
Documentation - CDC

RESUME

Réalisation de la 2ème activité Professionnelle

Christian LATOUR, Yoann CURTY, Léo LAROU-CHALOT

AP2 - Gestion de projet

# 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).

**PROBLEMATIQUE WEB**

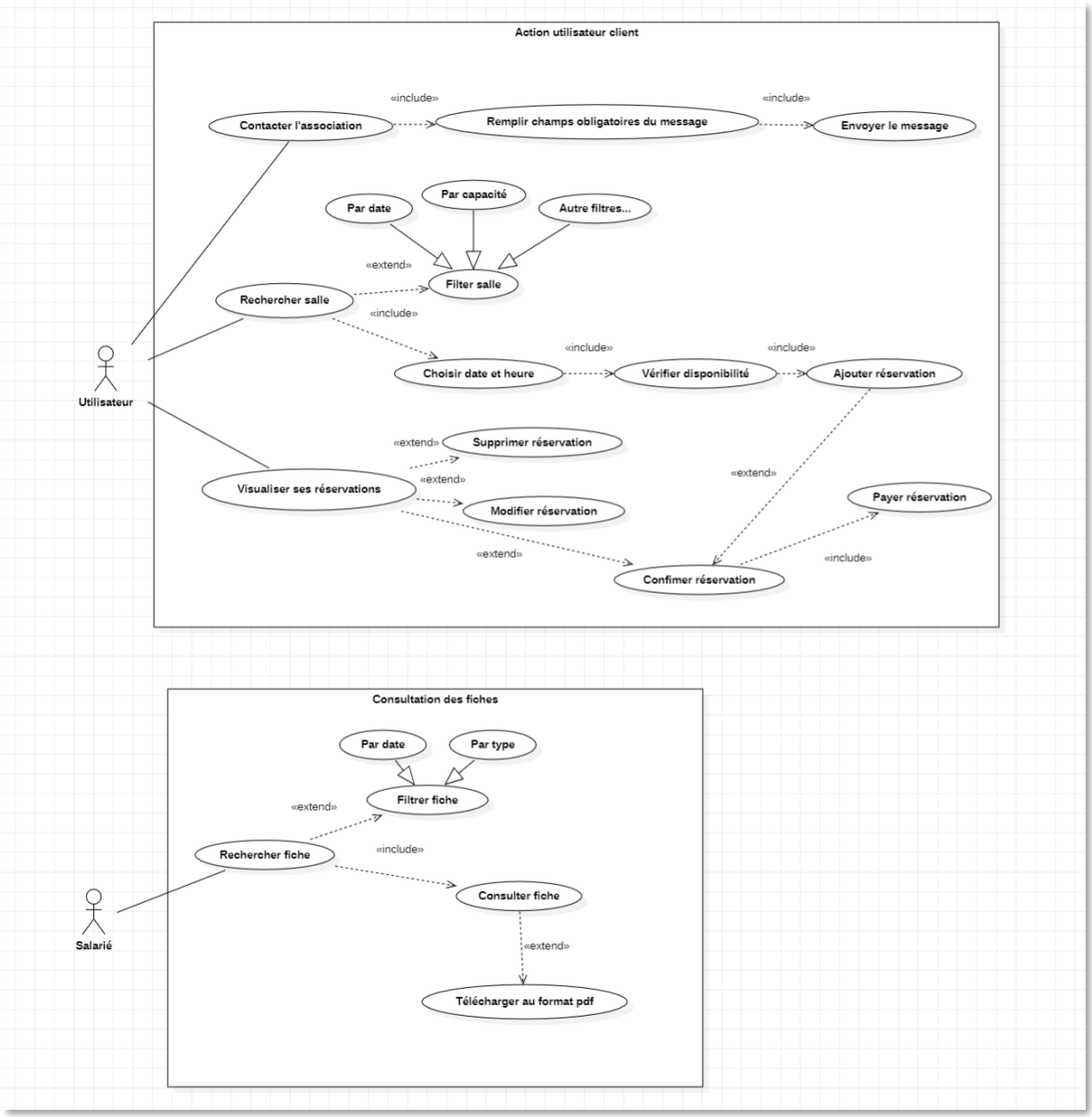
L'entreprise M2L souhaite faciliter la réalisation et la gestion des fiches de paies, et d'emploi pour les responsales de l'entreprise, ainsi que faciliter la réservation et la gestion des salles.

