

Cahier des charges

Basic-Fat

Table des matières

Contexte	
.....	2
Objectif du projet et de l'application	
.....	2
Cahier des charges	
.....	3
Interfaces Graphique	
.....	3
Identification	
.....	3
Gestion des bases de données	
.....	4
Partie	
Programmation	
.4	
Livrables et documents	
attendus	4
Organisation du projet	5
Gestion des bases de données	6
Maquettes	
.....	6
Diagramme du cas	
.....	11
Modèle conceptuel de données.	12
Implémentation de la base de données dans PhpMyAdmin	
.....	12
Mission 3 – Partie	15

Contexte :

L'organisation Basic-Fat nécessite une application Web afin de permettre aux adhérents et aux coachs de gérer les différentes activités qui seront proposées par la salle de sport. Pour réaliser ce projet, l'équipe de Basic-Fat souhaiterait s'appuyer sur une application web avec une interface simple et utilisable par les adhérents ainsi que par les coachs sportifs.

Objectif du projet et de l'application :

L'objectif du projet est la création d'une application web qui peut permettre aux coachs de la salle de sport Basic-Fat d'inscrire de nouveaux adhérents de la salle de sport, ainsi ces derniers pourront prendre des rendez-vous avec différents coachs selon l'emploi du temps / planning de ces derniers, de travailler un muscle en particulier, mais aussi permettre à l'adhérent de connaître sa masse grasseuse, son pourcentage de masse musculaire... Les coachs pourront en parallèle suivre les activités qui ont été réalisés par un adhérent, l'historique de leur activité ainsi que de leur séance. Les coaches pourront proposer aussi des séances adaptées et personnalisées selon leurs besoins. Pour réaliser ce projet, l'équipe de Basic-Fat souhaiterait s'appuyer sur une application web avec une interface simple et utilisable par les adhérents ainsi que les coachs sportifs.

Cahier des charges

Interfaces Graphique :

L'interface graphique de ce projet doit prendre en compte les fonctionnalités suivantes :

- Utilisation du clavier et/ou de la souris en fonction des traitements.
- Placement du curseur dans le premier champ de saisie à l'ouverture d'une fenêtre.
- Présentation des données en mode fiche suite à un double clic sur une ligne d'une table.
- Vues multifenêtres avec mise à jour simultanée.
- Interface responsive optimisée sur téléphone.

Identification

L'accès à l'application nécessite l'adresse mail de l'utilisateur ainsi que son mot de passe.

La création d'un nouveau compte se fait sur l'interface du coach donc en physique.

Gestion des bases de données

- Diagramme de classes.
- Élaboration d'un MCD.
- Réalisation du modèle relationnel.
- Implémentation de la base de données dans PostgreSQL.

Partie Programmation

- Préparer le diagramme de classe.
- Créer des interfaces graphiques avec WindowBuilder.
- Développer méthodes et constructeurs via Symfony.
- Proposer des fonctionnalités et une implémentation mobile (Android / IOS)

Prise de notes restructurée

Application :

- Optimisation sur interface mobile
- Récurrence des séances => possibilité de créneau fixe, sauvegarde de séances ?
- Horaire d'ouverture de la salle
- Authentification:
 - => adresse mail + mdp
- Enregistrement des exercices
 - => saisie par les coachs :
 - => Liste complète de tous les exercices
 - => libellé, temps, explication rapide
- Planning (vue par Coach et Adhérent) :
 - => Séance disponible, nb place, adhérent inscrit (vue par les coachs)

Coach

(numCoach, dispoCoach)

- Admin de l'application
- Suivi d'entraînement => via la fiche san, historique des séances
 - => Visualisation des séances de l'Adhérent :
 - => pas de modification possible
- Planification des séances de la semaine (date, jour, nb de places disponible)
 - => possibilité de suppression des séances sur le planning
- réalisation des séances
- Création des inscrits sur le site :
 - => nom, prénom, date de naissance
 - => Adresse mail + mdp afin de se connecter
 - => Possibilité de modification par les coachs

Adhérent

(numAdherent, adresseAdherent)

- Fiche sanitaire:

 - => Enregistrement dans la base de données

- participation :

