
Analyse du Système Informatique d'une application créée pour l'occasion : maNBA

Table des matières

Contexte

I. Analyse Technique

- Les différentes fonctionnalités
- Diagramme de cas d'utilisation
- Diagramme d'activité

II. Analyse de Développement

- Choix Techniques
- Diagramme des composants

Conclusion

Contexte

Sujet

Dans le cadre d'un projet personnel motivé par mon école, le but est :

- pour le 29 Octobre, de réaliser l'analyse d'un système de ventes de billets pour une salle de spectacle
- pour début décembre, de développer une application dans le même genre

Ayant cru que la partie développement était optionnel et à rendre en même temps que le rapport d'analyse, je l'ai donc déjà réalisé, l'application semble alors approximativement finie même s'il manque en effet quelques fonctionnalités et améliorations à implémenter.

Objectif

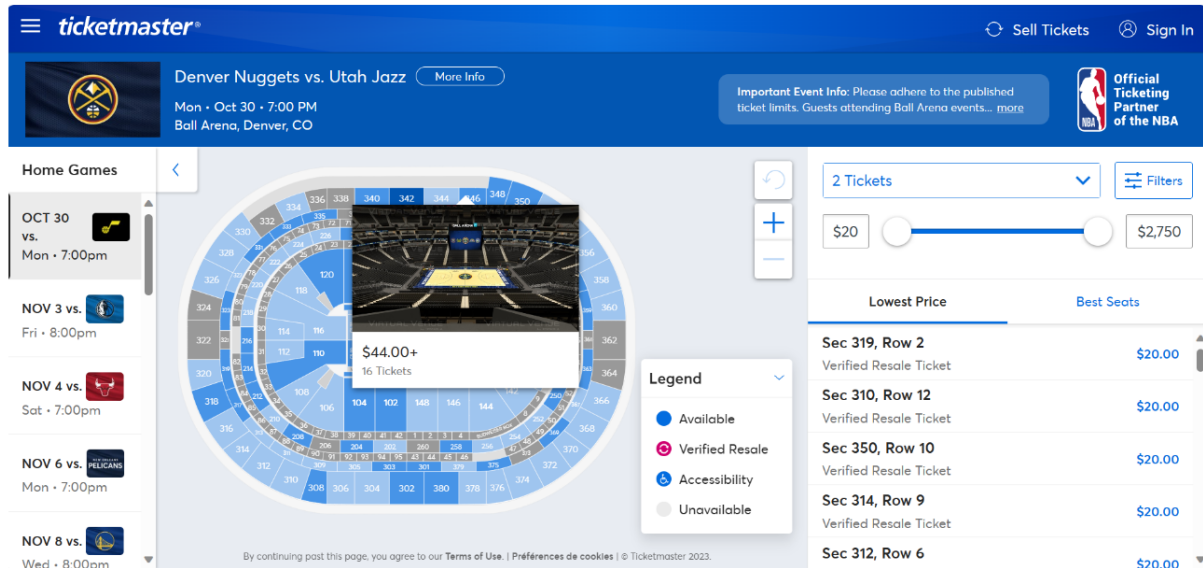
Mon but est personnellement double avec ce sujet. Le premier est d'apprendre à coder en ReactJS, sans oublier le plaisir du développement avec les différentes idées et fonctionnalités à implémenter, et le deuxième est de montrer ma motivation pour le choix de filière à venir : TSI - C.

Ce rapport fait alors l'analyse de ce site et me permet par ailleurs de comparer les différentes fonctionnalités entre site référence choisi et celui créé pour l'occasion. Ce qui me permettra d'estimer la qualité de l'application et de mettre en évidence ce qui pourrait être amélioré.

