



MAISON DES LIGUES

# Contexte

- La Maison des Ligues de Lorraine (M2L) a pour mission de fournir des espaces et des services aux différentes ligues sportives régionales et à d'autres structures hébergées. La M2L est une structure financée par le Conseil Régional de Lorraine dont l'administration est déléguée au Comité Régional Olympique et Sportif de Lorraine (CROSL).



# PROBLÉMATIQUE WEB

- L'entreprise M2L souhaite promouvoir le sport dans sa région en mettant en vente des produits sportifs dans un but non-lucratif.



# PROBLÉMATIQUE

## MOBILE A CHANGER

- L'entreprise M2L souhaite faciliter l'accès aux récapitulatif des fiches de paient ainsi que les réservation des salles du client.



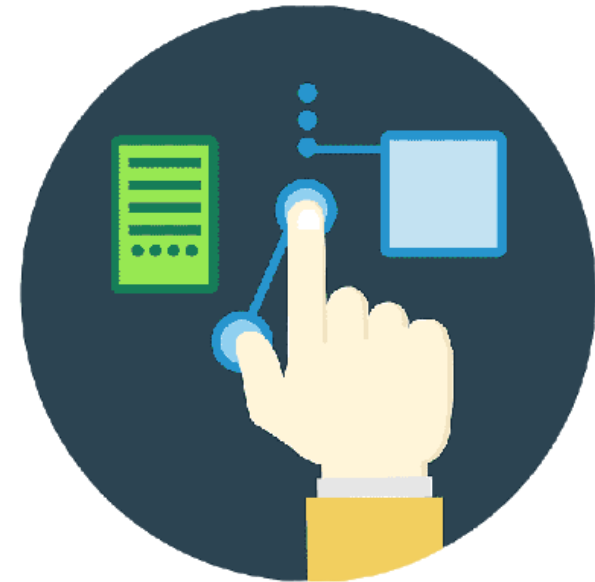
# INTERFACE WEB CÔTÉ ADMIN

- Gestion des utilisateurs
- Gestion des produits
- Gestion des commandes

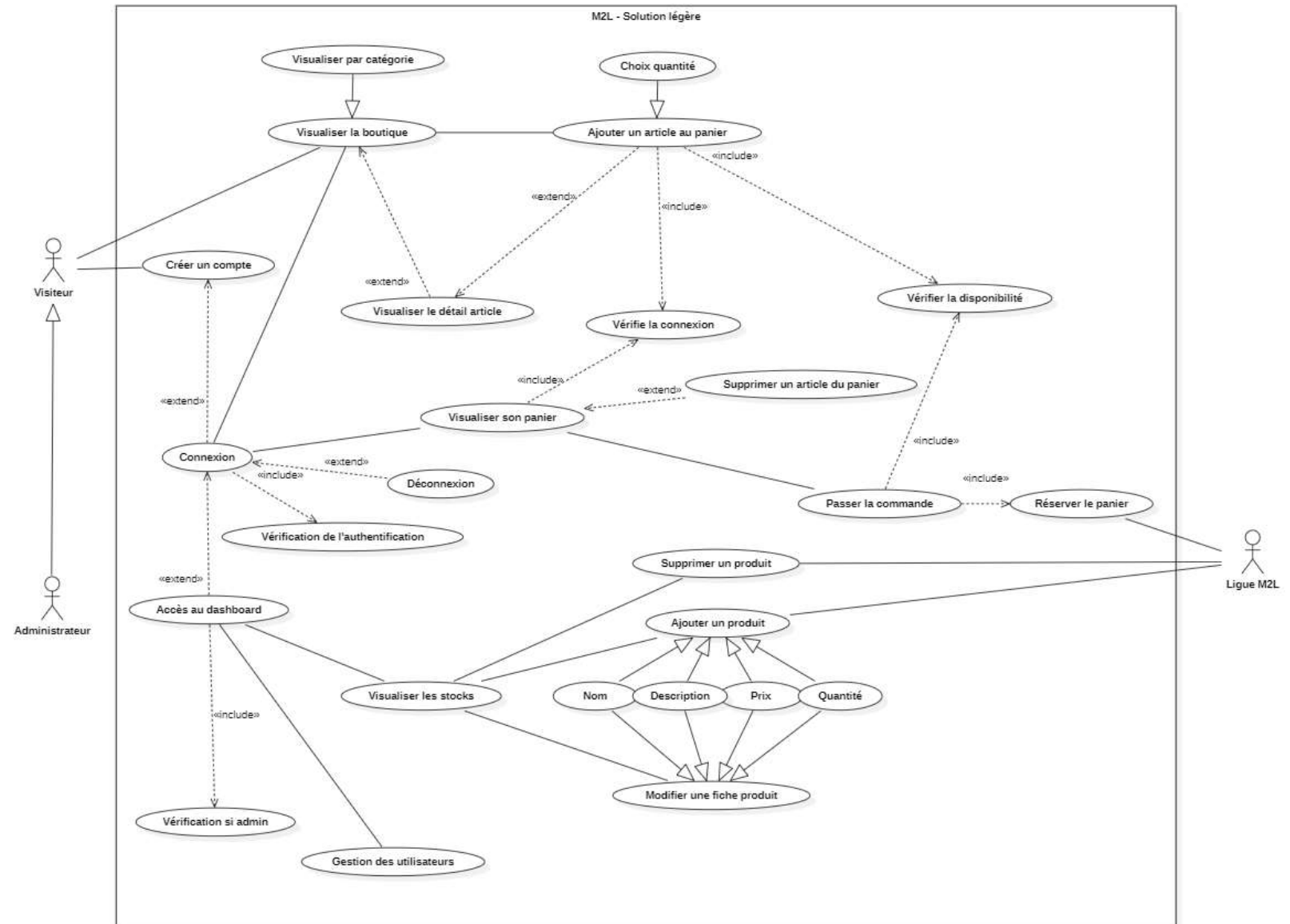


# INTERFACE WEB COTÉ USER

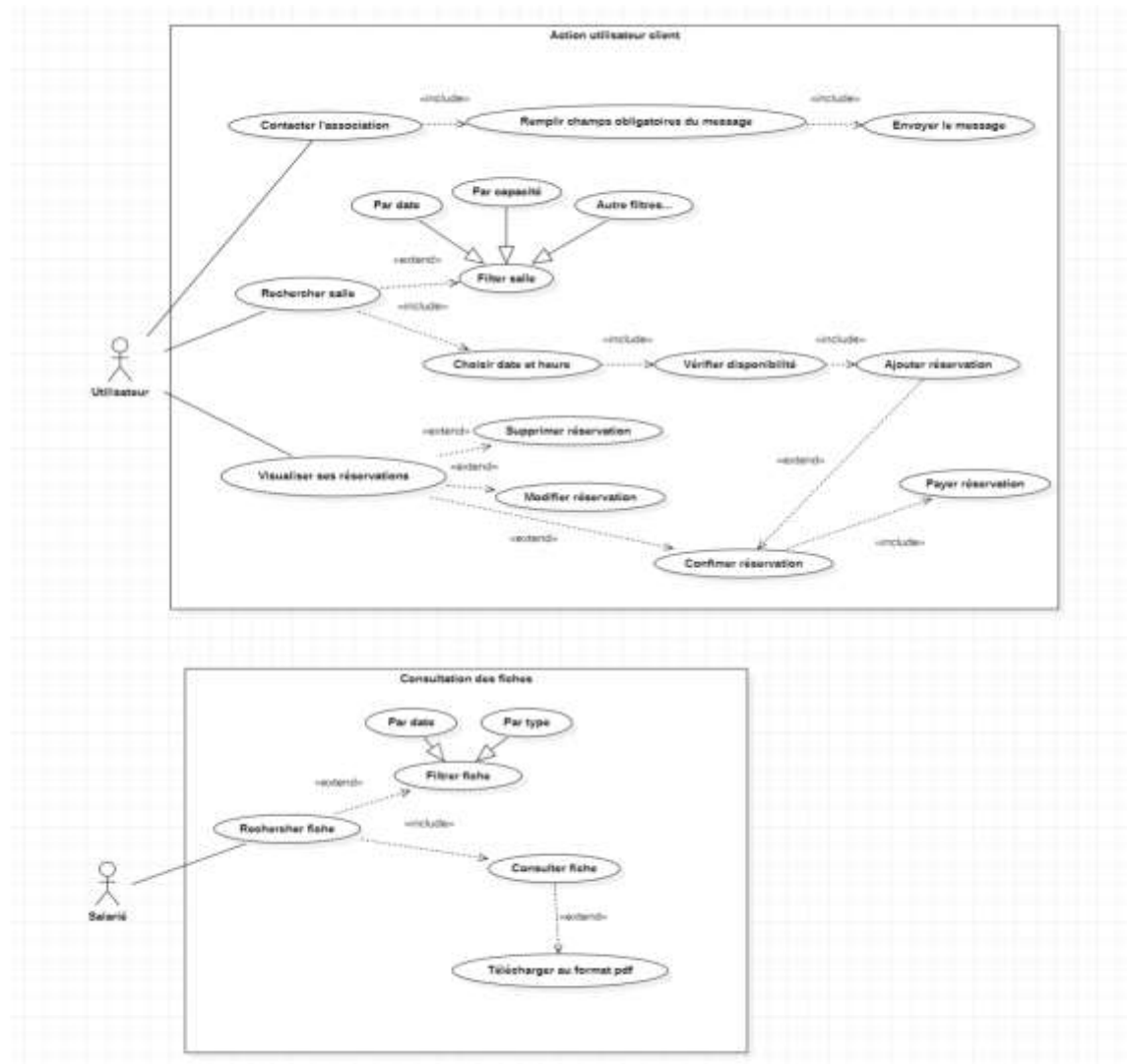
- Naviguer sur le site
- Acheter un produit
- Consulter son panier



