



# Projet de course sportive

Epreuve E5 BTS SIO SLAM

Réalisé par Léa Giron  
01/03/2025

## Table des matières

1) Introduction.....	4
1.1) Plan.....	4
2) Cahier des charges.....	4
2.1) Présentation du projet .....	4
2.2) Analyse des besoins.....	5
2.3) Description des fonctionnalités .....	5
2.4) Contraintes.....	5
2.5) Architecture technique.....	5
2.6) Planification du projet .....	6
2.7) Critères de validation.....	7
2.8) Evolution .....	8
3) Modélisation de la base de données .....	8
3.1) Le dictionnaire de données .....	8
3.2) Modèle conceptuel de données et modèle physique de données.....	8
3.3) Diagramme de cas d'utilisation et diagramme de séquence .....	10
4) Conception de la base de données.....	14
4.1) Insérer des données de test .....	16
4.2) Tester les relations entre les tables .....	16
5) Développement.....	17
5.1) Présentation de Git.....	18
5.2) La page d'accueil (index).....	19
5.3) La page d'inscription.....	20
5.4) La page de connexion des organisateurs .....	22
5.5) Le tableau de bord .....	24
5.6) La page pour accepter un participant .....	25
5.7) La page d'annulation d'une inscription.....	26
5.8) La page de modification des informations.....	26
5.9) La page pour restaurer une inscription .....	27
5.10) La page des participants .....	27
6) Tests.....	28
6.1) Tests d'interface .....	28
6.2) Tests fonctionnels .....	29
6.3) Tests de base de données .....	29
6.4) Tests d'erreurs .....	29

6.4.1) [ERR-06] [Tableau de bord - Modifier les informations] [Modifier le champ "email" en email valide] [Message d'erreur "L'email est invalide (format email.test@domaine.fr attendu)"]	29
6.4.2) [ERR-08] [Tableau de bord - Modifier les informations] [Saisir un numéro de téléphone invalide] [Message d'erreur "Le numéro de téléphone doit contenir exactement 10 chiffres."]	31
6.4.3) [ERR-09] [Tableau de bord - Modifier les informations] [Entrer " ' OR '1'='1 " dans le champ "nom"] [Message d'erreur "Le nom ne peut contenir que des lettres et des espaces."]	32
6.4.4) [ERR-10] [Inscription] [Saisir une adresse email invalide] [Message d'erreur "L'email est invalide (format email.test@domaine.fr attendu)"]	33
6.4.5) [ERR-11] [Inscription] [Saisir un email déjà utilisé] [Message d'erreur " Cet email est déjà utilisé "]	34
6.4.6) [ERR-12] [Inscription] [Saisir un numéro de téléphone invalide] [Message d'erreur "Le numéro de téléphone doit contenir exactement 10 chiffres"]	34
6.4.7) [ERR-14] [Inscription] [Entrer " ' OR '1'='1 " dans le champ "nom"] [Message d'erreur "Le nom ne peut contenir que des lettres et des espaces."]	35
6.5) Tests de compatibilité des navigateurs	36
7) Documentation technique v1.0	36
1) Introduction	36
1.1) Contexte du projet	36
1.2) Objectifs de la documentation	37
2) Sécurité et conformité RGPD	37
3) Architecture et conception	37
3.1) Technologies utilisées	37
3.2) Modélisation et diagrammes	37
4) Structure du projet	43
5) Prérequis	44
6) Tests	44
6.1) Stratégie de tests	44
6.2) Convention de nommage des tests	45
7) Evolutions	45
7.1) Futures évolutions	45
8) Documentation utilisateur v1.0	46
1) Introduction	46
1.1) Présentation de l'application	46
2) Accès à l'application web	46
2.1) Accès	46

3) Guide de l'utilisateur .....	46
3.1) Interface globale .....	46
3.2) Fonctionnalités détaillées .....	46
4) Dépannage.....	51
4.1) Problèmes fréquents et solutions .....	51
4.2) Contact pour le support.....	52

## 1) Introduction

### 1.1) Plan

Pour ce projet de création d'une plateforme de gestion d'une course à pied, je vais commencer par présenter brièvement le projet. Ensuite, je vais rédiger le cahier des charges, puis avec la modélisation et la conception de la base de données. J'expliquerais le développement du projet ainsi que les tests effectués. Enfin, je présenterais la documentation technique et utilisateur.

## 2) Cahier des charges

### 2.1) Présentation du projet

- **Contexte**

Une commune souhaite organiser une course à pied en solo proposant plusieurs distances à parcourir : 5km, 10km et 15km. Jusqu'à présent, l'événement se déroulait sans plateforme en ligne. L'organisation de la course était compliquée car les inscriptions étaient gérées manuellement ce qui entraînait des erreurs et des confusions. L'événement manquait également de visibilité et une expérience utilisateur limitée pour les participants, qui ne pouvaient pas s'inscrire en ligne, consulter les informations sur la course ou vérifier leurs inscriptions.

De plus, cette plateforme contenant des informations personnelles sur les participants de la course, devra respecter les normes sur la protection des données personnelles (RGPD). Il sera donc essentiel de garantir un stockage sécurisé et de respecter le droit des participants qui pourront demander l'accès, la rectification et la suppression de leurs données.

- **Objectifs**

La plateforme doit permettre aux participants de s'inscrire en ligne, d'obtenir des informations sur la course et aux organisateurs de gérer l'événement.

L'objectif est de simplifier l'organisation de la course en passant par une interface intuitive et fonctionnelle.

## 2.2) Analyse des besoins

La phase d'analyse des besoins dans un cahier des charges permet d'identifier les différents types d'utilisateurs du système et de définir les besoins fonctionnels. C'est-à-dire ce que le système doit faire pour satisfaire les besoins des utilisateurs. Puis, définir les besoins non fonctionnels comme la performance ou la sécurité du système.

D'abord, il y aura deux types d'acteurs dans ce système, qui sont les organisateurs de la course et les participants. Les organisateurs doivent pouvoir se connecter pour modifier les informations des participants, accepter ou annuler l'inscription d'un participant et visualiser l'ensemble des inscriptions à l'aide d'un tableau de bord. Un participant doit pouvoir s'inscrire à la course, contacter un organisateur par téléphone, visualiser l'ensemble des participants inscrits à la course.

De plus, la plateforme devra être sécurisée afin de pouvoir se conformer au Règlement Général sur la Protection des Données (RGPD) car le système enregistre des informations personnelles sur les données des participants et des organisateurs.

La plateforme sera performante et ergonomique pour que les acteurs aient accès à une plateforme facile à utiliser.

## 2.3) Description des fonctionnalités

Les fonctionnalités principales de la plateforme sont le formulaire d'inscription en ligne simple et rapide qui permet aux participants de s'inscrire à l'une des trois distances proposées (5km, 10km ou 15km) et en fournissant leurs informations personnelles. Une page descriptive de l'événement détaillant les informations essentielles de la course telles que les horaires, les consignes pratiques, des informations de contact. Les utilisateurs peuvent également voir la liste de tous les coureurs inscrits avec leur nom et leur distance choisie. Enfin, les organisateurs peuvent se connecter et accéder à un tableau de bord permettant de gérer les inscriptions, modifier les informations des participants, accepter, annuler ou restaurer l'inscription d'un participant.

## 2.4) Contraintes

Plusieurs contraintes sont à respecter, d'abord, le système doit être accessible via une application web. De plus, la plateforme doit être compatible avec différents navigateurs. En ce qui concerne les contraintes réglementaires, la plateforme doit respecter le Règlement Général sur la Protection des Données (RGPD). L'accès doit être géré pour que seuls les organisateurs aient accès au tableau de bord et donc modifier certaines informations. En termes de sécurité, les données doivent être protégées contre les attaques courantes telles que les injections SQL. L'interface doit être intuitive et facile à utiliser. Pour le budget, les technologies sont open source. Enfin, le respect des échéances précisées dans le planning doit être respecté pour livrer le projet avant le mois d'avril.

## 2.5) Architecture technique

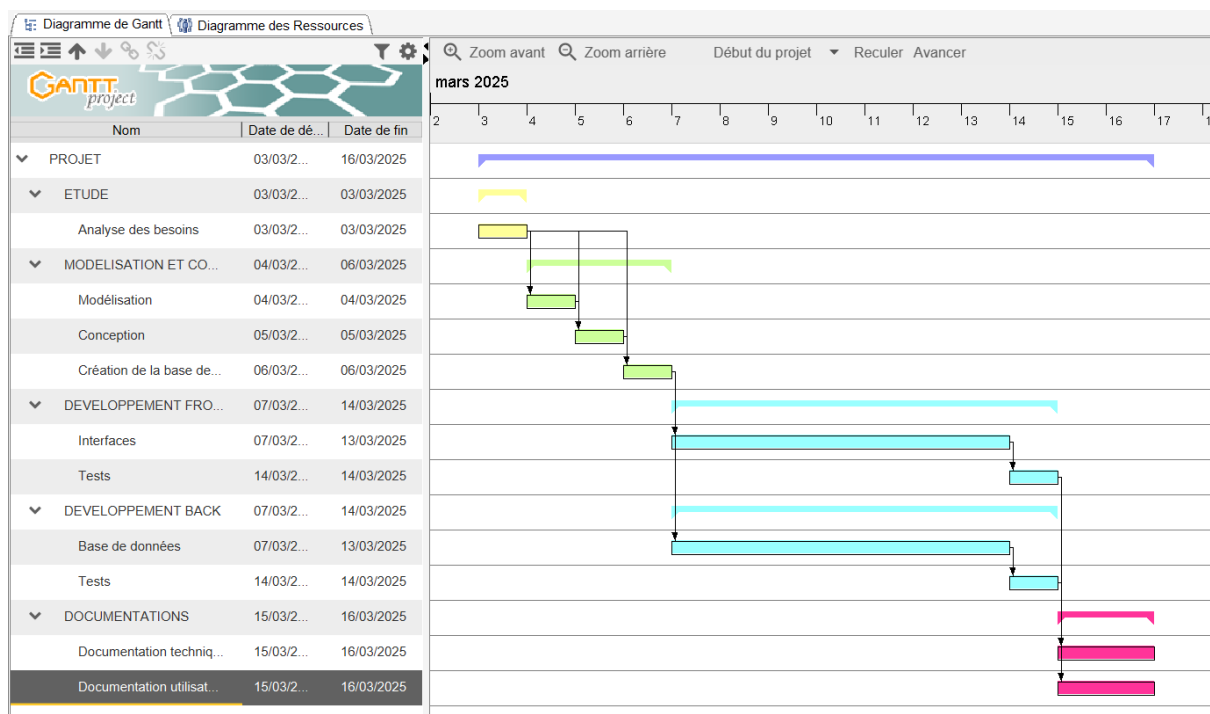
La plateforme est composée d'une architecture client-serveur avec une partie front-end pour l'interface utilisateur et d'une partie back-end pour le traitement des données.

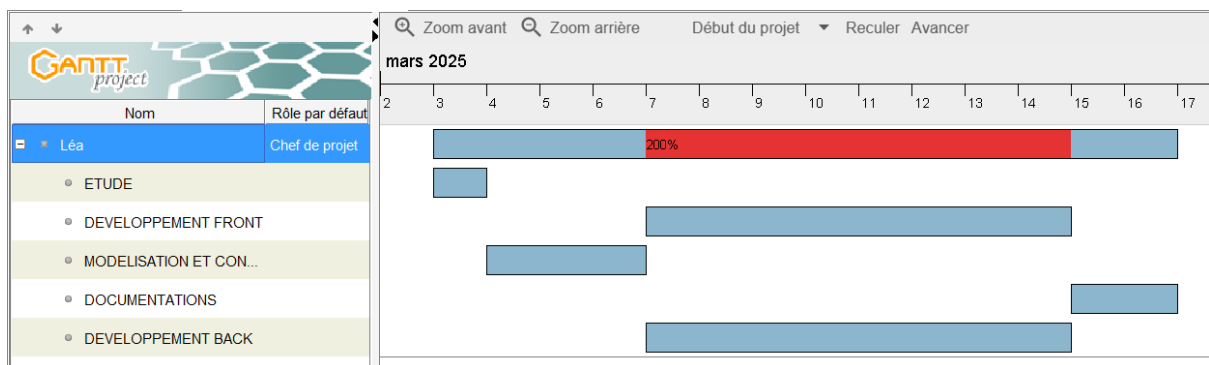
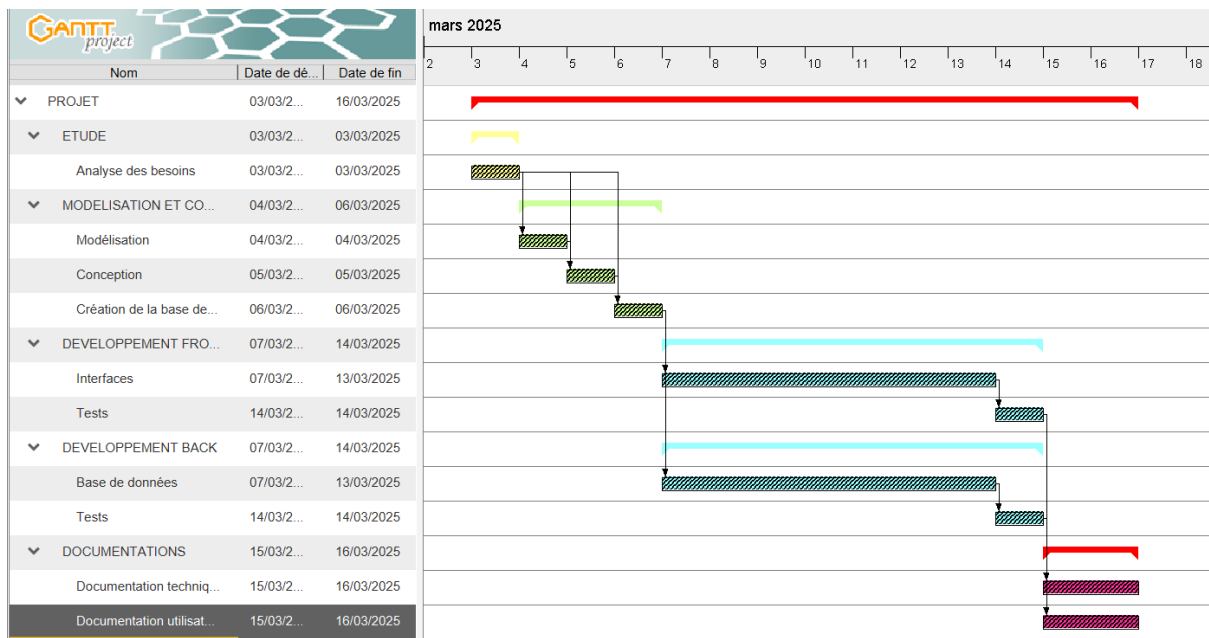
Les technologies utilisées sont, pour la partie front-end, HTML, CSS. Pour la partie back-end, PHP est utilisé. La base de données choisie est MySQL gérée via phpMyAdmin. Le serveur web local utilisé est XAMPP. Pour la sécurité, il faut utiliser des requêtes préparées pour éviter les injections SQL. L'accès doit être géré avec des comptes pour les organisateurs afin qu'il puisse modifier des informations sur les participants. Les composants communiquent entre eux, d'une part le front-end qui envoie les requêtes via des formulaires HTML et traités d'autre part par le back-end PHP. Les données sont récupérées et affichées via PHP. Enfin, la structure du code doit être organisée afin de permettre des modifications si nécessaire.

## 2.6) Planification du projet

Le projet débute le 3 mars 2025 et doit être finalisé le 16 mars 2025 pour une livraison du projet pour le début du mois d'avril 2025.

Je réalise le planning prévisionnel à l'aide de GanttProject. L'utilité de ce planning est d'avoir une vue d'ensemble des étapes nécessaires à la réalisation de ce projet. Je peux aussi avoir un visuel sur le chemin critique pour avoir les étapes obligatoires à la bonne réalisation du projet. Enfin, ce logiciel me permet d'affecter les tâches aux ressources humaines du projet. Ici, je suis chef de projet et la seule personne affectée à la réalisation de toutes ces tâches.





Les livrables attendus pour ce projet sont le cahier des charges, la modélisation et la conception de la base de données, le code source du projet, la documentation technique et d'utilisation, la documentation du projet, le rapport de tests et la plateforme fonctionnelle.

Les ressources mobilisées pour ce projet sont de type matérielles, logicielles et humaines. Elles se composent d'un ordinateur, d'un serveur local XAMPP, d'un logiciel de gestion de base de données phpMyAdmin ainsi qu'un éditeur de code VS Code, un outil collaboratif de gestion des versions Git, le langage de programmation PHP, un système de gestion de bases de données relationnelles MySQL, un navigateur web. Enfin, je serais seule à réaliser ce projet qui est individuel.

## 2.7) Critères de validation

Pour que le projet soit validé, il doit répondre aux exigences définies dans le cahier des charges. Les critères de validation incluent toutes les fonctionnalités prévues, le respect des contraintes techniques, les tests sont réussis. De plus, l'interface doit être intuitive et ergonomique et respecter les règles de protection des données.

## 2.8) Evolution

Ce projet est conçu pour être évolutif et pourra être amélioré pour les prochaines courses prévues. Par exemple, le système pourra envoyer une notification de confirmation par email lorsque l'inscription à la course est validée. Les participants pourraient avoir la possibilité de créer un compte et de modifier eux-mêmes leurs informations sans avoir à passer par un organisateur.

## 3) Modélisation de la base de données

Ici, je vais visualiser toutes les données dont je vais avoir besoin et les représenter sous forme d'entités et de relations grâce aux méthodes Merise et UML.

### 3.1) Le dictionnaire de données

Je crée le dictionnaire de données à l'aide d'un tableau Excel. Il permet de concevoir et gérer une base de données. Il est utile pour décrire les données, faciliter la communication entre développeurs, standardiser et comprendre la structure des données.