 - => Séance collective

 - => Séance libre

- Formulaire à remplir :

 - => Masse total

 - => Masse grasseuse

 - => Masse musculaire

 - => Possibilité de suppression par les coachs

 - => Validation du formulaire avec date, heure

=> Tableau d'évolution, // Graph d'évolution

 - => Progression personnel :

 - => masse total, masse musculaire, masse grasseuse

 - => possibilité de modification (erreur)

- Enregistrement des séances libres et collectives :

 - => Ajout des séances (nom des exos, temps de la séance)

 - => Modification des séances

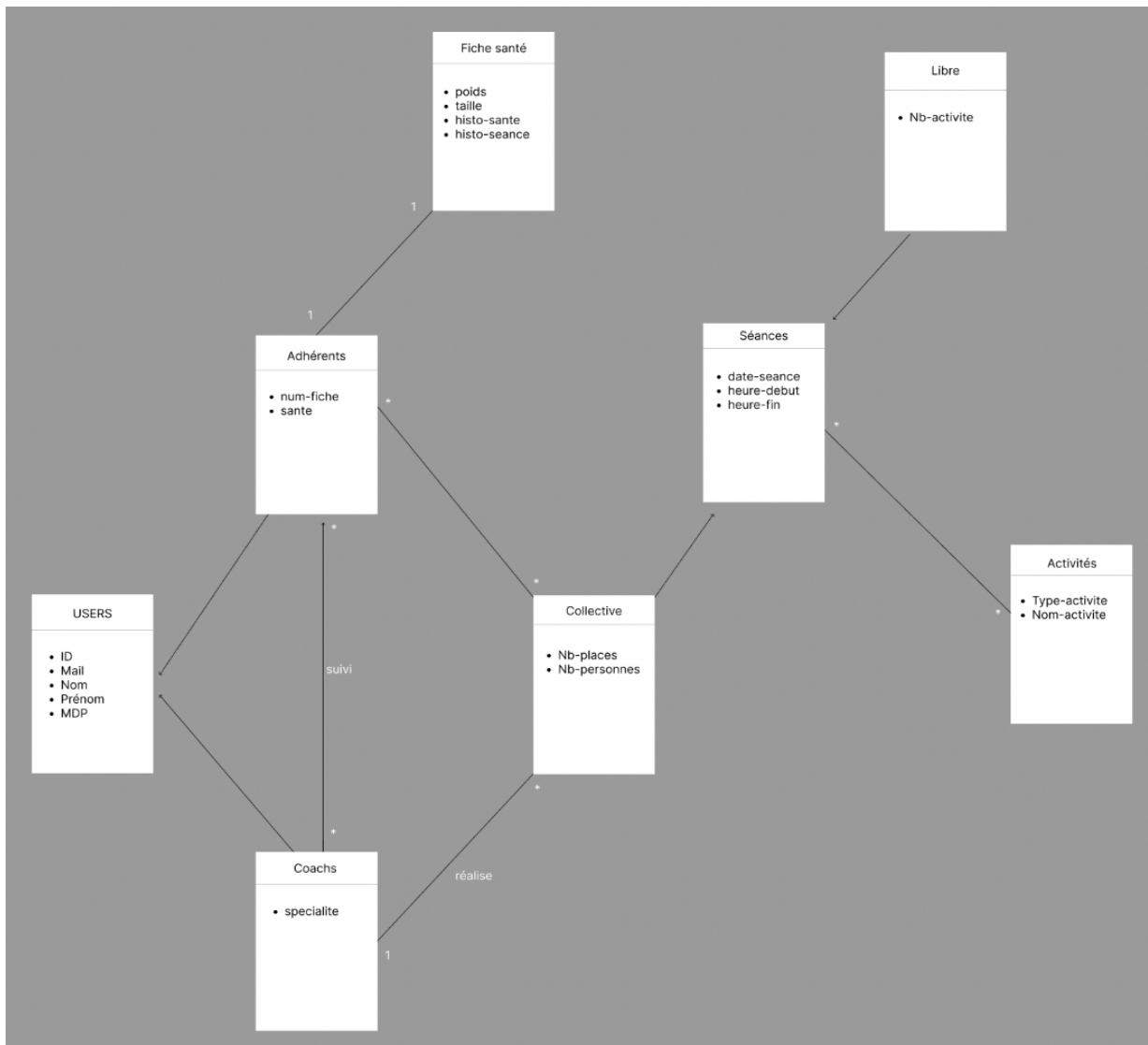
 - => Suppression des séances (si erreur)

