



UNIVERSITE DE PARIS



SUJET :

RAPPORT DU PROJET WEB GESTION D'UN CLUB SPORTIF

SPECIALITE :

Master 1 AMDS FI

FAIT PAR :

GHARBI MOHAMED

Année Scolaire : 2020/2021

TABLE DES MATIERES

Introduction :	1
Solution proposée	1
Partie 1 : Conception	2
Introduction :	2
Diagrammes utilisés :	2
Partie 2 : Les services	5
Service des entrainements :	5
Service des inventaires :	5
Service des ressources :	6
Partie 3 : Implémentation de la solution	6
Introduction :	6
L'architecture de l'application :	6
Les outils utilisés :	7
Les différentes bases de données :	8
Présentation de l'application :	9
Conclusion :	15

LISTE DES FIGURES :

FIGURE 1: DIAGRAMME DE CAS D'UTILISATION.....	2
FIGURE 2: DIAGRAMME DE CLASSES.....	3
FIGURE 3: FIGURE DES DIFFERENTS SERVICES DE L'APPLICATION	5
FIGURE 4 :ARCHITECTURE DE L'APPLICATION	6
FIGURE 5: LES DIFFERENTS SCHEMAS ET BASES DE DONNEES	8
FIGURE 6: AUTHENTIFICATION	9
FIGURE 7: LISTE DES EQUIPES	10
FIGURE 8: L'AJOUT D'UNE EQUIPE	10
FIGURE 9: LA MODIFICATION D'UNE EQUIPE	11
FIGURE 10: L'AJOUT D'UNE RESSOURCE	11
FIGURE 11: LISTE DES INVENTAIRES	12
FIGURE 12: LISTE DU MATERIEL	12
FIGURE 13: LA CREATION DU DOCKERFILE	13
FIGURE 14: GENERATION DU FICHIER TARGET	13
FIGURE 15 : GENERATION DU JAR	13
FIGURE 16: LA CREATION DU DOCKER	14
FIGURE 17: LA PUBLICATION ET L'EXECUTION DE L'IMAGE	14

Introduction :

De nos jours, la collecte de l'information a tellement évolué, en terme de volume et de processus que son informatisation est devenue inévitable.

Le club sportif IRHD n'échappe pas à la règle : De plus en plus de processus et procédures à gérer; ces dernières ont besoin d'être automatisés pour une gestion optimale.

Le projet à réaliser, fera le sujet de la conception et la réalisation d'une application web « gestion d'un club sportif » qui sera basée sur une architecture micro-service .

Le présent document sera divisé en trois parties :

- **Partie 1** : Conception de la solution.

Dans cette partie, nous allons utiliser un des outils de modélisations UML pour schématiser notre solution sous forme d'un diagramme de classe et de cas d'utilisation.

- **Partie 2** : La définition des trois micro-services.

La deuxième partie fera partie de la définition des trois micro-services de notre application.

- **Partie 3** : Réalisation et l'implémentation de la solution.

Cette partie est composée de deux parties. En premier lieu, nous présentons les différents outils utilisés pour la réalisation. Puis, nous présentons le travail réalisé à travers la création des bases de données, la connexion entre les différents services en illustrant avec des captures d'écran de l'application.

Solution proposée

Après avoir fait l'analyse et la collecte des besoins du gérant du club sportif « IRHD », nous pouvons à présent proposer une solution adéquate afin de répondre à la problématique à savoir :

- La décomposition de notre projet en trois sous-projets.
- La conception de la solution.
- La réalisation d'une application full web qui permettra à l'utilisateur de gérer le club.

Partie 1 : Conception

Introduction :

Dans cette partie nous allons aborder l'analyse et la conception de la solution qui sera schématisé sous forme de diagrammes en utilisant l'un des outils de modélisation qui est UML.

Diagrammes utilisés :

Diagramme de cas d'utilisation :