	A	B	C	D	E	F
1	TABLE	Nom	Nom logique	Type	Longueur	Observation
2	PARTICIPANTS	identifiant_participant	part_id	INT (numérique)	11	Clé primaire, identifiant unique pour chaque participant.
3		nom_participant	part_nom	VARCHAR (caractère)	50	Nom de famille du participant
4		prenom_participant	part_prenom	VARCHAR (caractère)	50	Prénom du participant
5		date_naissance	part_date_naissance	DATE		Date de naissance du participant
6		email_participant	part_email	VARCHAR (caractère)	100	Email du participant
7		telephone_participant	part_telephone	VARCHAR (numérique)	10	Numéro de téléphone du participant
8	ORGANISATEURS	identifiant_organisateur	orga_id	INT (numérique)	11	Clé primaire, identifiant unique pour chaque organisateur.
9		nom_organisateur	orga_nom	VARCHAR (caractère)	50	Nom de famille de l'organisateur
10		prenom_organisateur	orga_prenom	VARCHAR (caractère)	50	Prénom de l'organisateur
11		email_organisateur	orga_email	VARCHAR (caractère)	100	Email de l'organisateur
12		telephone_organisateur	orga_telephone	VARCHAR (numérique)	10	Numéro de téléphone de l'organisateur
13		mot_de_passe_organisateur	orga_mot_de_passe	VARCHAR (numérique)	255	Mot de passe de l'organisateur.
14	COURSES	identifiant_course	cour_id	INT (numérique)	11	Clé primaire, identifiant unique pour chaque course.
15		nom_course	cour_nom	VARCHAR (caractère)	50	Nom de la course
16		date_course	cour_date	DATE		Date de la course
17		distance_course	cour_distance	INT (numérique)	5	Distance de la course (5km, 10km, 15km)
18	INSCRIPTIONS	identifiant_inscription	inscr_id	INT (numérique)	11	Clé primaire, identifiant unique pour chaque inscription.
19		identifiant_course	cour_id	INT (numérique)	11	Clé étrangère, identifiant unique pour chaque course
20		identifiant_participant	part_id	INT (numérique)	11	Clé étrangère, identifiant unique pour chaque participant.
21		date_inscription	inscr_date	DATE		Date de l'inscription
22		statut_inscription	inscr_statut	VARCHAR (caractère)	20	Statut de l'inscription (en attente, validée ou annulée)

### 3.2) Modèle conceptuel de données et modèle physique de données

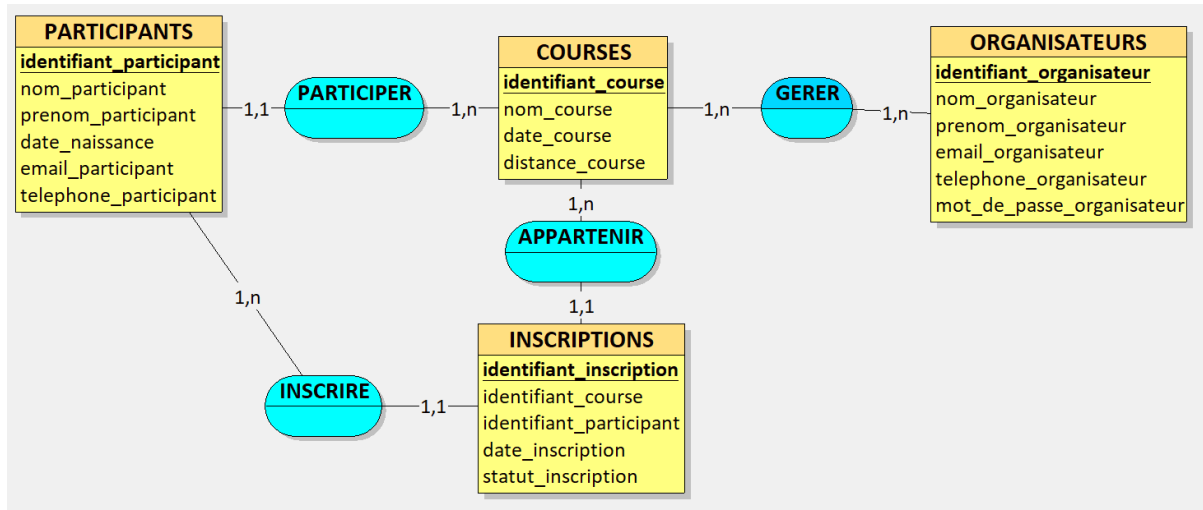
Je vais utiliser la méthode Merise utilisée pour créer des bases de données et modéliser les processus en informatique. D'abord, je conçois le modèle conceptuel de données (MCD) qui représente les données du système d'information de manière abstraite sous la forme d'entités, d'associations, d'attributs et de cardinalités.

Enfin, je crée le modèle physique de données (MPD) qui est adapté aux caractéristiques techniques du système de gestion de bases de données. Il utilise des index, des types de données et optimise la performance.

Pour concevoir ces modèles, je vais utiliser le logiciel Looping.

Voici le modèle conceptuel de données (MCD) :





Ensuite, je vais expliciter les relations.

#### PARTICIPANTS - COURSES

Un participant peut participer à une seule course.

Une course peut contenir plusieurs participants.

#### ORGANISATEURS - COURSES

Un organisateur peut être responsable de plusieurs courses.

Une course peut être gérée par plusieurs organisateurs.

#### PARTICIPANTS - INSCRIPTIONS

Un participant peut s'inscrire à un ou plusieurs événements.

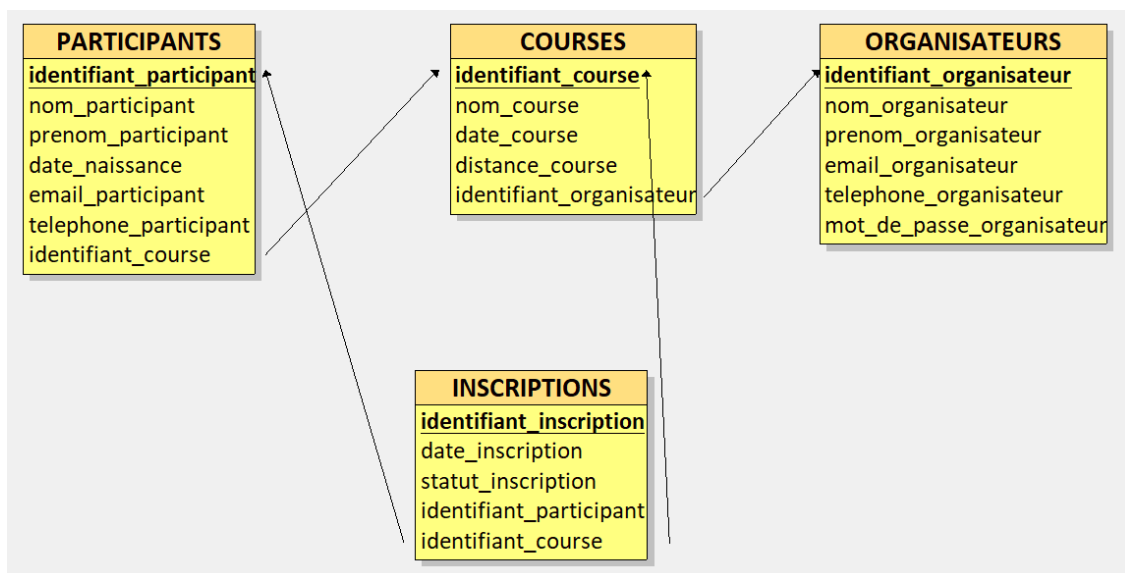
Une inscription contient un seul participant.

#### COURSES - INSCRIPTIONS

Une course peut avoir plusieurs inscriptions.

Une inscription appartient à une course.

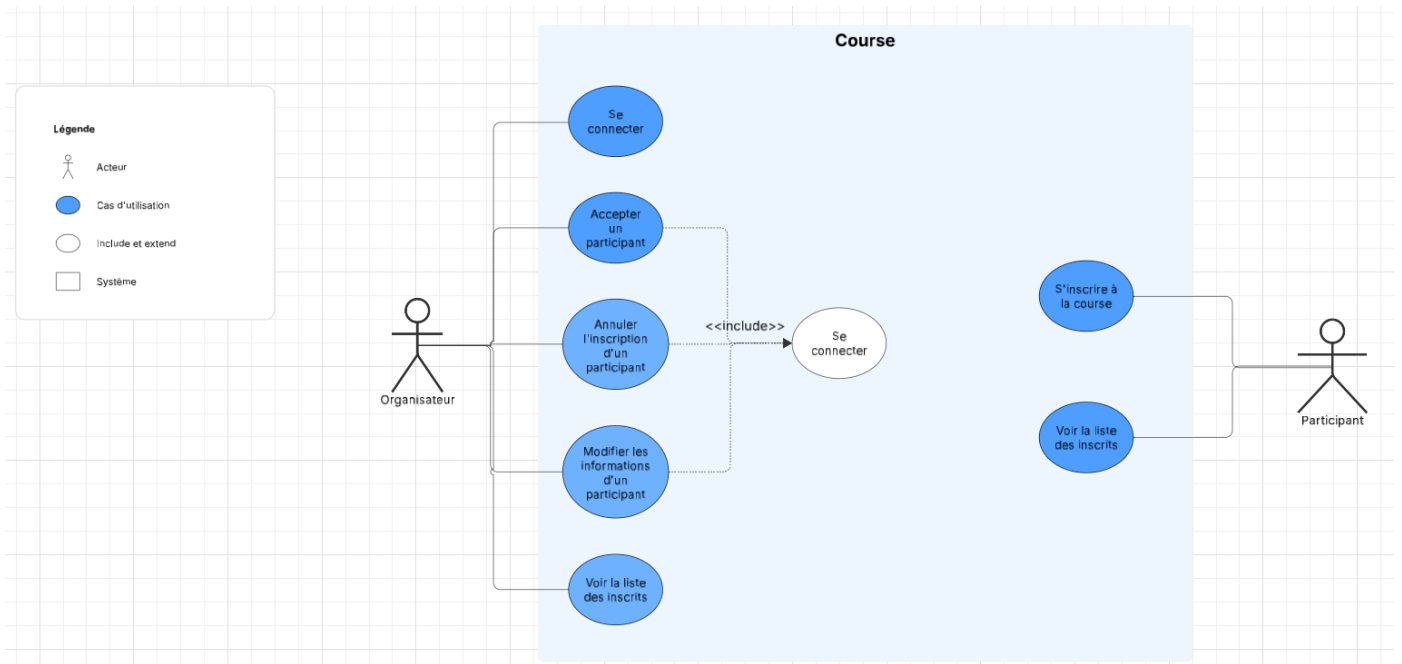
Ensuite, je réalise le modèle physique de données (MPD) :



### 3.3) Diagramme de cas d'utilisation et diagramme de séquence

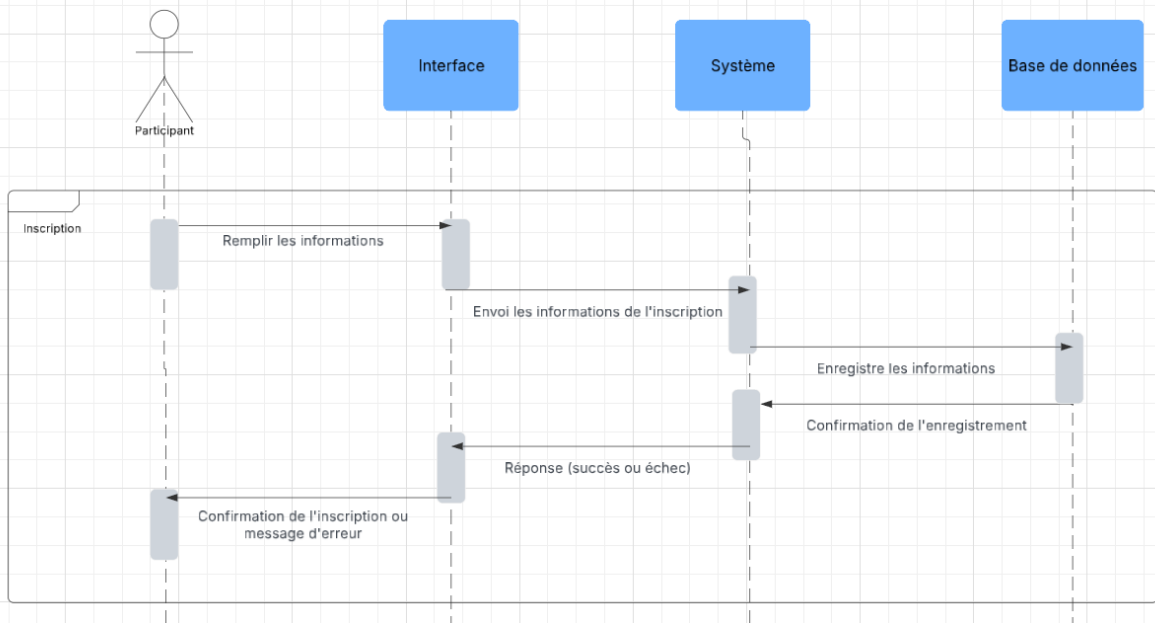
Le diagramme de cas d'utilisation est utilisé en langage de modélisation unifié (UML), il montre les interactions entre les utilisateurs et le système.

Je vais commencer par ce diagramme avec les participants et les organisateurs qui seront les utilisateurs. Pour cela, je m'aide de l'outil en ligne Lucidchart.

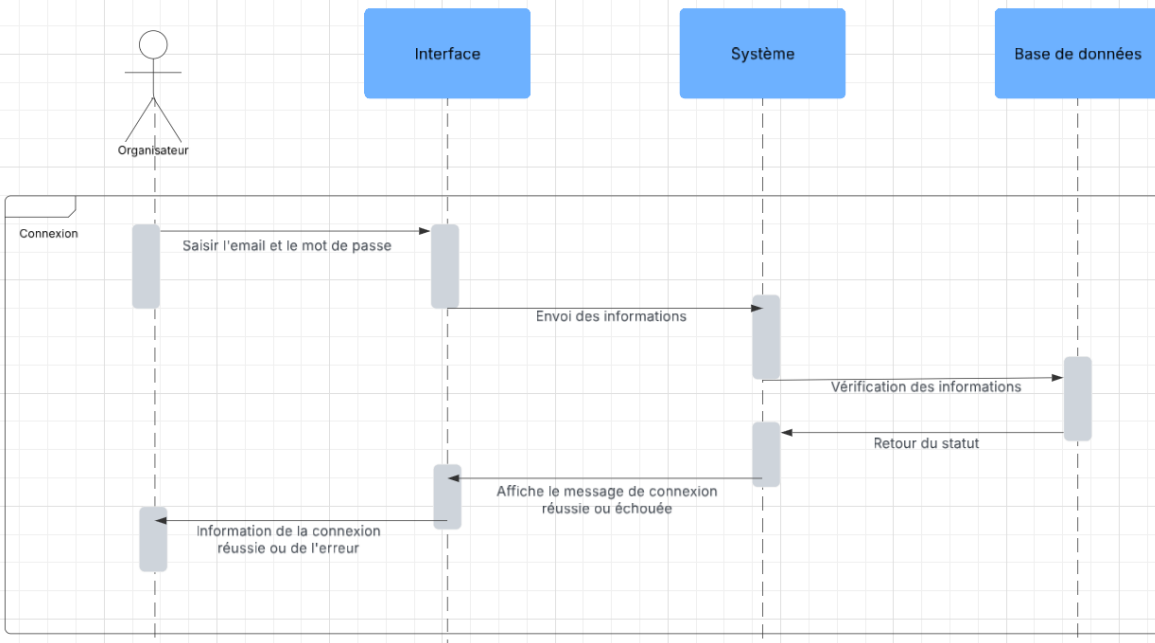


Le diagramme de séquence est également utilisé en langage de modélisation unifié (UML), il montre le déroulement des interactions entre les acteurs et le système. Ici, il y a plusieurs acteurs, le participant, l'organisateur, l'interface, le système et la base de données. Chaque diagramme représente une seule fonctionnalité dont l'inscription à la course, la connexion d'un organisateur, accepter ou annuler l'inscription d'un participant, modifier les informations d'un participant.