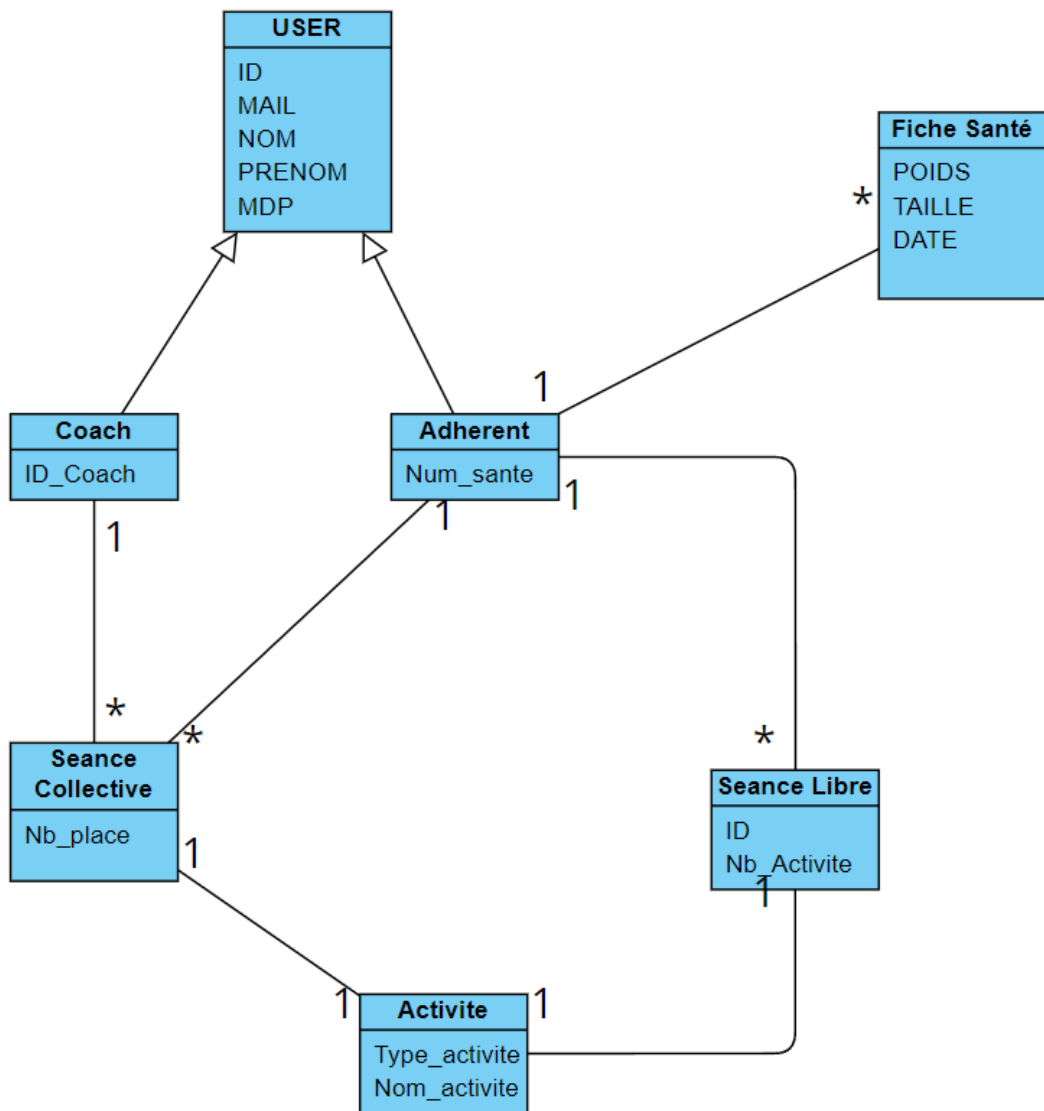
- Historique des séances (vue par Adhérent, Coach (barre de recherche):

