

DÉDICACES

REMERCIEMENTS

Avant de commencer la présentation de ce rapport, je tiens à exprimer mes respects et mes sincères remerciements aux personnels du OMINET et Je tiens à remercier le directeur de la société et mon encadreur YAssez BENALI qui choisit le sujet de mon projet, et qui a été plus qu'un maître de stage. il m'a guidée, critiquée, et a fait des suggestions.

Son encouragement permanent et son dynamisme organisateur m'ont énormément facilité la tâche. il m'a conseillée tout au cours de mon stage. il m'a relu et critiquée mon manuscrit Je remercie également toute l'équipe pour son accueil, son esprit convivial et chaleureux. Enfin, je tiens à remercier tous ceux qui m'ont assisté à élaborer ce rapport de stage.

TABLE DES MATIÈRES

Introduction Générale	1
1 Présentation du cadre du projet	2
Chapitre 1 : Présentation du cadre du projet	2
Introduction	2
I. Présentation de l'organisme d'accueil	2
1. Situation géographique :	2
2. Historique de Coccinet :	3
3. Organigramme	3
II. Etude préalable	4
1. Critique de l'existant	4
2. Problématique :	5
3. Solution proposée	5
III. Méthodologie adaptées	6
1. Méthodologie de modélisation et de conception	6
2. Methodologie de gestion de projet	6

Conclusion	8
2 Planification du projet	9
Chapitre 2 : Planification du projet	9
Introduction	9
I. Spécification des besoins	9
1. Besoins fonctionnels	9
2. Besoins non fonctionnels	10
II. Gestion de projet avec Scrum	11
1. Équipe SCRUM et rôles	11
2. Product Backlog	12
3. Planification de sprint	15
III. Diagrammes de cas d'utilisations général	15
1. Identification des acteurs	15
2. Diagramme de contexte statique	16
3. Diagramme de cas d'utilisation global	17
IV. Diagramme de classes général	17
V. Environnement de développement et choix techniques	18
1. Environnement matériel	18
2. Environnement logiciel	19
3. Les langages de programmation	20
4. Framework	21
5. Le système de gestion de base de données	21
6. Les bibliothèques et les outils	21
VI. Architecture adoptée	23

1.	Architecture logique :	23
2.	Architecture physique	24
3	RELEASE 1	26
	Chapitre 3 : RELEASE 1	26
	Introduction	26
I.	Organisation du sprint	26
II.	Sprint 1 : Configuration initiale et base du projet	26
1.	Objectif du sprint	26
III.	Sprint 2 : Authentification et gestion des utilisateurs	27
1.	Objectif du sprint	27
2.	Sprint backlog	27
3.	Implémentation	28
4	RELEASE 2	33
	Chapitre 4 : RELEASE 2	33
	Introduction	33
I.	Organisation du sprint	33
II.	Sprint 3 : Gestion des événements (CRUD)	33
1.	Objectif du sprint	33
2.	Sprint backlog	34
3.	Implémentation du sprint 3	34
	Conclusion Générale	36
	Références	36

TABLE DES FIGURES

1.1	L'organigramme de l'OMINET	4
1.2	Le processus de Scrum	8
2.1	L'équipe Scrum	12
2.2	Planification des sprints	15
2.3	Diagramme de contexte statique	16
2.4	Diagramme de cas d'utilisation globale	17
2.5	Diagramme de classes général de l'application	18
2.6	L'architecture de Model-View-Controller	24
2.7	L'architecture client-serveur	25
3.1	Cas d'utilisation « S'inscrire / S'authentifier » (Participant)	28
3.2	Cas d'utilisation « Gérer les utilisateurs » (Admin)	30
3.3	Séquence « Authentification / JWT »	31
3.4	Séquence « CRUD Utilisateurs » (Admin)	32
4.1	Diagramme des cas d'utilisation gestion d'événement "Administrateur"	35

LISTE DES TABLEAUX

2.1	Product Backlog	14
2.2	Identification des acteurs et de leurs fonctionnalités	16
2.3	Caractéristiques matérielles des ordinateurs utilisés	18
3.1	User Stories – Sprint 2 : Authentification & gestion des utilisateurs	27
3.2	Inscription participant	29
3.3	Authentification participant / gestionnaire	29
3.4	Gestion des utilisateurs (Admin)	30
4.1	User Stories - Sprint Gestion des événements	34

INTRODUCTION GÉNÉRALE

Depuis l'émergence de l'informatique, l'homme n'a cessé de développer des langages et des outils pour concevoir le web, mêlant créativité et logique afin de créer des expériences en ligne interactives et fonctionnelles. Aujourd'hui, l'informatique est devenue un pilier central dans la gestion de l'information et un atout majeur dans le monde professionnel en perpétuelle évolution.

Pour les étudiants, acquérir de l'expérience dès leurs études est crucial. Pourtant, la recherche de stages reste un véritable défi : trouver des opportunités alignées avec leurs compétences et aspirations, tout en naviguant dans un écosystème professionnel souvent complexe.

C'est dans cette optique que notre projet prend tout son sens. Nous proposons la création d'une plateforme innovante dédiée à la gestion d'événements. Cet outil offrira aux utilisateurs qu'ils soient simples participants, gestionnaires d'événements ou administrateurs une interface moderne et intuitive pour interagir efficacement.

Les utilisateurs pourront consulter les événements disponibles, s'inscrire, réserver des places en ligne, donner leur avis et échanger avec une communauté dynamique. De leur côté, les administrateurs disposeront d'outils avancés pour organiser et gérer les événements, superviser les inscriptions et analyser les retours des participants, garantissant ainsi une meilleure gestion et une expérience utilisateur optimisée.

CHAPITRE 1

PRÉSENTATION DU CADRE DU PROJET

Introduction

Dans ce présent chapitre nous avons présenté l'organisme d'accueil, et ses principales activités. Nous avons aussi évoqué la problématique, la solution, l'objectif à atteindre et la planification de notre projet.

I. Présentation de l'organisme d'accueil

1. Situation géographique :

- Entreprise : OMINET SARL
- Adresse : Centre Urbain Nord, Tunis.
- Horaires : De 8h à 12h et de 14h à 17h.

OMINET SARL, héritière de Coccinet fondée en 2014, est une agence offshore spécialisée dans la création de sites Internet et les solutions digitales sur mesure. Grâce à une équipe d'experts expérimentés, l'entreprise accompagne ses clients dans le développement de leur présence en ligne.

Classée comme une PME (Petite et Moyenne Entreprise), OMINET SARL dispose de plusieurs départements, dont un département commercial et un département technique. L'équipe, composée de [insérer nombre de collaborateurs et stagiaires], est dirigée par un gestionnaire

principal.

Depuis sa création, OMINET SARL s'engage à offrir des solutions adaptées aux besoins de ses clients, avec pour objectif principal de maximiser leur satisfaction. L'entreprise met tout en œuvre pour développer des outils digitaux innovants, permettant à ses clients de booster leur performance et d'atteindre leurs objectifs stratégiques.

2. Historique de Coccinet :

Coccinet est une société à responsabilité limitée (SARL), immatriculée sous le SIREN 479824914, et en activité depuis 18 ans. Basée à Paris (75011), elle se spécialise dans le secteur de la programmation informatique.

Avec un effectif de 6 à 9 salariés, l'entreprise a réalisé un chiffre d'affaires de 200 300 € en 2012, marquant une augmentation notable de 43,64 du total du bilan entre 2011 et 2012.

Depuis sa création, Coccinet a vu plusieurs évolutions importantes, notamment l'enregistrement de différents établissements et mandataires. Le dernier événement notable de l'entreprise remonte au 28 décembre 2021.

Actuellement, l'entreprise est dirigée par Maxence Caillaud, son gérant, qui continue de piloter Coccinet dans ses projets et son développement dans le domaine des solutions informatiques.