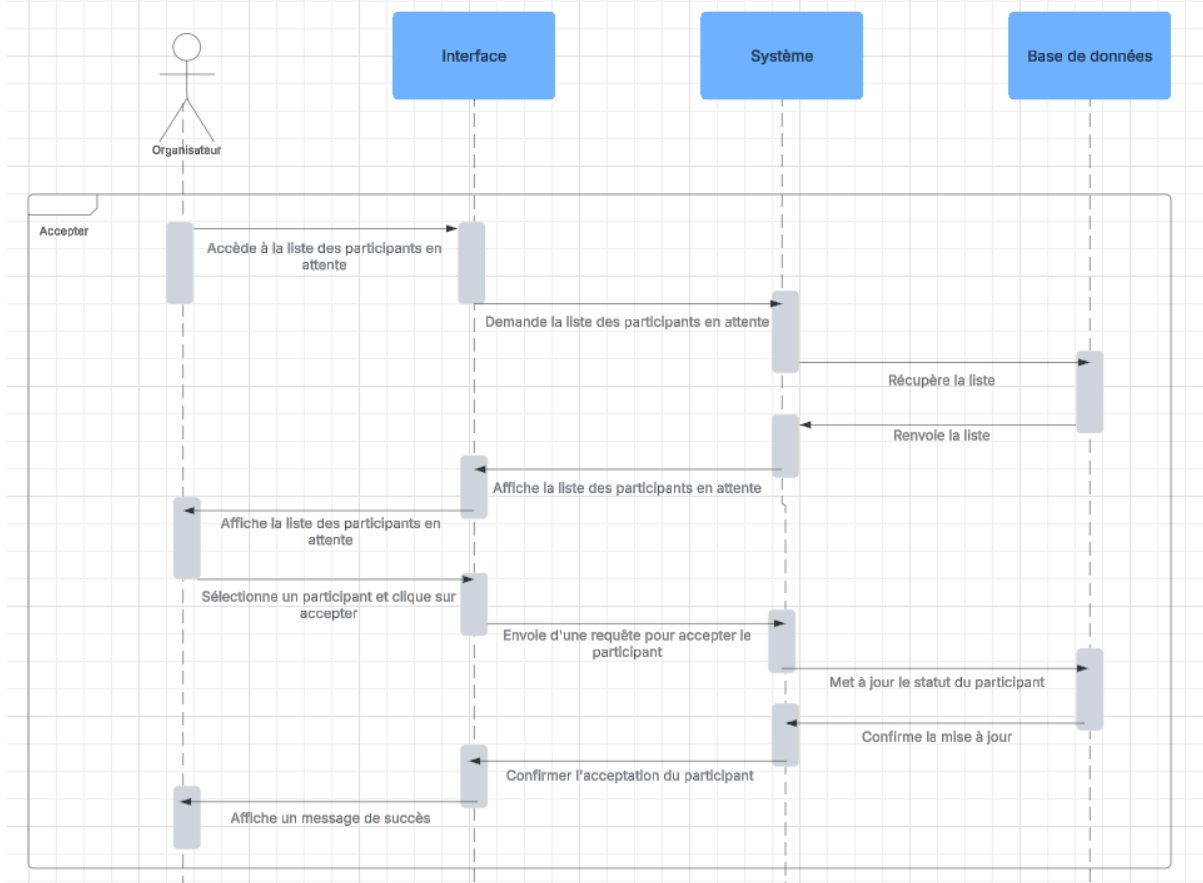
## 1. Inscription à la course



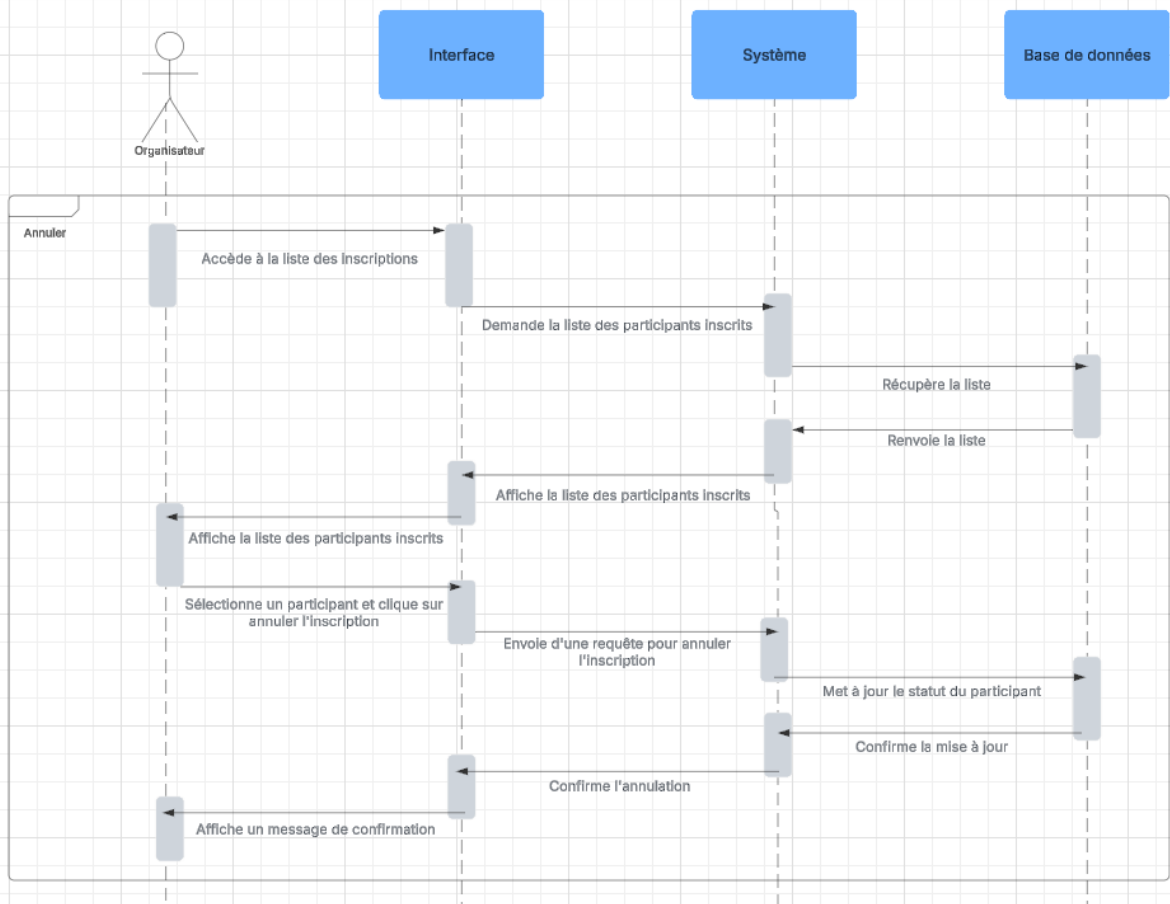
## 2. Connexion

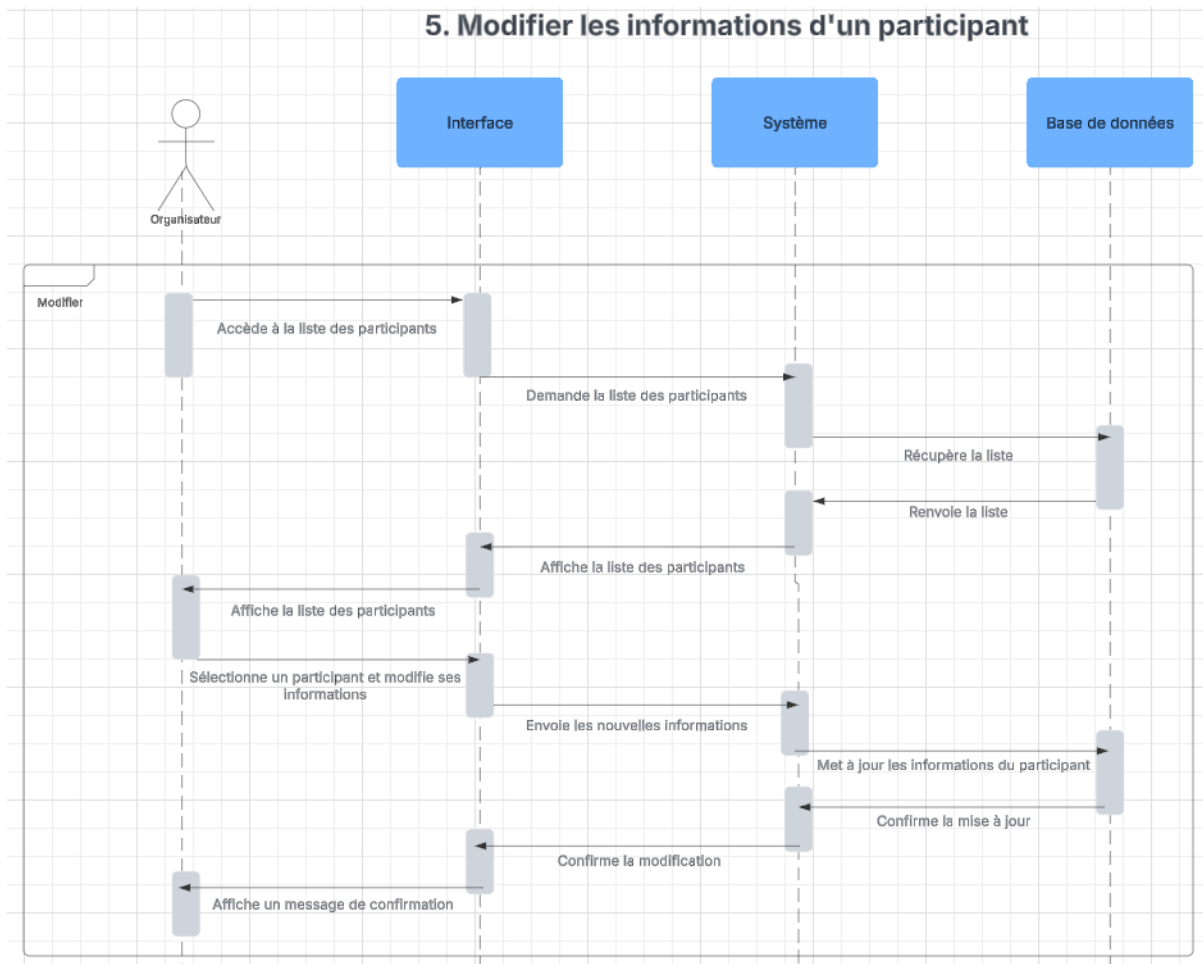


### 3. Accepter l'inscription d'un participant



#### 4. Annuler l'inscription d'un participant





## 4) Conception de la base de données

Je vais créer la base de données MySQL sur phpMyAdmin. Pour cela, je crée une nouvelle base de données et j'y ajoute des tables grâce à des requêtes SQL.

Ici, je crée la base de données :

```
1 CREATE DATABASE gestion_course;
```

Ensuite, je crée la première table "participants". Chaque participant est associé à un identifiant unique et les autres colonnes stockent les informations. Certaines colonnes ont des contraintes "NOT NULL" pour ne pas qu'elles soient nulles et "UNIQUE" pour l'email évitant ainsi les doublons.

```

1 CREATE TABLE participants (
2     part_id INT AUTO_INCREMENT PRIMARY KEY,
3     part_nom VARCHAR(50) NOT NULL,
4     part_prenom VARCHAR(50) NOT NULL,
5     part_date_de_naissance DATE NOT NULL,
6     part_email VARCHAR(100) UNIQUE NOT NULL,
7     part_telephone VARCHAR(10)
8 );

```

Je vais créer la table “organiseurs”. Chaque organisateur est associé à un identifiant unique et les autres colonnes stockent les informations. Certaines colonnes ont des contraintes “NOT NULL” pour ne pas qu’elles soient nulles et “UNIQUE” pour l’email évitant ainsi les doublons.

```

1 CREATE TABLE organisateurs (
2     orga_id INT AUTO_INCREMENT PRIMARY KEY,
3     orga_nom VARCHAR(50) NOT NULL,
4     orga_prenom VARCHAR(50) NOT NULL,
5     orga_email VARCHAR(100) UNIQUE NOT NULL,
6     orga_mot_de_passe VARCHAR(255) NOT NULL,
7     orga_telephone VARCHAR(10) NOT NULL
8 );

```

Je vais créer la table “courses”. Chaque course est associée à un identifiant unique. Certaines colonnes ont des contraintes “NOT NULL” pour ne pas qu’elles soient nulles.

```

1 CREATE TABLE courses (
2     cour_id INT AUTO_INCREMENT PRIMARY KEY,
3     cour_nom VARCHAR(50) NOT NULL,
4     cour_date DATE NOT NULL,
5     cour_distance INT NOT NULL
6 );

```

Enfin, je crée la table “inscriptions” pour gérer les inscriptions des participants aux courses. Chaque inscription est associée à un identifiant unique. Les clés étrangères “cour\_id” et “part\_id” relient une inscription à une course et un participant.

```

1 CREATE TABLE inscriptions (
2     inscr_id INT AUTO_INCREMENT PRIMARY KEY,
3     cour_id INT NOT NULL,
4     part_id INT NOT NULL,
5     inscr_date DATETIME NOT NULL,
6     inscr_statut ENUM('en attente', 'confirmée', 'annulée') NOT NULL,
7     FOREIGN KEY (cour_id) REFERENCES courses(cour_id),
8     FOREIGN KEY (part_id) REFERENCES participants(part_id)
9 );

```

## 4.1) Insérer des données de test

Toutes les tables sont créées, je vais insérer des données de test pour m'assurer que tout fonctionne correctement.

J'insère des données de test dans toutes les tables.

Pour la table "participants" :

```
1 INSERT INTO participants (part_nom, part_prenom, part_date_de_naissance, part_email, part_telephone)
2 VALUES
3 ('Garnier', 'Jean', '1990-05-12', 'jean.garnier@email.com', '0612345678'),
4 ('Martin', 'Sophie', '1985-09-25', 'sophie.martin@email.com', '0698765432'),
5 ('Bernard', 'Lucas', '1998-07-15', 'lucas.bernard@email.com', '0678456123'),
6 ('Leroy', 'Emma', '2000-03-30', 'emma.leroy@email.com', '0654321897'),
7 ('Morel', 'Thomas', '1995-11-22', 'thomas.morel@email.com', '0645678932');
8 |
```

Pour la table "organisateurs" :

```
1 INSERT INTO organisateurs (orga_nom, orga_prenom, orga_email, orga_telephone, orga_mot_de_passe) VALUES
2 ('Durand', 'Paul', 'paul.durand@email.com', '0623456789', 'mdp123'),
3 ('Lemoine', 'Julie', 'julie.lemoine@email.com', '0676543210', 'mdp456'),
4 ('Petit', 'Nicolas', 'nicolas.petit@email.com', '0611122233', 'mdp789'),
5 ('Garcia', 'Camille', 'camille.garcia@email.com', '0633445566', 'mdp753'),
6 ('Roux', 'Antoine', 'antoine.roux@email.com', '0655667788', 'mdp159');
7 |
```

Pour la table "courses". Il y a une seule course proposée par la commune qui se déroulera le 10 juin 2025. Les distances proposées sont de 5km, 10km et 15km.

```
1 INSERT INTO courses (cour_id, cour_nom, cour_date, cour_distance) VALUES
2 ('1', 'Course de la Ville', '2025-06-10', 5),
3 ('2', 'Course de la Ville', '2025-06-10', 10),
4 ('3', 'Course de la Ville', '2025-06-10', 15);
- |
```

Pour la table "inscriptions" :

```
1 INSERT INTO inscriptions (cour_id, part_id, inscr_date, inscr_statut) VALUES
2 (1, 1, '2025-02-25', 'confirmée'),
3 (2, 2, '2025-02-25', 'annulée'),
4 (3, 3, '2025-02-25', 'confirmée'),
5 (1, 4, '2025-02-25', 'en attente'),
6 (2, 5, '2025-02-25', 'en attente');
```

## 4.2) Tester les relations entre les tables

Je vais tester les relations entre les tables à l'aide de requêtes SQL pour vérifier que tout fonctionne. J'utilise des alias dans chaque requête pour une meilleure lisibilité. D'abord, j'affiche les inscriptions en associant les noms des participants et les courses correspondantes. Cette requête sélectionne l'identifiant de l'inscription, le nom et prénom du participant, le nom et la distance de la course, la date et le statut de l'inscription. La première jointure relie la table "inscriptions" à la table "participants" en associant l'identifiant du



participant dans chaque table. On récupère le nom et le prénom du participant inscrit. La seconde jointure relie la table “inscriptions” à la table “courses” en associant l’identifiant de la course. On peut obtenir le nom et la distance de la course correspondante.

```
1 SELECT i.inscr_id, p.part_nom, p.part_prenom, c.cour_nom, c.cour_distance, i.inscr_date, i.inscr_statut
2 FROM inscriptions i
3 JOIN participants p ON i.part_id = p.part_id
4 JOIN courses c ON i.cour_id = c.cour_id;
5 |
```

inscr_id	part_nom	part_prenom	cour_nom	cour_distance	inscr_date	inscr_statut
1	Garnier	Jean	Course de la Ville	5	2025-02-25 00:00:00	confirmée
4	Leroy	Emma	Course de la Ville	5	2025-02-25 00:00:00	en attente
2	Martin	Sophie	Course de la Ville	10	2025-02-25 00:00:00	annulée
5	Morel	Thomas	Course de la Ville	10	2025-02-25 00:00:00	en attente
3	Bernard	Lucas	Course de la Ville	15	2025-02-25 00:00:00	confirmée

Je vais voir le nombre d’inscriptions par course. Cette requête affiche le nom, la distance et le nombre total d’inscriptions pour la course à partir de la table “courses”. La jointure associe les courses à leurs inscriptions grâce à l’identifiant de la course. L’indication “GROUP BY” regroupe les résultats par course.

```
1 SELECT c.cour_nom, c.cour_distance, COUNT(i.inscr_id) AS nombre_inscriptions
2 FROM courses c
3 LEFT JOIN inscriptions i ON c.cour_id = i.cour_id
4 GROUP BY c.cour_id;
```

cour_nom	cour_distance	nombre_inscriptions
Course de la Ville	5	2
Course de la Ville	10	2
Course de la Ville	15	1

Je teste aussi le changement de statut d’une inscription. Cette requête met à jour l’inscription ayant pour identifiant ‘4’ dans la table “inscriptions” en changeant son statut en “confirmée”.

```
1 UPDATE inscriptions
2 SET inscr_statut = 'confirmée'
3 WHERE inscr_id = 4;
```

✓ 1 ligne affectée. (traitement en 0,0006 seconde(s).)

```
UPDATE inscriptions SET inscr_statut = 'confirmée' WHERE inscr_id = 4;
```

## 5) Développement

Pour rappel, l’objectif de cette plateforme de gestion d’une course à pied est de simplifier l’organisation de la course grâce à une interface intuitive et fonctionnelle qui va permettre aux organisateurs de gérer cet événement et aux participants de s’inscrire facilement à la course.

