



PROJET DE TECHNOLOGIE OBJET

Rapport

*ROBBANA Myriam DADOUN Ambre LÉCUYER Simon
FRYDMAN Alexandre EL AALOUCHE Youssef BAUX Hugo*

*Encadrants :
Crégut Xavier
Ouederni Meriem
François Gilles*

*Groupe EF04
1A Sciences du Numérique
ENSEEIH*

2024

Table des matières

1	Introduction	2
2	Principales fonctionnalités et objectifs	2
2.1	Elevator Pitch	2
2.2	Charte produit	2
2.3	User Stories	3
3	Architecture de l'application	4
3.1	Diagrammes de classe	4
3.2	Diagrammes Scénario	8
4	Choix de conception et réalisation	9
4.1	Organisation de l'équipe et mise en œuvre des méthodes agiles	9
4.2	Problèmes rencontrés et solutions apportées	10
5	Conclusion	10

1 Introduction

Ce rapport présente en détail le développement d'une plateforme de gestion d'événements divers, réalisée dans le cadre du projet de Technologie Objet. Cette plateforme vise à simplifier l'organisation et la participation à des événements de toutes sortes, en offrant une interface conviviale et des fonctionnalités adaptées aux besoins variés des utilisateurs. À travers ce rapport, nous détaillerons les principales fonctionnalités de notre application, son architecture, les choix de conception et de réalisation, ainsi que l'organisation de notre équipe de développement et l'application des méthodes agiles.

2 Principales fonctionnalités et objectifs

2.1 Elevator Pitch

- **Pour :** Tout le monde
- **Qui souhaitent :** Organiser ou assister à des évènements
- **Notre produit est :** Une plateforme de gestion d'événements divers
- **Qui :** Centralise et simplifie la gestion d'événement et l'achat de places
- **À la différence de :** Les autres gestionnaires d'événements classiques
- **Permet de :** Gérer de manière ergonomique et surtout universelle les évènements

2.2 Charte produit

Charte produit	
Vision produit	Notre vision est de révolutionner la gestion d'événements en proposant une plateforme unique qui simplifie l'organisation et la participation. Nous voulons être le partenaire de confiance pour tous ceux qui organisent ou assistent à des événements, en offrant une expérience utilisateur intuitive et des fonctionnalités adaptées à tous les besoins.
Clients/Utilisateurs	<p>Les clients/utilisateurs de notre plateforme de gestion d'événements divers sont :</p> <ol style="list-style-type: none">1. Clients : Tout individu intéressé par la participation à des événements de divers types, allant des amateurs de concerts et de spectacles aux professionnels assistant à des conférences et des salons professionnels.2. Organisateurs d'évènements : Les entreprises ou les individus qui organisent et vendent des places pour des événements à travers notre plateforme.3. Fournisseurs de services événementiels : Les prestataires de services tels que les traiteurs, les fournisseurs de matériel, les entreprises de location d'équipements, etc., qui peuvent utiliser la plateforme pour se connecter aux organisateurs d'événements et proposer leurs services.4. Sponsors et partenaires : Les entreprises ou les marques souhaitant sponsoriser des événements ou établir des partenariats avec des organisateurs d'événements pour accroître leur visibilité et leur engagement auprès du public cible.