3. Organigramme

L'organigramme de l'OMINET est représenté dans la figure ci-dessous :

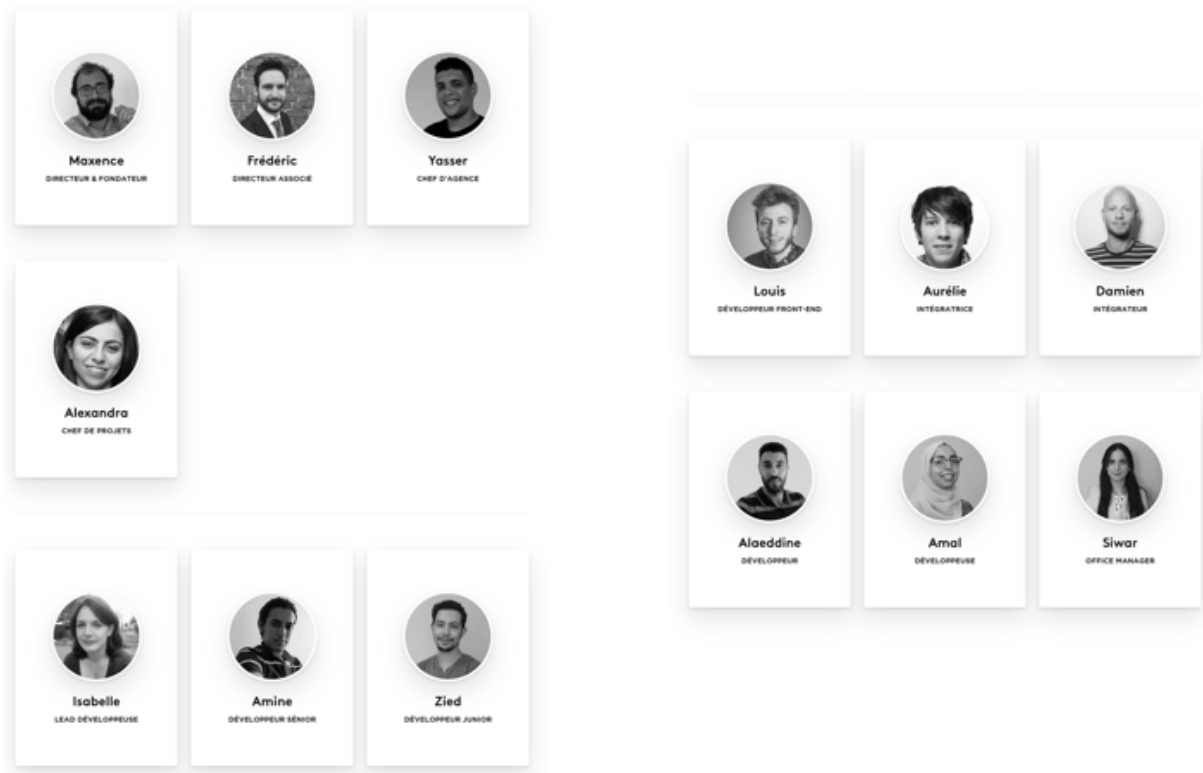


FIGURE 1.1 – L'organigramme de l'OMINET

II. Etude préalable

1. Critique de l'existant

En regardant de plus près, voici ce qui pose problème avec ces méthodes :

→ **Perte de temps et d'énergie :**

Les organisateurs et participants perdent beaucoup de temps à gérer ou chercher des informations. Et même avec les recherches, les résultats ne sont pas toujours fiables.

→ **Problème des places limitées :**

Quand un événement est complet, il n'y a pas de moyen simple pour gérer les gens en liste d'attente. Cela crée de la frustration, car des participants motivés passent à côté.

→ **Mauvaise communication :**

Sans système intégré, les annonces ou rappels ne sont pas toujours bien diffusés, ce qui complique la tâche pour tout le monde.

→ **Manque d'analyse :**

Il est difficile pour les organisateurs d'avoir une vue claire des statistiques, comme le nombre d'inscrits ou les retours des participants.

→ **Plateformes peu adaptées :**

Certaines solutions existantes ne sont pas faites pour de petites structures, soit parce qu'elles sont trop chères, soit parce qu'elles ne correspondent pas aux besoins.

2. Problématique :

Actuellement, les associations, clubs et autres communautés en Tunisie utilisent des méthodes classiques ou des outils simples pour gérer leurs événements, mais ces solutions ne répondent pas toujours à leurs besoins. Voici ce qu'on remarque :

1. **Méthodes traditionnelles :**

Les organisateurs gèrent souvent leurs inscriptions avec des fichiers Excel ou des listes papier. C'est simple mais vite inefficace quand il y a beaucoup de participants.

2. **Outils en ligne limités :**

Il existe des plateformes pour créer et gérer des événements, mais elles ne permettent pas forcément de gérer les inscriptions correctement, surtout quand les places sont limitées.

3. **Communication dispersée :**

Pour informer les participants, on passe par WhatsApp, des e-mails, ou des publications sur Facebook. C'est compliqué et parfois les gens ratent l'information importante.

4. **Peu de retours d'expérience :**

Les outils actuels n'aident pas vraiment à analyser les événements (comme voir combien de personnes se sont inscrites ou ce qui a bien fonctionné).

5. **Accessibilité :**

Certaines plateformes sont trop chères ou compliquées à utiliser pour les petites associations ou clubs.

3. Solution proposée

Notre plateforme apporte des solutions modernes pour simplifier la gestion des événements et répondre aux besoins des organisateurs :

★ **Inscriptions faciles :**

Une interface intuitive permet de gérer les inscriptions, les annulations, et les listes d'attente automatiquement.

★ **Communication rapide :**

Avec un système de messagerie intégré et des notifications push, les organisateurs peuvent rester connectés avec les participants en temps réel.

★ **Suivi clair :**

Un tableau de bord analytique fournit des statistiques sur les inscriptions et les retours des participants, aidant les organisateurs à mieux planifier.

★ **Accessibilité pour tous :**

Simple, abordable et adaptée même aux petites communautés, la plateforme est facile à prendre en main sans expertise technique.

★ **Gestion des places limitées :**

Les participants peuvent rejoindre une liste d'attente et être notifiés dès qu'une place est disponible.

III. Méthodologie adaptées

Dans cette section, nous examinons en détail les méthodes utilisées qui nous ont permis de progresser dans la réalisation de notre projet.

1. Méthodologie de modélisation et de conception

En utilise le méthodologie de modélisation et de conception pour mieux comprendre les besoins et planifier techniquement le projet .

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Son objectif est de concevoir et mettre en place des systèmes logiciels complexes en termes de structure et de comportement. Les applications de l'UML dépassent le domaine du développement logiciel, en particulier pour les flux de processus dans le secteur industriel.[1]

2. Methodologie de gestion de projet

Dans le domaine de developpement web , les projets informatiques deviennent de plus en plus complexes avec l'évolution rapide des technologies et des besoins , alors en va Développer une approche stratégique pour répondre au besoin du client car les anciennes méthodes de gestion rigides ne répondent plus à cette réalité . dans ce cas l'approche Agile s'est imposée comme une solution moderne et efficace.

La méthode agile est une méthode de gestion de projet. L'idée, lorsque l'on utilise cette approche, est d'apporter souplesse et performance à la gestion de projet. Centrée sur l'humain et la communication, elle permet aux clients de participer au développement d'un produit tout

au long de l'avancement du projet.[2] . Dans ce manière en utilise le méthode SCRUM pour ce projet

Scrum C'est la méthode de travail la plus répandue pour la gestion des projets agile et divise le travail dans un équipe .

Scrum est une structure Agile qui facilite la collaboration au sein des équipes et les aide à réaliser des tâches à haute valeur ajoutée. Elle propose un schéma de valeurs, rôles et directives pour leur permettre de se concentrer sur chaque itération et de s'améliorer en continu.[3]

Principes fondamentaux :

- **Collaboration :** Les équipes travaillent ensemble de manière collaborative.
- **Itération** Le travail est divisé en sprints (généralement de 2 à 4 semaines).
- **Adaptabilité** Les exigences peuvent évoluer en fonction des retours des utilisateurs.

2.1 Rôles dans Scrum :

Le methode scrum définit trois rôles principaux :

- ★ **Product Owner :** traduit les besoins du client en tâches concrètes pour l'équipe, tout en s'assurant que les priorités sont claires et alignées avec les objectifs du projet.
- ★ **Scrum Master :** il accompagne l'équipe au quotidien en veillant à ce que le cadre Scrum soit respecté, tout en aidant chacun à avancer malgré les blocages éventuels.
- ★ **Équipe de Développement :** qui transforme les idées en solutions concrètes, en collaborant activement pour livrer un produit fonctionnel et de qualité .

2.2 Artefacts de Scrum :

Les artefacts de scrum permet de de gérer le travail .