J'utilise plusieurs outils définis dans le cahier des charges. D'abord, la plateforme est développée en PHP. Ensuite, la base de données choisie est MySQL gérée via phpMyAdmin. Cette base de données relationnelle est robuste et évolutive. Elle est idéale pour gérer les données tout en assurant la sécurité de celles-ci. De plus, phpMyAdmin facilite l'administration de ma base de données, l'interface est intuitive et efficace ce qui ne demande pas de compétences avancées.

Pour gérer les versions du code, j'utilise l'outil Git qui permet de suivre l'évolution du code et permet donc de développer de façon organisée et de revenir sur une version du code en particulier si besoin.

Ces choix technologiques garantissent la stabilité, la sécurité et l'évolutivité de la plateforme tout en facilitant le développement et sa maintenance à long terme.

## 5.1) Présentation de Git

Comme défini dans le cahier des charges, un logiciel de gestion de versions doit être utilisé afin de garder une trace des modifications du code. Pour ce faire, je vais utiliser Git. Je vais pouvoir avoir un historique complet de mon projet, je pourrais revenir à la version précédente en cas d'erreurs et cela va me permettre de mieux m'organiser.

Je vais utiliser les commandes du terminal pour me rendre dans le dossier de mon projet, initialiser Git puis ajouter des fichiers. Je fais ensuite un commit dans le dépôt Git pour commenter les modifications.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\xampp\htdocs\gestion_course> git init
Initialized empty Git repository in C:/xampp/htdocs/gestion_course/.git/
PS C:\xampp\htdocs\gestion_course> git add .
PS C:\xampp\htdocs\gestion_course> git commit -m "Ajout des fichiers au projet de course sportive"
[master (root-commit) 49e2e62] Ajout des fichiers au projet de course sportive
12 files changed, 722 insertions(+)
create mode 100644 affichage-tableau-de-bord.php
create mode 100644 config.php
create mode 100644 connexion-bdd.php
create mode 100644 connexion.html
create mode 100644 connexion.php
create mode 100644 images/logo.png
create mode 100644 index.html
create mode 100644 inscription.html
create mode 100644 participants.php
create mode 100644 styles.css
create mode 100644 tableau-de-bord.php
create mode 100644 traitement-inscription.php
PS C:\xampp\htdocs\gestion_course>
```

Les fichiers Git sont stockés dans un fichier caché appelé .git et contient tout l'historique du projet, les commits et la configuration.

Nom	Modifié le	Type	Taille
hooks	07/02/2025 11:42	Dossier de fichiers	
info	07/02/2025 11:42	Dossier de fichiers	
logs	07/02/2025 11:44	Dossier de fichiers	
objects	07/02/2025 11:44	Dossier de fichiers	
refs	07/02/2025 11:42	Dossier de fichiers	
COMMIT_EDITMSG	07/02/2025 11:44	Fichier	1 Ko
config	07/02/2025 11:42	Fichier	1 Ko
description	07/02/2025 11:42	Fichier	1 Ko
HEAD	07/02/2025 11:42	Fichier	1 Ko
index	07/02/2025 11:44	Fichier	1 Ko

Ensuite, je décide d'envoyer ce projet sur GitHub afin de sauvegarder une fois de plus mon projet et de pouvoir m'organiser.


Je crée un nouveau dépôt GitHub qui s'intitule "gestion\_course" et j'utilise plusieurs commandes pour ajouter mes fichiers dans le dépôt.

```
PS C:\xampp\htdocs\gestion_course> git remote add origin https://github.com/LeaGiron/gestion_course.git
PS C:\xampp\htdocs\gestion_course> git branch -M main
PS C:\xampp\htdocs\gestion_course> git push -u origin main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (23/23), 34.95 KiB | 2.91 MiB/s, done.
Total 23 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/LeaGiron/gestion_course.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\xampp\htdocs\gestion_course>
```

Par exemple, je peux ajouter un fichier sur GitHub à chaque modification importante qui nécessite une nouvelle version. Pour cela, j'exécute ces commandes dans mon répertoire :

```
PS C:\Users\giron\OneDrive - CHARLES DE FOUCAULD\Bureau\BTS SIO 2\Bloc 2 SLAM\gestion_course> git add tableau-de-bord.php
PS C:\Users\giron\OneDrive - CHARLES DE FOUCAULD\Bureau\BTS SIO 2\Bloc 2 SLAM\gestion_course> git commit -m "Page tableau de bord OK"
[main 254dbc2] Page tableau de bord OK
1 file changed, 107 insertions(+), 27 deletions(-)
PS C:\Users\giron\OneDrive - CHARLES DE FOUCAULD\Bureau\BTS SIO 2\Bloc 2 SLAM\gestion_course> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.53 KiB | 783.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/LeaGiron/gestion_course.git
 fd18c23..254dbc2 main -> main
PS C:\Users\giron\OneDrive - CHARLES DE FOUCAULD\Bureau\BTS SIO 2\Bloc 2 SLAM\gestion_course>
```

Le fichier est ensuite mis à jour sur GitHub avec le nom du fichier "tableau-de-bord.php" et du commit "Page tableau de bord OK".

 tableau-de-bord.php	Page tableau de bord OK	1 minute ago
---	-------------------------	--------------

## 5.2) La page d'accueil (index)

La page d'accueil va permettre à tous de consulter les informations de la course. J'ajoute un bouton de connexion pour les organisateurs, un bouton "s'inscrire à la course" qui renvoie

vers le formulaire d'inscription et un bouton "voir la liste des inscrits" qui renvoie vers la liste de tous les participants inscrits.

Cette page est simple et intuitive. Toutes les actions sont faites depuis cette page pour les participants. Il suffit de moins de trois clics pour trouver l'information recherchée.

De plus, une catégorie contact est ajoutée pour permettre aux participants de contacter un organisateur par téléphone pour pouvoir modifier ses informations personnelles ou annuler son inscription.

**Course de la Ville 2025** [Connexion Organisateur](#)

## Bienvenue sur la plateforme officielle de la course

Participez à notre course en solo **le 10 juin 2025** avec des distances de 5km, 10km et 15km.

Inscrivez-vous dès maintenant et rejoignez la compétition !

### À propos

La Course de la Ville 2025 est un événement sportif unique qui rassemble des passionnés de course à pied. Que vous soyez débutant ou athlète confirmé, venez vivre une expérience inoubliable dans un cadre urbain dynamique.

[S'inscrire à la course](#) [Voir la liste des inscrits](#)

### Programme

- 09:00 - Ouverture des stands
- 10:00 - Départ de la course
- 13:00 - Remise des prix
- 14:00 - Cocktail et animations

### Contact

Pour toute question ou information, contactez l'un des organisateurs au 06 76 54 32 10.

© 2025 Course de la Ville

### 5.3) La page d'inscription

Cette page permet aux participants de remplir un formulaire pour s'inscrire à la course. Il y a plusieurs champs à compléter tels que le nom et prénom, la date de naissance, l'email, le numéro de téléphone et la distance.

De plus, un bouton permet d'être redirigé rapidement vers la page d'accueil.

Course de la Ville 2025

Page d'accueil

### Inscription à la course

Nom :

Prénom :

Date de naissance :

Email :

Numéro de téléphone :

Choisissez une course :

S'inscrire

© 2025 Course de la Ville

Pour enregistrer la distance de la course dans la table “inscriptions”, je vais créer une jointure entre la table “inscriptions” et la table “courses” ce qui évite de stocker une donnée redondante.