Valeur Métier	<p>La valeur métier ajoutée de notre plateforme de gestion d'événements divers réside dans les avantages suivants pour nos utilisateurs :</p> <ol style="list-style-type: none"> 1. Centralisation et simplification : Nous offrons une plateforme unique où les organisateurs peuvent gérer tous les aspects de leurs événements, de la planification à la vente de billets, en passant par la promotion et la gestion des participants. Cela simplifie considérablement le processus, économisant ainsi du temps et des ressources. 2. Facilité d'utilisation : Notre interface conviviale et intuitive permet à tous les utilisateurs, qu'ils soient organisateurs ou participants, de naviguer facilement sur la plateforme et d'accéder aux informations pertinentes sans effort. 3. Universalité : Notre plateforme est conçue pour gérer une grande variété d'événements, des petites réunions aux grands festivals, et pour répondre aux besoins divers des organisateurs et des participants, offrant ainsi une solution adaptée à toutes les situations. 4. Visibilité accrue : Les organisateurs d'événements bénéficient d'une visibilité accrue grâce à notre plateforme, ce qui peut entraîner une plus grande participation et un plus grand succès pour leurs événements. 5. Connectivité : Notre plateforme favorise la connexion entre organisateurs, participants, vendeurs de billets, fournisseurs de services et sponsors, créant ainsi un écosystème dynamique où toutes les parties prenantes peuvent interagir et collaborer efficacement. 6. Analyse et suivi : Nous offrons des outils d'analyse avancés qui permettent aux organisateurs d'événements de suivre les performances de leurs événements, de comprendre le comportement des participants et d'ajuster leur stratégie en conséquence pour des résultats optimaux.
Critères de succès (métriques)	<p>Les critères de succès de notre plateforme incluent :</p> <ol style="list-style-type: none"> 1. Interface visuelle satisfaisante : Une interface utilisateur attrayante et intuitive qui améliore l'expérience globale des utilisateurs et facilite la navigation sur la plateforme. 2. Gestion de Comptes : Fonctionnalités robustes de gestion des comptes permettant aux utilisateurs de créer, modifier et gérer facilement leurs profils et leurs informations. 3. Gestion d'événements : Outils complets pour la création, la gestion et la promotion des événements, incluant la vente de billets, la communication avec les participants et le suivi des performances.

2.3 User Stories

Gérant

1. En tant que gérant, je veux pouvoir accéder à la page de création d'un événement après avoir saisi mes informations personnelles (Nom, Prénom, Date de naissance, Solde Bancaire) afin de créer un nouvel événement. (Pas fini)
2. En tant que gérant, je veux pouvoir créer un nouvel événement en spécifiant son nom, sa date, son emplacement, sa catégorie, le prix des tickets et le nombre maximal de places achetées. (Pas fini)
3. En tant que gérant, je veux pouvoir accéder à la page de gestion des participants pour un événement créé, afin de voir la liste des participants et effectuer des actions telles que la suppression ou l'ajout de participants. (Pas fini)

4. En tant que gérant, je veux pouvoir supprimer un participant spécifique de la liste des participants pour un événement donné. (Pas fini)
5. En tant que gérant, je veux pouvoir ajouter un nouveau participant à la liste des participants pour un événement donné. (Pas fini)

Client

1. En tant que client, je veux pouvoir m'inscrire en fournissant mes informations personnelles (Nom, Prénom, Age, Solde Bancaire) afin de pouvoir acheter des tickets pour des événements. (Itération 1)
2. En tant que client, je veux pouvoir rechercher des événements par nom et lieu dans l'onglet Détails Événement de l'accueil client. (Pas fini)
3. En tant que client, je veux pouvoir sélectionner des tickets pour un événement spécifique dans l'onglet Billetterie, en précisant le nombre de tickets et le type de forfait. (Itération 3)
4. En tant que client, je veux pouvoir visualiser le récapitulatif de mon panier dans l'onglet Mon Panier, incluant le nombre de places, les détails de l'événement, le prix total et mon solde actuel. (Itération 2)
5. En tant que client, je veux pouvoir finaliser mon achat en confirmant le paiement, en entrant un code promo facultatif et en validant mon achat, afin d'obtenir mes tickets. (Pas fini)
6. En tant que client, je veux que mon achat soit refusé si je n'ai pas suffisamment d'argent sur mon solde. (Itération 2)
7. En tant que client, je veux que mon inscription soit refusée si j'utilise des informations déjà présentes dans la base de données, et être dirigé vers une connexion ou une suggestion de changement de nom. (Itération 2)
8. En tant que client, je veux que mes tentatives d'achat de plusieurs places sur un événement à places limitées soient limitées à une place maximum par personne. (Itération 3)
9. En tant que client, je veux que mes tentatives d'achat de places excédant le nombre de places disponibles pour un événement soient refusées. (Itération 2)

