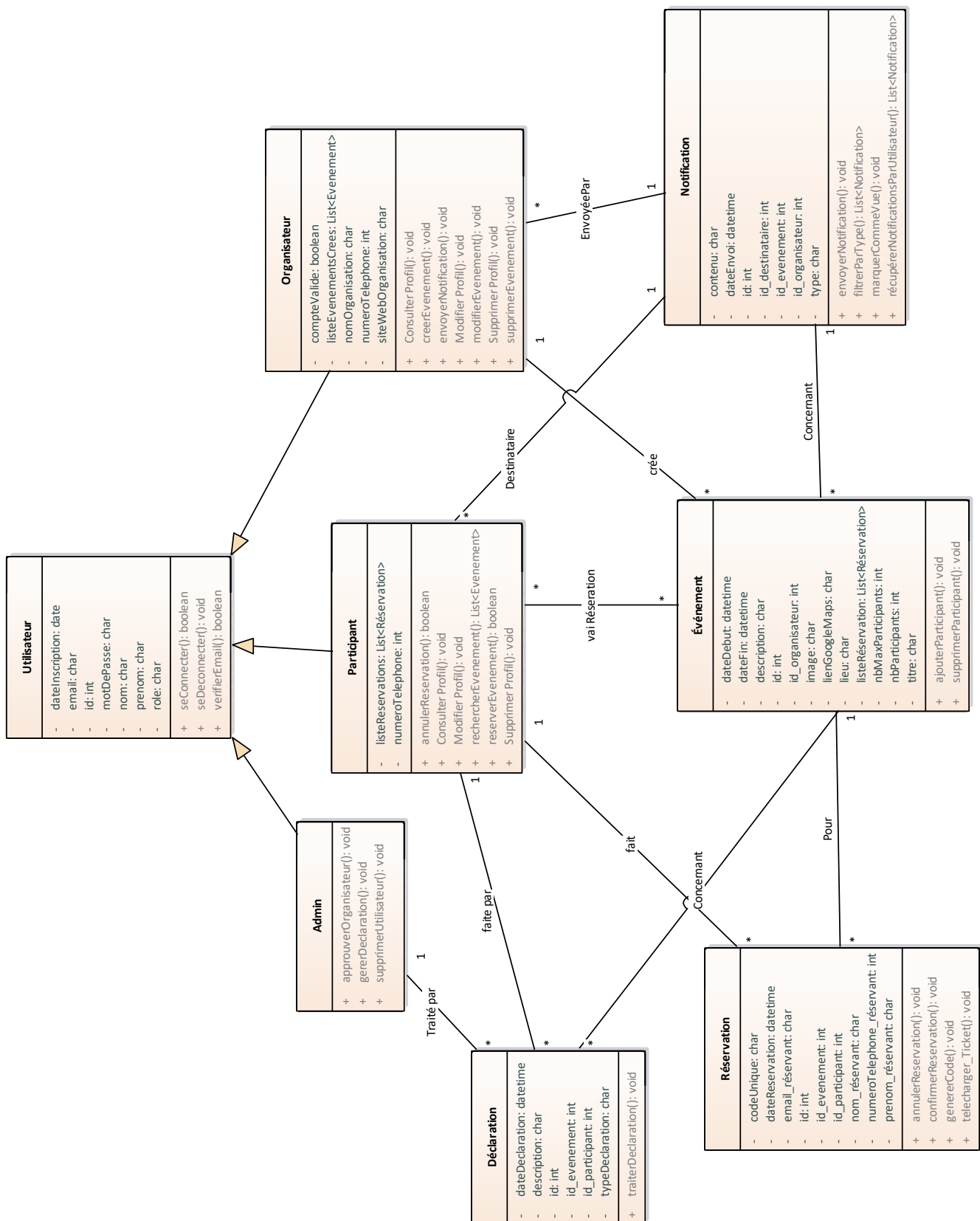


Figure 7 : Diagramme de classe

5. Conclusion :

Ce chapitre a présenté la modélisation conceptuelle de notre plateforme, mettant en évidence les interactions et la structure du système. Le prochain chapitre traitera de la réalisation de l'application.

Chapitre 4 : Réalisation

1. Introduction :

Dans ce chapitre, nous présentons les outils et technologies employés pour le développement de notre plateforme. Nous illustrons également les principales fonctionnalités réalisées à travers des captures d'écran, offrant une démonstration concrète de l'application.