# Solution Web : diagramme de cas d'utilisation

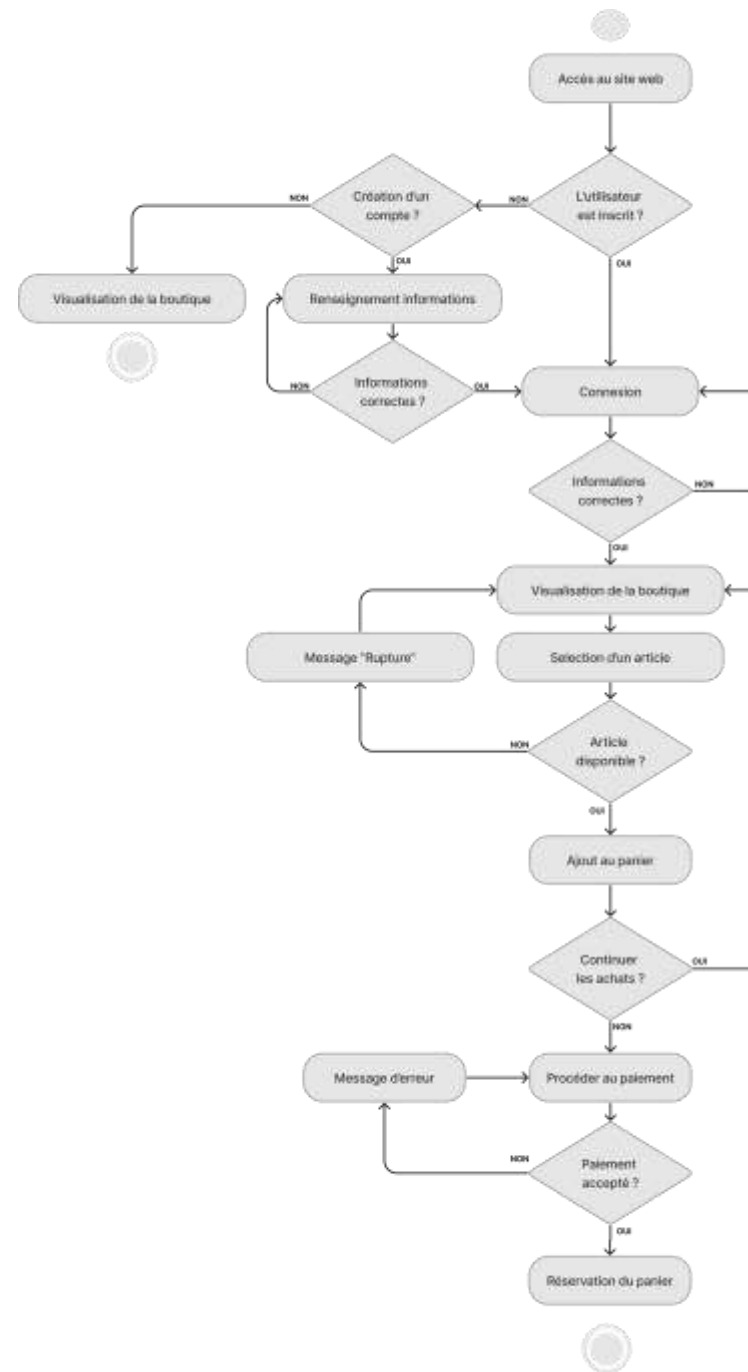


# Solution Mobile : diagramme de cas d'utilisation





# Diagramme d'activité



# Aspects techniques et Cahier des charges



**LES LANGAGES**



**NODE.JS**



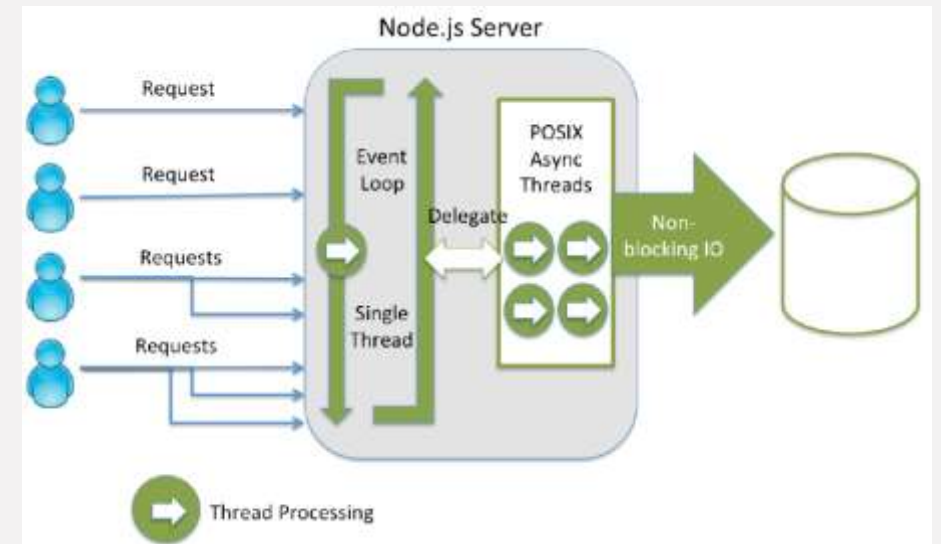
**REACT.JS**



**FLUTTER**

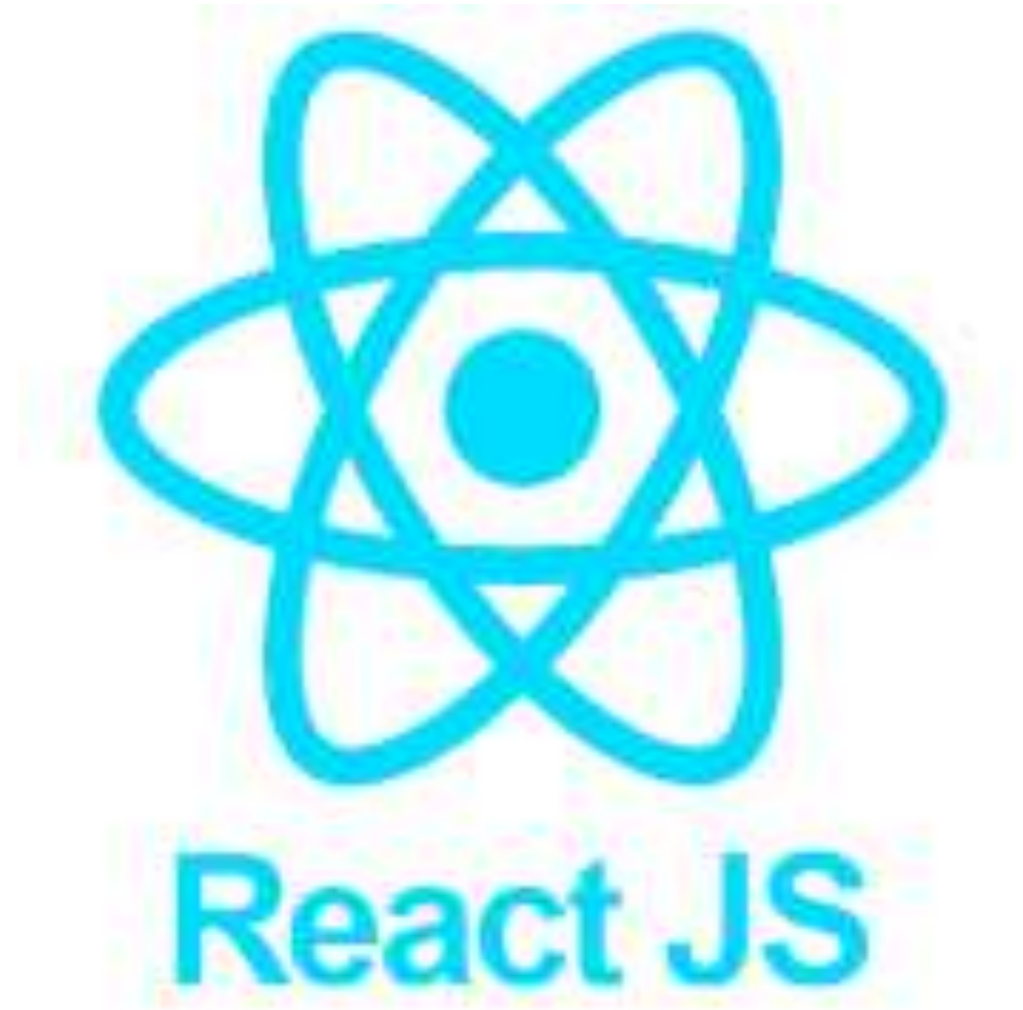
# Node.js

- Node.js (v.18.16.0) est un environnement d'exécution single-thread, open-source et multi-plateforme permettant de créer des applications rapides et évolutives côté serveur et en réseau. Il fonctionne avec le moteur d'exécution JavaScript V8 et utilise une architecture d'Entrée / Sortie non bloquante et pilotée par les événements, ce qui le rend efficace et adapté aux applications en temps réel.
- Le temps de développement peut être considérablement réduit avec Node.js. Le code JavaScript peut être partagé sur le frontend et le backend par un seul développeur.
- Grâce à sa communauté, Node.js est en évolution constante depuis son apparition et permet une flexibilité accrue par rapport à d'autres langages backend tels que PHP par exemple.