```

1 SELECT
2     i.inscr_id,
3     i.cour_id,
4     i.part_id,
5     i.inscr_date,
6     i.inscr_statut,
7     c.cour_distance
8 FROM inscriptions i
9 JOIN courses c ON i.cour_id = c.cour_id;

```

inscr_id	cour_id	part_id	inscr_date	inscr_statut	cour_distance
1	1	1	2025-02-25 00:00:00	confirmée	5
2	2	2	2025-02-25 00:00:00	annulée	10
3	3	3	2025-02-25 00:00:00	confirmée	15
4	1	4	2025-02-25 00:00:00	confirmée	5
5	2	5	2025-02-25 00:00:00	en attente	10

Je vais ensuite tester la fonctionnalité d’inscription pour vérifier que les informations s’enregistrent bien dans la base.

## Inscription à la course

Nom :

Test

Prénom :

Lea

Date de naissance :

02/04/2001

Email :

lea.g@test.com

Numéro de téléphone :

0123456789

Choisissez une course :

10 km

S'inscrire

				part_id	part_nom	part_prenom	part_date_de_naissance	part_email	part_telephone
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	Garnier	Jean	1990-05-12	jean.garnier@email.com	0612345678
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	Martin	Sophie	1985-09-25	sophie.martin@email.com	0698765432
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	Bernard	Lucas	1998-07-15	lucas.bernard@email.com	0678456123
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	Leroy	Emma	2000-03-30	emma.leroy@email.com	0654321897
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	Morel	Thomas	1995-11-22	thomas.morel@email.com	0645678932
<input type="checkbox"/>	Éditer	Copier	Supprimer	50	Test	Lea	2001-01-01	lea.g@test.com	0123456789

				inscr_id	cour_id	part_id	inscr_date	inscr_statut
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	1	1	2025-02-25 00:00:00	confirmée
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	2	2	2025-02-25 00:00:00	annulée
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	3	3	2025-02-25 00:00:00	confirmée
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	1	4	2025-02-25 00:00:00	confirmée
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	2	5	2025-02-25 00:00:00	en attente
<input type="checkbox"/>	Éditer	Copier	Supprimer	15	1	50	2025-03-07 11:42:30	en attente

Je remplis le formulaire et les données sont enregistrées dans la table “participants”. En cliquant sur le bouton s’inscrire, je m’inscris à la course et les données s’enregistrent dans la table “inscriptions”. Je retrouve mes informations dans la table “participants” et dans la table “inscriptions”.

### 5.4) La page de connexion des organisateurs

Cette page permet aux organisateurs de se connecter pour gérer les inscriptions et informations des participants. Le formulaire demande à l’organisateur de rentrer son email et son mot de passe pour se connecter. J’ajoute également un bouton qui permet de rediriger directement vers la page d’accueil.

Course de la Ville 2025

Page d'accueil

Connectez-vous

Email :

Mot de passe :

Se connecter

© 2025 Course de la Ville

Je décide de séparer la configuration (config.php) et la connexion à la base de données (connexion-bdd.php) dans deux fichiers distincts pour plus de sécurité. J'utilise PDO (PHP Data Objects) pour la connexion à la base de données. L'avantage d'utiliser PDO est la sécurité car elle permet de protéger contre les attaques par injection SQL. Aussi, la gestion des erreurs est plus robuste car elle permet de rendre plus facile le débogage et la gestion des erreurs. Ce choix est approprié car le cahier des charges définit que la plateforme doit être sécurisée car elle contient des données personnelles et le projet doit être facilement modifiable.

Je crée un utilisateur sécurisé avec des droits limités. Pour cela, je me rend sur phpMyAdmin et j'exécute ces requêtes. Je crée l'utilisateur 'admin\_course' avec un mot de passe. La seconde requête permet de donner tous les droits en lecture, écriture, modification et suppression à toutes les tables de la base de données "gestion\_course". Je recharge la table des privilèges avec "FLUSH PRIVILEGES" pour prendre en compte les modifications.
















Je vérifie ensuite dans l'onglet "Comptes utilisateurs" si mon utilisateur administrateur a tous les privilèges.

 **Aperçu des comptes utilisateurs**

	Nom d'utilisateur	Nom d'hôte	Mot de passe	Privilèges globaux 	Groupe d'utilisateurs	« Grant »	Action
<input type="checkbox"/>	N'importe quel	%	Non 	USAGE		Non	 Éditer les privilèges  Exporter  Verrouiller
<input type="checkbox"/>	admin_course	localhost	Oui	ALL PRIVILEGES		Oui	 Éditer les privilèges  Exporter  Verrouiller

Dans ma base de données, j'ai précédemment inséré des données de test pour quelques utilisateurs. Cependant, les mots de passe sont stockés en clair dans la table "organismateurs". Donc, cela indique une erreur quand j'essaie de me connecter avec un mot de passe en clair. Je vais donc mettre à jour les données de test en exécutant un script PHP qui va hasher ces mots de passe de tests.

Je crée un script "maj\_mot\_de\_passe" et j'y accède via cet url : [http://localhost/gestion\\_course/maj\\_mot\\_de\\_passe.php](http://localhost/gestion_course/maj_mot_de_passe.php)  
Je reçois une confirmation de mise à jour puis je vérifie en accédant à la table "organismateurs".

<div>←T→</div>			orga_id	orga_nom	orga_prenom	orga_email	orga_mot_de_passe	orga_telephone	
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	1	Durand	Paul	paul.durand@email.com	\$2y\$10\$ekDNmXDJmMC7MPELXjkFO.mebBulkfu.9/go9eqERtr...	0623456789
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	2	Lemoine	Julie	julie.lemoine@email.com	\$2y\$10\$ij52V6PcNXHR2X4McdNelOIkw02Vc8Hs/UI7N31p5Zl...	0676543210
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	3	Petit	Nicolas	nicolas.petit@email.com	\$2y\$10\$44n.AGnR8Wd9xrq6pKG17uMVErbYZGfllsEOS9d8f9A...	0611122233
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	4	Garcia	Camille	camille.garcia@email.com	\$2y\$10\$dk.CINyR0UvGCC6qnk2IQS2kMy3MV3hLk6i5qw1do...	0633445566
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	5	Roux	Antoine	antoine.roux@email.com	\$2y\$10\$2a.lyblG2she3y1IHHZT2ezXQ8dkOR7ZdmNXEHtEMz...	0655667788

Je supprime ensuite ce fichier pour qu'il ne soit pas accessible à des personnes malveillantes. Je peux maintenant me connecter avec l'email et le mot de passe d'un utilisateur de test sans avoir d'erreurs, ce qui me redirige vers le tableau de bord de l'organisateur.

## 5.5) Le tableau de bord

Le tableau de bord de l'organisateur permet à celui-ci de gérer en quelques clics toutes les inscriptions. Grâce aux sections de cette page, l'organisateur a une vue d'ensemble des participants à accepter, des participants inscrits et ceux qui ont annulé leur inscription.

Il suffit de cliquer sur :

- le bouton "accepter" pour que les informations du participant se retrouvent dans la section "liste de tous les participants inscrits"
- le bouton "modifier les informations" pour être redirigé vers un formulaire de modification
- le bouton "annuler l'inscription" pour que les informations du participant se retrouvent dans la section "participants avec inscription annulée"
- le bouton "restaurer l'inscription" pour que les informations du participant se retrouvent dans la section "liste de tous les participants inscrits"

Il peut aussi revenir à la page d'accueil ou se déconnecter en un clic.

Course de la Ville 2025

Page d'accueil

Bienvenue sur votre tableau de bord, Organisateur

Déconnexion

Participants à accepter

Voici la liste des participants qui ont demandé à s'inscrire à vos courses.

Nom : Garnier Jean

Course demandée : 5

Email : jean.garnier@email.fr

Accepter



### Liste de tous les participants inscrits

Voici la liste des participants inscrits à vos courses.

**Nom** : Leroy Emma  
**Course** : 5  
**Email** : emma.leroy@email.com

Modifier les informationsAnnuler l'inscription

**Nom** : Test Lea  
**Course** : 5  
**Email** : lea.g@test.com

Modifier les informationsAnnuler l'inscription

**Nom** : Morel Thomas  
**Course** : 10  
**Email** : thomas.morel@email.com

Modifier les informationsAnnuler l'inscription

### Participants avec inscription annulée

Voici la liste des participants dont l'inscription a été annulée.

**Nom** : Martin Sophie  
**Course** : 10  
**Email** : sophie.martin@email.com

Restaurer l'inscription

© 2025 Course de la Ville

## 5.6) La page pour accepter un participant

Cette section du tableau de bord permet de retrouver tous les participants en attente d'inscription. Il suffit à l'organisateur de cliquer sur le bouton "accepter" pour valider l'inscription.

Course de la Ville 2025

Page d'accueil

### Bienvenue sur votre tableau de bord, Organisateur

Déconnexion

#### Participants à accepter

Voici la liste des participants qui ont demandé à s'inscrire à vos courses.

**Nom** : Garnier Jean  
**Course demandée** : 5  
**Email** : jean.garnier@email.fr

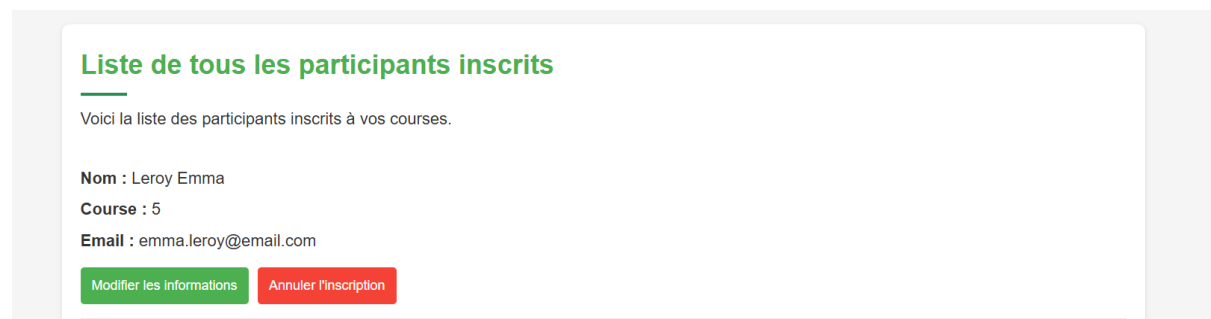
Accepter

Sur le tableau de bord de l'organisateur, les participants en attente d'inscription à la course sont affichés ici. Pour accepter l'inscription, il suffit de cliquer sur le bouton "Accepter" et les

informations du participants se retrouve dans la partie “liste de tous les participants inscrits” du tableau de bord.

## 5.7) La page d’annulation d’une inscription

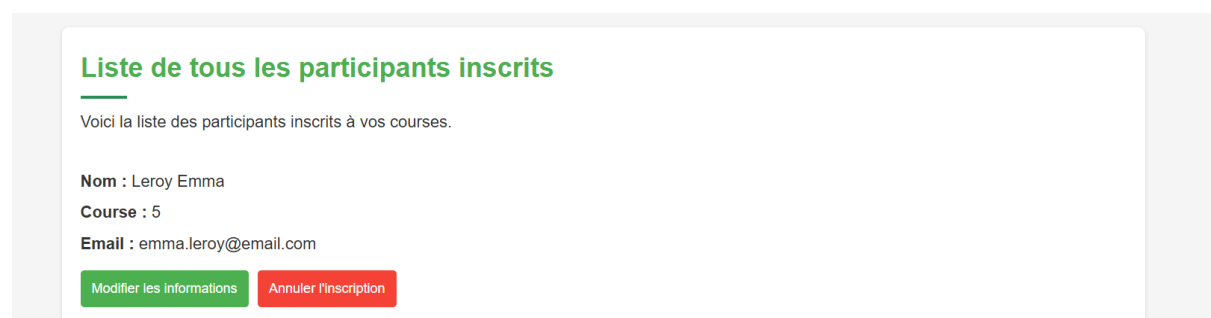
Cette section du tableau de bord permet de retrouver tous les participants inscrits à la course. Il suffit à l’organisateur de cliquer sur le bouton “annuler” pour annuler l’inscription ou de cliquer sur “modifier les informations” pour modifier les informations personnelles d’un participant.



Sur le tableau de bord de l’organisateur, les participants inscrits à la course sont affichés ici. Pour annuler l’inscription, il suffit de cliquer sur le bouton “Annuler” et les informations du participants se retrouve dans la partie “participants avec inscription annulée” du tableau de bord.

## 5.8) La page de modification des informations

Cette section du tableau de bord permet de modifier les informations personnelles d’un participant à partir d’un formulaire. Il suffit à l’organisateur de cliquer sur le bouton “modifier les informations” depuis le tableau de bord.



## Modifier les informations du participant

Nom :

Leroy

Prénom :

Emma

Date de naissance :

30/03/2000



Email :

emma.leroy@email.com

Téléphone :

0654321897

Sauvegarder les modifications

### 5.9) La page pour restaurer une inscription

Cette section du tableau de bord permet de restaurer l'inscription d'un participant qui a été précédemment annulée. Il suffit de cliquer sur le bouton "restaurer l'inscription" pour que celle-ci soit à nouveau validée.

#### Participants avec inscription annulée

Voici la liste des participants dont l'inscription a été annulée.

**Nom :** Martin Sophie

**Course :** 10

**Email :** sophie.martin@email.com

Restaurer l'inscription

### 5.10) La page des participants

Cette page permet de visualiser tous les participants inscrits à la course. Ainsi, chaque participant peut retrouver son inscription et donc voir si celle-ci a été validée par un organisateur. Un bouton permet d'être redirigé rapidement vers la page d'accueil.



## 6) Tests

Dans cette partie, je vais tester manuellement mon application web ainsi que toutes les pages et fonctionnalités. Le but est de valider le bon fonctionnement entre l'interface et la base de données.

Mes tests seront organisés de la façon suivante : l'identifiant du test et une suite claire et explicite [ID] : [Page] - [Action] - [Résultat attendu] - [Résultat obtenu] - [Statut], des tests par catégories et un tableau de suivi des tests.

### 6.1) Tests d'interface

L'objectif des tests d'interface (UI) est de vérifier l'affichage des pages et l'ergonomie de l'application.

	A	B	C	D	E	F
1	ID	Page	Action	Résultat attendu	Résultat obtenu	Statut
2	UI-01	Accueil	Vérifier l'alignement des boutons	Boutons bien alignés	OK	Réussi
3	UI-02	Accueil	Vérifier la lisibilité du texte	Texte lisible	OK	Réussi
4	UI-03	Accueil	Cliquer sur le bouton "connexion organisateur"	Redirection vers la page de connexion de l'organisateur	OK	Réussi
5	UI-04	Accueil	Cliquer sur le bouton "s'inscrire à la course"	Redirection vers le formulaire d'inscription à la course	OK	Réussi
6	UI-05	Accueil	Cliquer sur le bouton "voir la liste des inscrits"	Redirection vers la page des participants confirmés	OK	Réussi
7	UI-06	Accueil	Passer la souris sur le bouton "connexion organisateur"	Le bouton change de couleur	OK	Réussi
8	UI-07	Accueil	Passer la souris sur le bouton "s'inscrire à la course"	Le bouton change de couleur	OK	Réussi
9	UI-08	Accueil	Passer la souris sur le bouton "voir la liste des inscrits"	Le bouton change de couleur	OK	Réussi
10	UI-09	Connexion	Vérifier l'alignement des boutons	Boutons bien alignés	OK	Réussi
11	UI-10	Connexion	Vérifier la lisibilité du texte	Texte lisible	OK	Réussi
12	UI-11	Connexion	Cliquer sur le bouton "page d'accueil"	Redirection vers la page d'accueil	OK	Réussi
13	UI-12	Connexion	Passer la souris sur le bouton "page d'accueil"	Le bouton change de couleur	OK	Réussi
14	UI-13	Connexion	Passer la souris sur le bouton "se connecter"	Le bouton change de couleur	OK	Réussi
15	UI-14	Tableau de bord	Vérifier l'alignement des boutons	Boutons bien alignés	OK	Réussi
16	UI-15	Tableau de bord	Vérifier la lisibilité du texte	Texte lisible	OK	Réussi
17	UI-16	Tableau de bord	Cliquer sur le bouton "page d'accueil"	Redirection vers la page d'accueil	OK	Réussi
18	UI-17	Tableau de bord	Cliquer sur le bouton "déconnexion"	Redirection vers la page d'accueil	OK	Réussi
19	UI-18	Tableau de bord	Cliquer sur le bouton "modifier les informations"	Redirection vers le formulaire de modification du participant	OK	Réussi
20	UI-19	Tableau de bord	Passer la souris sur le bouton "page d'accueil"	Le bouton change de couleur	OK	Réussi
21	UI-20	Inscription	Vérifier l'alignement des boutons	Boutons bien alignés	OK	Réussi
22	UI-21	Inscription	Vérifier la lisibilité du texte	Texte lisible	OK	Réussi
23	UI-22	Inscription	Cliquer sur le bouton "page d'accueil"	Redirection vers la page d'accueil	OK	Réussi
24	UI-23	Inscription	Passer la souris sur le bouton "page d'accueil"	Le bouton change de couleur	OK	Réussi
25	UI-24	Inscription	Passer la souris sur le bouton "s'inscrire"	Le bouton change de couleur	OK	Réussi
26	UI-25	Inscription	Vérifier que les placeholders sont visibles	Placeholders présents	OK	Réussi
27	UI-26	Participants	Vérifier l'alignement des boutons	Boutons bien alignés	OK	Réussi
28	UI-27	Participants	Vérifier la lisibilité du texte	Texte lisible	OK	Réussi
29	UI-28	Participants	Cliquer sur le bouton "page d'accueil"	Redirection vers la page d'accueil	OK	Réussi
30	UI-29	Participants	Passer la souris sur le bouton "page d'accueil"	Le bouton change de couleur	OK	Réussi

## 6.2) Tests fonctionnels

Les tests fonctionnels (FUNC) consistent à valider le fonctionnement de l'inscription, de la connexion. Il faut vérifier que chaque fonctionnalité remplit son rôle.

ID	Page	Action	Résultat attendu	Résultat obtenu	Statut
FUNC-01	Connexion	Se connecter avec des identifiants valides	Connexion réussie	OK	Réussi
FUNC-02	Tableau de bord	Cliquer sur "modifier les informations" et modifier un champ	Sauvegarder les modifications	OK	Réussi
FUNC-03	Tableau de bord	Cliquer sur "accepter"	Le participant est déplacé dans la "Liste de tous les participants inscrits"	OK	Réussi
FUNC-04	Tableau de bord	Cliquer sur "annuler l'inscription"	Le participant est déplacé dans la "Liste des participants avec inscription annulée"	OK	Réussi
FUNC-05	Tableau de bord	Cliquer sur "restaurer l'inscription"	Le participant est déplacé dans la "Liste de tous les participants inscrits"	OK	Réussi
FUNC-06	Inscription	Remplir le formulaire correctement et valider	Utilisateur enregistré	OK	Réussi

## 6.3) Tests de base de données

Les tests de base de données (DB) consistent à vérifier que les données sont bien enregistrées, modifiées ou supprimées après une action sur le site.

ID	Page	Table	Action	Résultat attendu	Résultat obtenu	Statut
DB-01	Tableau de bord	inscriptions	Accepter l'inscription d'un participant	Le statut de l'inscription doit être "confirmée"	OK	Réussi
DB-02	Tableau de bord	inscriptions	Annuler l'inscription d'un participant	Le statut de l'inscription doit être "annulée"	OK	Réussi
DB-03	Tableau de bord	inscriptions	Restaurer l'inscription d'un participant	Le statut de l'inscription doit être "confirmée"	OK	Réussi
DB-04	Tableau de bord	participants	Modifier les informations d'un participant	Les informations changées doivent être modifiées	OK	Réussi
DB-05	Inscription	participants et inscriptions	Remplir le formulaire d'inscription	Nouveau participant et nouvelle inscription ajoutée avec les bonnes informations	OK	Réussi

## 6.4) Tests d'erreurs

L'objectif des tests d'erreurs (ERR) est de vérifier la robustesse du système face aux erreurs et attaques potentielles.

ID	Page	Action	Résultat attendu	Résultat obtenu	Statut	Commentaire
ERR-01	Connexion	Se connecter avec un compte inexistant	Message d'erreur "Aucun utilisateur trouvé avec cet email"	OK	Réussi	
ERR-02	Connexion	Laisser le champ "email" vide et valider	Message d'erreur "Veuillez renseigner ce champ"	OK	Réussi	
ERR-03	Connexion	Laisser le champ "mot de passe" vide et valider	Message d'erreur "Veuillez renseigner ce champ"	OK	Réussi	
ERR-04	Connexion	Saisir un mot de passe incorrect pour un email existant	Message d'erreur "Email ou mot de passe incorrect"	OK	Réussi	
ERR-05	Connexion	Saisir un email incorrect pour un mot de passe existant	Message d'erreur "Email ou mot de passe incorrect"	OK	Réussi	
ERR-06	Tableau de bord - Modifier les informations	Modifier le champ "email" en email invalide	Message d'erreur "L'email est invalide (format email.test@domaine.fr attendu)"	KO	échoué	Email testé : jean.garnier@e -> aucune erreur affichée et les modifications se sont enregistrées
ERR-07	Tableau de bord - Modifier les informations	Laisser le champ "numéro de téléphone" vide et valider	Message d'erreur "Veuillez renseigner ce champ"	OK	Réussi	
ERR-08	Tableau de bord - Modifier les informations	Saisir un numéro de téléphone invalide (0123 et 0123abc)	Message d'erreur "Le numéro de téléphone doit contenir exactement 10 chiffres"	KO	échoué	Numéro de téléphone testé : 0123 et 0123abc -> aucune erreur affichée et les modifications se sont enregistrées
ERR-09	Tableau de bord - Modifier les informations	Entrer " " OR "1" dans le champ "nom"	Message d'erreur "Le nom ne peut contenir que des lettres et des espaces"	KO	échoué	Le champ nom est testé " OR "1" -> aucune erreur affichée et l'inscription est enregistrée
ERR-10	Tableau de bord - Modifier les informations	Entrer " " OR "1" dans le champ "prénom"	Message d'erreur "Le prénom ne peut contenir que des lettres et des espaces"	OK	Réussi	
ERR-11	Tableau de bord - Modifier les informations	Entrer " " OR "1" dans le champ "email"	Message d'erreur "L'email est invalide (format email.test@domaine.fr attendu)"	OK	Réussi	
ERR-12	Inscription	Laisser le champ "email" vide et valider	Message d'erreur "Veuillez renseigner ce champ"	OK	Réussi	
ERR-13	Inscription	Saisir une adresse email invalide (test@com)	Message d'erreur "L'email est invalide (format email.test@domaine.fr attendu)"	KO	échoué	Email testé : test@com -> aucune erreur affichée et l'inscription est enregistrée
ERR-14	Inscription	Saisir un email déjà utilisé	Message d'erreur "Cet email est déjà utilisé"	KO	échoué	Email testé : jean.garnier@gmail.fr -> aucune erreur affichée et l'inscription est enregistrée
ERR-15	Inscription	Saisir un numéro de téléphone invalide (0123 et 0123abc)	Message d'erreur "Le numéro de téléphone doit contenir exactement 10 chiffres"	KO	échoué	Numéro de téléphone testé : 0123 -> aucune erreur affichée et l'inscription est enregistrée
ERR-16	Inscription	Ne pas entrer de date de naissance et valider	Message d'erreur "Veuillez renseigner ce champ"	OK	Réussi	
ERR-17	Inscription	Entrer " " OR "1" dans le champ "nom"	Message d'erreur "Le nom ne peut contenir que des lettres et des espaces"	KO	échoué	Le champ nom est testé " OR "1" -> aucune erreur affichée et l'inscription est enregistrée
ERR-18	Inscription	Entrer " " OR "1" dans le champ "prénom"	Message d'erreur "Le prénom ne peut contenir que des lettres et des espaces"	OK	Réussi	
ERR-19	Inscription	Entrer " " OR "1" dans le champ "email"	Message d'erreur "L'email est invalide (format email.test@domaine.fr attendu)"	OK	Réussi	

### 6.4.1) [ERR-06] [Tableau de bord - Modifier les informations] [Modifier le champ "email" en email valide] [Message d'erreur "L'email est invalide (format email.test@domaine.fr attendu)"]

Ce test consiste à modifier le champ "email" en un email invalide. Le premier email testé est "jean.garnier" qui affiche une erreur "Veuillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "jean.garnier".

Le deuxième email testé est "jean.garnier@" qui affiche une erreur "Veuillez saisir la partie manquante après le symbole "@". L'adresse "jean.garnier@" est incomplète."

Le troisième email testé est jean.garnier@e” et n’affiche pas d’erreur. Cet email invalide s’enregistre dans les informations modifiées de l’utilisateur et dans la base de données ce qui ne devrait pas arriver.

Il a fallu modifier le code PHP de la page “modifier-participants.php”.

```
// Validation de l'email avec une expression régulière
if (!preg_match("/^[a-z0-9._-]+@[a-z0-9.-]+\.[a-z]{2,6}$/i",
$email)) {
    $email_error = "L'email est invalide (format
email.test@domaine.fr attendu).";
} else {
    // Mise à jour des informations dans la base de données si l'email
est valide
    $part_nom = $_POST['part_nom'];
    $part_prenom = $_POST['part_prenom'];
    $part_date_de_naissance = $_POST['part_date_de_naissance'];
    $part_telephone = $_POST['part_telephone'];
```

```
<label for="part_email">Email :</label>
<input type="email" id="part_email" name="part_email" value="<?=
htmlspecialchars($participant['part_email']) ?>" required>

<?php if (isset($email_error)) { echo "<p>$email_error</p>"; }
?>
```

J’ajoute une expression régulière (regex) avec preg\_match() qui permet de vérifier si une chaîne de caractères respecte une structure définie.

Par exemple, l’expression : !preg\_match("/^[a-z0-9.\_-]+@[a-z0-9.-]+\.[a-z]{2,6}\$/i", \$email)) signifie que l’email accepte une ou plusieurs lettres minuscules, chiffres, points, tirets et underscore [a-z0-9.\_-].

Il est obligatoire d’ajouter le caractère @.

Pour le domaine, le regex accepte une ou plusieurs lettres minuscules, chiffres, points et tirets [a-z0-9.-].

Il est obligatoire d’ajouter un point avant l’extension du domaine \.

Pour l’extension (com, fr, net, etc), il accepte entre 2 et 6 lettres [a-z]{2,6}.

Le symbole \$ indique la fin de la chaîne et donc qu’il ne doit rien y avoir avec l’extension.

Enfin, la vérification est insensible à la casse /i.

Après modification, un test d’erreur a de nouveau été effectué. Maintenant, l’email testé est “jean.garnier@e” qui affiche l’erreur “L’email est invalide (format email.test@domaine.fr attendu)”. Le champ “email” du formulaire n’accepte plus que l’email de test “jean.garnier@email.fr”.

Le test d’erreur 6 de la page de tableau de bord pour modifier les informations d’un participant en entrant un email invalide est résolu. Le test est maintenant concluant.

Email :

jean.garnier@email.fr

L'email est invalide (format email.test@domaine.fr attendu).

#### 6.4.2) [ERR-08] [Tableau de bord - Modifier les informations] [Saisir un numéro de téléphone invalide] [Message d'erreur "Le numéro de téléphone doit contenir exactement 10 chiffres."]

Ce test consiste à modifier le champ "numéro de téléphone" en un numéro de téléphone invalide. Le premier numéro testé est "0123" qui s'enregistre dans les informations modifiées de l'utilisateur et dans la base de données ce qui ne devrait pas arriver.

Le deuxième numéro testé est "0123abc" et celui-ci s'enregistre également dans les modifications et la base de données.

Il a fallu modifier le code PHP de la page "modifier-participants.php".