2. Environnement de travail :

Dans cette section, nous décrivons les principales technologies utilisées pour le développement de notre plateforme web de gestion d'événements :

➤ **Laravel :**

Laravel est un framework PHP open-source qui facilite le développement d'applications web robustes et sécurisées. Il offre un ensemble d'outils intégrés, tels que l'ORM Eloquent, le système de routage avancé et les migrations de base de données, pour simplifier le développement backend.



Figure 8 : Logo de Laravel

➤ **React.js :**

React.js est une bibliothèque JavaScript utilisée pour la création d'interfaces utilisateur dynamiques et interactives. Grâce à son approche basée sur les composants et à son utilisation du DOM virtuel, React garantit des performances optimales pour les applications modernes.



Figure 9 : Logo de React.js

➤ TypeScript :

TypeScript est un sur-ensemble de JavaScript qui ajoute des types statiques, rendant le code plus structuré et facile à maintenir. Il est utilisé dans notre projet pour améliorer la qualité et la robustesse du frontend développé avec React.



Figure 10 : Logo de TypeScript

➤ Tailwind CSS :

Tailwind CSS est un framework de design utilitaire qui permet de créer rapidement des interfaces utilisateur élégantes et personnalisables. Grâce à ses classes CSS prédéfinies, il simplifie grandement la mise en forme et l'adaptation du design aux différents écrans.



Figure 11 : Logo de Tailwind CSS

➤ MySQL :

MySQL est un système de gestion de bases de données relationnelles utilisé pour stocker et gérer les informations de notre plateforme, telles que les utilisateurs, les événements, les réservations et les notifications.



Figure 12 : Logo de MySQL

➤ Postman :

Postman est un outil utilisé pour tester les API de manière efficace. Il a été utilisé pour valider les endpoints de notre backend développé avec Laravel.



POSTMAN

Figure 13 : Logo de Postma

4.Description des interfaces de l'application :

Dans cette section, nous décrirons les principales interfaces de notre application, mettant en avant leur rôle et leur interaction avec les utilisateurs. Voici un aperçu des interfaces clés :

- **Page d'accueil :**

La page d'accueil de la plateforme offre une vue globale des fonctionnalités principales et des informations générales. Elle est conçue pour accueillir les visiteurs avec une interface conviviale et intuitive.