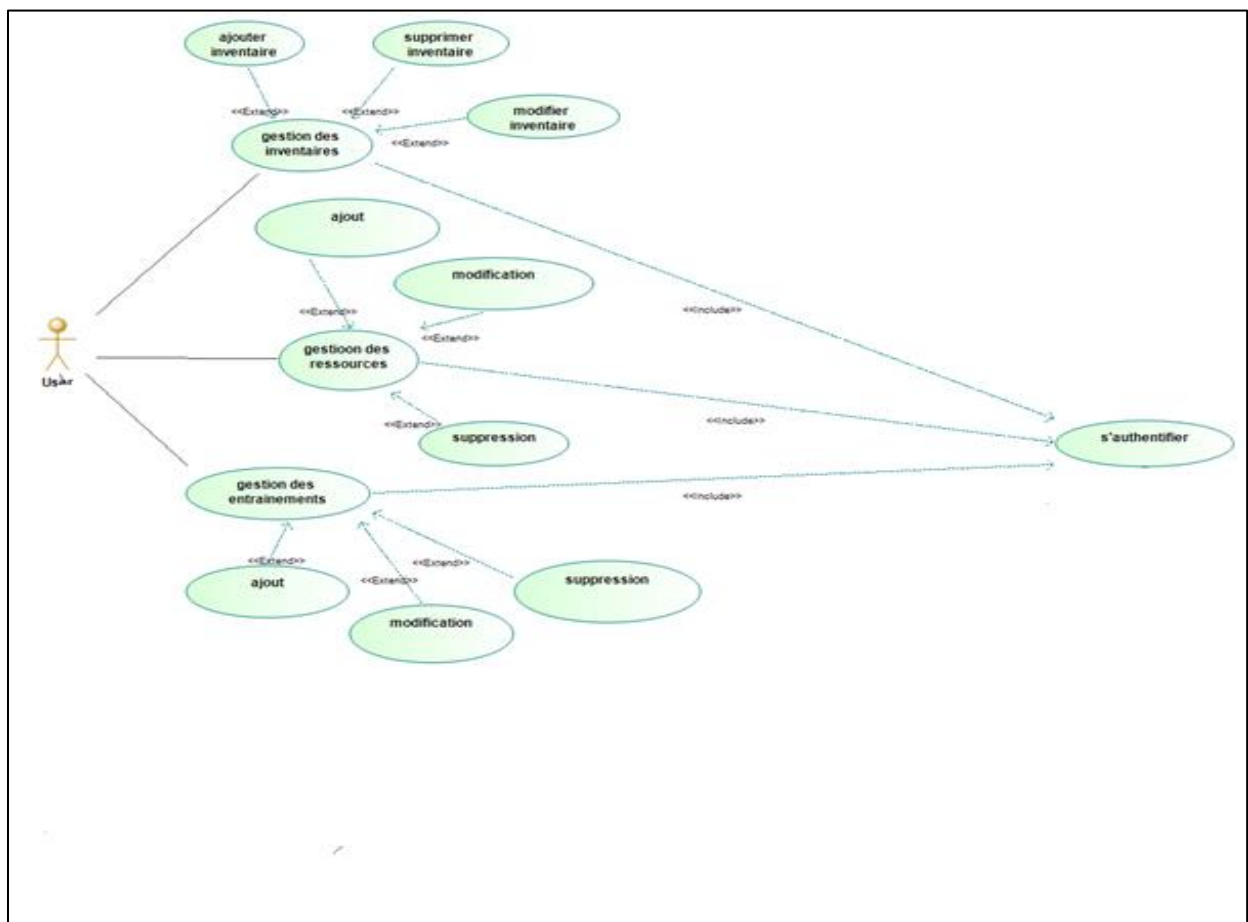


Figure 1: Diagramme de cas d'utilisation

Description du diagramme

Acteur1 : User

Cas1 : Gestion des entrainements :

- L'ajout et la modification d'une équipe ou d'un entraineur.
- L'ajout, la suppression et la modification d'un joueur, d'une équipe ou d'un entraineur.

Cas2 : Gestion des ressources:

- L'ajout, la modification et la suppression d'un investisseur.

Cas3 : Gestion des inventaires:

- L'ajout, la modification et la suppression d'un article.
- L'ajout et la modification d'un inventaire.

Toutes les fonctionnalités seront faites après avoir authentifier.

Diagramme de classe :

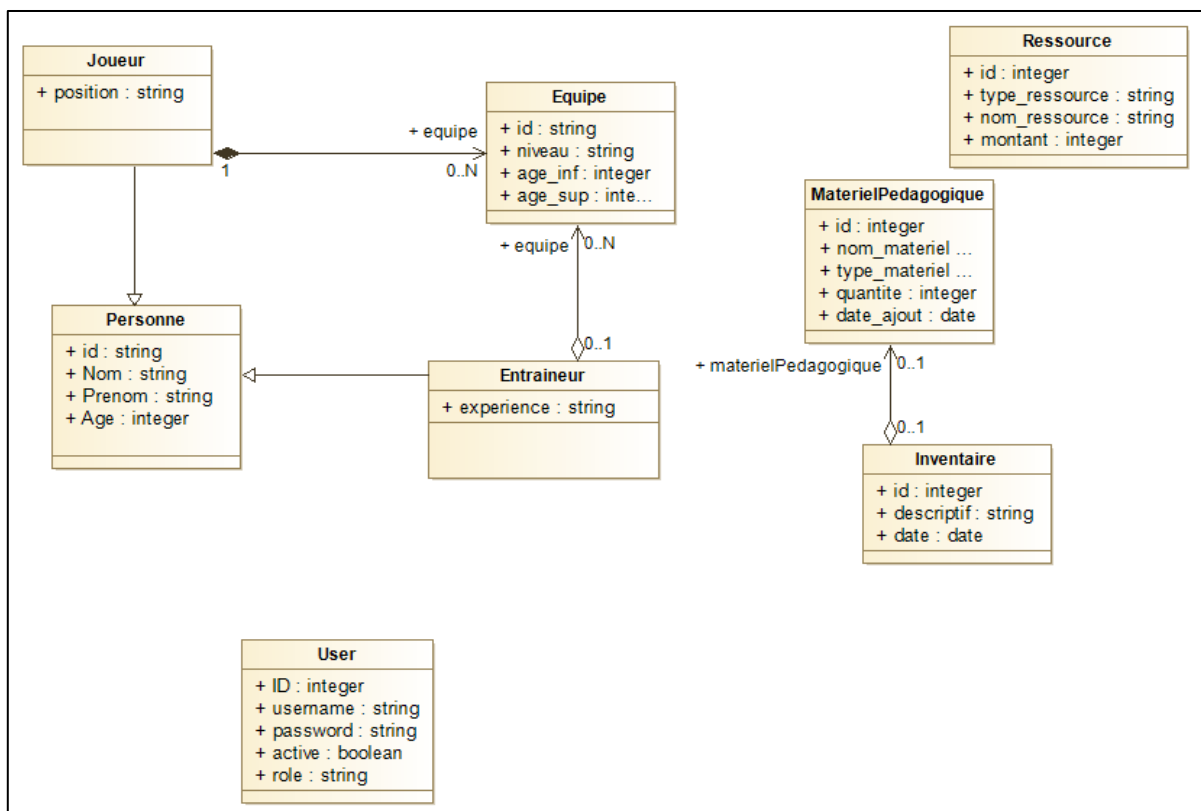


Figure 2: Diagramme de classes

D'après le diagramme de classe nous concluons que :

- L'entraîneur peut avoir une à plusieurs équipes.
- L'équipe est composée d'un ou plusieurs joueurs
- Un inventaire a un ou plusieurs matériaux pédagogiques.