User : Coach + Adherent

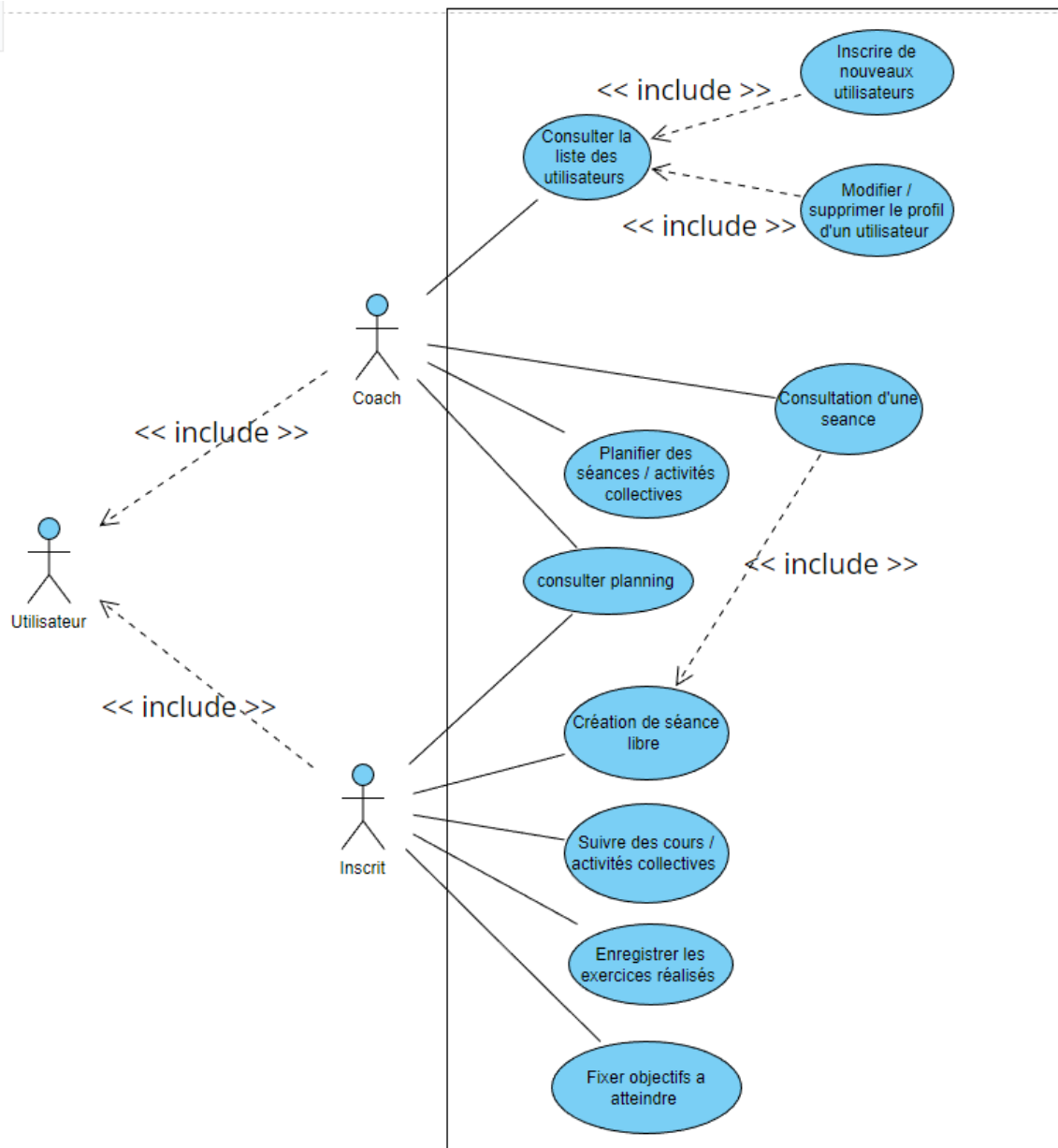
(id, nom, prenom, mail, mdp, numeroTel, sexe)