**PROBLEMATIQUE MOBILE**

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

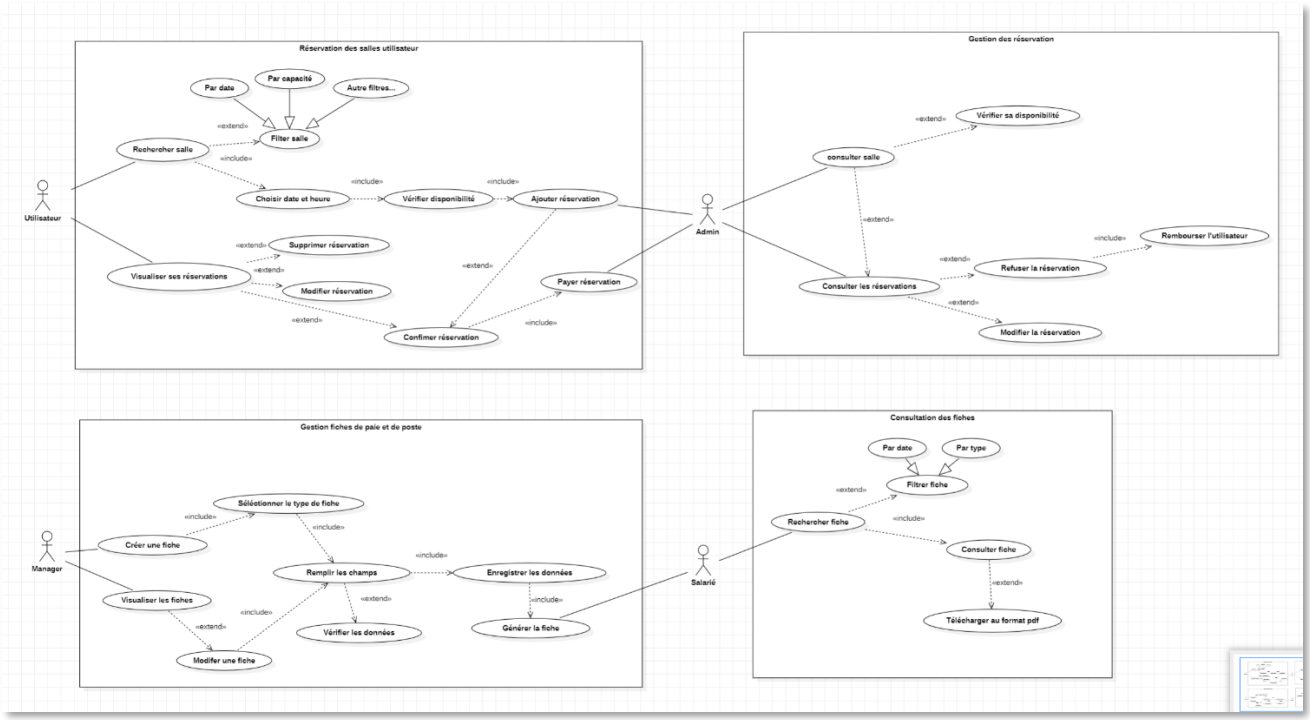
# Interface Web

## Cas d'utilisation

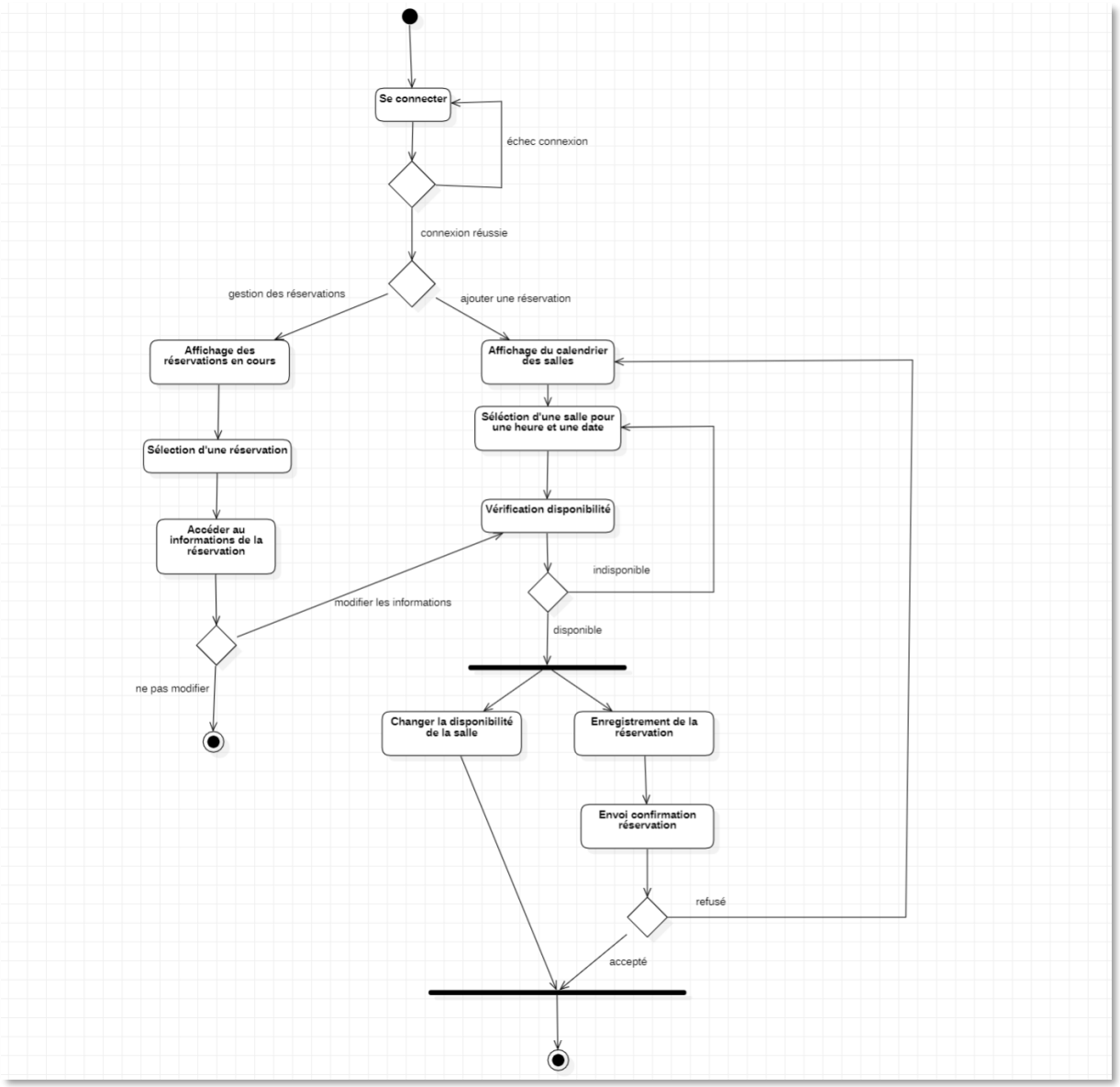


# Application Mobile

## Cas d'utilisation



# Exemple de diagramme d'activité



# Devis

03/05/2023

**Devis N°87234**

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

**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 € |
| **Développement Frontend** | | 1 | 7.000 € | 20% | 8.400 € |
| **Développement Backend** | | 1 | 13.250 € | 20% | 15.900 € |
| **Correction, amélioration et maintenance (for. 30j)** | | 1 | 2.700 € | 20% | 3.240 € |

*Montant Total HT*

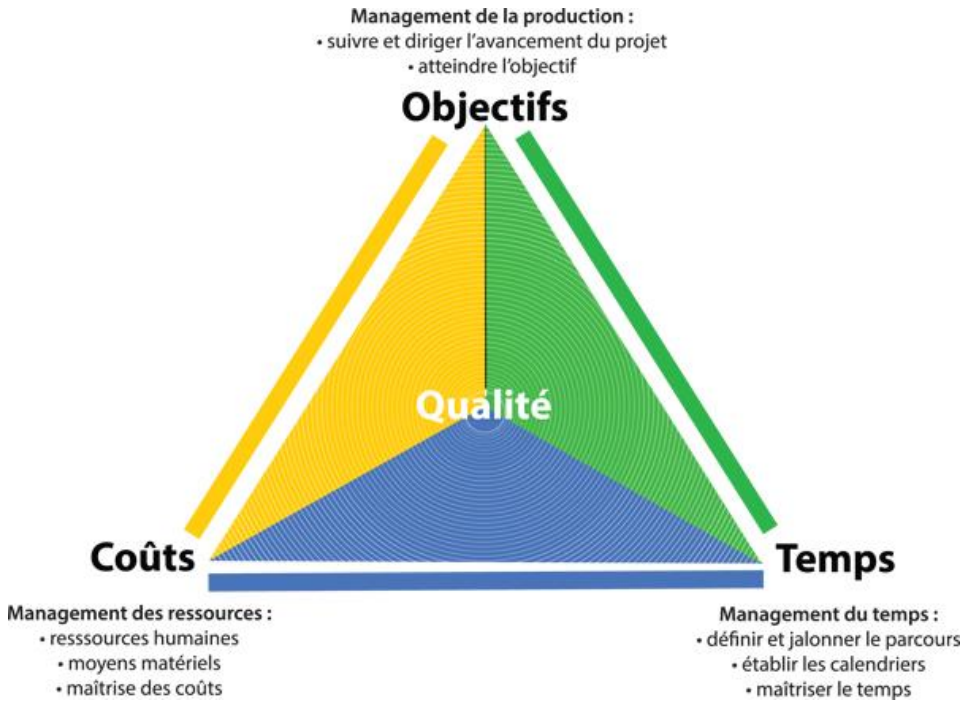
| | |
|---|---|
| *Total Net HT* | 25.650 € |
| *Total TVA* | 20% |
| *Montant Total TTC* | 30.780 € |

La loi n°92/1442 du 31 décembre 1992 nous fait l'obligation de vous indiquer que le non-respect des conditions de paiement entraine 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.

# Triangles de projets

Lien vers la doc

Le Triangle de Projet consiste à représenter les trois vecteurs principaux de nos projets. Avec cet outils, il est primordial d'identifier l'élément directeur.




Dans notre situation, c'est le temps qui ne pourra pas être modifier. Nous pourrons jouer sur la gestion des coûts (cela aura alors un impact sur les objectifs), ou bien sur les objectifs (cela aura alors un impact sur les coûts).