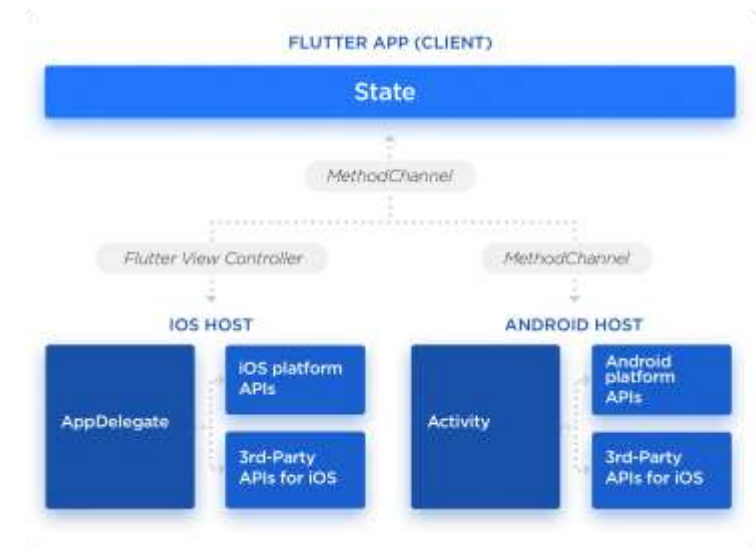
# React.js

- React.js (v.18.2.0), communément appelé simplement React, est une bibliothèque JavaScript utilisée pour construire des interfaces utilisateur. Chaque application web React est composée de composants réutilisables qui constituent des parties de l'interface utilisateur – nous pouvons avoir un composant distinct pour notre barre de navigation, un pour le pied de page, un autre pour le contenu principal, et ainsi de suite. Vous comprendrez mieux cela lorsque nous arriverons à la section où nous devons travailler avec des composants.
- Le fait de disposer de ces composants réutilisables facilite le développement car nous n'avons pas à répéter le code récurrent. Il nous suffit de créer sa logique et d'importer le composant dans n'importe quelle partie du code où il est nécessaire.
- React est également une application à page unique. Ainsi, au lieu d'envoyer une requête au serveur à chaque fois qu'une nouvelle page doit être rendue, le contenu de la page est chargé directement à partir des composants React. Cela conduit à un rendu plus rapide sans rechargement de la page.



# Flutter

- Flutter (v.3.7.1) est un Framework Dart de développement d'applications mobiles open source de Google. La principale raison de sa popularité est qu'il prend en charge la création d'applications multiplateformes. Flutter est également utilisé pour créer des apps interactives qui s'exécutent sur des pages web ou sur le bureau.
- Flutter a montré son potentiel quelques années seulement après son lancement officiel. Ce Framework permet non seulement d'accélérer le processus de développement d'apps, mais aussi de gagner du temps et de l'argent.
- Comparé aux autres Framework de développement d'applications multiplateformes, Flutter est plus rentable. Il répond à toutes les exigences de chaque entreprise, quels que soient son modèle et sa taille.
- Grâce à sa fonction de réutilisation du code, Flutter permet aux développeurs de gagner du temps pour plusieurs plateformes. Le concept « Write Once, Run Everywhere » (écrire une fois, exécuter partout) se vérifie puisqu'un seul code est utilisé pour développer
- Les petites et moyennes entreprises peuvent choisir cette plateforme pour créer des applications rapides avec les fonctionnalités souhaitées et d'excellentes conceptions. Ici, le coût du développement d'une application reste faible car les applications Flutter prennent peu de temps à développer.
- Contrairement aux langages tels que Swift (iOS) et Kotlin/Java (Android) ce langage nous permet le développement simultané sur toutes nos plateformes.