- ★ **Product Backlog :** Il contient toutes les idées, fonctionnalités et améliorations à venir. Il se met à jour au fur et à mesure que le projet avance, afin de s'assurer que les priorités restent alignées avec les besoins réels du client.
- ★ **Sprint Backlog :** C'est la liste des tâches que l'équipe doit accomplir en fonction de ses capacités et des priorités du moment, choisies directement à partir du Product Backlog
- ★ **Increment :** C'est un livrable qui doit être fonctionnel et prêt à être déployé, garantissant que chaque itération apporte une réelle valeur ajoutée au projet.

2.3 Processus Scrum :

Le Processus Scrum comporte plusieurs étapes.

- ★ **sprint meeting planning** : Est un moment où l'équipe trier les éléments du product backlog à réaliser durant le sprint. et définies les tâches spécifiques pour chaque élément sélectionné.
- ★ **Daily Scrum** : faire une réunion quotidienne de 15 minutes pour cadrer l'équipe et chaque membre partage son travail et présente les difficultés rencontrées.
- ★ **Le sprint review** : le Product Owner invite l'équipe Scrum et les parties prenantes, et l'équipe presente ce qu'il a fait. Les parties prenantes donnent leur avis, pour permet d'améliorer le produit et d'ajuster le backlog.
- ★ **Sprint Rétrospective** : c'est la dernier réunions ,on discute sur ce qui a bien fonctionné et ce qui peut être amélioré .

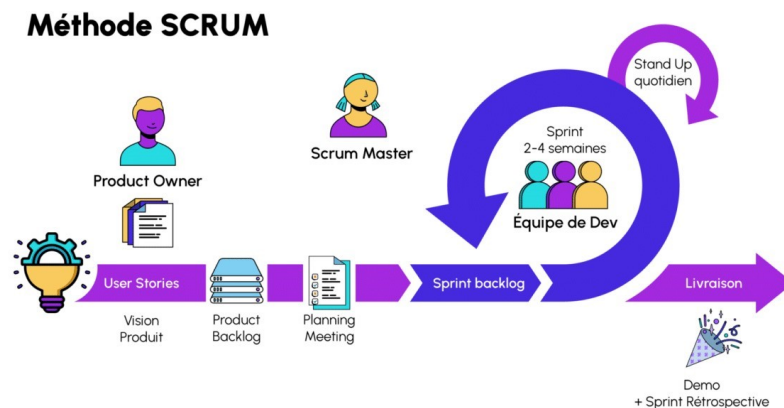


FIGURE 1.2 – Le processus de Scrum

Conclusion

Au cours de ce chapitre, nous avons présenté la société d'accueil « OMINET » ainsi que l'étude préalable, les méthodologies utilisées. Dans le prochain chapitre, notre objectif est la planification du projet

CHAPITRE 2

PLANIFICATION DU PROJET

Introduction

Dans ce chapitre, nous allons en premier lieu définir les acteurs en détaillant les besoins fonctionnels et non fonctionnels de notre application. En second lieu, nous allons présenter l'architecture de notre plateforme.

I. Spécification des besoins

1. Besoins fonctionnels

Notre application permet à Administrateur :

- Gérer les utilisateurs.
- Gérer les événements.
- Gérer les catégories.
- Consulter les statistiques.

Notre application permet à Gestionnaire des événements :

- Créer un compte.
- Mettre à jour son profil.
- Gérer les événements.
- Gérer les inscriptions.
- Envoyer des rappels.

- Accéder aux statistiques des événements.

Notre application permet à Utilisateur :

- Créer un compte.
- Mettre à jour son profil.
- Lister les événements.
- S'inscrire à un événement.
- Envoyer des messages pour demande d'information.
- Faire un commentaire.

2. Besoins non fonctionnels

Performance :

- La plateforme doit supporter jusqu'à 500 utilisateurs simultanés sans dégrader les performances.

Sécurité :

- Les mots de passe doivent être cryptés pour garantir leur sécurité.
- Des règles strictes d'accès aux données doivent être définies selon les rôles.
- La plateforme doit être protégée contre les attaques comme les injections SQL et XSS.

Fiabilité :

- La plateforme doit être disponible 99,9% du temps avec un minimum de temps d'arrêt.
- Les données doivent être sauvegardées quotidiennement pour éviter leur perte.

Scalabilité :

- L'architecture doit évoluer pour supporter une augmentation d'utilisateurs et d'événements.
- Il doit être facile d'ajouter de nouvelles fonctionnalités à la plateforme.

Accessibilité :

- La plateforme doit respecter les normes d’accessibilité (WCAG 2.1).
- Elle doit être compatible avec les lecteurs d’écran et utiliser des couleurs contrastées.

Maintenabilité :

- Le code doit être bien structuré, documenté et facile à modifier.
- Des tests automatisés doivent être utilisés pour détecter rapidement les erreurs.

Portabilité :

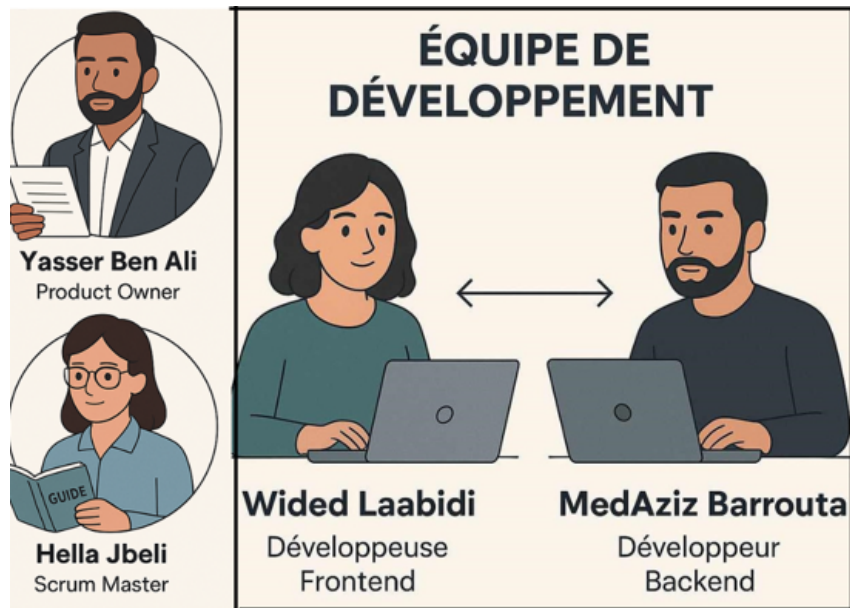
- La plateforme doit être compatible avec les navigateurs principaux (Chrome, Firefox, Edge, Safari).
- L’hébergement doit se faire sur des serveurs cloud (AWS, Azure).

II. Gestion de projet avec Scrum

1. Équipe SCRUM et rôles

SCRUM définit trois rôles, comme reflété dans la figure ci-dessous :

- Product Owner : Mr. Yasser Ben Ali
- Scrum Master : Mme. Hella Jebali
- L’équipe de développement : Wided Laabidi et Mohamed Aziz Barrouta

**FIGURE 2.1** – L'équipe Scrum

2. Product Backlog

C'est le composant le plus basique dans le processus Scrum. Nous l'avons utilisé pour Planifier la réalisation de chaque sprint. Donc, il contient des fonctionnalités requises pour la construction d'un produit, ainsi que tous les éléments nécessitant L'intervention de l'équipe. Ces éléments sont classés par ordre de priorité et de Leur dépendance, ce qui permet de spécifier l'ordre de leur réalisation.