Diagramme de classe

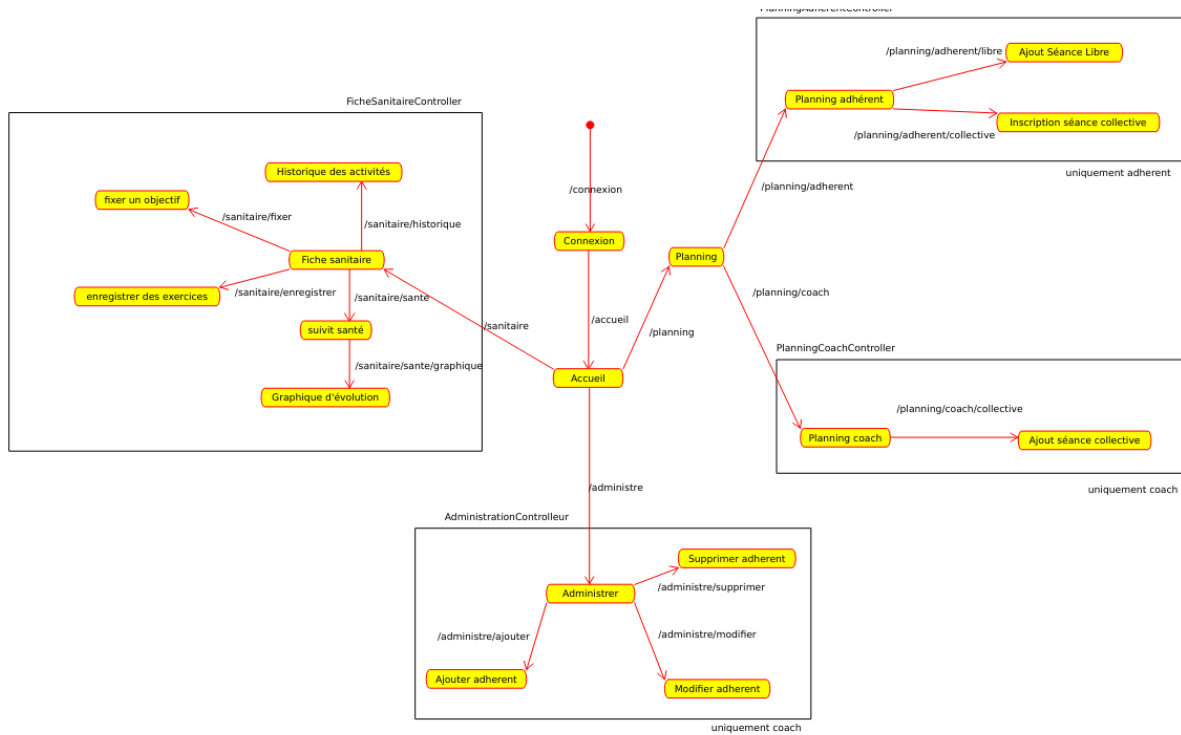




UseCase

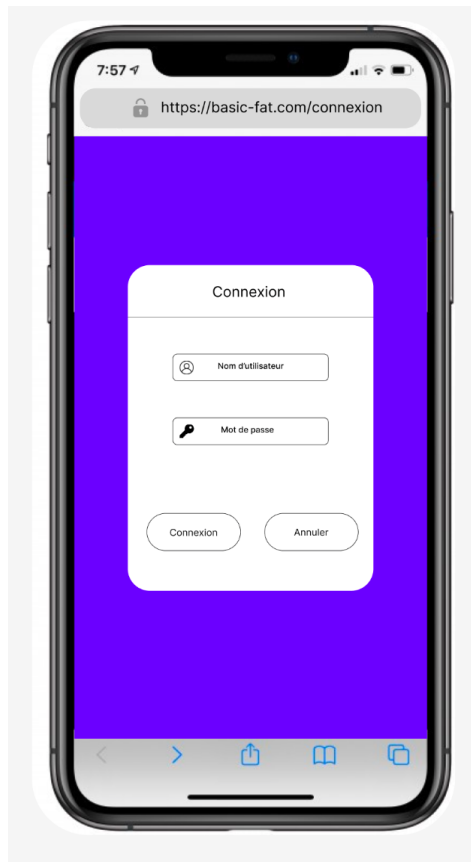