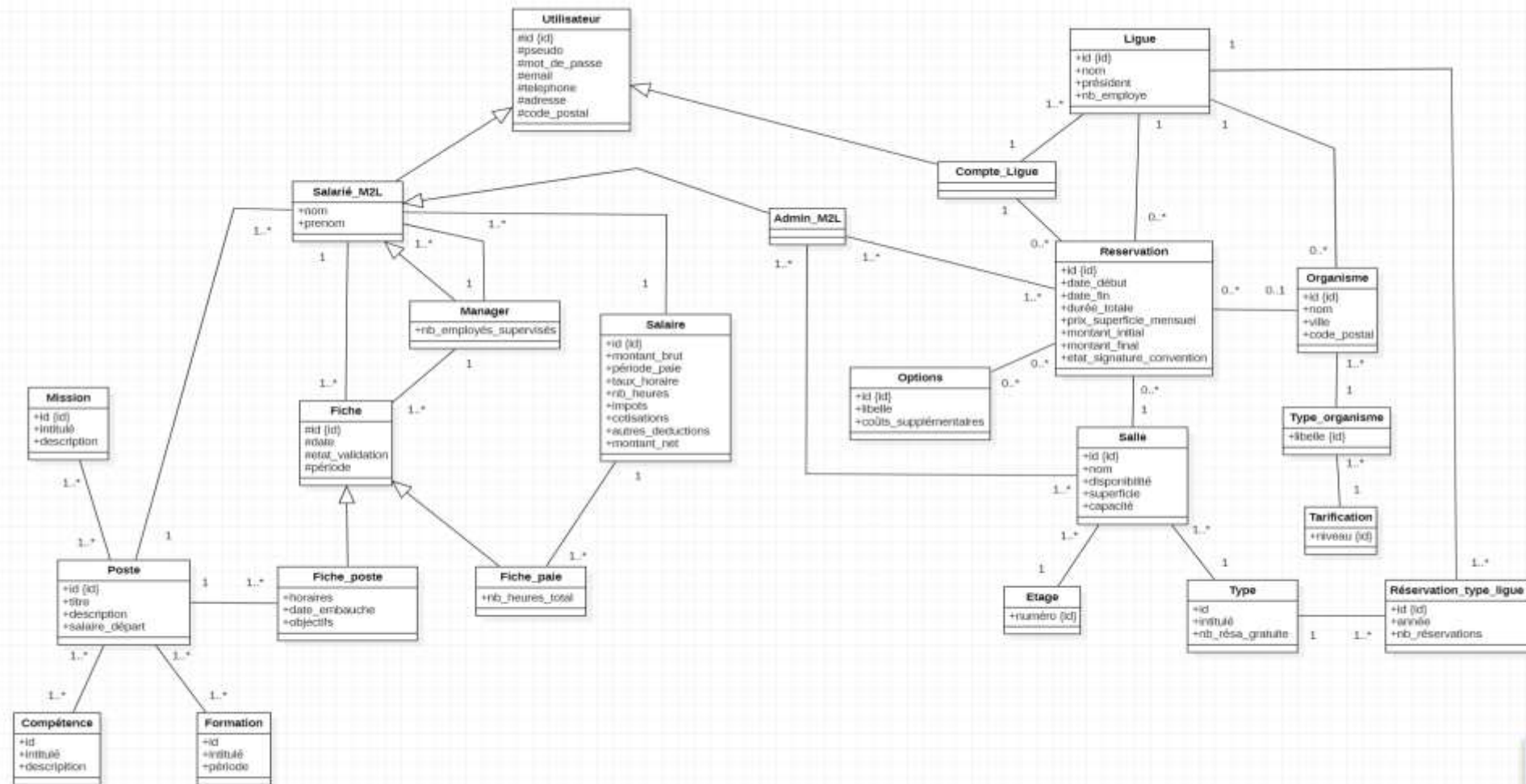


# En conclusion

- Pour le développement de nos solutions, il était nécessaire de choisir des langages nous permettant d'optimiser notre temps de travail et les coûts associés. React et Flutter, associés à Node, répondent à ce besoin, ce qui explique notre choix pour les projets à venir.