```
// Validation du numéro de téléphone (10 chiffres requis)
if (!preg_match('/^\d{10}$/', $part_telephone)) {
    $errors['telephone'] = "Le numéro de téléphone doit contenir
exactement 10 chiffres.";
}
```

```
<label for="part_telephone">Téléphone :</label>
<input type="tel" id="part_telephone" name="part_telephone"
value="<?= htmlspecialchars($participant['part_telephone']) ?>"
required>
<?php if (isset($errors['telephone'])) { echo
"<p>{$errors['telephone']}</p>"; } ?>
```

Cette expression régulière `!preg_match('/^\d{10}$/', $part_telephone)` valide un numéro de téléphone composé de 10 chiffres.

Le symbole `^` indique le début de la chaîne ce qui empêche d'avoir des caractères avant. Ensuite, le numéro de téléphone doit être composé de chiffres `\d`.

Il doit aussi y avoir exactement 10 chiffres `{10}`. Le symbole `$` indique la fin de la chaîne et donc empêche d'avoir des caractères après.

Après modification, un test d'erreur a de nouveau été effectué. Maintenant, le numéro de téléphone testé est "0123" qui affiche l'erreur "Le numéro de téléphone doit contenir exactement 10 chiffres". Le champ "numéro de téléphone" du formulaire n'accepte plus qu'un numéro valide.

Le test d'erreur 8 de la page de tableau de bord pour modifier les informations d'un participant en entrant un numéro de téléphone invalide est résolu. Le test est maintenant concluant.

### 6.4.3) [ERR-09] [Tableau de bord - Modifier les informations] [Entrer " ' OR '1'='1 " dans le champ "nom"] [Message d'erreur "Le nom ne peut contenir que des lettres et des espaces."]

Ce test consiste à entrer une injection SQL dans le champ nom du formulaire du tableau de bord. Cette erreur valide le formulaire et le nom ' OR '1'='1 s'enregistre dans la base de données.

Il a fallu modifier le fichier "modifier-participant.php" et "traitement-modifier-participant.php" :

modifier-participant.php :

```
<label for="part_nom">Nom :</label>
<input type="text" id="part_nom" name="part_nom" value="<?=
htmlspecialchars($participant['part_nom']) ?>" required>
<?php if (isset($errors['nom'])) { echo "<p
class='error'>{$errors['nom']}</p>"; } ?>

<label for="part_prenom">Prénom :</label>
<input type="text" id="part_prenom" name="part_prenom"
value="<?= htmlspecialchars($participant['part_prenom']) ?>" required>
<?php if (isset($errors['prenom'])) { echo "<p
class='error'>{$errors['prenom']}</p>"; } ?>
```

traitement-modifier-participant.php :

```
// Vérification des caractères pour nom et prénom (lettres et espaces
uniquement)
if (!preg_match('/^[a-zA-ZÀ-ÿ\s-]+$/', $nom)) {
    $errors['nom'] = "Le nom ne peut contenir que des lettres, des
espaces et des tirets.";
}

if (!preg_match('/^[a-zA-ZÀ-ÿ\s-]+$/', $prenom)) {
    $errors['prenom'] = "Le prénom ne peut contenir que des lettres,
des espaces et des tirets.";
}
```

Cette expression régulière !preg\_match('/^[a-zA-ZÀ-ÿ\s-]+\$/', \$nom) ou !preg\_match('/^[a-zA-ZÀ-ÿ\s-]+\$/', \$prenom) est utilisée pour valider un nom ou un prénom.

Le symbole ^ indique le début de la chaîne ce qui empêche d'avoir des caractères avant. Ensuite, l'expression accepte les lettres minuscules et majuscules de l'alphabet a-zA-Z, les lettres accentuées À-ÿ, autorise les espaces \s et le tiret -.

Le symbole + indique qu'il doit y avoir au moins un caractère parmi ceux autorisés et le symbole \$ indique la fin de la chaîne ce qui évite d'ajouter des caractères interdits après.

Après modification, un test d'erreur a de nouveau été effectué. Maintenant, le nom testé est ' OR '1'='1 qui affiche l'erreur "Le nom ne peut contenir que des lettres et des espaces".



Le test d'erreur 14 de la page d'inscription pour s'inscrire à la course en entrant une injection SQL dans le champ nom est résolu. Le test est maintenant concluant.

#### 6.4.4) [ERR-10] [Inscription] [Saisir une adresse email invalide] [Message d'erreur "L'email est invalide (format email.test@domaine.fr attendu)"]

Ce test consiste à modifier le champ "email" en un email invalide. Le premier email testé est "test" qui affiche une erreur "Veuillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "test".

Le deuxième email testé est "test@" qui affiche une erreur "Veuillez saisir la partie manquante après le symbole "@". L'adresse "test@" est incomplète."

Le troisième email testé est "test@com" et n'affiche pas d'erreur. Cet email invalide s'enregistre dans les informations modifiées de l'utilisateur et dans la base de données ce qui ne devrait pas arriver.

Il a fallu modifier le code PHP de la page "inscription.php" et le fichier "traitement-inscription.php" :

inscription.php :

```
<?php
session_start();

// Si des erreurs existent en session, on les récupère
if (isset($_SESSION['errors'])) {
    $errors = $_SESSION['errors'];
    unset($_SESSION['errors']);
}
?>
```

```
<label for="email">Email :</label>
<input type="email" id="email" name="email" required
placeholder="Votre email" />
<?php if (isset($errors['email'])) {echo "<p>" . $errors['email']
. "</p>";}?>
```

traitement-inscription.php :

```
// Initialisation de $errors
$errors = [];
```

```
// Validation de l'email
if (!preg_match("/^[a-z0-9._-]+@[a-z0-9.-]+\.[a-z]{2,6}$/i",
$email)) {
    $errors['email'] = "L'email est invalide (format attendu :
email.test@domaine.fr).";
}
```

L'expression régulière utilisée est la même que pour l'erreur 6.

Après modification, un test d'erreur a de nouveau été effectué. Maintenant, l'email testé est "test@com" qui affiche l'erreur "L'email est invalide (format email.test@domaine.fr attendu)". Le champ "email" du formulaire n'accepte plus que l'email de test "lucas@test.com".

Le test d'erreur 10 de la page d'inscription pour s'inscrire à la course en entrant un email invalide est résolu. Le test est maintenant concluant.

#### 6.4.5) [ERR-11] [Inscription] [Saisir un email déjà utilisé] [Message d'erreur "Cet email est déjà utilisé"]

Ce test consiste à entrer un email déjà utilisé. L'email entré est "jean.garnier@email.fr". Cet email invalide permet de valider le formulaire mais ne s'enregistre pas dans la base de données.

Il a fallu modifier le code PHP du fichier "traitement-inscription.php" :

```
if ($participant) {  
    // Si l'email existe déjà, on ajoute l'erreur  
    $errors['email'] = "Cet email est déjà utilisé.";  
}
```

Après modification, un test d'erreur a de nouveau été effectué. Maintenant, l'email testé est "jean.garnier@email.fr" qui affiche l'erreur "Cet email est déjà utilisé".

Le test d'erreur 11 de la page d'inscription pour s'inscrire à la course en entrant un email déjà utilisé est résolu. Le test est maintenant concluant.

#### 6.4.6) [ERR-12] [Inscription] [Saisir un numéro de téléphone invalide] [Message d'erreur "Le numéro de téléphone doit contenir exactement 10 chiffres"]

Ce test consiste à entrer un numéro de téléphone invalide (0123 et 0123abc) dans le champ "numéro de téléphone" de la page d'inscription. Cette erreur valide le formulaire et le numéro s'enregistre dans la base de données.

Il a fallu modifier le code PHP du fichier "inscription.php" et "traitement-inscription.php" :

inscription.php :

```
<label for="telephone">Numéro de téléphone :</label>  
<input type="tel" id="telephone" name="telephone" required  
placeholder="Votre numéro de téléphone" />  
<?php if (isset($errors['telephone'])) { echo  
"<p>{$errors['telephone']}</p>"; } ?>
```

traitement-inscription.php :

```
// Validation du numéro de téléphone (doit contenir exactement 10  
chiffres)  
if (!preg_match('/^\d{10}$/', $telephone)) {  
    $errors['telephone'] = "Le numéro de téléphone doit contenir  
exactement 10 chiffres.";
```

```
}
```

Cette expression régulière est la même que pour l'erreur 8.

Après modification, un test d'erreur a de nouveau été effectué. Maintenant, les numéros de téléphone testés sont "0123" et "0123abc" qui affiche l'erreur "Le numéro de téléphone doit contenir exactement 10 chiffres".

Le test d'erreur 12 de la page d'inscription pour s'inscrire à la course en entrant un numéro de téléphone est résolu. Le test est maintenant concluant.

#### 6.4.7) [ERR-14] [Inscription] [Entrer " ' OR '1'='1 " dans le champ "nom"] [Message d'erreur "Le nom ne peut contenir que des lettres et des espaces."]

Ce test consiste à entrer une injection SQL dans le champ nom de la page d'inscription. Cette erreur valide le formulaire et le nom ' OR '1'='1 s'enregistre dans la base de données.

Il a fallu modifier le fichier "inscription.php" et "traitement-inscription.php" :

inscription.php :

```
<label for="nom">Nom :</label>
<input type="text" id="nom" name="nom" required placeholder="Votre
nom" value="<?php echo isset($_POST['nom']) ?
htmlspecialchars($_POST['nom']) : ' '; ?>" />
<?php if (isset($errors['nom'])) { echo "<p
class='error'>{$errors['nom']}</p>"; } ?>

<label for="prenom">Prénom :</label>
<input type="text" id="prenom" name="prenom" required
placeholder="Votre prénom" value="<?php echo isset($_POST['prenom']) ?
htmlspecialchars($_POST['prenom']) : ' '; ?>" />
<?php if (isset($errors['prenom'])) { echo "<p
class='error'>{$errors['prenom']}</p>"; } ?>
```

traitement-inscription.php :

```
// Vérifier que le nom et prénom ne contiennent que des lettres et
des espaces
if (!preg_match('/^[a-zA-Z\s]+$/ ', $nom)) {
    $errors['nom'] = "Le nom ne peut contenir que des lettres et des
espaces.";
}