Analyse technique

a) Les différentes fonctionnalités

Observons d'abord les différentes fonctionnalités que le site de vente de tickets TicketMaster a à proposer lors d'une réservation d'un match NBA :



Capture d'écran de la réservation d'un match NBA sur TicketMaster

la barre de recherche permet de :

- chercher un match
- chercher une équipe

l'onglet Home Games à gauche permet de voir les prochains matchs à domicile

la carte interactive permet de :

- choisir des places
- voir les places disponibles
- afficher le point de vue de la section
- faire la liaison entre prix et places

la partie Ticket permet de :

- faire la recherche d'une place en fonction du prix, du nombre de places
- lancer la réservation
- annuler une réservation

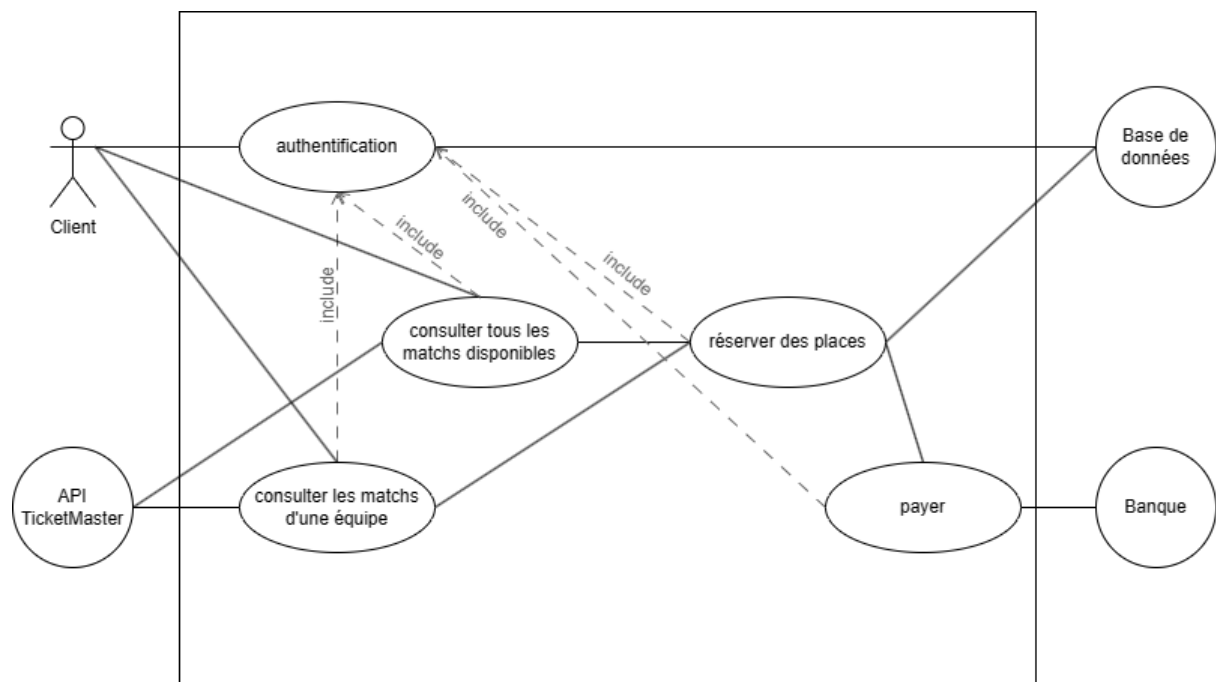
Dans maNBA, l'onglet Teams permet de voir les matchs à venir d'une équipe en particulier, à domicile ou à l'extérieur. L'onglet Games permet de voir tous les matchs à venir, qu'importe les équipes.

L'application ne comporte pas de carte interactive, mais réussit en quelque sorte à choisir sa section puis ses places pour un match NBA. Il est également possible de voir les places déjà réservées et d'annuler les réservations. Les paiements sont toujours acceptés.