# Aspects techniques et Cahier des charges

## Les langages

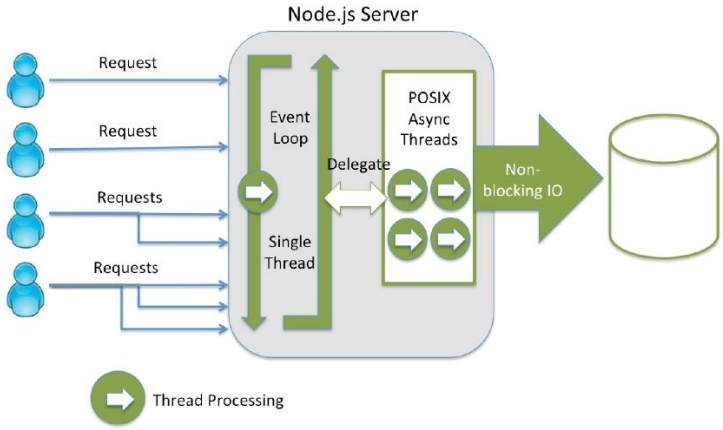
### Node.js

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 sont apparition et permet une flexibilité accrue par rapport à d'autres langages backend tels que Php par exemple.

Source : Kinsta.com




Visiter le site de Node : nodejs.org

### React.js

React.js (v.18.2.0), communément appelé 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.

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.

Source : Kinsta.com

Visiter le site de React : [React.dev](React.dev)

## 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.

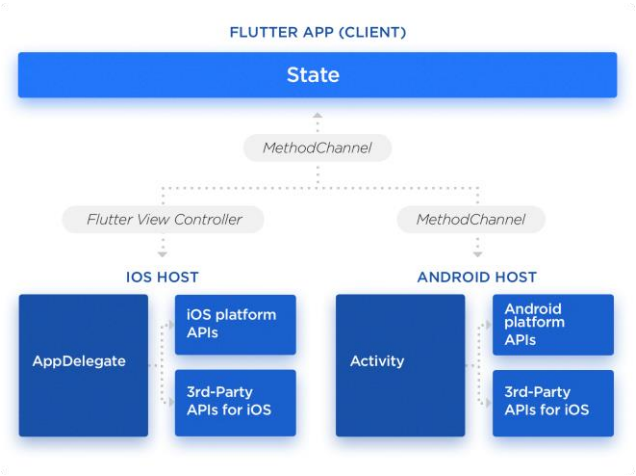
Comparé aux autres Framework de développement d'applications multiplateformes, Flutter est plus rentable. Il répond à toutes les exigences de chaque entreprise, quels que soient son modèle et sa taille.

Grâce à sa fonction de réutilisation du code, Flutter permet aux développeurs de gagner du temps. Le concept « Write Once, Run Everywhere » (écrire une fois, exécuter partout) se vérifie puisqu'un seul code est utilisé pour développer une application pour plusieurs plateformes.

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.

Source : [Mobiskill.fr](Mobiskill.fr)