ID	Sprint	User Story	Priorité	Complexité
2	Authentification et gestion des utilisateurs	En tant qu'utilisateur, je souhaite créer un compte.	Élevée	Moyenne
		En tant qu'utilisateur, je souhaite m'authentifier..	Élevée	Moyenne
		En tant que gestionnaire, je souhaite créer un compte.	Élevée	Moyenne
		En tant que gestionnaire, je souhaite m'authentifier...	Élevée	Moyenne
		En tant qu'administrateur, je veux créer un utilisateur pour la gestion des utilisateurs.	Élevée	Moyenne
		En tant qu'administrateur, je veux modifier un utilisateur pour la gestion des utilisateurs.	Élevée	Moyenne
		En tant qu'administrateur, je veux supprimer un utilisateur pour la gestion des utilisateurs.	Élevée	Moyenne
		En tant qu'administrateur, je veux consulter un utilisateur pour la gestion des utilisateurs.	Élevée	Moyenne
3	Gestion des événements	En tant qu'administrateur, je veux modifier des événements.	Élevée	Moyenne
		En tant qu'administrateur, je veux supprimer des événements.	Élevée	Moyenne
		En tant qu'administrateur, je veux consulter des événements.	Élevée	Moyenne
		En tant qu'administrateur, je veux approuver des événements.	Élevée	Moyenne
		En tant que gestionnaire, je veux créer des événements.	Élevée	Moyenne
		En tant que gestionnaire, je veux modifier des événements.	Élevée	Moyenne
		En tant que gestionnaire, je veux supprimer des événements.	Élevée	Moyenne

4	Inscription et interaction utilisateur	En tant qu'utilisateur, je veux m'inscrire à un événement.	Élevée	Moyenne
		En tant qu'utilisateur, je veux recevoir une notification après inscription.	Élevée	Moyenne
		En tant qu'utilisateur, je veux laisser un feedback après participation.	Élevée	Moyenne
5	Gestion avancée et tableaux de bord	En tant qu'utilisateur, je veux consulter les statistiques globales.	Élevée	Moyenne
		En tant que gestionnaire, je veux accéder aux statistiques des événements.	Élevée	Moyenne

TABLE 2.1 – Product Backlog

3. Planification de sprint

La préparation des sprints revêt une importance capitale dans la concrétisation des projets SCRUM. Nous diviserons notre projet en 4 releases qui seront constituées de 6 sprints. La figure suivante illustre la division des releases en sprints.

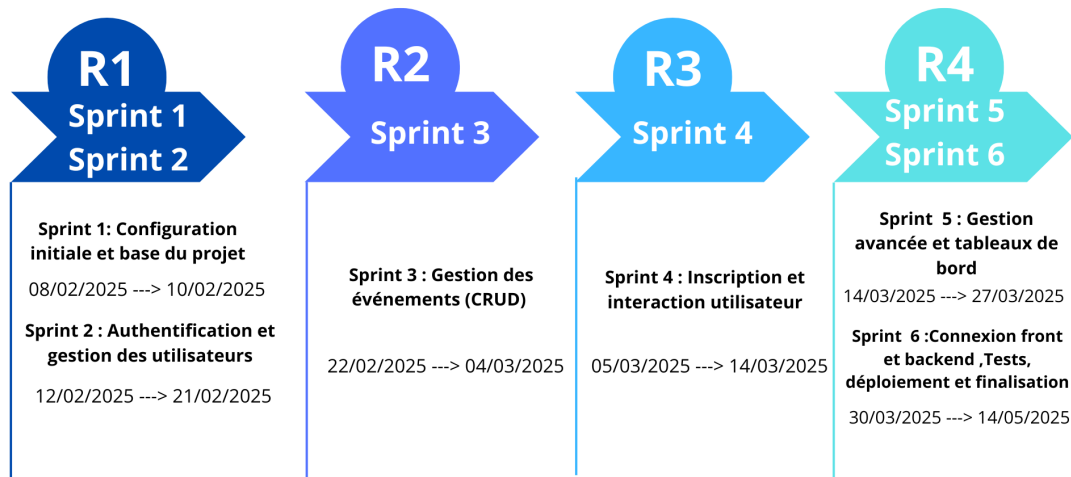



FIGURE 2.2 – Planification des sprints

III. Diagrammes de cas d'utilisations général

1. Identification des acteurs

Dans cette section, nous définissons les acteurs du système ainsi que leurs rôles respectifs. Le tableau 2.2 ci-dessous récapitule ces informations.

Acteur	Rôle	Fonctionnalités/Services
 Administrateur	Gérer et administrer la plateforme	<ul style="list-style-type: none"> — Gérer les utilisateurs. — Gérer les événements. — Consulter les statistiques.



 Utilisateur	Participer aux événements	<ul style="list-style-type: none"> — S’inscrire aux événements. — Envoyer des messages. — Poster des commentaires.
 Gestionnaire des événements	Organiser et gérer les événements	<ul style="list-style-type: none"> — Créer et modifier des événements. — Gérer les inscriptions. — Suivre les statistiques des événements.

TABLE 2.2 – Identification des acteurs et de leurs fonctionnalités

2. Diagramme de contexte statique

Ce diagramme UML montre la relation des différents acteurs avec le système.

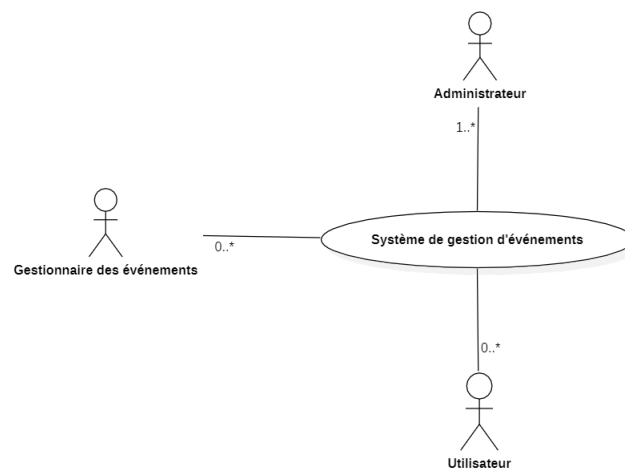


FIGURE 2.3 – Diagramme de contexte statique

3. Diagramme de cas d'utilisation global

Voici le diagramme de cas d'utilisation global de notre application :

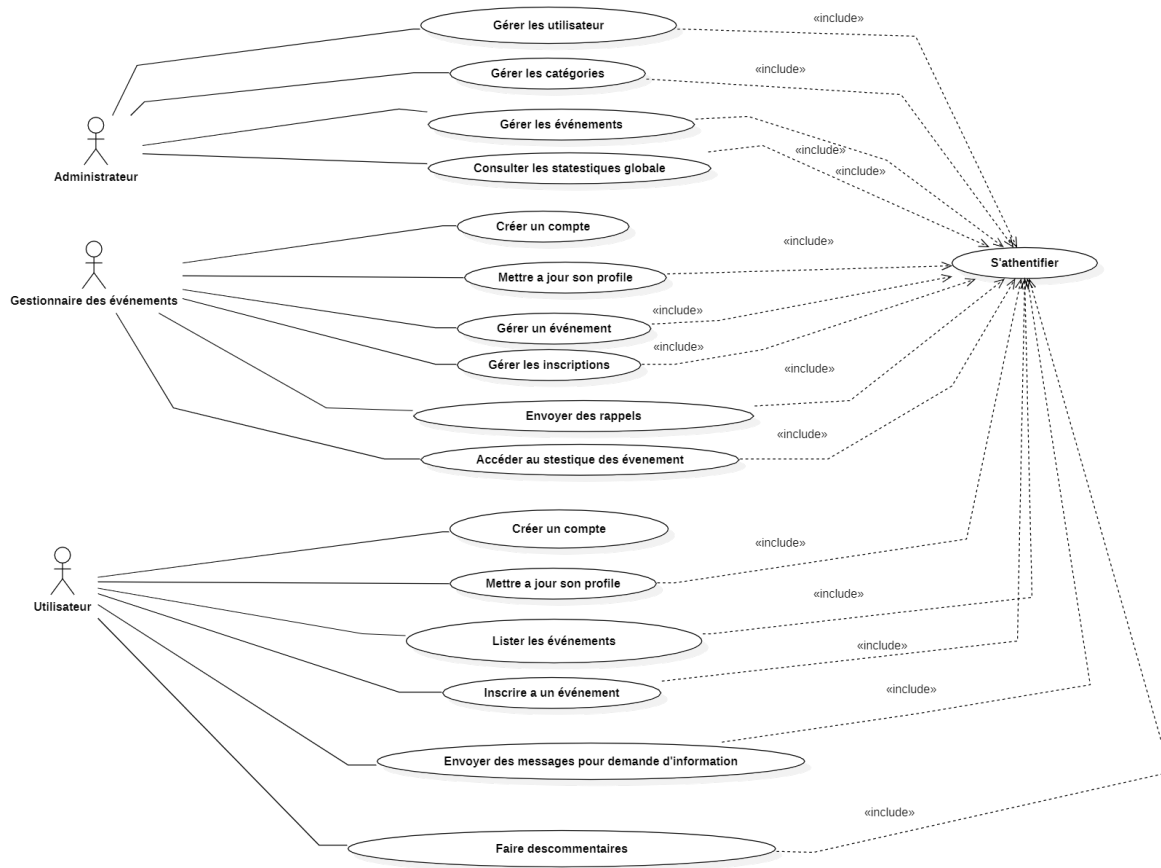


FIGURE 2.4 – Diagramme de cas d'utilisation globale

IV. Diagramme de classes général

Ce diagramme UML représente la structure de données de notre application.