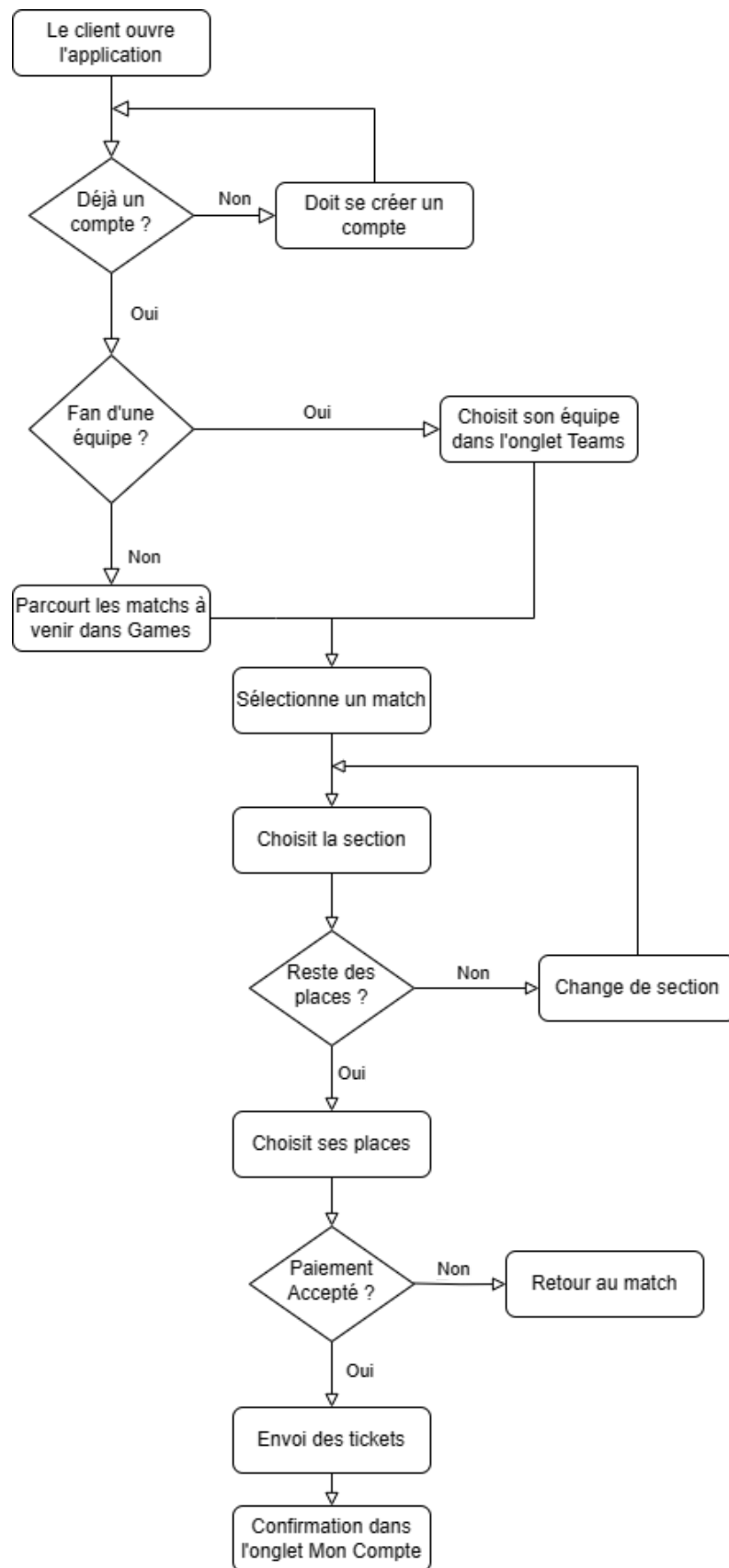
Pour avoir les informations du client, une authentification est requise dès son arrivée pour avoir accès au reste de l'application.

b) Diagramme de cas d'utilisation

Ces différentes fonctionnalités peuvent être résumées dans ce diagramme de cas d'utilisation :