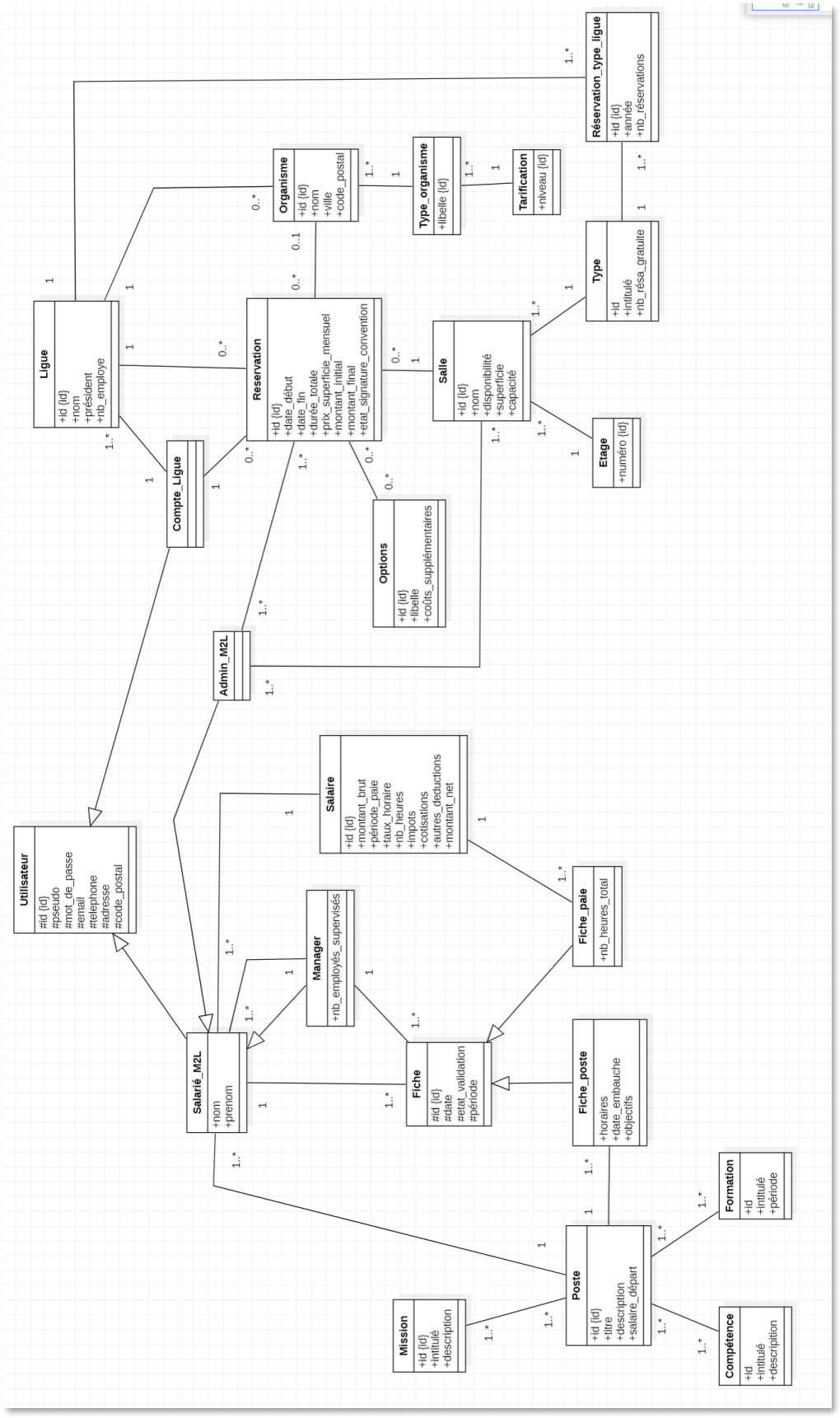



Visiter le site de Flutter : [Flutter.dev](Flutter.dev)

## En conclusion

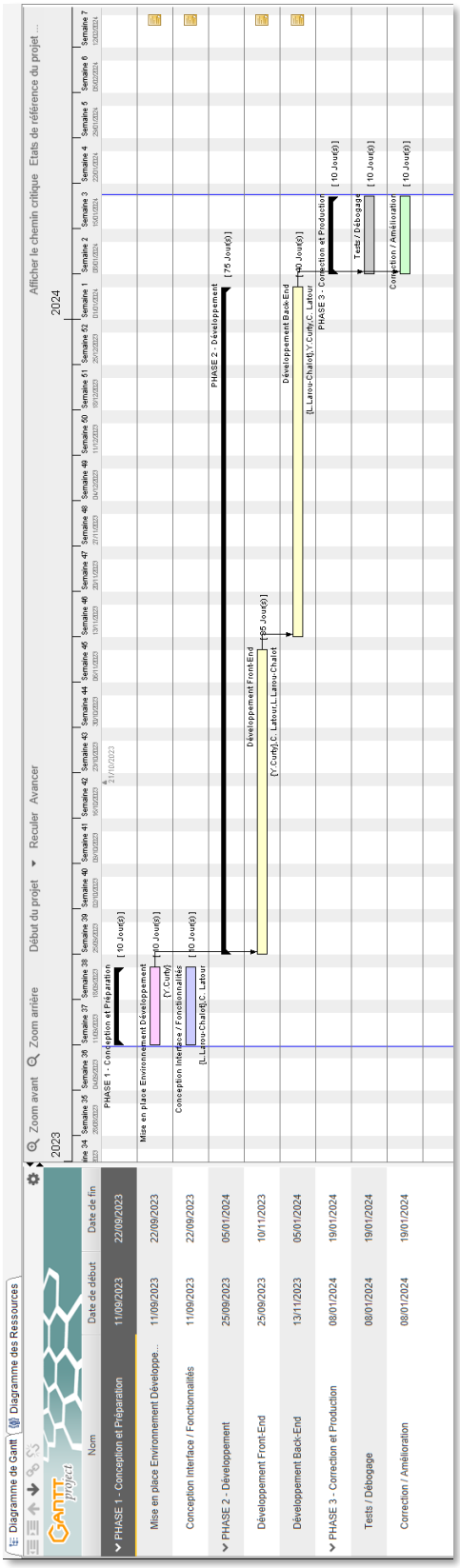
Pour le développement de nos solutions, il était nécessaire de choisir des langages nous permettant d'optimiser notre temps de travail et les coûts associés. React et Flutter, associés à Node, répondent à ce besoin, ce qui explique notre choix pour les projets à venir.

