

Pour la réalisation de ce projet l'équipe se composait de trois membres : Adem, Ines et Meriem. Nous avons pour sujet la réalisation d'un programme permettant à des organisateurs de festival de gérer les différentes scènes du festival CY'FEST. Les participants doivent payer et réserver leurs places pour les différents concerts.

Au niveau de la dynamique de l'équipe, Adem ayant des facilités au niveau du codage il s'en occupait principalement tout en expliquant le processus au reste du groupe. Meriem et Ines s'occupait du rapport. Également le flux de travail était régulier, tous les jours environ 2 à 3 heures .

Nous avons commencé par le mode festivalier qui était plus complexe à coder que le mode manager. En effet en tant que festivalier on peut réserver un ou plusieurs sièges (ou emplacements si jamais le concert possède une fosse) dans certaines salles. Ces salles seront affichées par le programme où des concerts sont prévu et en cours de validité. Lorsque l'on réserve, le programme affiche le plan de la salle en indiquant les sièges libres ainsi que leur catégorie. Ensuite il affichera le prix des places réservées. Comme on peut le voir le programme est très complexe car de nombreux choix sont à faire et il faut prévoir les erreurs possibles de l'utilisateur. Il faut afficher les plans également ce qui est plutôt laborieux puisqu'il faut faire appel à `afficherPlanSalle` qui est une procédure que l'on est venu créer après. On fait appel à plusieurs autres procédures que l'on crée après comme `lireEntier()`, `lireCaractere()` ou `reserverSiege()`.

Pour le mode manager nous avons mis en place un système de gestion de salles de concert interactif, qui permet de créer, modifier, sauvegarder et observer des salles de concert et leurs configurations. Pour cela, nous avons créé la procédure **`menuManager()`** qui accepte deux paramètres : un tableau de pointeurs vers des objets **`SalleDeConcert`** et un pointeur vers un entier représentant le nombre de salles.

Pour répondre aux différentes difficultés que nous avons pu rencontrer dans ce mode, nous avons dû apporter des solutions spécifiques.

Par exemple pour améliorer la gestion des salles de concert de l'application, nous avons restructuré le code pour permettre une gestion flexible des catégories de sièges, en créant des fonctions robustes pour définir, modifier et afficher les salles de concert avec des configurations personnalisées de rangées et de sièges. Nous avons Implémenté un système de sauvegarde et de chargement des données, garantissant que les informations sur les salles soient correctement enregistrées et restaurées. En optimisant la lecture des entrées utilisateurs et en intégrant des validations pour éviter les erreurs, nous avons contribué à une interface utilisateur plus intuitive et à une meilleure expérience utilisateur. Ces améliorations ont permis de répondre aux besoins des utilisateurs en offrant une gestion plus précise et efficace des concerts et des réservations.

En conclusion, nous avons réussi à développer un programme complet pour la gestion des scènes de festival et des réservations de sièges, avec une collaboration efficace entre les membres. Les défis ont été surmontés grâce à des solutions spécifiques, améliorant ainsi l'expérience utilisateur et répondant aux besoins des organisateurs et des festivaliers.

Les nouvelles améliorations peuvent être le fait de mieux gérer l'heure pour bloquer un client qui veut prendre une place de concert à la dernière minute ou en cours de concert.