if (!preg_match('/^[a-zA-Z\s]+$/ ', $prenom)) {
    $errors['prenom'] = "Le prénom ne peut contenir que des lettres
et des espaces.";
}
```

Cette expression régulière est la même que pour l'erreur 9.

Après modification, un test d'erreur a de nouveau été effectué. Maintenant, le nom testé est ' OR '1'='1 qui affiche l'erreur "Le nom ne peut contenir que des lettres et des espaces".

Le test d'erreur 14 de la page d'inscription pour s'inscrire à la course en entrant une injection SQL dans le champ nom est résolu. Le test est maintenant concluant.

## 6.5) Tests de compatibilité des navigateurs

Ces tests (NAV) consistent à vérifier que le site fonctionne correctement et s'affiche correctement sur différents navigateurs. Ici, les navigateurs testés sont Chrome, Firefox et Edge.

ID	Page	Navigateur	Résultat attendu	Résultat obtenu	Statut
NAV-01	Accueil	Chrome	La page d'accueil s'affiche et fonctionne correctement	OK	Réussi
NAV-02	Accueil	Firefox	La page d'accueil s'affiche et fonctionne correctement	OK	Réussi
NAV-03	Accueil	Edge	La page d'accueil s'affiche et fonctionne correctement	OK	Réussi
NAV-04	Connexion	Chrome	La page de connexion s'affiche et fonctionne correctement	OK	Réussi
NAV-05	Connexion	Firefox	La page de connexion s'affiche et fonctionne correctement	OK	Réussi
NAV-06	Connexion	Edge	La page de connexion s'affiche et fonctionne correctement	OK	Réussi
NAV-07	Inscription	Chrome	Le formulaire d'inscription s'affiche et fonctionne correctement	OK	Réussi
NAV-08	Inscription	Firefox	Le formulaire d'inscription s'affiche et fonctionne correctement	OK	Réussi
NAV-09	Inscription	Edge	Le formulaire d'inscription s'affiche et fonctionne correctement	OK	Réussi
NAV-10	Participants	Chrome	La page des participants s'affiche et fonctionne correctement	OK	Réussi
NAV-11	Participants	Firefox	La page des participants s'affiche et fonctionne correctement	OK	Réussi
NAV-12	Participants	Edge	La page des participants s'affiche et fonctionne correctement	OK	Réussi
NAV-13	Tableau de bord	Chrome	Le tableau de bord s'affiche et fonctionne correctement	OK	Réussi
NAV-14	Tableau de bord	Firefox	Le tableau de bord s'affiche et fonctionne correctement	OK	Réussi
NAV-15	Tableau de bord	Edge	Le tableau de bord s'affiche et fonctionne correctement	OK	Réussi

# 7) Documentation technique v1.0

## 1) Introduction

### 1.1) Contexte du projet

Une commune souhaite organiser une course à pied en solo proposant plusieurs distances à parcourir : 5km, 10km et 15km. Jusqu'à présent, l'événement se déroulait sans plateforme en ligne. L'organisation de la course était compliquée car les inscriptions étaient gérées manuellement ce qui entraînait des erreurs et des confusions. L'événement manquait également de visibilité et une expérience utilisateur limitée pour les participants, qui ne pouvaient pas s'inscrire en ligne, consulter les informations sur la course ou vérifier leurs inscriptions.

De plus, cette plateforme contenant des informations personnelles sur les participants de la course, devra respecter les normes sur la protection des données personnelles (RGPD). Il sera donc essentiel de garantir un stockage sécurisé et de respecter le droit des participants qui pourront demander l'accès, la rectification et la suppression de leurs données.

## 1.2) Objectifs de la documentation

Cette documentation technique du projet d'organisation d'une course sportive comporte plusieurs objectifs tels que l'architecture, l'installation et la configuration des différents éléments, le fonctionnement, les tests et le déploiement, la maintenance et les évolutions de l'application web.

## 2) Sécurité et conformité RGPD

Les mesures de sécurité mises en place sont l'utilisation de requêtes préparées pour éviter les injections SQL, l'accès est restreint aux organisateurs pour la vérification et l'accès aux données, le stockage des informations de test fictives est sécurisé.

En termes de conformité du Règlement Général sur la Protection des Données (RGPD), les données personnelles fictives sont traitées en respectant ces normes. La plateforme permet aux utilisateurs d'exercer leurs droits d'accès, de rectification et de suppression de leurs données.

## 3) Architecture et conception

### 3.1) Technologies utilisées

La plateforme est composée d'une architecture client-serveur avec une partie front-end pour l'interface utilisateur et d'une partie back-end pour le traitement des données.

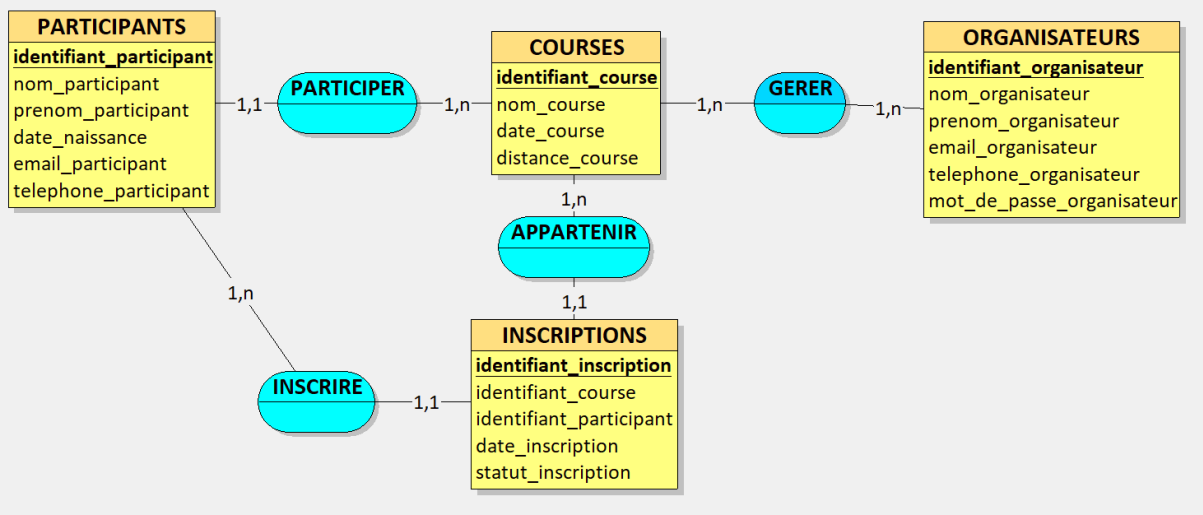
Les technologies utilisées sont, pour la partie front-end, HTML, CSS. Pour la partie back-end, PHP est utilisé. La base de données choisie est MySQL gérée via phpMyAdmin. Le serveur web local utilisé est XAMPP. Pour la sécurité, il faut utiliser des requêtes préparées pour éviter les injections SQL. L'accès doit être géré avec des comptes pour les organisateurs afin qu'il puisse modifier des informations sur les participants. Les composants communiquent entre eux, d'une part le front-end qui envoie les requêtes via des formulaires HTML et traités d'autre part par le back-end PHP. Les données sont récupérées et affichées via PHP. Enfin, la structure du code doit être organisée afin de permettre des modifications si nécessaire.

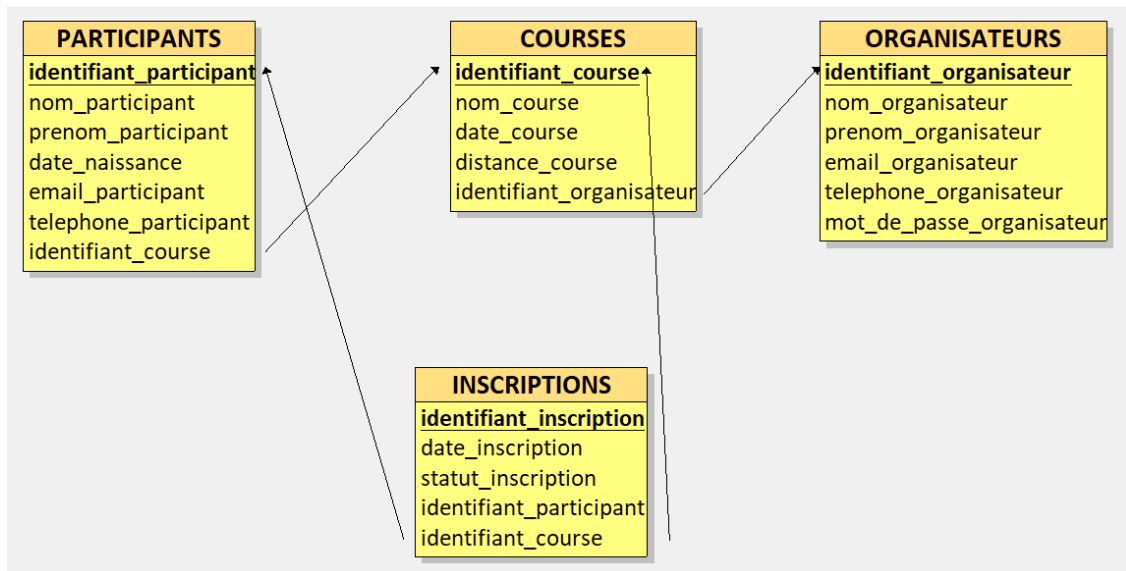
### 3.2) Modélisation et diagrammes

Pour concevoir et gérer la base de données, il faut créer le dictionnaire de données.

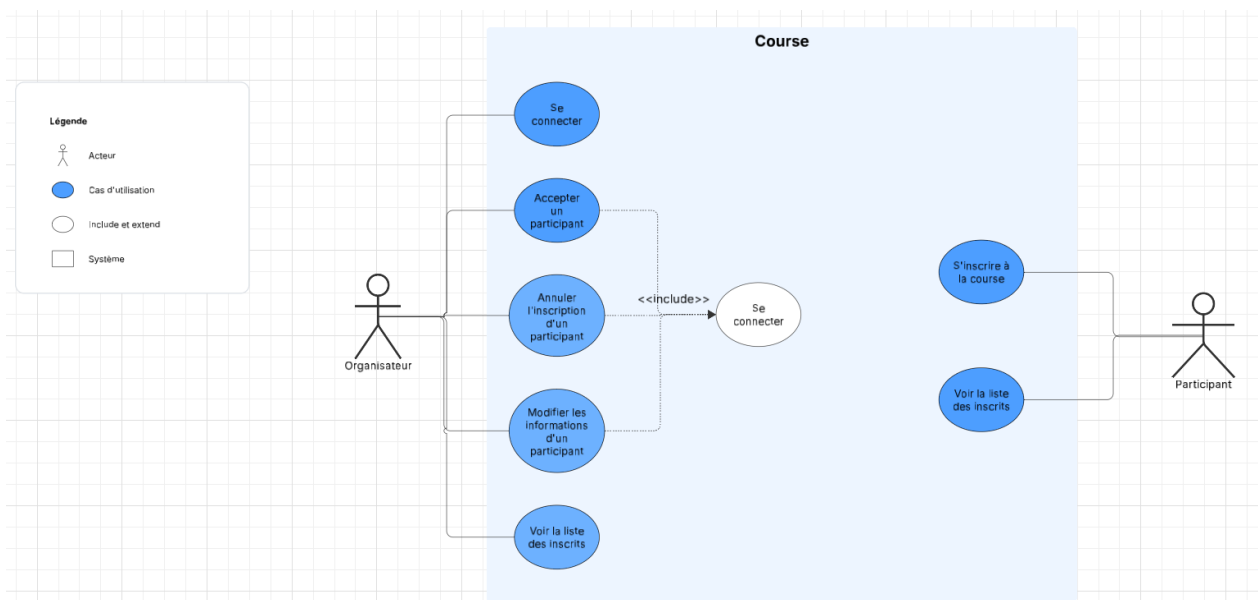
	A	B	C	D	E	F
	TABLE	Nom	Nom logique	Type	Longueur	Observation
1	<b>PARTICIPANTS</b>	identifiant_participant	part_id	INT (numérique)	11	Clé primaire, identifiant unique pour chaque participant.
2		nom_participant	part_nom	VARCHAR (caractère)	50	Nom de famille du participant
3		prenom_participant	part_prenom	VARCHAR (caractère)	50	Prénom du participant
4		date_naissance	part_date_naissance	DATE		Date de naissance du participant
5		email_participant	part_email	VARCHAR (caractère)	100	Email du participant
6		telephone_participant	part_telephone	VARCHAR (numérique)	10	Numéro de téléphone du participant
7	<b>ORGANISATEURS</b>	identifiant_organisateur	orga_id	INT (numérique)	11	Clé primaire, identifiant unique pour chaque organisateur.
8		nom_organisateur	orga_nom	VARCHAR (caractère)	50	Nom de famille de l'organisateur
9		prenom_organisateur	orga_prenom	VARCHAR (caractère)	50	Prénom de l'organisateur
10		email_organisateur	orga_email	VARCHAR (caractère)	100	Email de l'organisateur
11		telephone_organisateur	orga_telephone	VARCHAR (numérique)	10	Numéro de téléphone de l'organisateur
12	<b>COURSES</b>	identifiant_course	cour_id	INT (numérique)	11	Clé primaire, identifiant unique pour chaque course.
13		nom_course	cour_nom	VARCHAR (caractère)	50	Nom de la course
14		date_course	cour_date	DATE		Date de la course
15		distance_course	cour_distance	DECIMAL(5,2)	5	Distance de la course (5km, 10km, 15km)
16	<b>INSCRIPTIONS</b>	identifiant_inscription	inscr_id	INT (numérique)	11	Clé primaire, identifiant unique pour chaque inscription.
17		identifiant_course	cour_id	INT (numérique)	11	Clé étrangère, identifiant unique pour chaque course
18		identifiant_participant	part_id	INT (numérique)	11	Clé étrangère, identifiant unique pour chaque participant.
19		date_inscription	inscr_date	DATE		Date de l'inscription
20		statut_inscription	inscr_statut	VARCHAR (caractère)	20	Statut de l'inscription (en attente ou validée)
21						

Le modèle conceptuel de données (MCD) représente les données du système d'information de manière abstraite sous la forme d'entités, d'associations, d'attributs et de cardinalités. Enfin, je crée le modèle physique de données (MPD) qui est adapté aux caractéristiques techniques du système de gestion de bases de données. Il utilise des index, des types de données et optimise la performance.



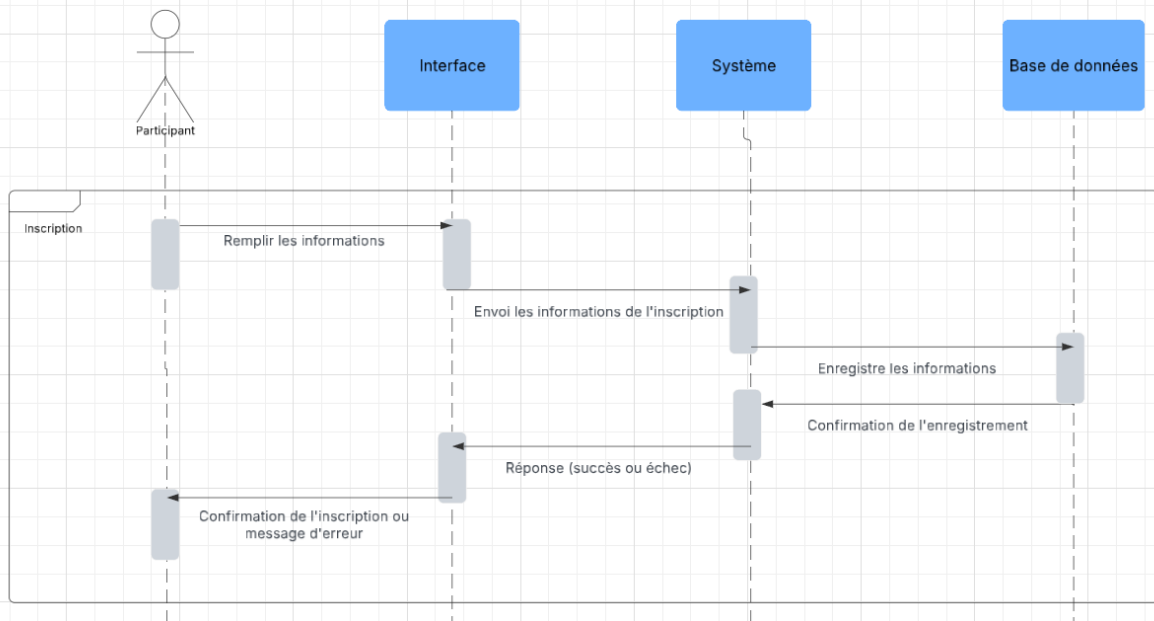


Le diagramme de cas d'utilisation est utilisé en langage de modélisation unifié (UML), il montre les interactions entre les utilisateurs et le système. Ici, les acteurs sont le participant et l'organisateur.

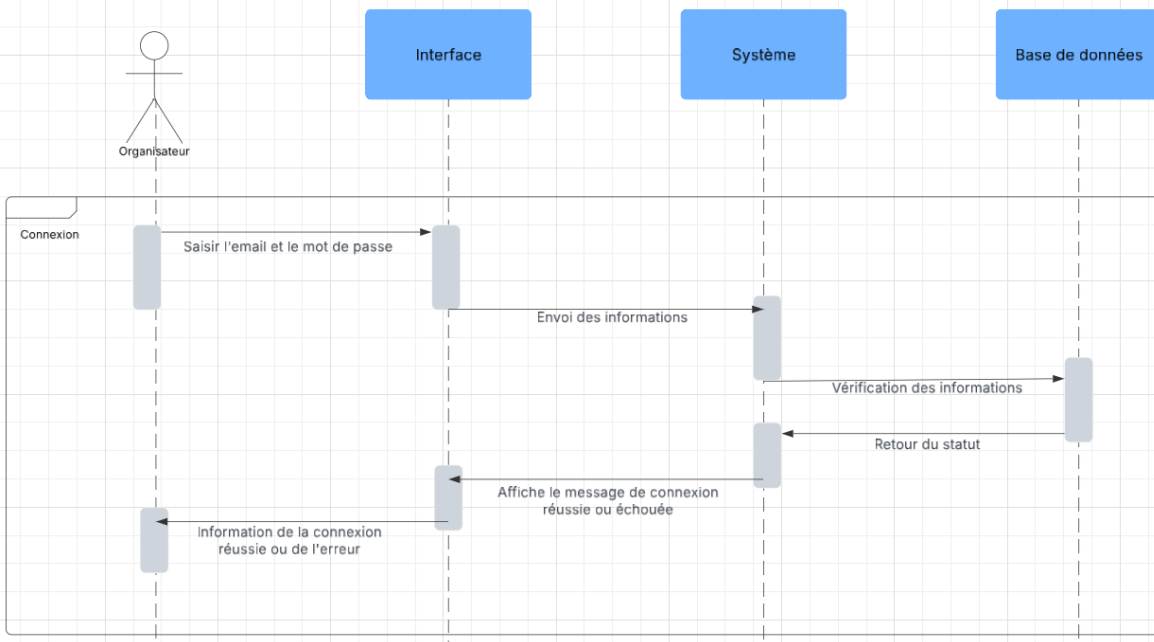


Le diagramme de séquence est également utilisé en langage de modélisation unifié (UML), il montre le déroulement des interactions entre les acteurs et le système. Ici, il y a plusieurs acteurs, le participant, l'organisateur, l'interface, le système et la base de données. Chaque diagramme représente une seule fonctionnalité dont l'inscription à la course, la connexion d'un organisateur, accepter ou annuler l'inscription d'un participant, modifier les informations d'un participant.

## 1. Inscription à la course

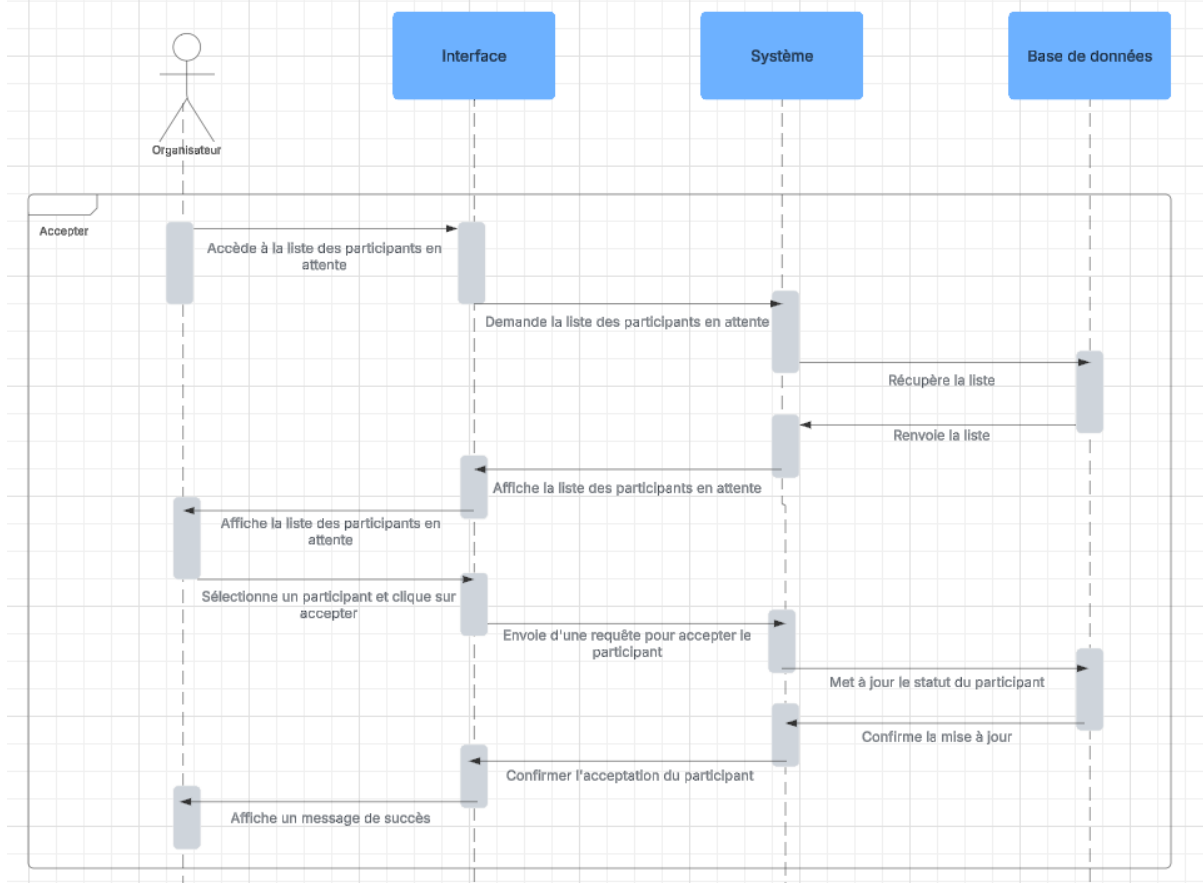


## 2. Connexion

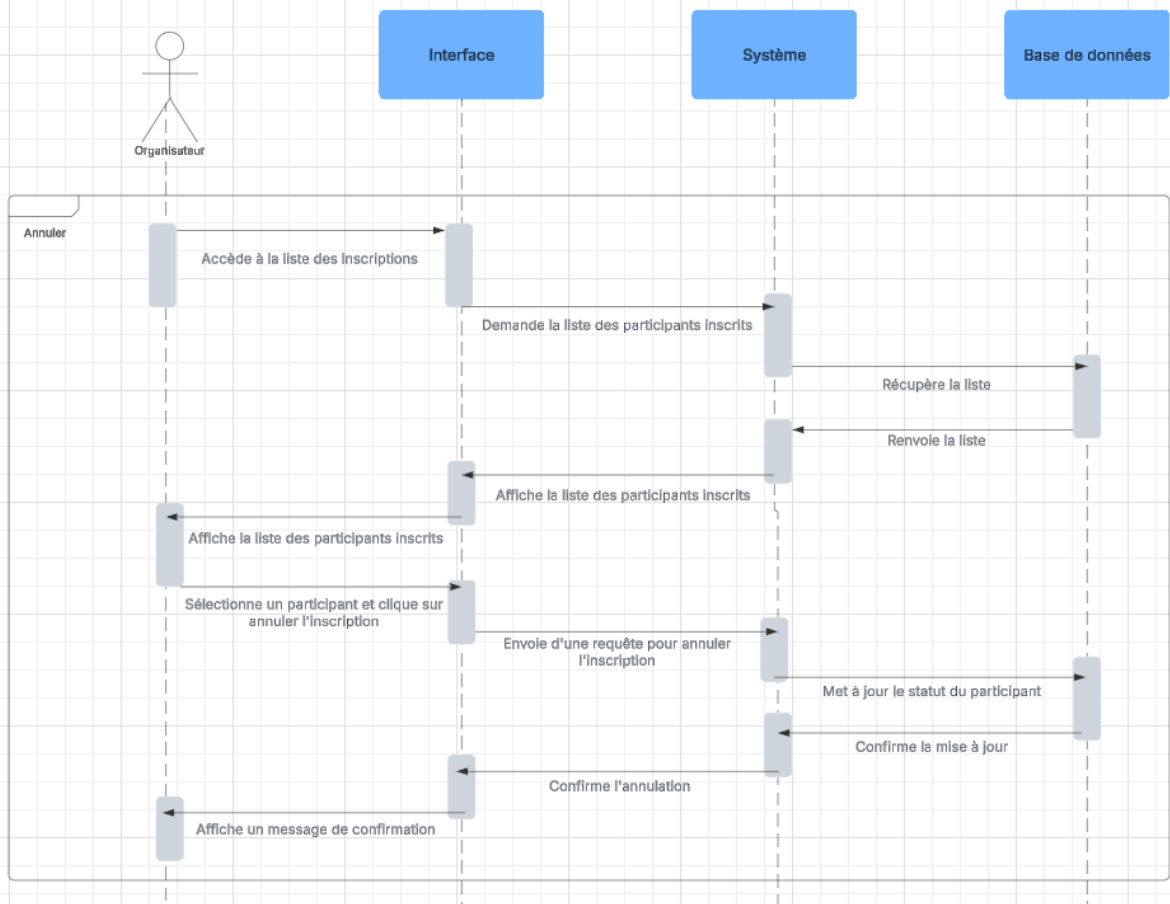


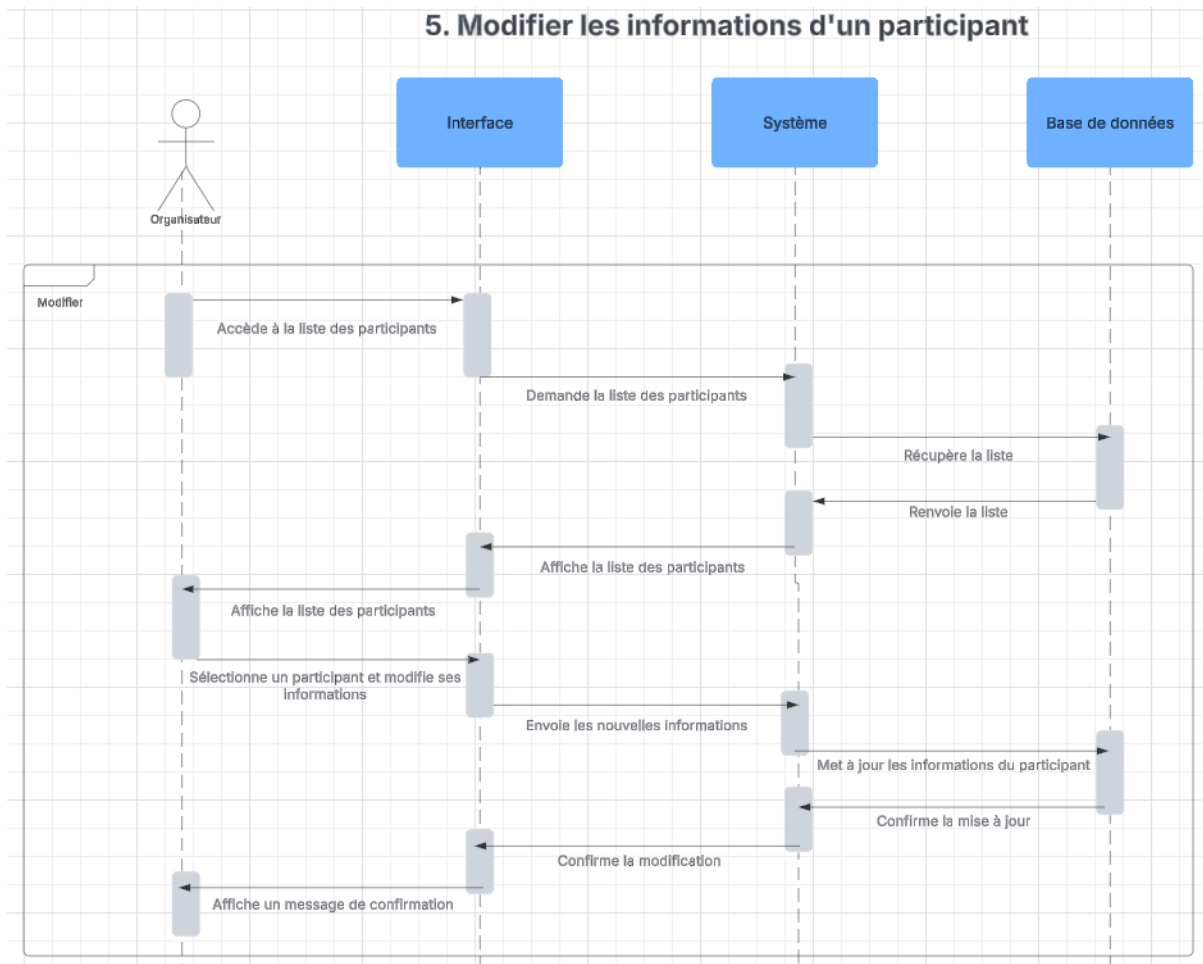


### 3. Accepter l'inscription d'un participant



#### 4. Annuler l'inscription d'un participant





## 4) Structure du projet

Voici la liste des principaux fichiers de code créés :

- index.html : page d'accueil du projet
- connexion.html : page de connexion pour les organisateurs
- connexion.php : page de connexion PHP
- tableau-de-bord.php : page de tableau de bord de l'organisateur
- deconnexion.php : permet à l'organisateur de se déconnecter
- accepter-participant.php : permet à l'organisateur d'accepter l'inscription d'un participant
- annuler-participant.php : permet à l'organisateur d'annuler l'inscription d'un participant
- modifier-participant.php : permet à l'organisateur de modifier les informations d'un participant
- traitement-modifier-participant.php : traitement PHP de modification des informations d'un participant
- restaurer-participant.php : permet à l'organisateur de restaurer l'inscription d'un participant
- inscription.php : formulaire d'inscription à la course
- traitement-inscription.php : traitement PHP du formulaire d'inscription
- participants.php : liste de tous les participants inscrits à la course
- config.php : configuration de la base de données

- connexion-bdd.php : connexion à la base de données
- styles.css : page de styles CSS
- gestion\_course.sql : fichier SQL de structure de base de données
- Readme.md : présentation du projet

J'utilise une convention de nommage de fichiers explicite décrivant la fonctionnalité comme index.html pour la page d'accueil, la configuration pour config.php, etc.

## 5) Prérequis

Dans le cadre de ce projet, le langage principal utilisé est PHP. Le développement s'effectue à l'aide de l'environnement de VS Code qui facilite l'écriture et le débogage du code. La base de données choisie est MySQL gérée via phpMyAdmin qui facilite la création, la modification et la suppression des tables et des données. Le serveur web local utilisé est XAMPP. Ce logiciel regroupe Apache, MySQL, PHP et Perl permettant de tester localement le projet et de reproduire les conditions d'un serveur de production. Enfin, le projet est conçu pour être facilement modifiable si des évolutions sont nécessaires grâce à l'utilisation du logiciel de gestion de versions Git.

## 6) Tests

### 6.1) Stratégie de tests

Les tests ont été réalisés sur l'ensemble de la plateforme afin d'assurer son bon fonctionnement et sa conformité.