Les règles de passages du diagramme de classe au modèle relationnel

Dans ce qui suit, nous allons présenter les différentes règles de passage, qui nous ont servis lors de l'élaboration du modèle relationnel des données :

- Affecter une table à chaque classe.
- La migration de la clé primaire de la table mère à la table fille, due à une association « un à plusieurs ».
- La représentation d'une association « plusieurs à plusieurs » est effectuée par une table ayant pour clé primaire la concaténation des clés primaires des deux tables associées.
- La représentation d'un héritage complet se fait en intégrant tous les attributs de la classe mère dans ses classes filles si la classe mère est une classe abstraite.

Le schéma relationnel

Ressource(id, type-ressource, nom_ressource, montant) ;

User(id, username, password, active, role);

Entraîneur(id_personne, nom, prenom, age, experience) ;

Joueur(id_personne, nom, prenom, age, position, id_equipe*) ;

Equipe(id_equipe, niveau, age_inf, age_sup, id_personne*);

Materielpedagogique(id, nom_materiel, type_materiel, quantite, date_ajout, idinv*) ;

Inventaire(id, date_ajout, descriptif) ;

Partie 2 : Les services

Pour notre projet, nous avons opté pour une gestion d'un club sportif composée de plusieurs services. Afin de comprendre les fonctionnalités du projet nous avons introduit ce schéma :



Figure 3: Figure des différents services de l'application

Service des entrainements :

Ce service a pour but la gestion des entrainements, il permet à l'utilisateur d'ajouter, de supprimer ou de modifier une équipe, un entraîneur et un joueur.

Tel que l'ajout d'un joueur ou d'une équipe suivent certaines règles à savoir :

- **L'ajout d'un joueur :**

S'assurer que la liste des équipes n'est pas vide.

- **L'ajout d'une équipe :**

S'assurer que la liste des entraîneurs n'est pas vide.

Service des inventaires :

Ce service offre à l'utilisateur la possibilité d'ajouter, modifier ou supprimer un inventaire et un matériel.

Le service des inventaires traite la gestion des inventaires et donne à l'utilisateur une idée sur ses dépenses et ses achats effectués { matériel pédagogique, matériel de rénovation ...etc.} tout au long de l'année.

Service des ressources :

Dans ce service nous traitons la gestion des ressources et qui a pour but l'ajout, la suppression et la modification d'une ressource.

Connaitre les ressources, revient à savoir le montant annuel versé par les différents bénévoles du le club.

Partie 3 : Implémentation de la solution

Introduction :

Après avoir terminé la conception, nous entamons la partie réalisation et la mise en œuvre de la solution. Pour cela, il a été indispensable d'utiliser un certain nombre d'outils et mettre en place des environnements d'exécution.

L'architecture de l'application :

Pour notre approche nous avons choisi l'approche de liaison direct des différents micro-services et le frontend.

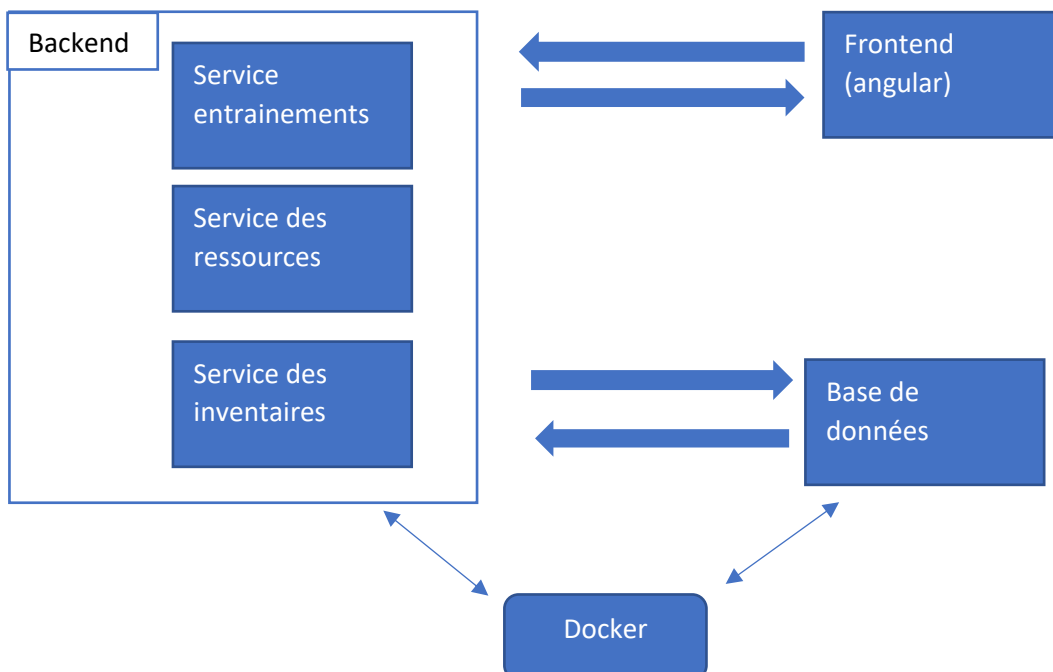


Figure 4 :Architecture de l'application

D'après l'architecture ci-dessus nous avons vu que notre application web est conçue de plusieurs parties à savoir :

- Partie frontend :

Cette partie a été faite en utilisant HTML, CSS, JQuery et angular.