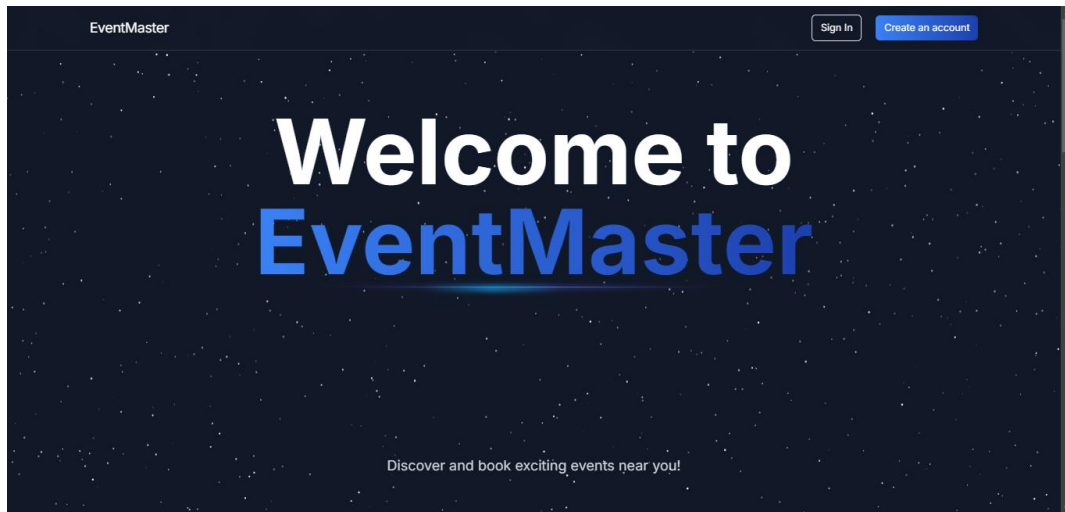


Figure 12 : Logo de MySQL

- **Interface d'inscription :**

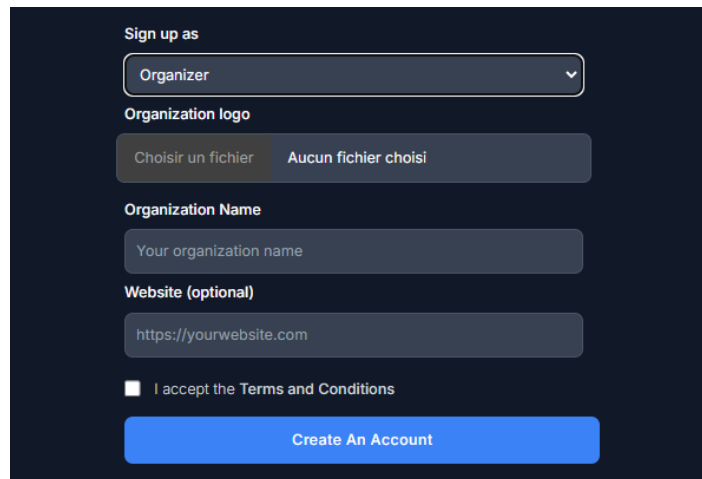
L'interface d'inscription offre une expérience fluide et intuitive pour permettre aux nouveaux utilisateurs de rejoindre la plateforme. Elle propose un formulaire où les utilisateurs peuvent saisir des informations telles que leur nom, prénom, adresse e-mail, numéro de téléphone et mot de passe.

Après leur inscription, les utilisateurs accèdent à une interface dédiée à la vérification de leur adresse e-mail pour confirmer leur compte. Un code unique est envoyé à leur adresse e-mail, qu'ils doivent saisir dans cette interface. En cas de non-réception du code, une option leur permet de demander un nouvel envoi.

Figure 12 : Logo de MySQL

Figure 12 : Logo de MySQL

En outre, une liste déroulante permet de sélectionner le rôle de l'utilisateur, soit Participant, soit Organisateur. Lorsqu'un utilisateur choisit le rôle Organisateur, son inscription nécessite une validation supplémentaire par l'administrateur afin de garantir la sécurité et la fiabilité des organisateurs sur la plateforme.

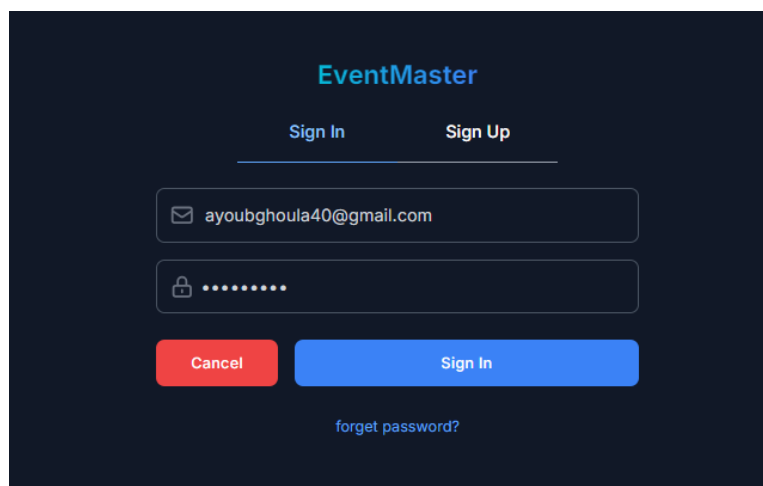


The screenshot shows a 'Sign up as' form with a dropdown menu set to 'Organizer'. Below this is a section for 'Organization logo' with a button 'Choisir un fichier' and the text 'Aucun fichier choisi'. The 'Organization Name' field contains 'Your organization name'. The 'Website (optional)' field contains 'https://yourwebsite.com'. There is a checkbox for 'I accept the Terms and Conditions' and a blue 'Create An Account' button at the bottom.

Figure 12 : Logo de MySQL

- **Interface d'authentification :**

L'interface d'authentification de notre plateforme offre une méthode simple et sécurisée pour que les utilisateurs puissent accéder à leur compte. Elle permet aux utilisateurs de se connecter en saisissant leur adresse e-mail et leur mot de passe. En cas d'oubli du mot de passe, une option dédiée leur permet de récupérer l'accès à leur compte via leur adresse e-mail.



The screenshot shows the 'EventMaster' login interface. It has a header with 'EventMaster' in blue. Below the header are two tabs: 'Sign In' (active) and 'Sign Up'. The 'Sign In' form has an email field with 'ayoubghoula40@gmail.com' and a password field with masked characters. There are 'Cancel' and 'Sign In' buttons. A 'forget password?' link is at the bottom.