3 Architecture de l'application

3.1 Diagrammes de classe

Nous avons dans les premières phases du développement envisagé une architecture comme celle-ci :

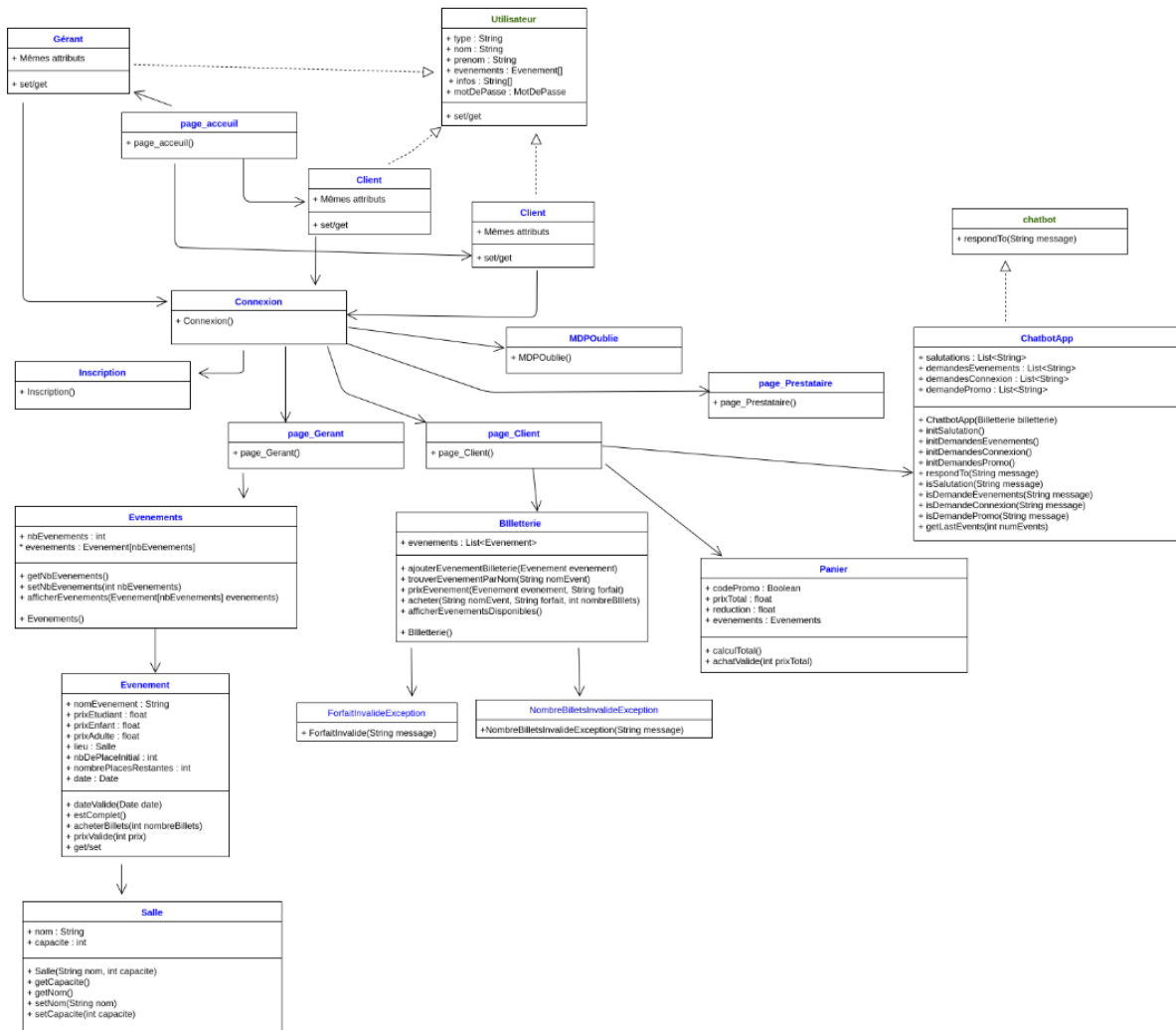


FIGURE 1 – Premier diagramme UML

Cependant, suite à des difficultés rencontrées en cours de développement et une plus ample réflexion, cette architecture a évolué.

TABLE 2 – Classe Billet

Classe Billet	Représente un billet pour un évènement
Attributs privés :	Méthodes publiques :
nomEvenement : String dateEvenement : Date Lieu : Lieu numéroSiège : int prixBillet : Float	getNomEvenement() : String setNomEvenement(nomEvenement : String) : void getDateEvenement() : Date setDateEvenement(dateEvenement : Date) : void getLieu() : Lieu setLieu(lieu : Lieu) : void getNumeroSiege() : int setNumeroSiege(numeroSiege : int) : void getPrixBillet() : Float

TABLE 3 – Classe Billetterie

Classe Billetterie	Représente la billetterie de l'événement
Attributs privés :	Méthodes publiques :
evenements : List<Evenement>	ajouterEvenementBilletterie(Evenement evenement) : void trouverEvenementParNom(String nomEvent) : Evenement prixEvenement(Evenement evenement, String forfait) : Float acheter(String nomEvent, Client client, int nombreBillets) : void afficherEvenementsDisponibles() : void getEvenements() : List<Evenement>

TABLE 4 – Classe MesPrestations

Classe MesPrestations	Représente les prestations disponibles
Attributs privés :	Méthodes publiques :