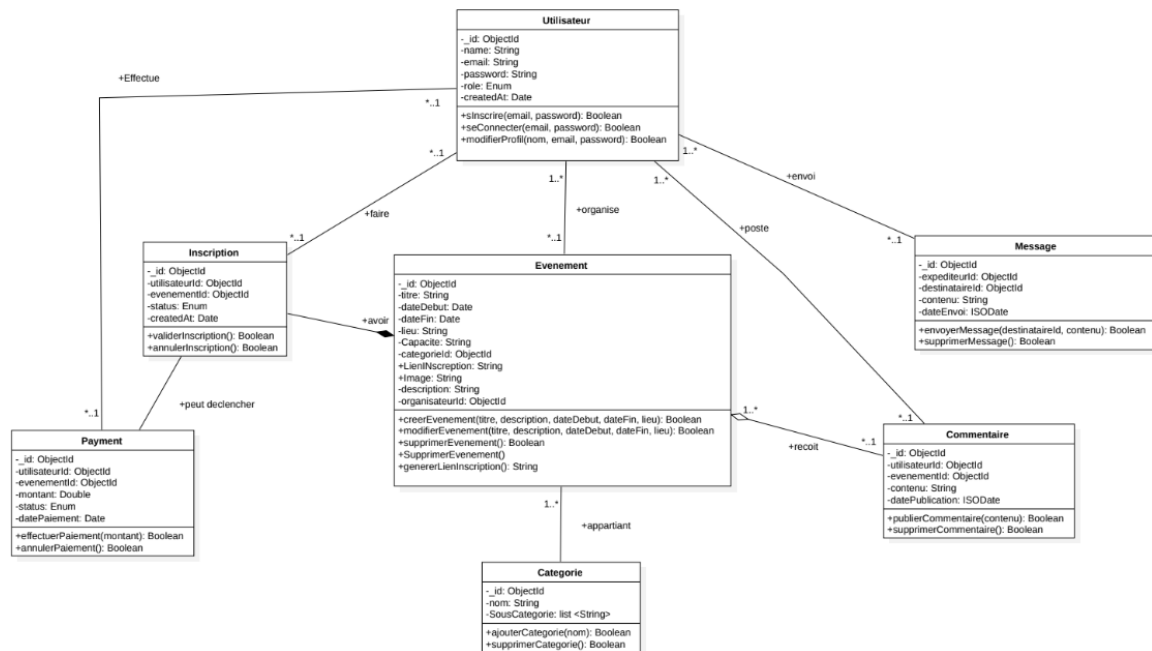


FIGURE 2.5 – Diagramme de classes général de l'application

V. Environnement de développement et choix techniques

Dans cette section, nous présentons l'environnement matériel ainsi que l'environnement logiciel utilisés pour implémenter la solution informatique.

1. Environnement matériel

Nous détaillons ici les caractéristiques matérielles, notamment le processeur, la mémoire vive (RAM), le disque dur et le système d'exploitation de chaque ordinateur utilisé. Toutes les informations sont récapitulées dans le Tableau 2.3.

Ordinateur	1	2
Propriétaire	Laabidi Wided	Barroua Mouhamed Aziz
Processeur	Intel Core i5	AMD Ryzen 5
RAM	16 Go	8 Go
Disque Dur	476 Go SSD	512 Go SSD
Système d'exploitation	Windows 10 (64 bits)	Windows 11 (64 bits)

TABLE 2.3 – Caractéristiques matérielles des ordinateurs utilisés

2. Environnement logiciel

Dans cette sous-section, nous décrivons les différents logiciels informatiques que nous avons utilisé pour mener à terme notre projet.



Visual Studio Code (VS Code).

Est un éditeur de code source et un environnement de développement intégré (IDE) de Microsoft. Il est open-source et cross-platform, c'est-à-dire qu'il fonctionne sur Windows, Linux et Mac. Il a été conçu pour les développeurs web, mais il prend en charge de nombreux autres langages de programmation tels que C++, C, Python, Java, etc.



Postman.

Est une plateforme API tout-en-un pour la création et l'utilisation d'API. Elle simplifie chaque étape du cycle de vie des API : de la conception et des tests à la livraison et à la surveillance. [5]



Node Js.

Est un environnement d'exécution JavaScript gratuit, open source et multiplateforme qui permet aux développeurs de créer des serveurs, des applications Web, des outils de ligne de commande et des scripts. [6]



GitHub.

Est une plateforme open source de gestion de versions et de collaboration destinée aux développeurs de logiciels. [7]



Overleaf.

Est un éditeur LaTeX en ligne et collaboratif. Il inclut un environnement LaTeX complet, prêt à l'emploi et permet de produire des documents scientifiques de haute qualité. [8]



Canva.

Lancé en 2013, Canva est un outil de design et de communication visuelle en ligne dont la mission est de permettre à tout le monde de concevoir et de publier selon ses envies.[27]

3. Les langages de programmation

HTML



HTML5.

Signifie « HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure[9]

CSS



CSS.

Le CSS pour Cascading Style Sheets est un langage informatique utilisé sur Internet pour la mise en forme de fichiers et de pages HTML.[10]



JavaScript

JavaScript.

Est un langage de script qui vous permet de créer du contenu mis à jour dynamiquement, de contrôler le multimédia, d'animer des images. [11]

**LaTeX.**

Est un système logiciel de composition de documents créé par Leslie Lamport. Plus exactement, il s'agit d'une collection de macro-commandes destinées à faciliter l'utilisation du « processeur de texte » TeX de Donald Knuth. Il a été créé en 1985. [12]

4. Framework**Tailwind-CSS.**

Est un framework CSS open source. La fonctionnalité principale de cette bibliothèque est, contrairement à d'autres frameworks CSS comme Bootstrap, qu'elle ne procure pas une série de classes prédéfinies pour des éléments tels que des boutons ou des tables.[13]

**Express.Js.**

Est le framework backend le plus populaire pour Node.js, et il fait partie intégrante de l'écosystème JavaScript. Il est conçu pour construire des applications web monopages, multipages et hybrides, il est également devenu la norme pour le développement d'applications backend avec Node.js, et il constitue la partie backend de ce que l'on appelle la pile MEVN. [14]

5. Le système de gestion de base de données**Mongodb.**

Est une base de données orientée documents. En clair, vous bénéficiez de la scalabilité et de la flexibilité que vous voulez, avec les fonctions d'interrogation et d'indexation qu'il vous faut. [15]

6. Les bibliothèques et les outils**React.JS.**

Est une bibliothèque JavaScript utilisée pour construire des interfaces utilisateur. Chaque application web React est composée de composants réutilisables qui constituent des parties de l'interface utilisateur.[16]

**Vite.React.**

Vite (mot français pour « rapide », prononcé/vit/, comme « veet »), est un outil de développement visant à accélérer et simplifier le développement des projets web modernes.[24]

<Recharts />

Recharts.Js.