- Partie Backend :

Nous avons utilisé Spring boot.

- Partie base de données

Dans cette partie, nous avons utilisé Le SGBD Postgresql et une fonctionnalité de Spring boot « JPA Repository »

- Partie Docker :

Nous avons utilisé Docker desktop et docker hub pour créer les images de chaque service.

Les outils utilisés :

➤ Spring boot

Spring Boot est un Framework qui facilite le développement d'applications web basé sur le langage Java. Les applications fondées sur Spring offrent des outils permettant d'obtenir une application packagée en jar, totalement autonome. Principalement utilisé pour produire des pages Web dynamiques via un serveur http.

➤ Postgresql :

La base de données a été implémentée sous PostgreSQL qui est un SGBD puissant, open source avec plus de 30 ans de développement actif qui lui a valu une solide réputation de fiabilité, de robustesse des fonctionnalités et de performances. En plus, c'est le SGBD utilisé au sein du département (systèmes existants). PostgreSQL est un système de gestion de base de données relationnelles et objets (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD.

➤ **Angular :**

Angular est une plateforme de développement qui permet de créer des applications web dynamiques et immersives.

➤ **Docker :**

Docker est un outil qui peut emballer une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur.

Les différentes bases de données :



Figure 5: Les différents schémas et bases de données

Présentation de l'application :

Authentification :

Afin de pouvoir accéder et naviguer dans l'application, l'utilisateur doit s'authentifier en utilisant comme nom d'utilisateur « admin » et mot de passe « admin ».

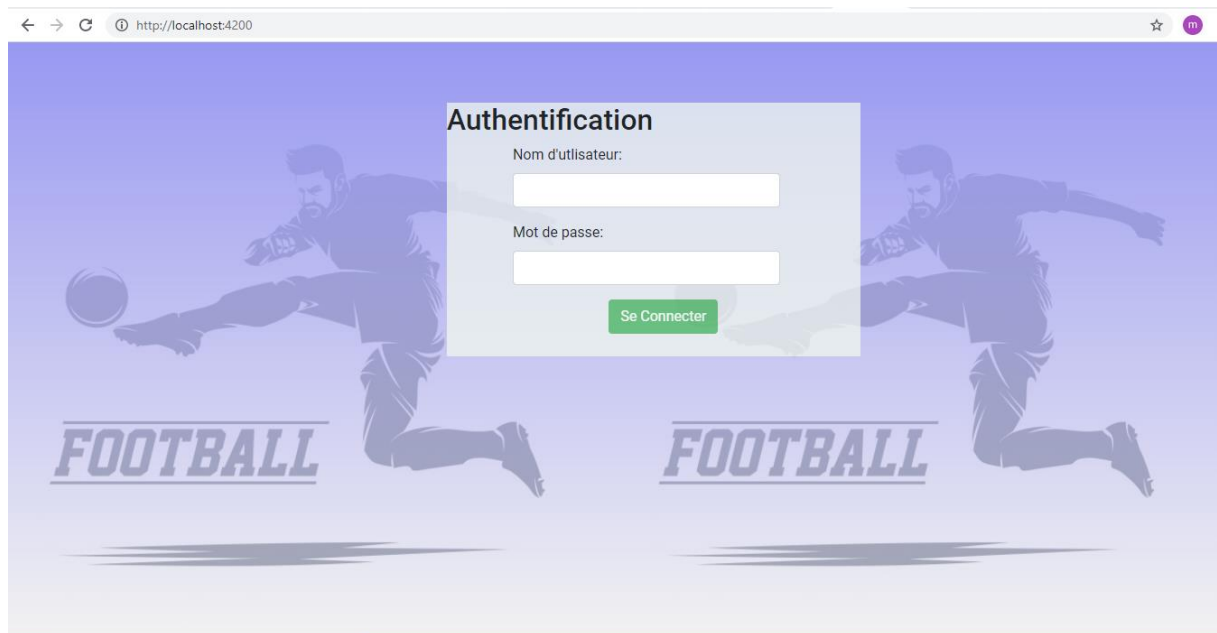


Figure 6: Authentification

Gestion de l'entraînement :

Dans ce volet, nous pouvons naviguer entre les différents volets « équipe, entraîneur, Joueurs » avec la possibilité de faire un ajout, modification et une suppression.

Pour l'ajout, nous devons au moins avoir un seul entraîneur pour ajouter une équipe et une seule équipe pour ajouter un joueur.

Dans les figures suivantes, nous allons appliquer ces fonctionnalités pour les équipes.

➤ Liste des équipes :



Figure 7: Liste des équipes

➤ L'ajout d'une équipe :

Afin d'ajouter une équipe, nous devons fournir toutes les informations de l'équipe et aussi l'identifiant de l'entraîneur.