prestations : List<Prestation>	ajouterPrestation(Prestation prestation) : void trouverPrestationParNom(String nomPrestation) : Prestation afficherPrestationsDisponibles() : void getPrestations() : List<Prestation>

TABLE 5 – Classe MesTickets

Classe MesTickets	Représente les tickets achetés par un client
Attributs privés :	Méthodes publiques :
billets : List<Billet>	ajouterBillet(Billet billet) : void ToString() : String getBillets() : List<Billet> getClient() : Client

TABLE 6 – Classe Prestation

Classe Prestation	Représente une prestation liée à un événement
Attributs privés :	Méthodes publiques :
nomEvenement : String dateEvenement : Date lieu : Lieu prixPrestation : Float	setNomEvenement(String nomEvenement) : void getNomEvenement() : String getDateEvenement() : Date setDateEvenement(Date dateEvenement) : void getLieu() : Lieu setLieu(lieu : Lieu) : void getPrixPrestation() : Float

TABLE 7 – Classe Salle

Classe Salle	Représente une salle où se déroule un événement
Attributs privés :	Méthodes publiques :
nom : String type : String description : String capacité : int disponibilités : boolean[] prixDeLocation : int[] adresse : String ville : String code Postale : String pays : String placesPrises : String[]	Les mêmes que celles de la classe Lieu

TABLE 8 – Les Exceptions

Exceptions	Description
AgeInvalideException	Exception levée en cas d'âge invalide lors de l'achat de billets
ForfaitInvalideException	Exception levée en cas de forfait invalide lors de l'achat de billets
NombreBilletsInvalideException	Exception levée en cas de nombre de billets invalide lors de l'achat