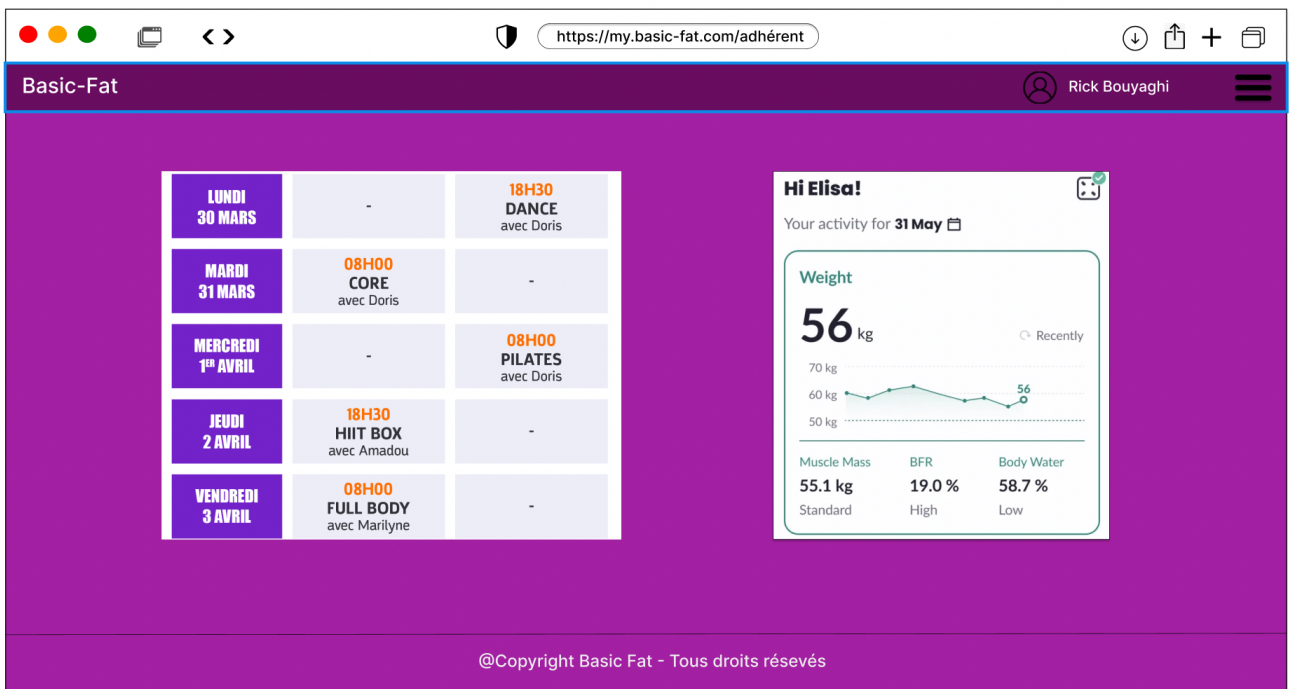
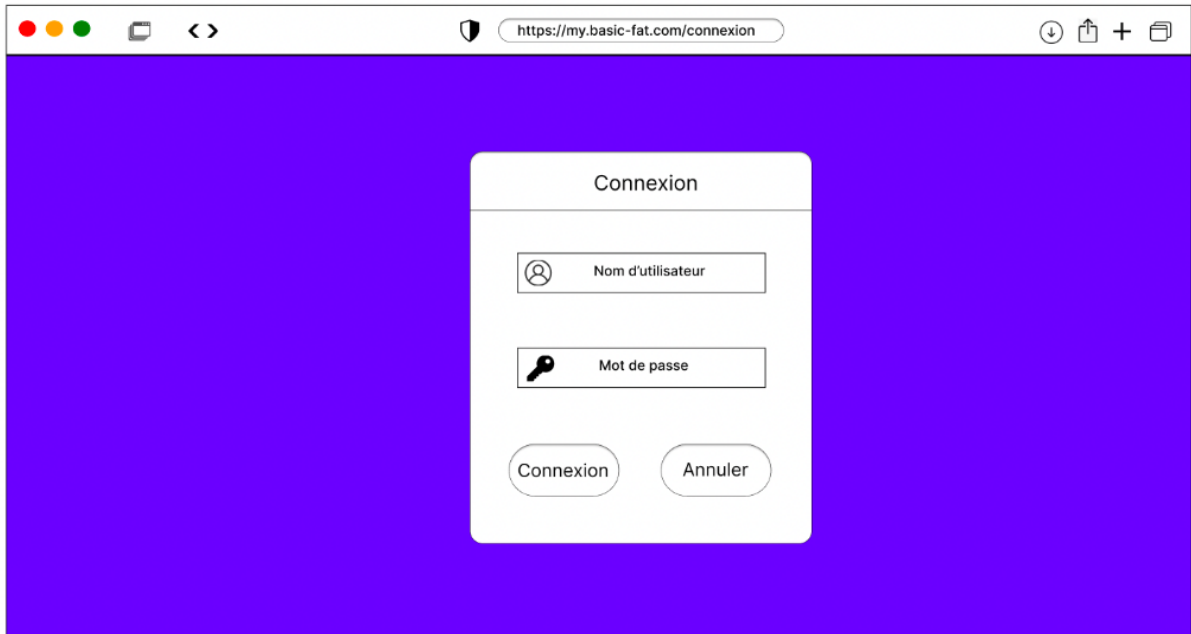
Cinématique des écrans :