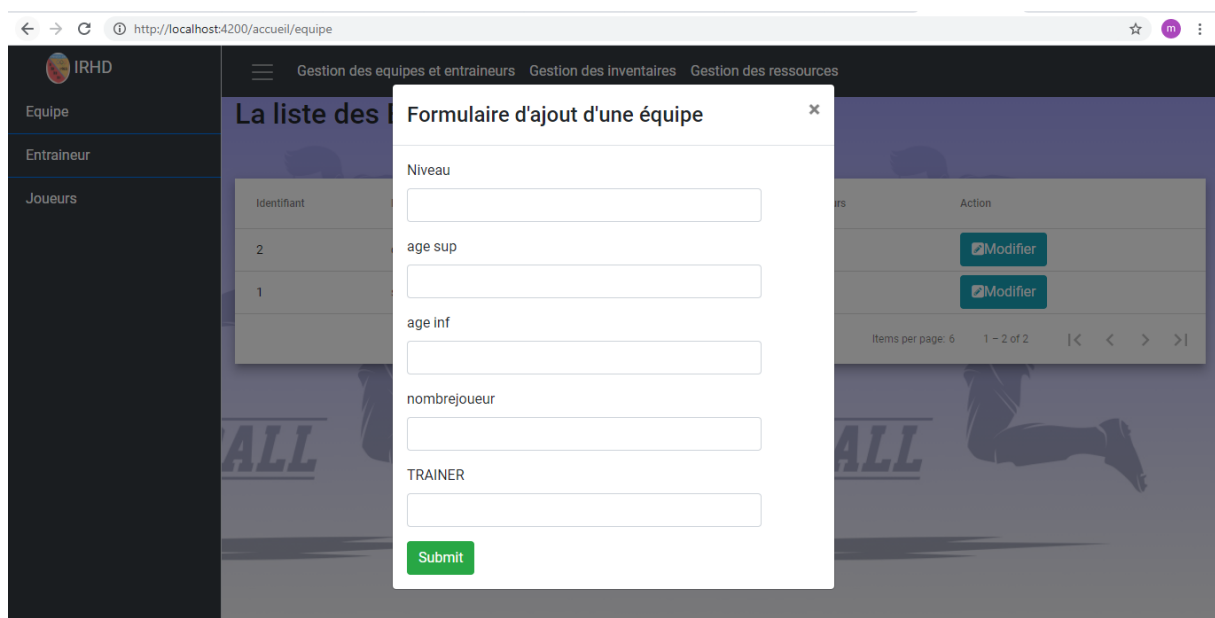


Figure 8: L'ajout d'une équipe

➤ La modification de l'équipe :

En cliquant sur le bouton « Modifier » nous aurons un Modal contenant les informations de l'équipe et vous pouvez les modifier et puis valider vos modifications.

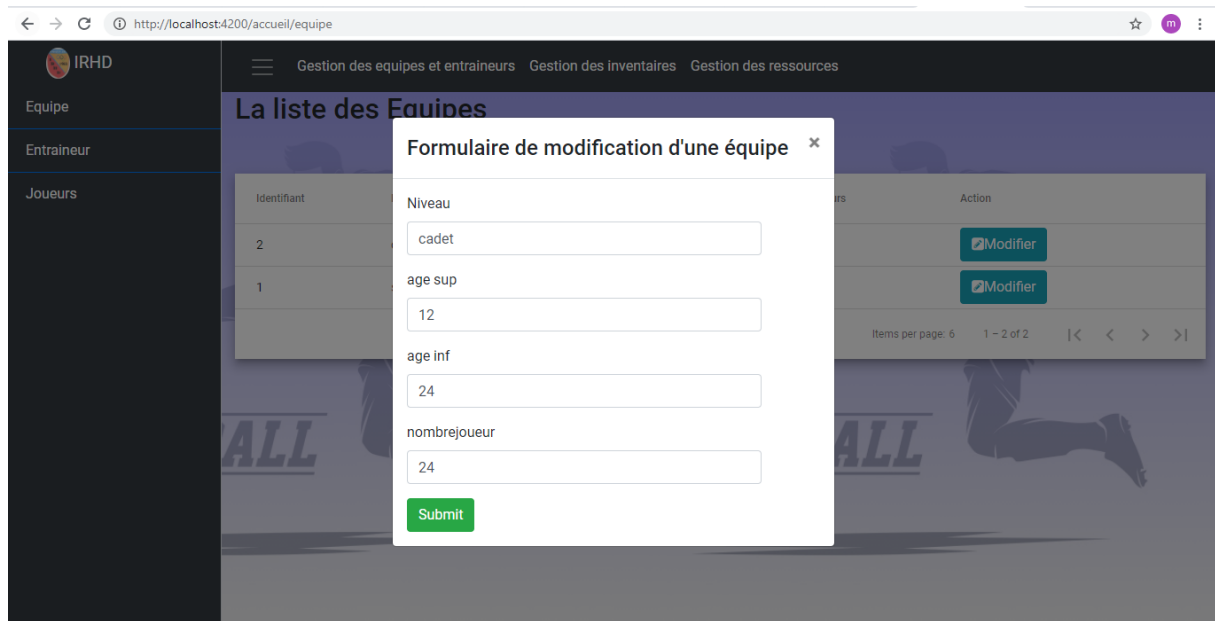


Figure 9: La modification d'une équipe

Gestion des ressources :

Ce volet traite les différentes fonctionnalités sur les ressources du club.