Figure 12 : Logo de MySQL

- **Interface Discover Events :**

L'interface *Discover Events* permet aux participants de découvrir les événements disponibles. Les utilisateurs peuvent rechercher des événements en fonction de critères comme le titre, la date ou le type d'événement. Chaque événement est affiché sous forme de carte.

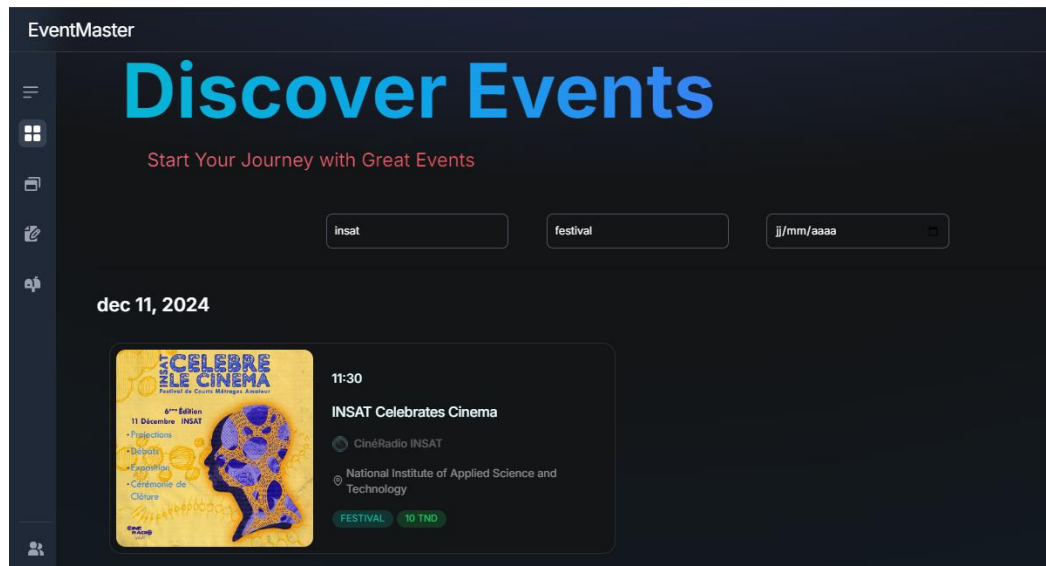


Figure 12 : Logo de MySQL

- **Interface de consultation des détails d'un événement :**

Cette interface permet aux utilisateurs d'accéder à des informations détaillées sur un événement sélectionné. Elle affiche le titre, la description complète, la date, l'emplacement. Les participants peuvent également réserver un ticket directement depuis cette interface ou signaler un problème avec l'événement.



Figure 12 : Logo de MySQL

- **Interface de réservation des tickets :**

L'interface de réservation des tickets offre une méthode pratique pour que les participants puissent confirmer leur participation à un événement. Elle inclut un formulaire pour saisir les informations personnelles (nom, prénom, numéro de téléphone, et email) et un résumé des informations liées à l'événement. Les participants peuvent effectuer le paiement et télécharger leur ticket avec un code unique après la validation.