# Diagramme de classe

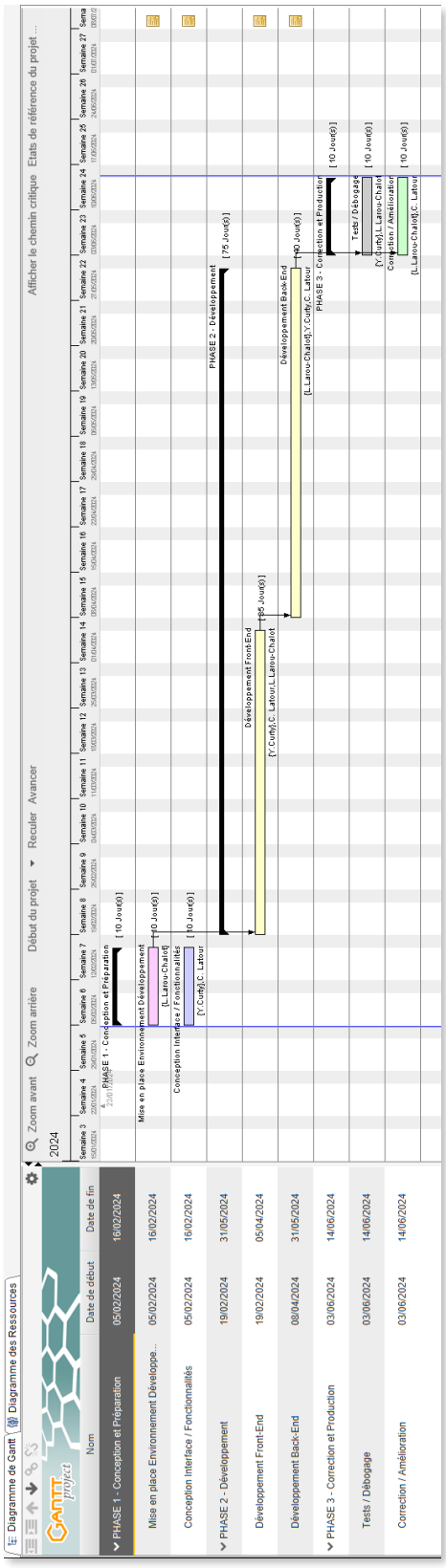


# Diagrammes de Gantt

React




Flutter



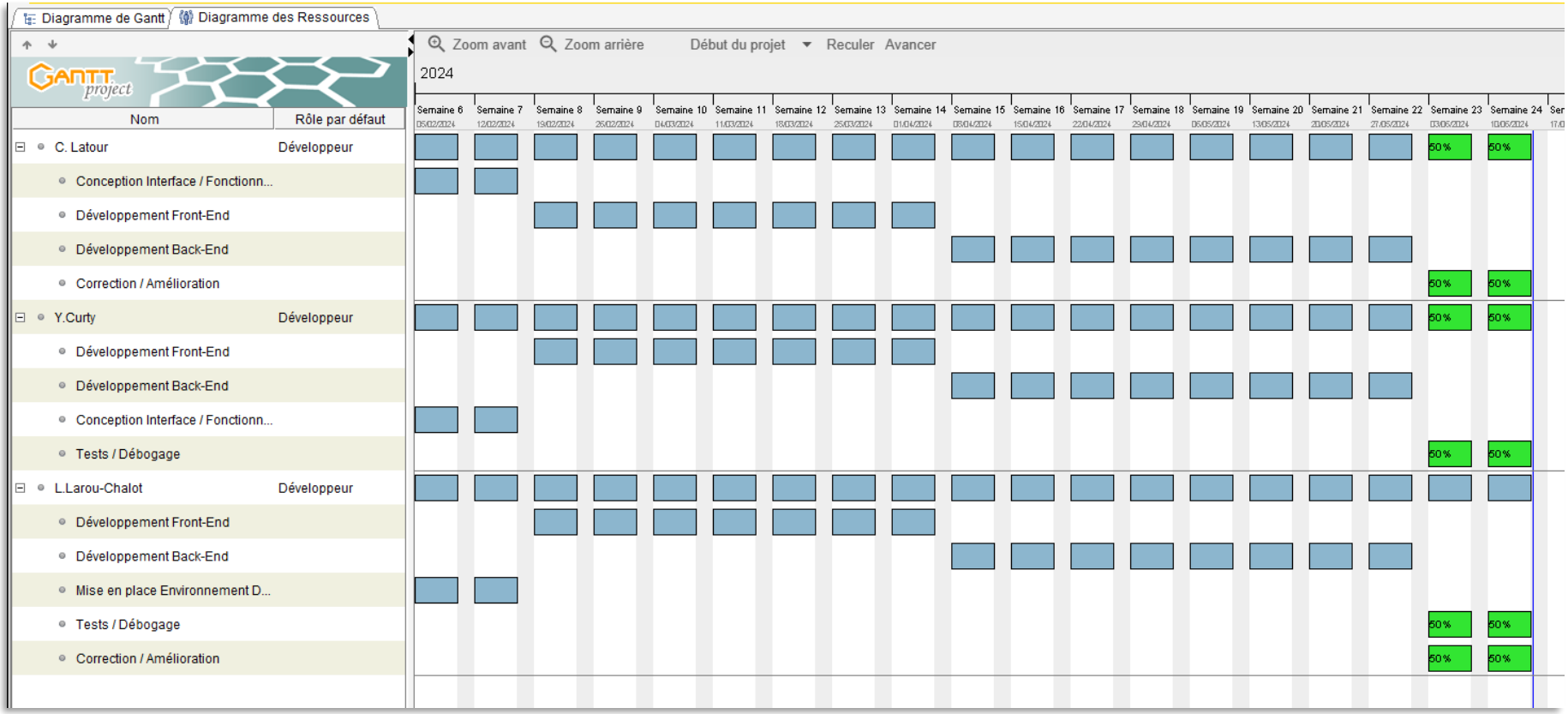


Diagramme de Gantt
Diagramme des Ressources
Zoom avant
Zoom arrière
Début du projet
Reculer
Avancer
2024
Nom
Rôle par défaut
C. Latour
Développeur
Conception Interface / Fonctionn...
Développement Front-End
Développement Back-End
Correction / Amélioration
Y.Curty
Développeur
Développement Front-End
Développement Back-End
Conception Interface / Fonctionn...
Tests / Débogage
L.Larou-Chalot
Développeur
Développement Front-End
Développement Back-End