Maquette du site internet

Page de Connexion sur mobile :





Connexion Gitea

gitea.btssio-poitiers.fr

Connexion base de donnée

admin_sport

<https://gitea.btssio-poitiers.fr/SioCorp-BasicFat/BasicFat.git>

```
* @Route("/", name="app_activite_index", methods={"GET"})  
* @Route("/{id}", name="app_activite_show", methods={"GET"})  
* @Route("/{id}/edit", name="app_activite_edit", methods={"GET", "POST"})  
* @Route("/{id}", name="app_activite_delete", methods={"POST"})  
  
* @Route("/", name="app_adherents_index", methods={"GET"})  
* @Route("/new", name="app_adherents_new", methods={"GET", "POST"})
```

A FAIRE !!!

**PAGE COACH : AJOUTER UNE ACTIVITÉ + AVOIR
LES SÉANCES DES ADHÉRENTS + FICHES
SANITAIRES DES ADHÉRENTS + MODIFIER LE**

**PLANNING DE LA SALLE (SEANCE , ACTIVITÉ
DISPONIBLE ETC...)**

**PAGE ADHERENTS : LIEN SEANCE ET ADHÉRENTS
+ MODIFIER FICHE SANITAIRE + AJOUTER UN
OBJECTIF , SUIVI ACTIVITE**

```
<section id="trainers">
  <h3>Emploi du Temps</h3>
  <div class="container">
    <table class="table-fill">
      <thead>
        <tr>
          <th>Numéro</th>
          <th>Activité</th>
          <th>Type</th>
          <th>Heure de début</th>
          <th>Heure de fin</th>
          <th>Jour de la semaine</th>
          <th>Actions</th>
        </tr>
      </thead>
      <tbody>
        {% for emploi_temp in emploi_temps %}
          <tr>
            <td>{{ emploi_temp.id }}</td>
```

```

        <td>{{ emploi_temp.nom }}</td>
        <td>{{ emploi_temp.type }}</td>
        <td>{{ emploi_temp.heureDebut ?
emploi_temp.heureDebut|date('H:i') : '' }}</td>
        <td>{{ emploi_temp.heureFin ?
emploi_temp.heureFin|date('H:i') : '' }}</td>
        <td>{{ emploi_temp.jourSemaine ?
emploi_temp.jourSemaine|date('d-m-Y') : '' }}</td>
        <td>
            <a href="{{
path('app_emploi_temps_edit', {'id': emploi_temp.id})
}}">Modifier</a>
        </td>
    </tr>
    {% else %}
    <tr>
        <td colspan="7">rien de prévu
        ....</td>
    </tr>
    {% endfor %}
</tbody>
</table>
<button>
    <a href="{{ path('app_emploi_temps_new')
    }}">Réserver</a>
</button> </div>
</section>

```