L'objectif des tests d'interface (UI) est de vérifier l'affichage des pages et l'ergonomie de l'application. La méthode utilisée était de vérifier le rendu sur différents navigateurs, de vérifier les couleurs, la lisibilité du texte et les boutons.

Un exemple de test d'interface pourrait être : [ID : UI-01] [Page : Accueil] [Action : Vérifier l'alignement des boutons] [Résultat attendu : Boutons bien alignés] [Résultat obtenu : OK] [Statut : Réussi].

L'objectif des tests fonctionnels (FUNC) est de vérifier que chaque fonctionnalité remplit son rôle. La méthode était de tester chaque action utilisateur tel que la connexion, la validation d'un participant, le bon fonctionnement des formulaires et s'assurer que les boutons déclenchent les actions attendues.

Un exemple de test fonctionnel pourrait être : [ID : FUNC-01] [Page : Connexion] [Action : Se connecter avec des identifiants valides] [Résultat attendu : Connexion réussie] [Résultat obtenu : OK] [Statut : Réussi].

L'objectif des tests de base de données (DB) est de vérifier la cohérence des données stockées. La méthode était de vérifier l'insertion, la suppression des données dans la base, tester les relations entre les tables, vérifier la gestion des erreurs.

Un exemple de test de base de données pourrait être : [ID : DB-01] [Page : Tableau de bord] [Table : inscriptions] [Action : Accepter l'inscription d'un participant] [Résultat attendu : Le statut de l'inscription doit être "confirmée"] [Résultat obtenu : OK] [Statut : Réussi].

L'objectif des tests d'erreurs (ERR) est de vérifier la robustesse du système face aux erreurs et attaques potentielles. La méthode était de laisser des champs obligatoires vides, injecter des requêtes SQL dans les formulaires et effectuer des tentatives d'accès à la connexion sans identifiants valides.

Un exemple de test d'erreur pourrait être : [ID : ERR-01] [Page : Connexion] [Action : Se connecter avec un compte inexistant] [Résultat attendu : Message d'erreur "Aucun utilisateur trouvé avec cet email"] [Résultat obtenu : OK] [Statut : Réussi] [Commentaire : /]

L'objectif des tests de compatibilité de navigateurs (NAV) est de vérifier qu'un site web fonctionne et s'affiche correctement sur différents navigateurs. La méthode était d'accéder aux différentes pages du site sur Chrome, Firefox et Edge.

Un exemple de test de compatibilité de navigateur pourrait être : [ID : NAV-01] [Page : Accueil] [Navigateur : Chrome] [Résultat attendu : La page d'accueil s'affiche et fonctionne correctement] [Résultat obtenu : OK] [Statut : Réussi].

## 6.2) Convention de nommage des tests

L'ensemble des tests sont réunis dans un fichier Excel.

La convention de nommage choisie est la suivante :

- Pour les tests d'interface (UI), et fonctionnels (FUNC)
  - [ID] [Page] [Action] [Résultat attendu] [Résultat obtenu] [Statut]
- Pour les tests de base de données (DB)
  - [ID] [Page] [Table] [Action] [Résultat attendu] [Résultat obtenu] [Statut]
- Pour les tests d'erreurs (ERR)
  - [ID] [Page] [Action] [Résultat attendu] [Résultat obtenu] [Statut] [Commentaire]
- Pour les tests de compatibilité navigateur (NAV)
  - [ID] [Page] [Navigateur] [Résultat attendu] [Résultat obtenu] [Statut]

De plus, les tests d'erreurs sont détaillés pour chaque erreur trouvée avec les étapes mises en place à sa résolution et le nouveau test effectué.

## 7) Evolutions

### 7.1) Futures évolutions

Ce projet est conçu pour être évolutif et pourra être amélioré pour les prochaines courses prévues. Par exemple, le système pourra envoyer une notification de confirmation par email lorsque l'inscription à la course est validée. Les participants pourraient avoir la possibilité de créer un compte et de modifier eux-mêmes leurs informations sans avoir à passer par un organisateur.

## 8) Documentation utilisateur v1.0

### 1) Introduction

#### 1.1) Présentation de l'application

Cette application web permet aux utilisateurs de s'inscrire à une course et aux organisateurs de gérer les inscriptions.

Les fonctionnalités principales sont :

- Inscription à la course
- Accès à un tableau de bord pour les organisateurs
- Gestion des participants pour les organisateurs (validation, annulation, modification des informations du participant)

### 2) Accès à l'application web

#### 2.1) Accès

L'application est accessible en local pour ce projet.

Les navigateurs compatibles sont : Chrome, Firefox et Edge.

### 3) Guide de l'utilisateur

#### 3.1) Interface globale

En accédant à l'application, plusieurs actions sont possibles selon le type d'utilisateur :

- Utilisateur classique (participant) : s'inscrire à la course et voir la liste de tous les participants inscrits
- Organisateur : se connecter et gérer les inscriptions et informations des participants

#### 3.2) Fonctionnalités détaillées

Ici, les actions courantes de l'application web vont être décrites étapes par étapes.

S'inscrire à la course

- 1. Accédez à l'accueil
- 2. Cliquez sur "s'inscrire à la course"
- 3. Remplir le formulaire
- 4. Cliquez sur "s'inscrire"

Course de la Ville 2025

Connexion Organisateur

Bienvenue sur la plateforme officielle de la course

Participez à notre course en solo avec des distances de 5km, 10km et 15km.

Inscrivez-vous dès maintenant et rejoignez la compétition !

À propos

La Course de la Ville 2025 est un événement sportif unique qui rassemble des passionnés de course à pied. Que vous soyez débutant ou athlète confirmé, venez vivre une expérience inoubliable dans un cadre urbain dynamique.

S'inscrire à la course

Voir la liste des inscrits

S'inscrire à la course

Inscription à la course

Nom :

Votre nom

Prénom :

Votre prénom

Date de naissance :

jj/mm/aaaa

Email :

Votre email

Numéro de téléphone :

Votre numéro de téléphone

Choisissez une course :

5 km

S'inscrire

Voir la liste de tous les participants

- 1. Accédez à l'accueil
- 2. Cliquez sur "voir la liste des inscrits"

Course de la Ville 2025

Connexion Organisateur

Bienvenue sur la plateforme officielle de la course

Participez à notre course en solo avec des distances de 5km, 10km et 15km.

Inscrivez-vous dès maintenant et rejoignez la compétition !

À propos

La Course de la Ville 2025 est un événement sportif unique qui rassemble des passionnés de course à pied. Que vous soyez débutant ou athlète confirmé, venez vivre une expérience inoubliable dans un cadre urbain dynamique.

S'inscrire à la course

Voir la liste des inscrits

Voir la liste des inscrits

Se connecter en tant qu'organisateur

- 1. Accédez à la page de connexion
- 2. Entrez vos identifiants
- 3. Cliquez sur "se connecter"
- 4. Accédez au tableau de bord

Course de la Ville 2025

Page d'accueil

Connectez-vous

Email :

Mot de passe :

Se connecter

© 2025 Course de la Ville

Se connecter

Course de la Ville 2025

Page d'accueil

Bienvenue sur votre tableau de bord, Organisateur

Déconnexion

Participants à accepter

Voici la liste des participants qui ont demandé à s'inscrire à vos courses.

Nom : Test Adrien

Course demandée : 15

Email : adrien@test.com

Accepter

Liste de tous les participants inscrits

Voici la liste des participants inscrits à vos courses.

Nom : Garnier Jean

Course : 5



Modifier les informations d'un participant (organisateur)

- 1. Se connecter en tant qu'organisateur
- 2. Accédez au tableau de bord
- 3. Cliquez sur "modifier les informations"
- 4. Modifiez le formulaire
- 5. Cliquez sur "sauvegarder les modifications"

Course de la Ville 2025

Page d'accueil

Connectez-vous

Email :

Mot de passe :

Se connecter

© 2025 Course de la Ville

Liste de tous les participants inscrits

Voici la liste des participants inscrits à vos courses.

Nom : Garnier Jean

Course : 5

Email : jean.garnier@email.fr

Modifier les informations

Annuler l'inscription

Modifier les informations

Modifier les informations du participant

Nom :

Garnier

Prénom :

Jean

Date de naissance :

01/05/2005

Email :

jean.garnier@email.fr

Téléphone :

0123785189

Sauvegarder les modifications

Sauvegarder les modifications

Accepter l’inscription d’un participant (organisateur)

- 1. Se connecter en tant qu’organisateur
- 2. Accédez au tableau de bord
- 3. Cliquez sur “accepter”

Course de la Ville 2025

Page d'accueil

Connectez-vous

Email :

Mot de passe :

Se connecter

© 2025 Course de la Ville

Course de la Ville 2025

Page d'accueil

Bienvenue sur votre tableau de bord, Organisateur

Deconnexion

Participants à accepter

Voici la liste des participants qui ont demandé à s'inscrire à vos courses.

Nom : Test Adrien

Course demandée : 15

Email : adrien@test.com

Accepter

Liste de tous les participants inscrits

Voici la liste des participants inscrits à vos courses.

Nom : Garnier Jean

Course : 5

Accepter

Annuler l’inscription d’un participant (organisateur)

- 4. Se connecter en tant qu’organisateur
- 5. Accédez au tableau de bord
- 6. Cliquez sur “annuler l’inscription”

Course de la Ville 2025

Page d'accueil

Connectez-vous

Email :

Mot de passe :

Se connecter

© 2025 Course de la Ville

Liste de tous les participants inscrits

Voici la liste des participants inscrits à vos courses.

Nom : Garnier Jean

Course : 5

Email : jean.garnier@email.fr

Modifier les informations

Annuler l'inscription

	<div>Annuler l'inscription</div>
<div><div>Restaurer l'inscription d'un participant (organisateur)</div><div><div>7. Se connecter en tant qu'organisateur</div><div>8. Accédez au tableau de bord</div><div>9. Cliquez sur "restaurer l'inscription"</div></div></div>	<div><div>Course de la Ville 2025</div><div>Page d'accueil</div><div><div>Connectez-vous</div><div>Email :</div><div><div></div></div><div>Mot de passe :</div><div><div></div></div><div>Se connecter</div></div><div>© 2025 Course de la Ville</div><div><div>Participants avec inscription annulée</div><div>Voici la liste des participants dont l'inscription a été annulée.</div><div><div>Nom : Leroy Emma</div><div>Course : 5</div><div>Email : emma.leroy@email.com</div><div>Restaurer l'inscription</div></div></div><div>Restaurer l'inscription</div></div>

## 4) Dépannage

### 4.1) Problèmes fréquents et solutions

Problème	Solution
Mon inscription ne s'affiche pas dans la liste de tous les participants inscrits.	Un organisateur doit valider votre inscription et elle apparaîtra ensuite dans la liste de tous les participants inscrits à la course.
Je veux modifier une information personnelle.	Veuillez contacter un organisateur par email ou par téléphone, celui-ci pourra modifier votre profil.
Je veux annuler mon inscription.	Veuillez contacter un organisateur par email ou par téléphone, celui-ci pourra annuler l'inscription.

Mon inscription a été annulée mais je souhaite me réinscrire.	Veuillez contacter un organisateur par email ou par téléphone, celui-ci pourra restaurer l'inscription.
---	---

#### 4.2) Contact pour le support

Email de contact : [julie.lemoine@email.com](mailto:julie.lemoine@email.com)

Numéro de téléphone : 06 76 54 32 10

Disponibilité : du mardi au jeudi de 10h à 17h