Est Une bibliothèque de graphiques composables construite sur des composants React.Créez rapidement vos graphiques avec des composants React découplés et réutilisables. [17]

**React Router.**

Est un package npm permettant d'implémenter le routage dynamique dans une application web. Il permet d'afficher des pages et de permettre aux utilisateurs de les parcourir. Il s'agit d'une bibliothèque de routage complète côté client et serveur pour React. [18]

**Axios.**

Axios, client HTTP basé sur les promesses, fonctionne de manière identique dans node.js et les navigateurs. [19]

**TanStack Query.**

Est une bibliothèque de gestion de l'état du serveur dans les applications React, permettant une gestion efficace des données asynchrones comme les requêtes API. [20]

**Material-ui.**

Est une bibliothèque de composants React open-source qui met en œuvre le design Material de Google. Il est complet et peut être utilisé dans la production à partir du carton. [21]

**lucide-react.**

Est une bibliothèque d'icônes open source proposant plus de 1 000 fichiers vectoriels (SVG) pour l'affichage d'icônes et de symboles dans des projets numériques et non numériques. [22]

**JSON Web Token.**

Est un standard ouvert qui permet un échange sécurisé de tokens entre différentes parties. La sécurité de l'échange se traduit par la vérification de l'intégrité et de l'authenticité des données. [23]

**AOS Animate On Scroll.**

Est une bibliothèque open source d'animation par défilement créée par Michał Sajnog. Elle a été créée pour optimiser les performances en utilisant uniquement du CSS pour les animations et en réservant JavaScript à la gestion de la logique. [25]

**React-hot-toast .**

Est une bibliothèque de notifications légère et open source pour React. Comme les autres bibliothèques React Toast, elle est conçue pour imiter les notifications push popularisées par les systèmes d'exploitation natifs, tels qu'iOS et Android, dans les applications web [26]

VI. Architecture adoptée

1. Architecture logique :

Avant d'entamer la conception et le développement de tout système informatisé, il est primordial d'élaborer son architecture. Dans le cadre de notre projet, nous avons choisi MVC

(modèle-Vue-Contrôleur) en tant que patron d'architecture logicielle. Ce dernier permet d'organiser globalement une interface graphique. En d'autres termes il permet de bien séparer le code de l'interface graphique de la logique applicative. D'ailleurs, c'est un choix populaire pour la conception d'application web. En général ce modèle permet de distinguer trois entités qui sont :

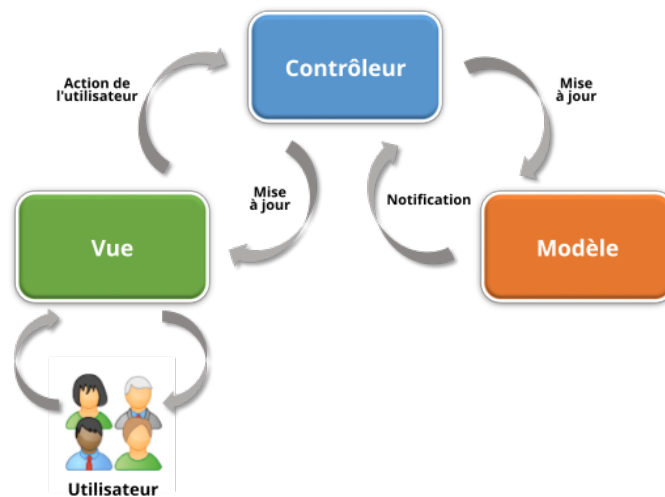


FIGURE 2.6 – L'architecture de Model-View-Controller

- **Modèle :**

contient les données manipulées par le programme. En effet il peut s'agir d'un ensemble de fonctions (Modèle procédural) ou de classes (Modèle orienté objet).

- **Vue :**

Fait l'interface avec l'utilisateur puisque elle donne plusieurs vues, elle peut aussi présenter la possibilité à l'utilisateur de changer de vue..

- **Contrôleur :**

Un contrôleur contient la logique concernant les actions effectuées par l'utilisateur..

2. Architecture physique

Pour assurer de bonnes performances, nous avons choisi une architecture **client-serveur**.

Cette architecture est illustrée dans le schéma ci-dessous :

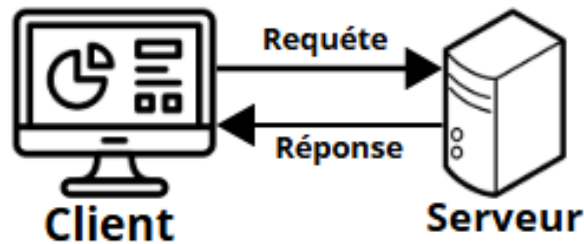


FIGURE 2.7 – L’architecture client-serveur

- **Le client** : que ce soit actif ou bien esclave lance la communication avec le serveur en adressant des demandes ou bien des requête
- **Le serveur** : Considéré comme passif ou maître, il répond aux demandes du client

Grâce à cette séparation, chaque partie peut effectuer son travail de manière efficace et améliorer le système. plus rapide et plus organisé

Conclusion

Dans ce chapitre, nous avons présenté la planification de notre travail Alors nous avons dégagé le premier artéfact le backlog de produit qui comporte les tous les liste des fonctionnalités de notre système, Par la suite nous avons dégagé ainsi les différents rôles de l’équipe dans le projet. Enfin nous avons choisi le concept de découpage pour la réalisation des sprints qui vont suivre, Nous allons enchaîner à présent avec notre premier sprint dans le chapitre qui suit.

CHAPITRE 3

RELEASE 1

Introduction

Après avoir exposé en détail les exigences de notre projet à travers un backlog produit, nous entamons dans ce chapitre la première version du projet, qui comprend deux sprints : le sprint 1 et le sprint 2. Chaque sprint couvre l'analyse, la conception et la réalisation.

I. Organisation du sprint

Notre release est composée de deux sprints :

- **Sprint 1** : Configuration initiale et base du projet.
- **Sprint 2** : Authentification et gestion des utilisateurs.

II. Sprint 1 : Configuration initiale et base du projet

1. Objectif du sprint

L'objectif de ce sprint est de mettre en place l'environnement MERN, la configuration backend et frontend. Les tâches principales sont :

- l'initialisation du dépôt Git ;
- la création du serveur Express avec une route de test ;

- la modélisation de la base de données MongoDB ;
- la configuration de React.js et des routes principales.

III. Sprint 2 : Authentification et gestion des utilisateurs

1. Objectif du sprint

L'objectif de ce sprint est de mettre en place un système d'authentification par JWT et une gestion des rôles (participant, gestionnaire, administrateur). **Tâches principales :**

- Modélisation Utilisateur avec rôle et état de compte ;
- Routes d'inscription, d'authentification et gestion de session (JWT) ;
- Interface React pour l'inscription, la connexion et le feedback d'erreur.

2. Sprint backlog

Période : 12 février – 21 février

TABLE 3.1 – User Stories – Sprint 2 : Authentification & gestion des utilisateurs

ID	Sprint	User Story	Priorité	Complexité
2	Authentification et gestion des utilisateurs	En tant que participant, je souhaite créer un compte.	Élevée	Moyenne
		En tant que participant, je souhaite me connecter.	Élevée	Moyenne
		En tant que gestionnaire, je souhaite obtenir un compte après validation par l'administrateur.	Élevée	Moyenne
		En tant que gestionnaire, je souhaite me connecter.	Élevée	Moyenne
		En tant qu'administrateur, je veux créer un utilisateur.	Élevée	Moyenne
		En tant qu'administrateur, je veux modifier un utilisateur.	Élevée	Moyenne
		En tant qu'administrateur, je veux supprimer un utilisateur.	Élevée	Moyenne
		En tant qu'administrateur, je veux consulter la liste des utilisateurs.	Élevée	Moyenne

3. Implémentation

3.1 Politique de création de comptes

Seuls les **participants** peuvent créer un compte via le formulaire. Les rôles gestionnaire et admin sont créés uniquement par un administrateur qui doit :

- recevoir une demande de création (email officiel) ;
- vérifier la légitimité du gestionnaire ;
- créer manuellement le compte et notifier le gestionnaire.

3.2 Spécification des besoins

On détaille ici trois types de cas d'utilisation : – Inscription / Authentification (participant & gestionnaire) – Gestion de comptes (administrateur) Chaque cas est accompagné de son diagramme et de sa description textuelle.