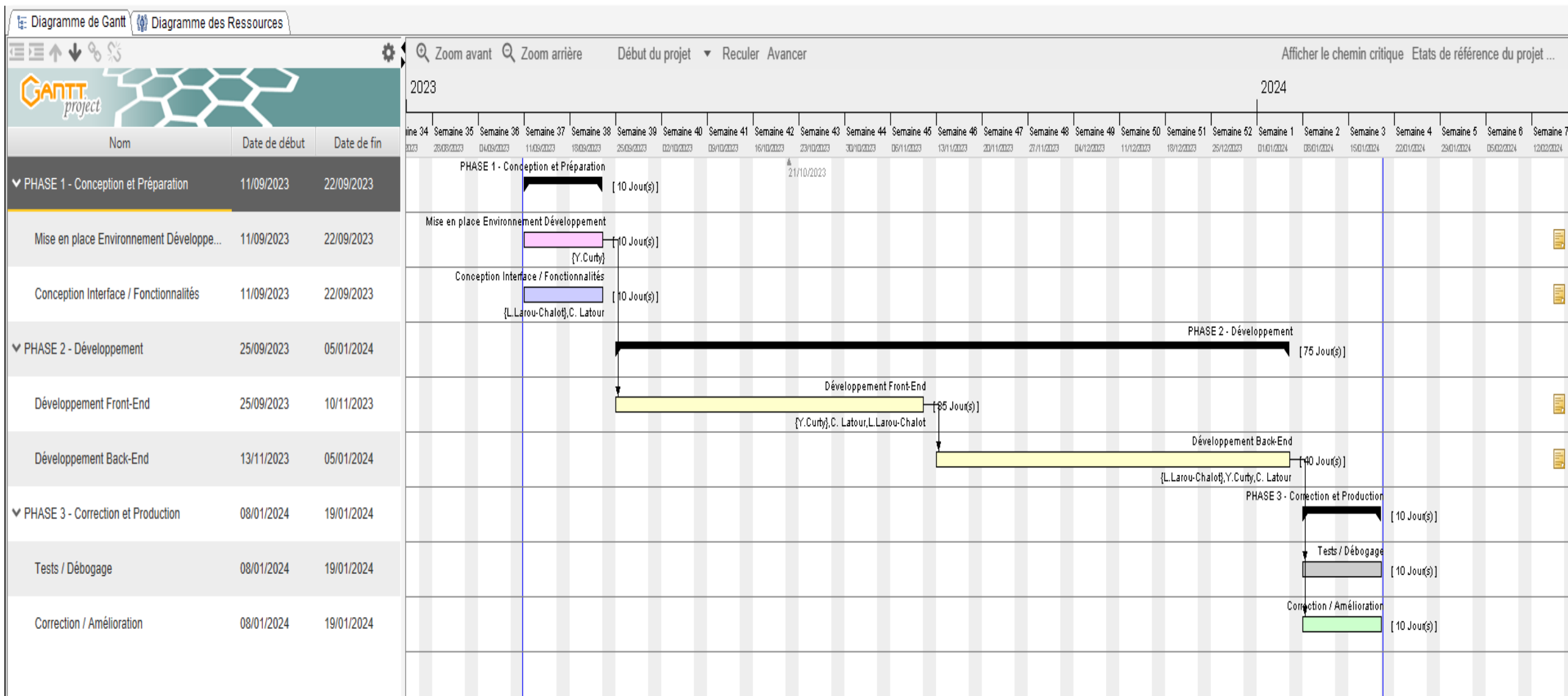
# Diagramme de classe





# Diagrammes de Gantt

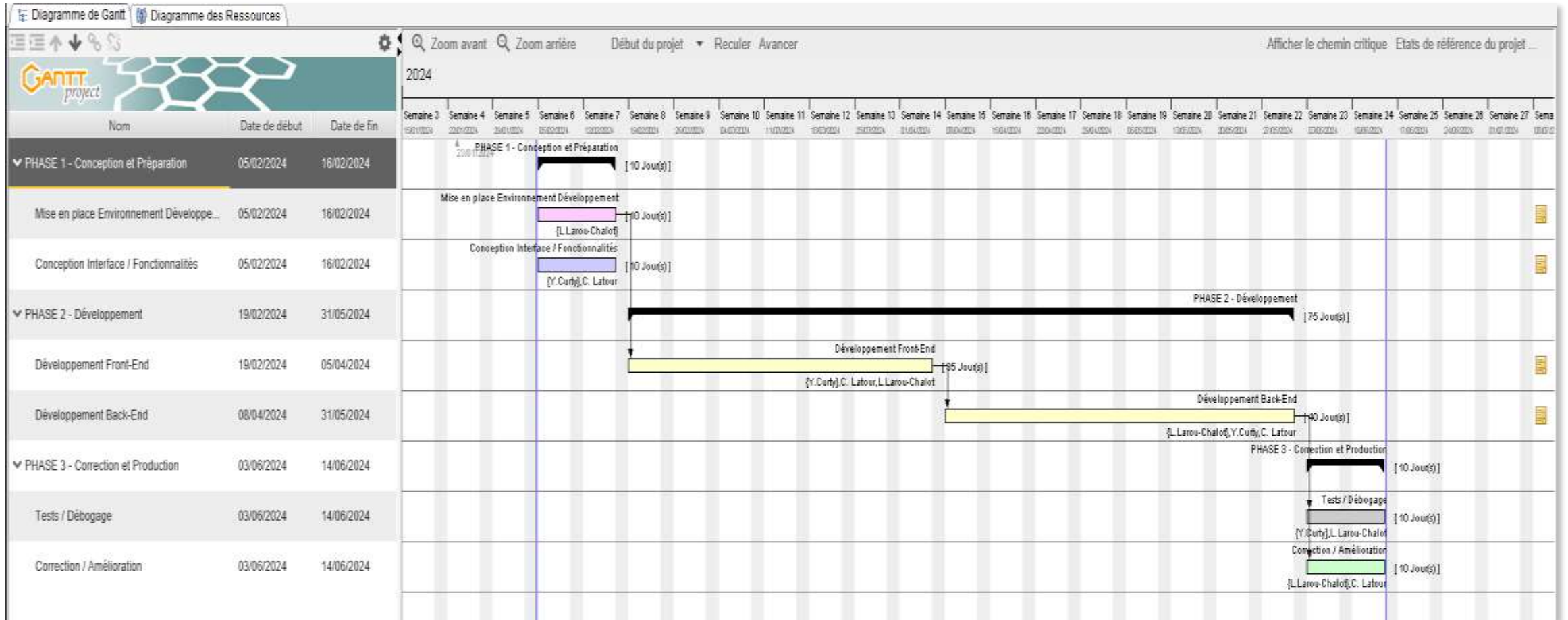
## React



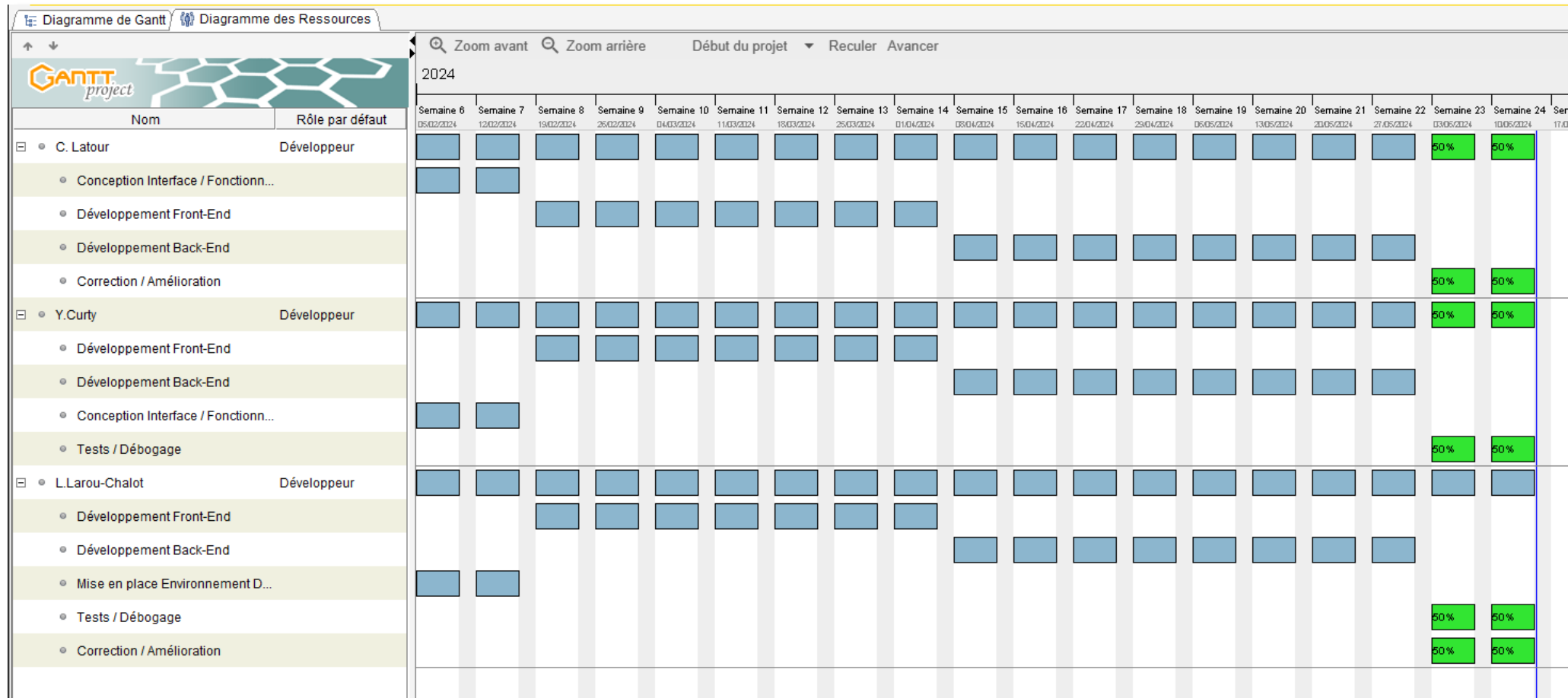


# Diagrammes de Gantt

Flutter




# Diagramme de ressource



# Devis





**Horizon Dev Web**  
25 rue Claude Tillier  
75012, Paris  
01.23.45.67.89  
Horizon.dev@contact.fr

03/05/2023

**Devis N°87234**

**Maison des Ligues de Lorraine**  
13 rue Jean Moulin  
54510, Tomblaine

*Devis détaillé – Développement de la solution Web*

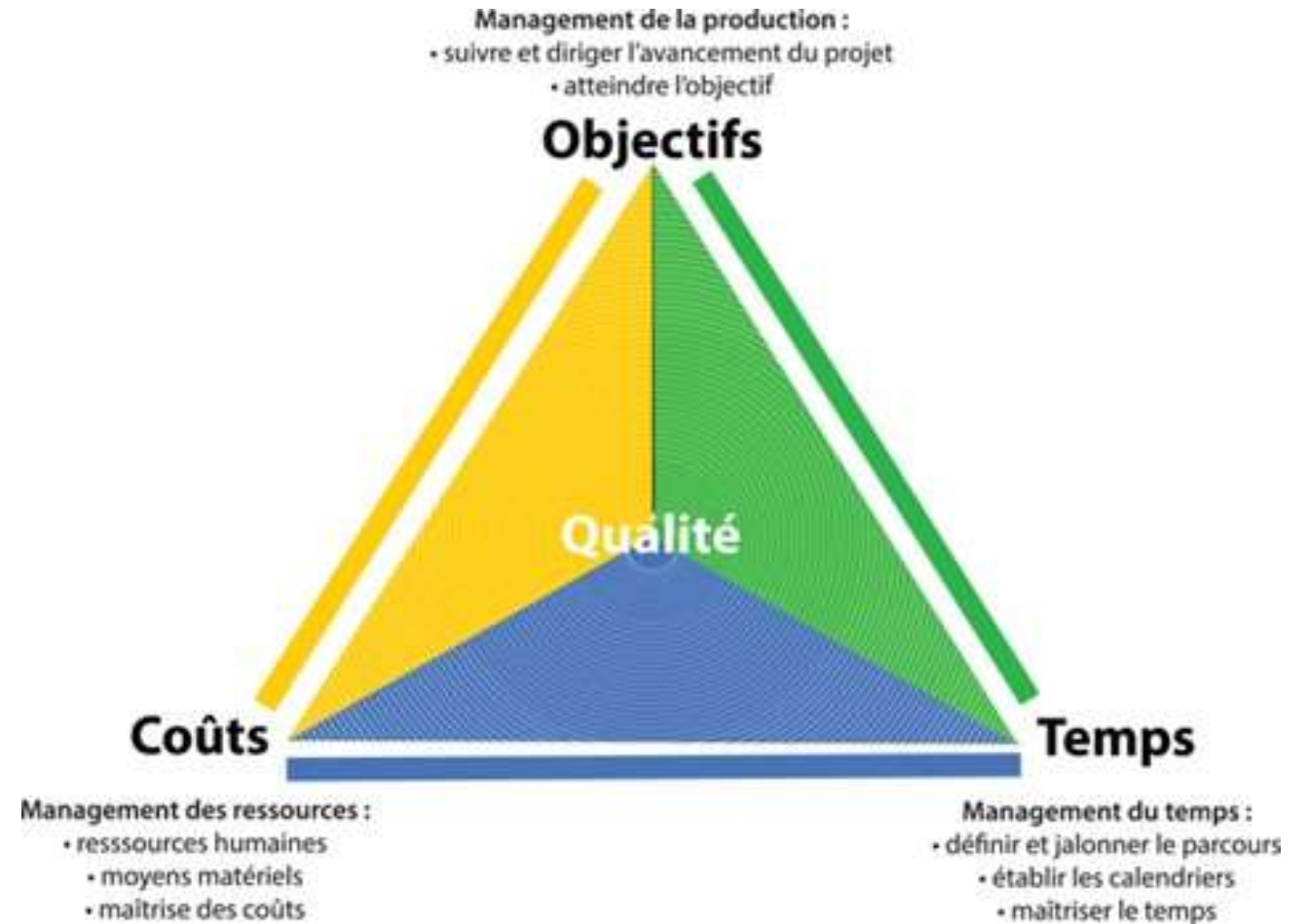
Description	Unité	Quantité	Prix Unitaire HT	TVA	Total HT