TABLE 9 – Les liens entre les classes

Liens	Description
Composition : Billet - MesTickets	Relation de composition entre les billets et les tickets du client, chaque ticket contient un ou plusieurs billets.
Association : Billetterie - Événement	Lien d'association entre la billetterie et les événements proposés, permettant à la billetterie de gérer les événements disponibles.
Utilisation : Billetterie - NombreBilletsInvalideException, AgeInvalideException	La billetterie utilise ces exceptions pour gérer les erreurs lors de l'achat de billets.
Association : MesPrestations - Prestataire	Association entre les prestations et les prestataires, permettant à chaque prestataire de proposer des prestations spécifiques.
Association : MesPrestations - Prestation	Relation entre les prestations et la liste des prestations disponibles, chaque prestation est répertoriée dans la liste des prestations.
Association : MesTickets - Client	Lien entre les tickets et les clients, chaque ticket est associé à un client qui l'a acheté.
Héritage : Salle - Lieu	Relation d'héritage entre les salles et les lieux, permettant aux salles de partager les propriétés et méthodes définies dans la classe Lieu.

3.2 Diagrammes Scénario

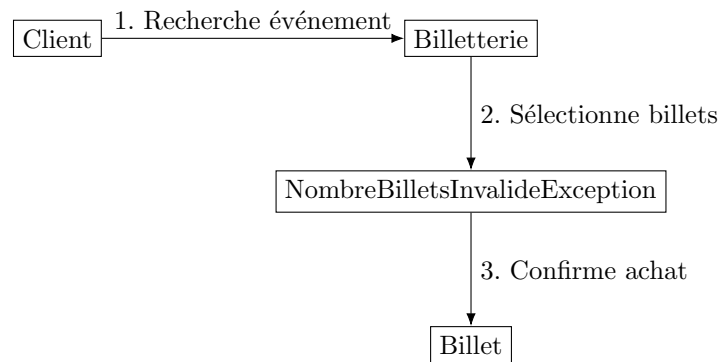


FIGURE 2 – Diagramme de séquence : Acheter un billet (Client → Billetterie)

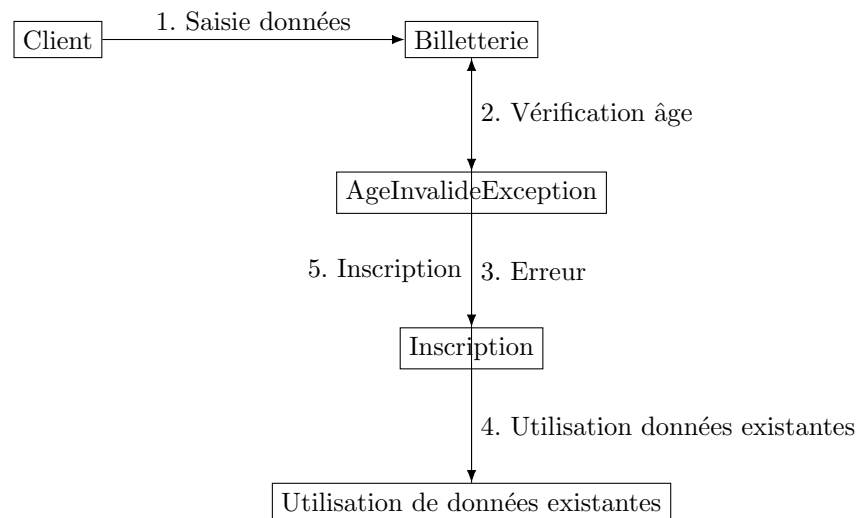


FIGURE 3 – Diagramme de séquence : Inscription (Client → Billetterie)

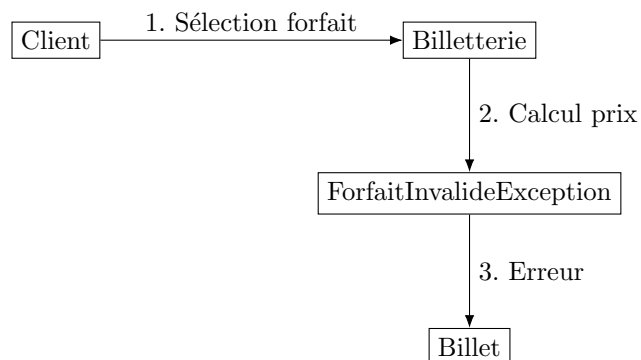


FIGURE 4 – Diagramme de séquence : Achat billet avec forfait invalide (Client → Billetterie)