Mise en place Environnement D...
Tests / Débogage
Correction / Amélioration
Semaine 6
Semaine 7
Semaine 8
Semaine 9
Semaine 10
Semaine 11
Semaine 12
Semaine 13
Semaine 14
Semaine 15
Semaine 16
Semaine 17
Semaine 18
Semaine 19
Semaine 20
Semaine 21
Semaine 22
Semaine 23
Semaine 24
50 %

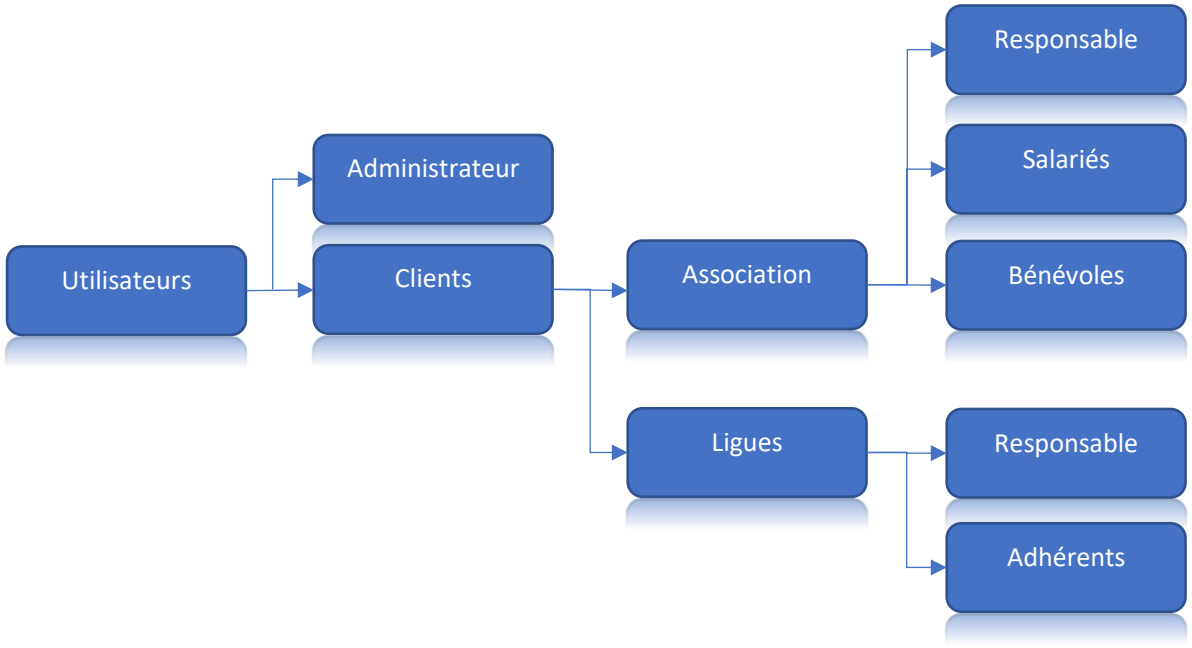
# La gestion des risques

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 devrons 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.




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 mal sera alors envoyé avec **un lien pour définir le mot de passe**.