Figure 10: L'ajout d'une ressource

Gestion des inventaires :

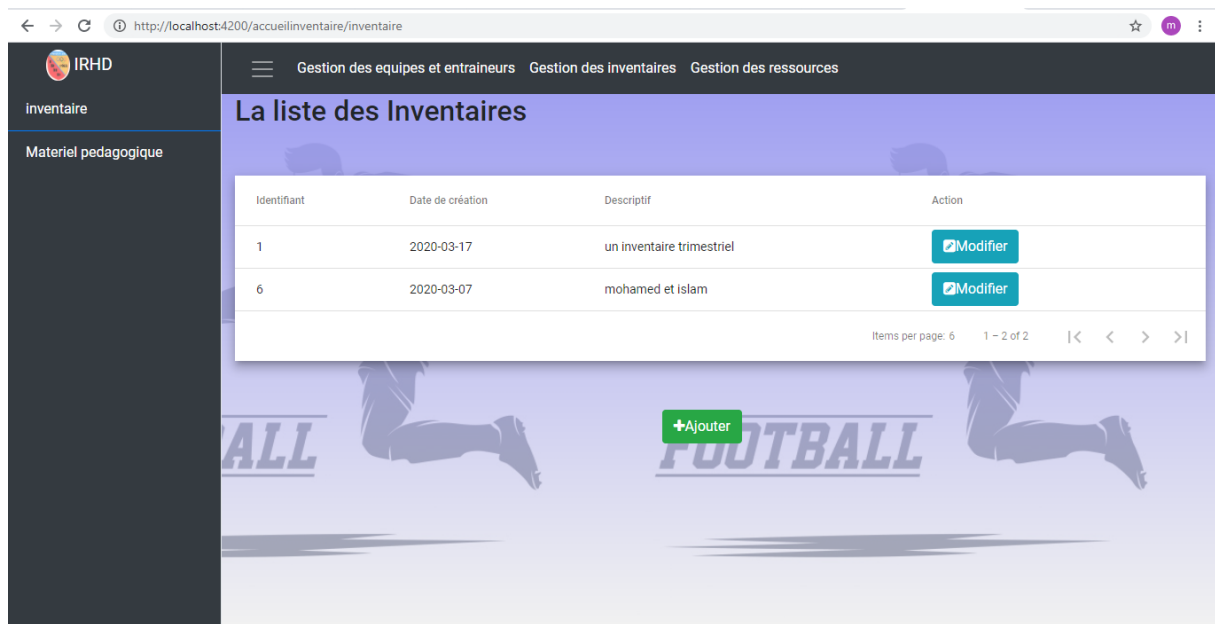


Figure 11: Liste des inventaires

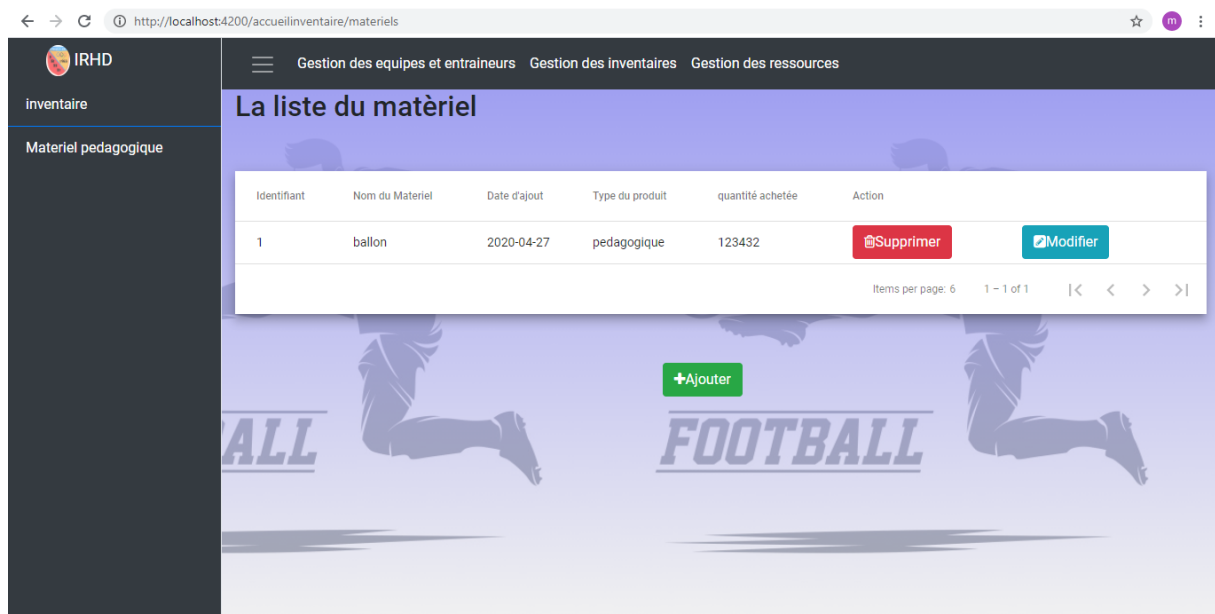


Figure 12: Liste du matériel

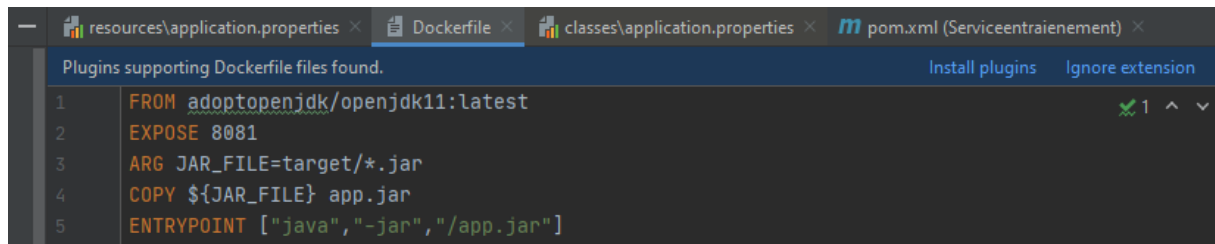
Les dockers :

Grace à docker, un logiciel libre permettant de lancer des applications dans des conteneurs logiciels, nous avons créé les images de chaque micro-service ainsi que le service front.

Dans cette étape nous allons détailler l'utilisation du docker en prenant comme exemple un des services « service des entrainements ».

Service des entrainements :

La création du docker file:



```
1 FROM adoptopenjdk/openjdk11:latest
2 EXPOSE 8081
3 ARG JAR_FILE=target/*.jar
4 COPY ${JAR_FILE} app.jar
5 ENTRYPOINT ["java", "-jar", "/app.jar"]
```

Figure 13: La création du dockerfile

La génération du jar en utilisant l'instruction « mvn clean install »

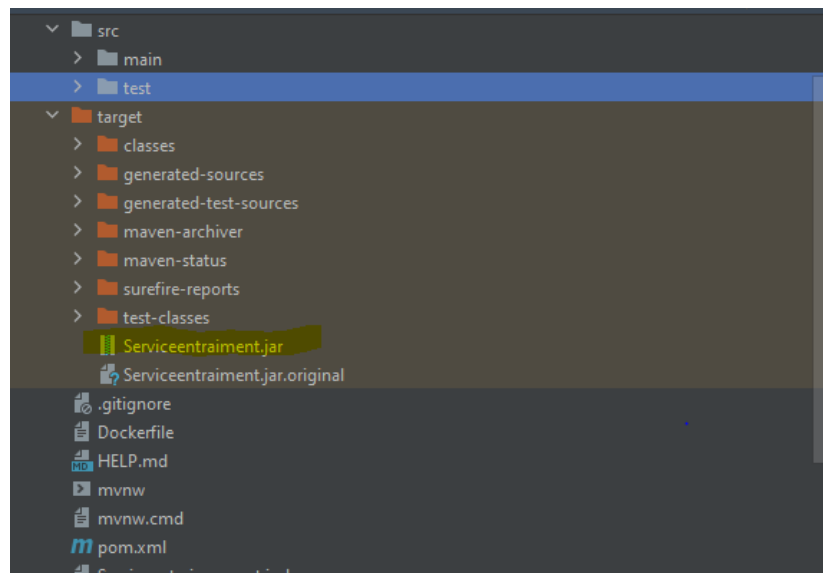


Figure 14: Génération du fichier target

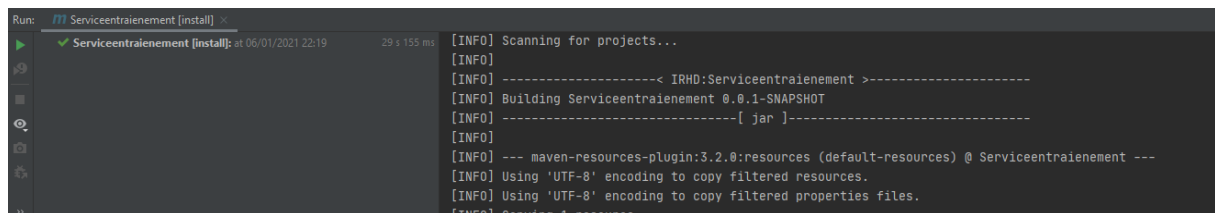


Figure 15 : génération du jar

La création du docker « docker-trainings »

```
PS C:\Users\gharbi\Desktop\gestion-club-sportif\Serviceentraînement> docker build -t docker-trainings .
[+] Building 0.3s (2/2) FINISHED
=> [internal] load build definition from Dockerfile                                0.3s
=> => transferring dockerfile: 255B                                              0.3s
=> [internal] load .dockerignore                                                  0.1s
=> => transferring context: 2B                                                  0.0s