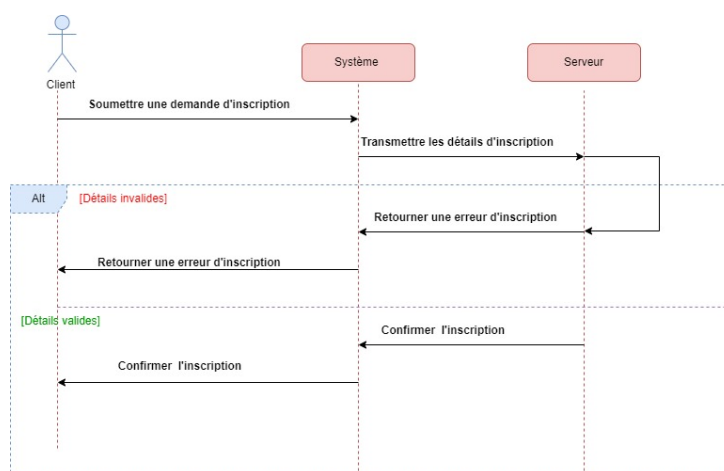


FIGURE 3.1 – Cas d'utilisation « S'inscrire / S'authentifier » (Participant)

Diagramme et description : Inscription & Authentification

Cas d'utilisation	S'inscrire (participant)
Acteur	Participant
Pré-condition	Le formulaire d'inscription est accessible
Post-condition	Compte créé, session ouverte

Scénario principal	<ol style="list-style-type: none"> 1. Affichage du formulaire d'inscription ; 2. Saisie des informations ; 3. Soumission et validation ; 4. Création du compte et feedback de succès.
Scénario alternatif	Email déjà utilisé ou données invalides → message d'erreur

TABLE 3.2 – Inscription participant

Cas d'utilisation	S'authentifier (participant & gestionnaire)
Acteur	Participant / Gestionnaire
Pré-condition	Le compte existe
Post-condition	Token JWT renvoyé, redirection selon rôle
Scénario principal	<ol style="list-style-type: none"> 1. Affichage du formulaire de connexion ; 2. Saisie des identifiants ; 3. Soumission, vérification et génération de JWT ; 4. Stockage du JWT (cookie HTTPOnly).
Scénario alternatif	Identifiants invalides → message d'erreur

TABLE 3.3 – Authentification participant / gestionnaire

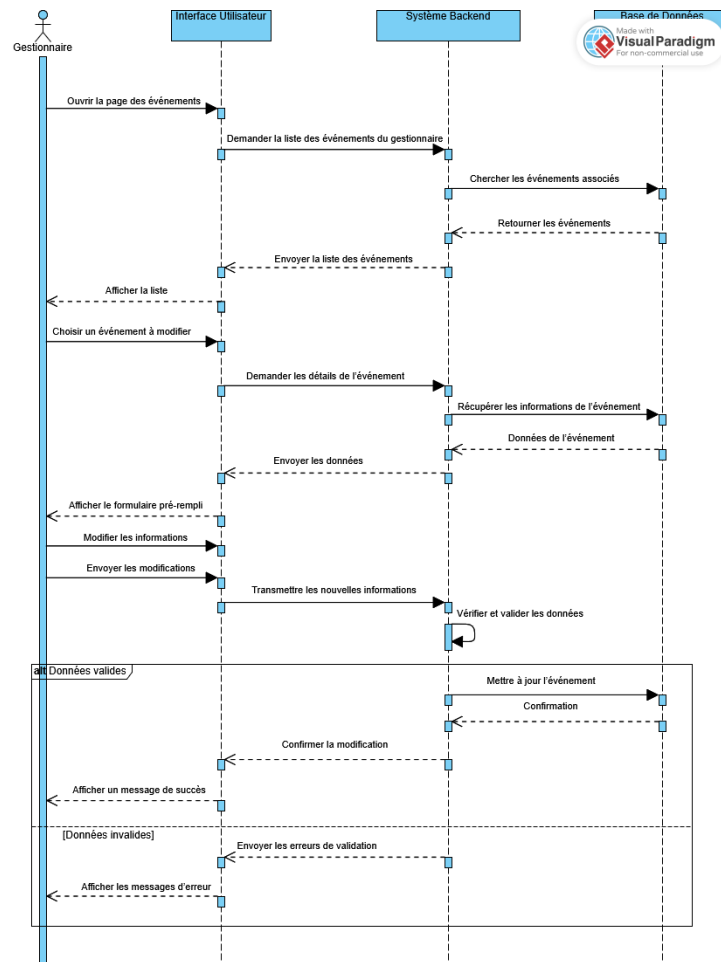


FIGURE 3.2 – Cas d'utilisation « Gérer les utilisateurs » (Admin)

Diagramme et description : Gestion des utilisateurs

Cas d'utilisation	Gérer les utilisateurs
Acteur	Administrateur
Pré-condition	Admin connecté
Post-condition	Opération CRUD réalisée
Scénario principal	<ol style="list-style-type: none"> 1. Affichage du tableau des utilisateurs ; 2. Choix d'une action (Créer, Modifier, Supprimer, Consulter) ; 3. Exécution de l'opération et feedback.
Scénario alternatif	Erreur de validation ou doublon email → message d'erreur

TABLE 3.4 – Gestion des utilisateurs (Admin)

3.3 Diagrammes de séquence systèmes

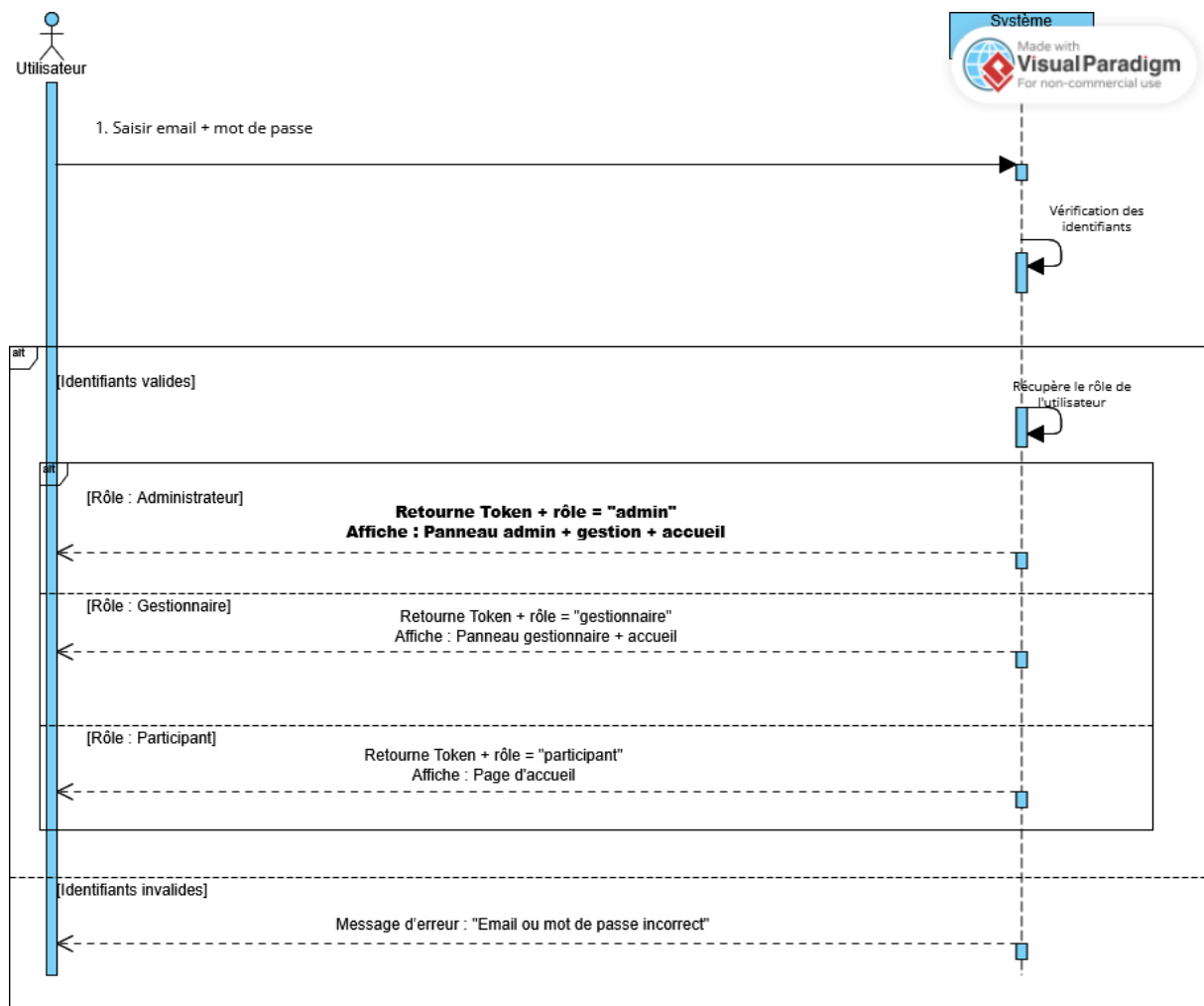


FIGURE 3.3 – Séquence « Authentication / JWT »