4 Choix de conception et réalisation

4.1 Organisation de l'équipe et mise en œuvre des méthodes agiles

Dans le cadre du développement de notre projet, nous avons adopté le framework de développement "Scrum", qui est une méthode agile bien établie dans le domaine du développement logiciel. Cette approche met l'accent sur la flexibilité, la communication et la collaboration au sein de l'équipe pour livrer des produits de manière itérative et incrémentielle.

Dans notre équipe, nous avons défini des rôles clés conformément à la méthodologie Scrum. Myriam Robbana a été désignée en tant que Product Owner, assumant la responsabilité de définir les exigences du projet, de prioriser les fonctionnalités à développer et de s'assurer que le produit final répond aux besoins des utilisateurs.

Simon Lécuyer a initialement occupé le rôle de Scrum Master, chargé de faciliter le processus de développement agile en mettant en place l'infrastructure technique nécessaire, telle que la configuration de GitHub pour le suivi des versions et la gestion du code source, ainsi que la création d'un espace de travail collaboratif sur Discord pour la communication et la coordination entre les membres de l'équipe. Cependant, en raison d'une période d'indisponibilité, Ambre Dadoun a repris ce rôle par la suite, se concentrant davantage sur la gestion et la coordination managériales de l'équipe tout au long du développement du projet.

Cette répartition des rôles a permis une mise en œuvre efficace de la méthode agile, avec une Product Owner déterminant les objectifs du projet, un Scrum Master facilitant le processus de développement, et une équipe de développement travaillant en collaboration pour réaliser les fonctionnalités définies dans chaque itération. Grâce à cette approche agile et à la collaboration étroite entre les membres de l'équipe, nous avons pu progresser de manière significative dans le développement de notre projet tout en répondant aux besoins changeants et en assurant la qualité du produit final.

Les membres de l'équipe ont également pris en charge des aspects spécifiques du développement :

- **Hugo Baux** : Mise en place des fonctionnalités d'inscription, de connexion et de gestion des profils utilisateurs.
- **Ambre Dadoun *Scrum Master*** : Gestion managériale du projet et développement des fonctionnalités liées à l'inscription et à la connexion des utilisateurs.
- **Youssef El Aalouche** : Développement des pages d'accueil et de gestion des rôles utilisateur.
- **Alexandre Frydman** : Développement du ChatBot pour la communication avec les clients et des interfaces des boutons clients.
- **Simon Lécuyer** : Mise en place des outils de collaboration et développement de l'interface utilisateur et rédaction du rapport.
- **Myriam Robbana *Product Owner*** : Définition des besoins du projet, conception de l'architecture logicielle et développement des fonctionnalités de gestion des événements et des billets, rédaction du manuel utilisateur.

Dans l'ensemble, chaque membre actif de l'équipe a joué un rôle essentiel dans le développement du projet, contribuant à sa progression grâce à son expertise, son dévouement et sa collaboration active.

Initialement, notre stratégie de gestion du projet reposait sur l'utilisation de GitHub, offrant des fonctionnalités avancées telles que les projets et les organisations, pour le suivi du développement, la gestion des tâches et la collaboration sur le code source. GitHub était un choix naturel en raison de sa popularité, permettant une traçabilité complète des changements apportés au code, une répartition claire des tâches et une collaboration efficace entre les membres de l'équipe. Cependant, en ce qui concerne la communication interne de l'équipe, nous avons initialement opté pour Discord en raison de ses fonctionnalités de messagerie instantanée et de voix, ainsi que de la facilité de création de salons thématiques pour différents sujets de discussion. Cependant, au fil du temps, l'usage organique a préféré un autre outil : WhatsApp. Bien que WhatsApp soit moins adapté pour l'échange de fichiers et la gestion des discussions à long terme, il s'est avéré être l'outil de communication le plus utilisé par l'équipe, offrant une accessibilité et une réactivité supérieures.