failed to solve with frontend dockerfile.v0: failed to create LLB definition: dockerfile parse error line 2: unknown ins
truction: //RUN
PS C:\Users\gharbi\Desktop\gestion-club-sportif\Serviceentraînement> docker build -t docker-trainings .
[+] Building 4.7s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                1.0s
=> => transferring dockerfile: 170B                                              0.7s
=> [internal] load .dockerignore                                                  0.1s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/adoptopenjdk/openjdk11:latest          2.0s
=> [auth] adoptopenjdk/openjdk11:pull token for registry-1.docker.io            0.0s
=> CACHED [1/1] FROM docker.io/adoptopenjdk/openjdk11:latest@sha256:446754c82808a76c88232d1808e374cbb8d07b34d  0.0s
=> [internal] load build context                                                  0.0s
=> => transferring context: 70B                                                  0.0s
=> [2/2] COPY target/*.jar app.jar                                              0.1s
=> exporting to image                                                            0.1s
=> => exporting layers                                                            0.1s
=> => writing image sha256:2f25f9078783cf75ad639e5f36d77012c0f04bb8731d5f77b0b082277e9ecf6e                0.0s
=> => naming to docker.io/library/docker-trainings                             0.0s
PS C:\Users\gharbi\Desktop\gestion-club-sportif\Serviceentraînement> docker tag docker-trainings mohamedgharbi/projetwe
```