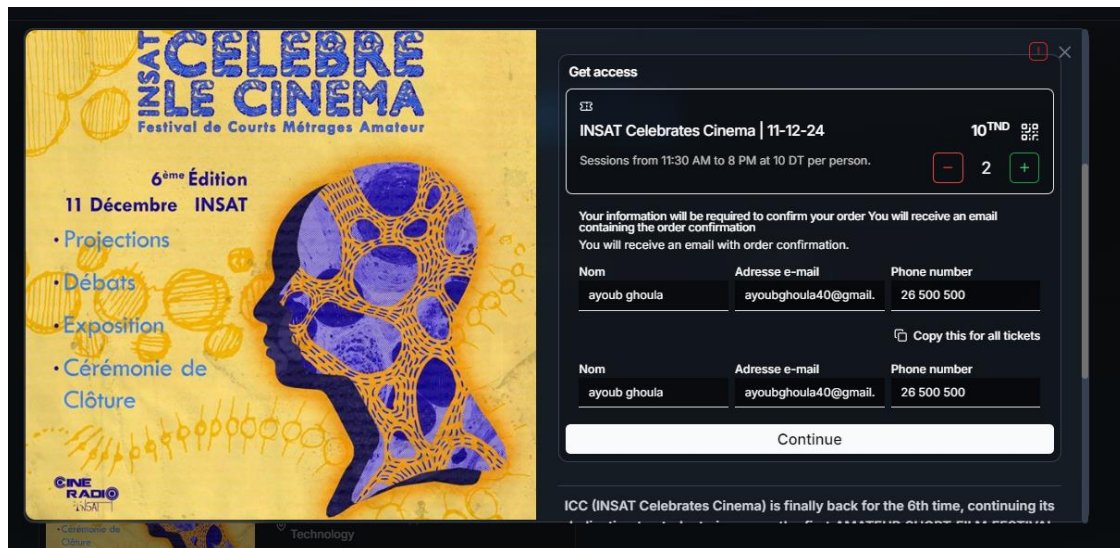


Figure 12 : Logo de MySQL

- **Interface de création d'événement :**

Cette interface est conçue pour permettre aux organisateurs de créer facilement de nouveaux événements. Ils peuvent remplir un formulaire avec des champs obligatoires tels que le titre, la description, la date, le type d'événement, le lieu, et le nombre maximum de participants. Une option pour télécharger une image de l'événement est également disponible.

Figure 12 : Logo de MySQL

5. Conclusion :

Dans ce chapitre consacré à la réalisation de notre plateforme, nous avons exploré différents aspects liés à l'environnement de travail, aux technologies utilisées, ainsi qu'à la description des interfaces principales accompagnées de captures d'écran.

Conclusion Générale

Au cours de ce projet de fin d'année, nous avons conçu et développé une plateforme web de gestion d'événements, permettant aux utilisateurs (participants et organisateurs) de s'inscrire, de gérer des événements, de réserver des billets et de recevoir des notifications. Ce travail a impliqué l'intégration de plusieurs technologies, dont Laravel pour le backend, React.js et TypeScript pour le frontend, ainsi que MySQL pour la gestion des données.

Tout au long de la réalisation de ce projet, nous avons fait face à plusieurs défis techniques et fonctionnels. L'un des plus grands défis a été l'optimisation des processus de réservation et de gestion des événements, afin de garantir une expérience utilisateur fluide et intuitive. La mise en place des notifications et des différents niveaux d'accès pour les utilisateurs (organisateur, participants, administrateurs) a également nécessité une attention particulière pour assurer la sécurité et la fiabilité de la plateforme.

Les obstacles rencontrés, tels que la gestion des autorisations complexes et l'intégration d'une fonctionnalité de paiement sécurisé pour les réservations, ont nécessité des ajustements techniques et une phase de test rigoureuse pour assurer la performance et la sécurité de la plateforme. De plus, l'adaptation à des exigences spécifiques en matière de conception d'interface utilisateur, tout en maintenant la simplicité et l'ergonomie, a constitué un autre défi majeur.

Pour l'avenir, plusieurs pistes d'amélioration ont été identifiées. L'ajout de fonctionnalités telles que la gestion des événements récurrents, la personnalisation des notifications pour les utilisateurs, ainsi que l'intégration de services tiers pour les paiements (par exemple, Konnect ou Flouci) pourrait améliorer l'expérience utilisateur et enrichir la plateforme. De plus, le développement de versions mobiles de l'application (en utilisant des technologies comme Flutter ou React Native) pourrait accroître l'accessibilité et étendre l'audience de la plateforme.

Ainsi, ces évolutions visent à renforcer les capacités de la plateforme, à élargir ses fonctionnalités et à améliorer son adoption sur le marché, tout en répondant de manière plus précise aux besoins des utilisateurs. L'intégration continue de nouveaux outils et technologies permettra également de maintenir la plateforme à jour et compétitive dans un environnement numérique en constante évolution.

BIBLIOGRAPHIE

1. **UML :**
[URL1]: <https://www.omg.org/spec/UML>
2. **Enterprise Architect :**
[URL2]: <https://www.sparxsystems.com/products/ea/>
3. **Laravel :**
[URL3]: <https://laravel.com/docs>
4. **React.js :**
[URL4]: <https://reactjs.org/docs/getting-started.html>
5. **TypeScript :**
[URL5]: <https://www.typescriptlang.org/docs/>
6. **MySQL :**
[URL6]: <https://dev.mysql.com/doc/>
7. **Tailwind CSS :**
[URL7]: <https://tailwindcss.com/docs>
8. **Postman :**
[URL8]: <https://www.postman.com/docs/>