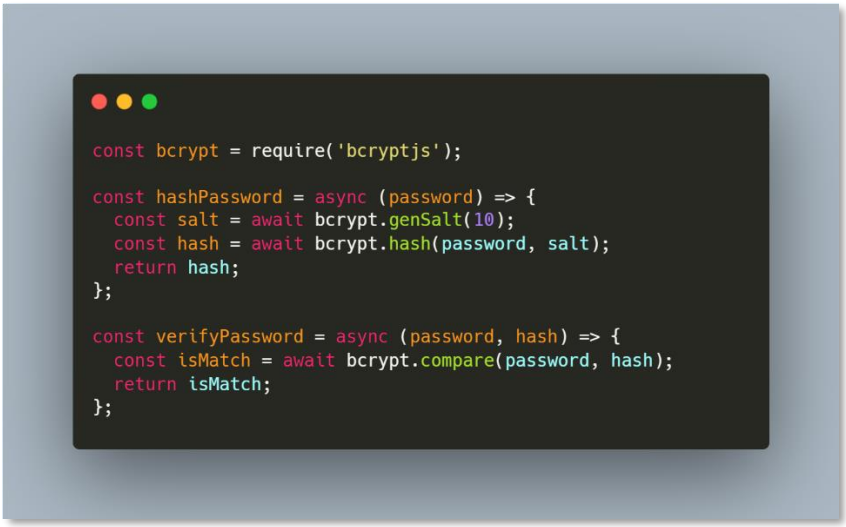
Cette pratique aura pour but d'éviter le partage du mot de passe en clair par voie électronique.

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

## *Code*



```
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;
};
```

La fonction « hashPassword » génère un sel de 10 tours de hashage, puis hache le mot de passe d'entrée avec le sel généré. Ensuite, il retourne le hachage résultant.

La fonction « verifyPassword » vérifie si un mot de passe d'entrée correspond au hachage stocké. Il utilise la fonction « compare » de « bcryptjs » pour comparer le hachage et le mot de passe d'entrée, et retourne un booléen indiquant si les deux correspondent.

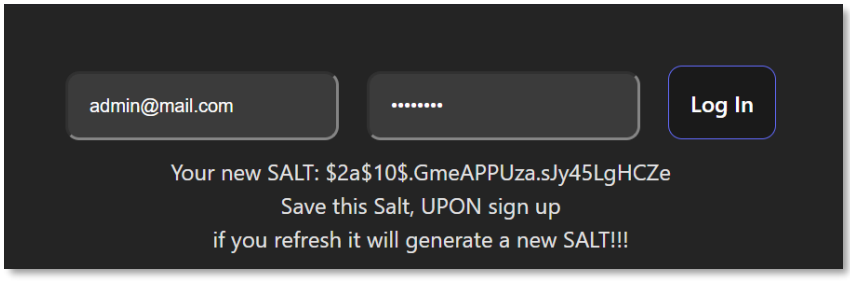
Il est important de noter que ces fonctions doivent être utilisées de manière sécurisée. Par exemple, les mots de passe ne doivent jamais être stockés en texte brut dans la base de donnée.

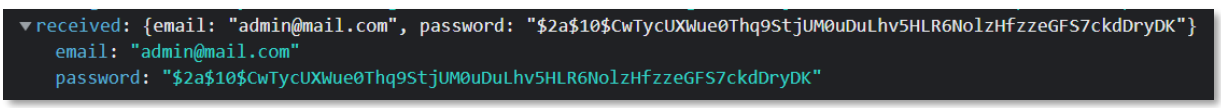
# Connexion depuis un réseau public

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.

## *Code*

Exemple de hashage lors de la connexion depuis un réseau public



```
import "./Form.css";
import { useRef } from "react";
import bcrypt from "bcryptjs";

export default function Form() {
    // SALT should be created ONE TIME upon sign up
    const salt = bcrypt.genSaltSync(10);
    // example =>  $2a$10$CwTycUXWue0Thq9StjUM0u => to be added always to the password hash
    const emailInputRef = useRef();
    const passwordInputRef = useRef();

    function handleLoginForm() {
        const email = emailInputRef.current.value;
        const password = passwordInputRef.current.value;
        const hashedPassword = bcrypt.hashSync(password, "$2a$10$CwTycUXWue0Thq9StjUM0u"); // hash created previously created upon sign
up

        fetch("https://api.sampleapis.com/beers/ale", {
            method: "POST",
            headers: {
                Accept: "application/json",
                "Content-Type": "application/json",
            },
            body: JSON.stringify({
                email: email,
                password: hashedPassword,
            }),
        });
    }

    return (
        <>
            <form>
                <input
                    style={{ padding: "15px", borderRadius: "10px", margin: "10px" }}
                    ref={emailInputRef}
                    type="email"
                    placeholder="Email"
                />
                <input
                    style={{ padding: "15px", borderRadius: "10px", margin: "10px" }}
                    ref={passwordInputRef}
                    type="password"
                    placeholder="Password"
                />
                <button
                    type="submit"
                    style={{ padding: "15px", borderRadius: "10px", margin: "10px" }}
                    onClick={(e) => {
                        e.preventDefault();
                        handleLoginForm();
                    }}
                >
                    Log In
                </button>
            </form>
            <span>Your new SALT: {salt}</span>
            <br />
            <span>
                Save this Salt, UPON sign up <br /> if you refresh it will generate a new SALT!!!
            </span>
        </>
    );
}
```

## 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.

# Documents sources et modèles

## Fiche de paye

## Fiche de poste