c) Diagramme d'activité

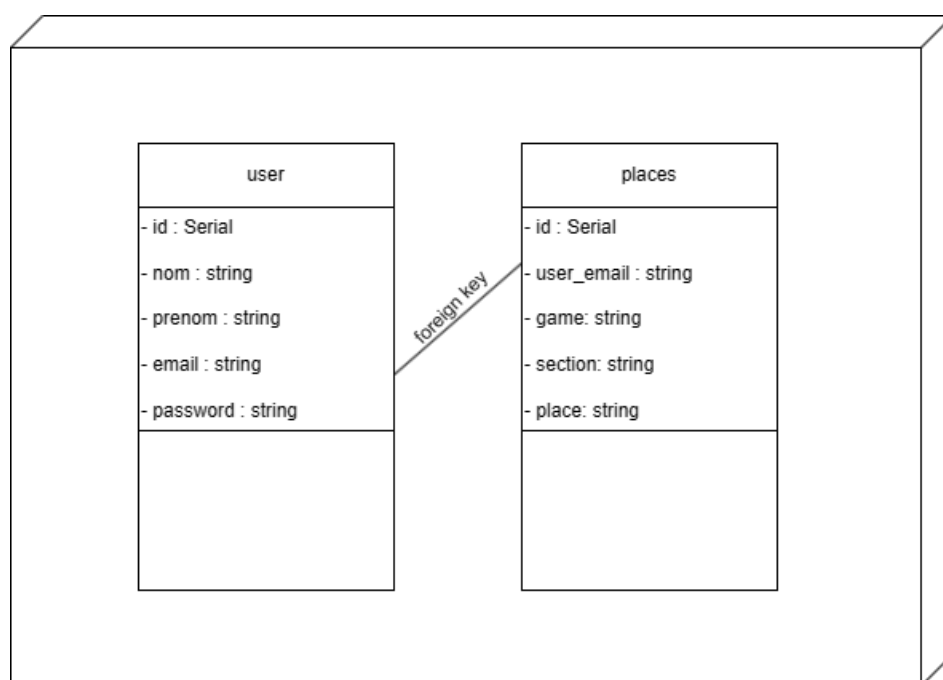


a) Choix techniques

Un de mes objectifs était d'apprendre à coder en ReactJS, en tout cas de créer ma première vraie application à l'aide de ce framework. De plus, je voulais utiliser une API pour que les "spectacles" que je voulais mettre en scène soient toujours d'actualité, facile d'accès, sans besoin de les créer etc. Les avantages sont nombreux. C'est alors en la cherchant que je suis tombé sur l'API TicketMaster qui m'ouvrait toutes les possibilités de catégories de spectacle. Etant donné cette passion pour le basket et la NBA, la voie était toute trouvée.

Il me restait alors à étudier les paramètres et options que cette API avait à me proposer pour fetch selon l'envie : les matchs d'une équipe en particulier, ou tous, dans l'ordre chronologique, la taille de la réponse, selon la page... etc. J'arrive ainsi, que cela soit dans l'onglet Teams, ou Games, à afficher les matchs souhaités. Ou gérer la pagination par exemple.

Concernant l'authentification, l'objectif était de trouver un moyen simple d'accéder aux users ou aux places que je voulais enregistrer. Pour cela, j'ai suivi un tutoriel pour mettre en place une API Express qui gère flexiblement ces deux tables dans la base de données. Pour résumer rapidement, cette API crée la liaison avec la base de données, définit les tables et les attributs qu'elles vont contenir, puis à l'aide de axios et de méthodes (get, post, put, delete...) les requêtes vont chercher, ajouter une ligne à la table ou peut détruire une ligne en une requête. Cette API me permet d'être vraiment flexible, à l'aise avec ces deux tables que voici :



b) Diagramme des composants

D'un point de vue composant maintenant, on a :

- ApiNBATickets qui crée les différentes méthodes de fetch (une équipe, tous, etc) où le nom d'une équipe peut par exemple passer en paramètre : teams présentent les 30 équipes et selon l'équipe choisit, on affiche le composant GamesOfOneTeam où les matchs sont les résultats de la requête du fetchGamesOfOneTeam.
- l'API Express gère users et places dans ses méthodes et permet notamment d'afficher les places d'une personne en particulier dans la page Mon Compte de l'utilisateur ou permet d'afficher les places déjà prises pour un match et une section données.

Concernant les routes, l'utilisateur arrive premièrement sur l'authentification avec SignIn. Un passage obligatoire pour accéder au service proposé. Ainsi, d'un point de technique, il aurait été possible de créer une route publique, accessible à tous, sans authentification nécessaire et une route privée pour ceux qui s'authentifient. La route publique se terminant forcément si le client veut acheter une place.