Conception Interface Utilisateur et Fonctionnalités		1	2.700 €	20 %	3.240 €
<b>Développement Frontend</b>		1	7.000 €	20%	8.400 €
<b>Développement Backend</b>		1	13.250 €	20%	15.900 €
<b>Correction, amélioration et maintenance (for. 30j)</b>		1	2.700 €	20%	3.240 €

*Montant Total HT*

<i>Total Net HT</i>	25.650 €
<i>Total TVA</i>	20%
<i>Montant Total TTC</i>	30.780 €

La loi n°92/1442 du 31 décembre 1992 nous fait l'obligation de vous indiquer que le non-respect des conditions de paiement entraîne des intérêts de retard suivant modalités et taux défini par la loi. Une indemnité forfaitaire de 40€ sera due pour frais de recouvrement en cas de retard de paiement.

# Triangle de projet



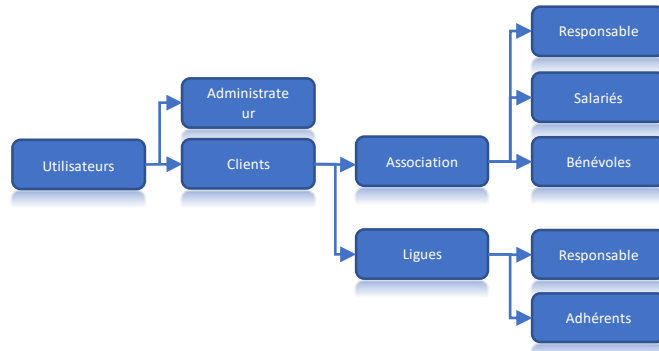
# La gestion des risques

Pour la réalisation de ces deux projets, certaines mesures de sécurité devront être mises en œuvre.

- Diffusion et gestion des identifiants
- Puisqu'il s'agit d'applications à l'attention des différentes ligues et au personnel de l'association, nous devons mettre en place un système d'identification pour orienter les choix d'affichage en fonction du type d'utilisateur.
- Pour chaque création de compte un couple identifiant/mot de passe devra être défini.