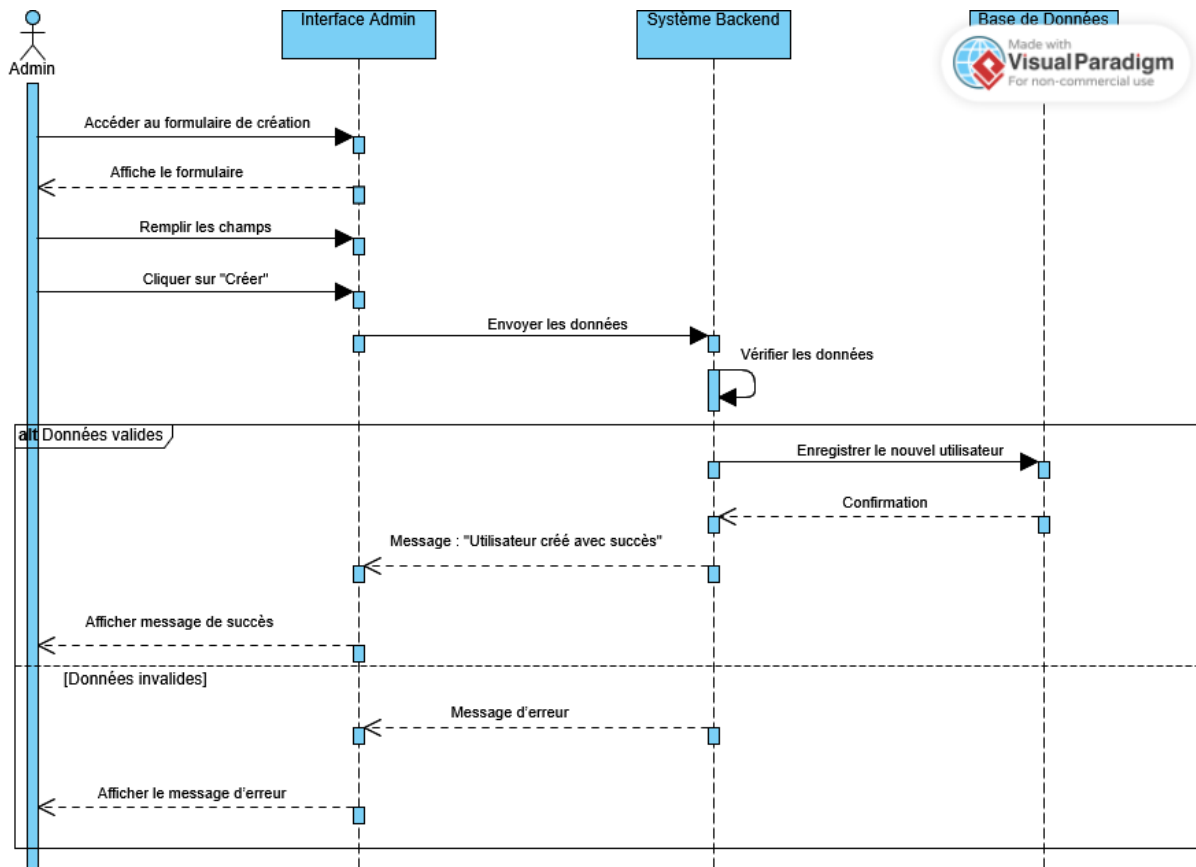


FIGURE 3.4 – Séquence « CRUD Utilisateurs » (Admin)

3.4 Diagramme de classe (à compléter)

3.5 Réalisation

Formulaires (inscription / connexion) et **tableau d'administration** implémentés en React, avec gestion des erreurs et feedback visuel.

Backend : routes Express sécurisées, génération et vérification de JWT, middleware rôles.

Frontend : React + Zustand pour l'état d'authentification, Axios pour les requêtes.

CHAPITRE 4

RELEASE 2

Introduction

Dans ce chapitre, nous explorerons en détail le deuxième release de notre projet, qui contient la troisième sprint.

I. Organisation du sprint

Notre release est composée par :

- **Sprint 3** : Gestion des événements (CRUD).

II. Sprint 3 : Gestion des événements (CRUD)

1. Objectif du sprint

L'objectif de ce sprint est la création, modification, suppression et affichage des événements. Les tâches principales sont :

- Implémentation des routes backend pour événements
- Création du modèle événement
- Validation des champs d'événements
- Affichage dynamique des événements

2. Sprint backlog

Le 3ème sprint s'étend du 22 février au 04 mars. Le tableau suivant représente le backlog de ce sprint :

ID	Sprint	User Story	Priorité	Complexité
3	Gestion des événements	En tant qu'administrateur, je veux modifier des événements.	Élevée	Moyenne
3	Gestion des événements	En tant qu'administrateur, je veux supprimer des événements.	Élevée	Moyenne
3	Gestion des événements	En tant qu'administrateur, je veux consulter des événements.	Élevée	Moyenne
3	Gestion des événements	En tant qu'administrateur, je veux approuver des événements.	Élevée	Moyenne
3	Gestion des événements	En tant que gestionnaire, je veux créer des événements.	Élevée	Moyenne
3	Gestion des événements	En tant que gestionnaire, je veux modifier des événements.	Élevée	Moyenne
3	Gestion des événements	En tant que gestionnaire, je veux supprimer des événements.	Élevée	Moyenne

TABLE 4.1 – User Stories - Sprint Gestion des événements

3. Implémentation du sprint 3

3.1 Spécification des besoins

Dans cette section, nous identifions les besoins de notre sprint, à travers :

- les diagrammes de cas d'utilisation,
- les descriptions textuelles associées,
- les diagrammes de séquences système.

La figure ci-dessous représente le diagramme de cas d'utilisation de ce sprint par rapport à l'administrateur.

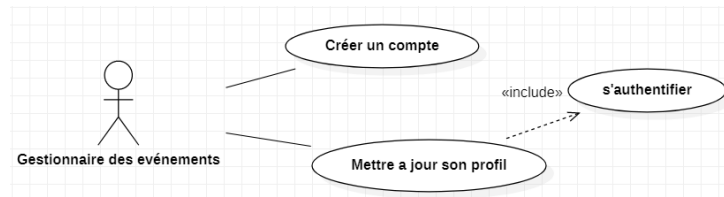


FIGURE 4.1 – Diagramme des cas d'utilisation gestion d'événement "Administrateur"

Ce diagramme décrit le processus de gestion d'événement pour l'administrateur. Nous détaillons ci-après ce cas d'utilisation sous forme textuelle :

CONCLUSION GÉNÉRALE

BIBLIOGRAPHIE

- [1] <https://www.lucidchart.com/pages/tutorial/uml>
- [2] <https://slack.com/intl/fr-fr/blog/collaboration/methode-agile>
- [3] <https://asana.com/fr/resources/what-is-scrum>.
- [4] <https://bility.fr/definition-visual-studio-code/>
- [5] <https://www.postman.com/product/what-is-postman/>
- [6] <https://nodejs.org/en/>
- [7] <https://www.lemagit.fr/definition/GitHub>
- [8] <https://scosi.univ-littoral.fr/assistance/catalogue-de-services/overleaf/#:~:text=Overleaf%20est%20un%20%C3%A9diteur%20LaTeX,documents%20scientifiques%20de%20hautes%20qualit%C3%A9s>.
- [9] <https://developer.mozilla.org/fr/docs/Web/HTML>
- [10] <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [11] <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [12] <https://www.techno-science.net/glossaire-definition/LaTeX.html>
- [13] https://fr.wikipedia.org/wiki/Tailwind_CSS
- [14] <https://kinsta.com/fr/base-de-connaissances/qu-est-express-js/>
- [15] <https://www.mongodb.com/fr-fr/company/what-is-mongodb>
- [16] <https://kinsta.com/fr/base-de-connaissances/qu-est-react-js/>
- [17] <https://recharts.org/en-US>
- [18] <https://www.geeksforgeeks.org/what-is-react-router-dom/>
- [19] <https://axios-http.com/docs/intro>
- [20] <https://medium.com/@ignatovich.dm/tanstack-query-a-powerful-tool-for-data-managem>
- [21] <https://mui.com/>
- [22] <https://lucide.dev/guide/>

- [23] <https://jwt.io/introduction>
- [24] <https://vite.dev/guide/>
- [25] <https://shanelonergan.github.io/animate-on-scroll-react/>
- [26] <https://refine.dev/blog/react-hot-toast/#what-is-react-hot-toast>
- [27] https://www.canva.com/fr_fr/about/