Les outils proposés par GitHub s'intègrent parfaitement avec l'utilisation de méthodes agiles, en effet, ils permettent une gestion transparente et efficace du flux de travail. La fonctionnalité des projets GitHub offre une vue d'ensemble claire des tâches en cours, de leur état d'avancement et des responsabilités de chacun au sein de l'équipe. De plus, la gestion des problèmes (issues) et des pull requests sur GitHub facilite la collaboration en permettant aux membres de l'équipe de signaler des problèmes, de proposer des solutions et de discuter des changements de manière

structurée et traçable. Ces outils favorisent la communication et la coordination au sein de l'équipe, permettant une réactivité rapide aux changements et une adaptation fluide aux exigences du projet, éléments essentiels des méthodes agiles.

4.2 Problèmes rencontrés et solutions apportées

Durant le développement du projet, nous avons rencontré divers défis, notamment en termes d'organisation de l'équipe. Un aspect particulièrement délicat a été la contribution inattendue de certains membres. Malgré les assurances initiales de participation active de tous les membres, nous avons constaté des disparités significatives dans la charge de travail. Cette situation a mis en évidence les défis organisationnels auxquels nous avons dû faire face et a nécessité une réévaluation constante de nos ressources et de nos priorités pour assurer la progression du projet dans les délais impartis.

De plus, au cours du développement du projet, nous avons été confrontés à des demandes changeantes et à des contraintes imprévues (prévisibles mais imprévues dues à notre inexpérience), qui ont nécessité une adaptation constante de l'architecture du programme. Cette évolution dynamique a souvent entraîné la nécessité de repenser en partie la structure du code pour répondre aux nouveaux besoins. Ce qui a entraîné des retards et des complications dans le processus de développement. Cette nécessité de réadapter de grandes parties du code a mis en lumière l'importance d'une conception flexible et modulaire, ainsi que d'une documentation claire et précise pour faciliter les modifications ultérieures.

Un autre défi auquel nous avons été confrontés a été l'utilisation non systématique de la fonctionnalité de projet et d'organisation sur GitHub. Initialement envisagée comme un outil de suivi et de gestion de projet, cette fonctionnalité a rapidement montré ses limites. En effet, elle est rapidement devenue redondante avec l'utilisation de Hudson pour la gestion du code source. Cette redondance a causé des confusions et des problèmes, en particulier lorsque plusieurs membres de l'équipe travaillaient simultanément sur le code. La synchronisation des tâches entre GitHub et Hudson n'était pas optimale, ce qui a entraîné des retards et des erreurs dans le développement. L'utilisation du repository de Github n'a donc pas été forcément idéale mais l'utilisation des backlogs et des fonctionnalités agiles l'a été. Cette expérience a souligné la nécessité d'une utilisation plus rigoureuse et cohérente des outils de gestion de projet pour assurer une collaboration efficace et harmonieuse au sein de l'équipe.

5 Conclusion

En conclusion, notre projet de développement de la plateforme de gestion d'événements divers a été une expérience enrichissante qui nous a permis de mettre en pratique nos compétences en matière de conception logicielle, de programmation et de gestion de projet. Grâce à l'application des méthodes agiles et à une collaboration étroite au sein de notre équipe, nous avons réussi à surmonter les obstacles rencontrés et à livrer une application fonctionnelle répondant aux besoins de nos utilisateurs.

Au cours de ce projet, nous avons appris l'importance de la flexibilité et de l'adaptabilité dans le processus de développement logiciel, ainsi que l'importance de la communication et de la collaboration pour assurer le succès d'un projet. Nous avons également pris conscience des défis et des compromis inhérents à la conception et à la réalisation d'une application logicielle, et nous avons acquis de précieuses compétences qui nous seront utiles dans nos futures carrières.