# La gestion des risques



- Le compte administrateur appartiendra à la personne compétente en charge de l'application. Ce compte sera créé par défaut.
- Pour les autres types de comptes, lors de l'enregistrement de ceux-ci, **il ne sera demandé qu'un nom d'utilisateur et une @mail** en plus de certaines informations relatives à l'utilisateur. Un mail sera alors envoyé avec **un lien pour définir le mot de passe**.
- Cette pratique aura pour but d'éviter le partage du mot de passe en clair par voie électronique.
- Ce nouveau mot de passe **devra être hashé puis sauvegardé en base de donnée**.



## Traitement de mots de passe côté serveur

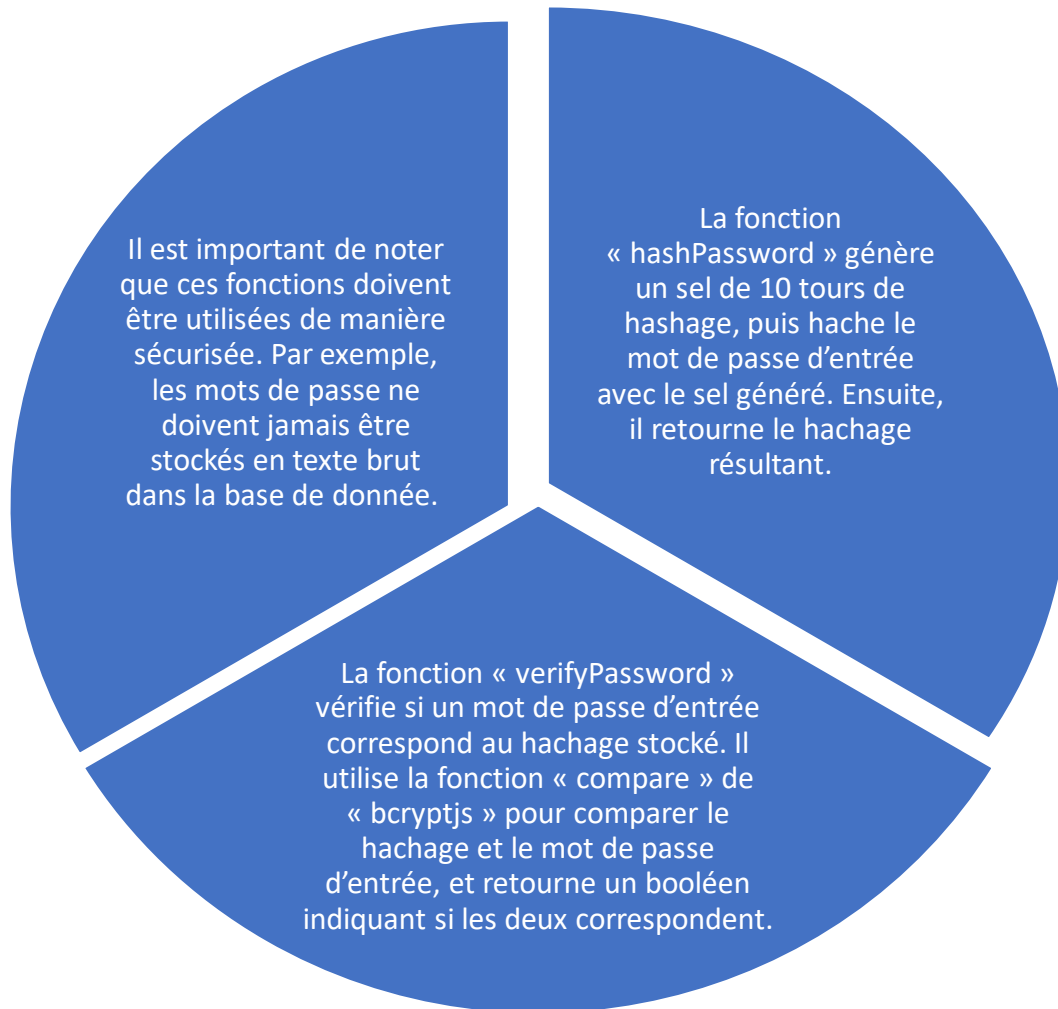
- Il est à noter que le hash des mots de passe est conseillé côté serveur, avec NodeJs nous pourrons utiliser le code suivant pour ce faire

```
const bcrypt = require('bcryptjs');

const hashPassword = async (password) => {
  const salt = await bcrypt.genSalt(10);
  const hash = await bcrypt.hash(password, salt);
  return hash;
};

const verifyPassword = async (password, hash) => {
  const isMatch = await bcrypt.compare(password, hash);
  return isMatch;
};
```

# Fonction hashpassword





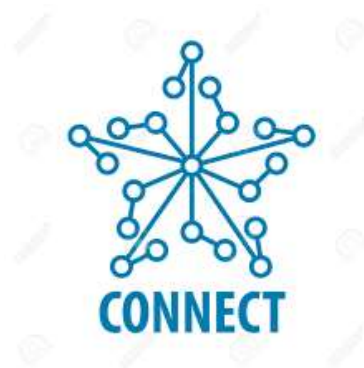
admin@mail.com      .....      Log In

Your new SALT: \$2a\$10\$.GmeAPPUza.sJy45LgHCZe  
Save this Salt, UPON sign up  
if you refresh it will generate a new SALT!!!

```
* received: {email: "admin@mail.com", password: "$2a$10$.GmeAPPUza.sJy45LgHCZe"}
* email: "admin@mail.com"
* password: "$2a$10$.GmeAPPUza.sJy45LgHCZe"
```

## Connexion depuis un réseau public

- En cas de **connexion depuis un réseau public**, pour éviter tout acte malveillant, l'utilisation de la bibliothèque bcryptJs pourra être utilisée pour éviter que le mot de passe ne soit récupéré.
- Exemple de hashage lors de la connexion depuis un réseau public



An abstract background on the left side of the slide. It features a close-up, angled view of a computer keyboard. Several keys are highlighted with glowing arrows pointing right. The arrows are in various colors: white, light blue, and orange. The overall color scheme transitions from dark blue at the top to a bright orange glow at the bottom, creating a sense of digital movement and energy.

# Injections HTML/SQL

- Lors de la connexion, une vérification devra être réalisée aussi bien pour le mail que pour le mot de passe pour éviter tout type d'injections SQL / Html. Cela évitera tout type de fuite en cas d'acte non désiré.