Figure 16: La création du docker

La publication du docker sur docker hub et son lancement

```
PS C:\Users\gharbi\Desktop>gestion-club-sportif\Serviceentraînement> docker push mohamedgharbi/projetweb
Using default tag: latest
The push refers to repository [docker.io/mohamedgharbi/projetweb]
702eb4ac535e: Layer already exists
2cb4ead5895f: Mounted from adoptopenjdk/openjdk11
1d1a1121ec80: Mounted from adoptopenjdk/openjdk11
f6253634dc78: Mounted from adoptopenjdk/openjdk11
9069f84dbbe9: Mounted from adoptopenjdk/openjdk11
bacd3af13903: Mounted from adoptopenjdk/openjdk11
latest: digest: sha256:4e7f836522ea82d8edc4a9273065e1d4a6d49aad7df6e54d757a0bec66dda65d size: 1580
PS C:\Users\gharbi\Desktop>gestion-club-sportif\Serviceentraînement> docker run -p 8080:8081 mohamedgharbi/projetweb

  ____ _
 / ___ \ | |
| |   \| |_| |
| |___) | | |
|_____||_|

=====|=====|==/=/=/
:: Spring Boot ::                (v2.4.0)

2021-01-06 21:32:11.159 INFO 1 --- [main] I.S.ServiceentraînementApplication : Starting Serviceentra
nementApplication v0.0.1-SNAPSHOT using Java 11.0.9.1 on 914a49283225 with PID 1 (/app.jar started by root in /)
2021-01-06 21:32:12.555 INFO 1 --- [main] I.S.ServiceentraînementApplication : No active profile set,
falling back to default profiles: default
2021-01-06 21:32:17.687 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Da
ta JPA repositories in DEFERRED mode.
2021-01-06 21:32:17.763 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data re
pository scanning in 61 ms. Found 3 JPA repository interfaces.
2021-01-06 21:32:26.157 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with
port(s): 8081 (http)
2021-01-06 21:32:27.379 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomca
t]
2021-01-06 21:32:27.379 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine
: [Apache Tomcat/9.0.39]
2021-01-06 21:32:29.294 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring emb
bedded WebApplicationContext
2021-01-06 21:32:29.295 INFO 1 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext
ext: initialization completed in 16642 ms
2021-01-06 21:32:33.913 INFO 1 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorSe
rvice 'applicationTaskExecutor'
```

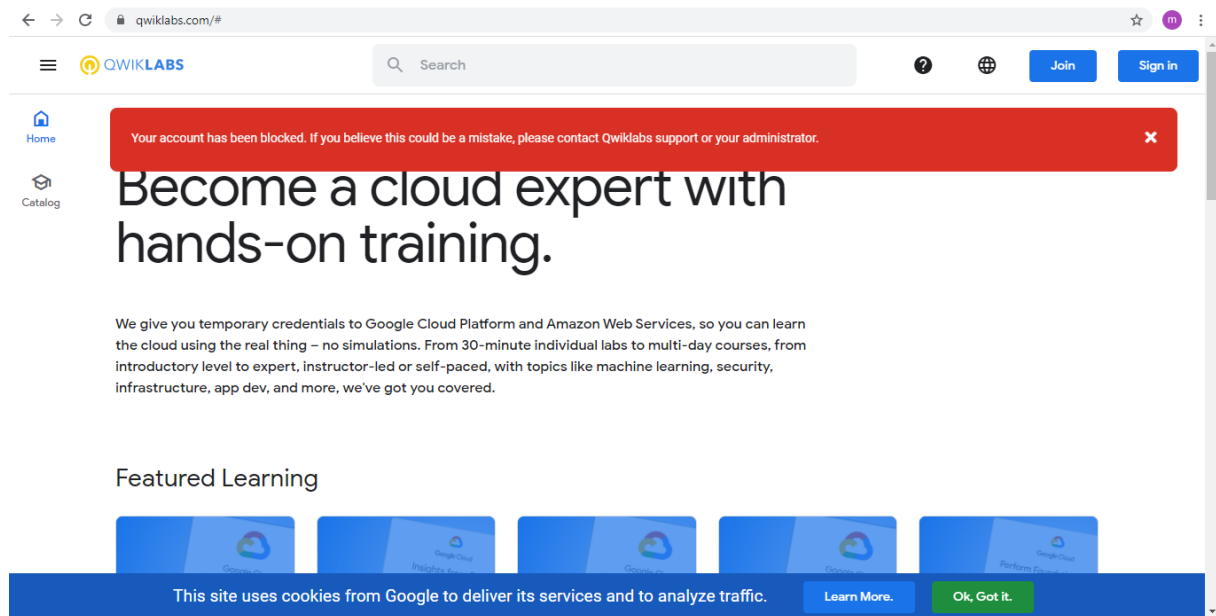
Figure 17: La publication et l'exécution de l'image

Conclusion :

La réalisation de ce projet, nous a permis de renforcer nos connaissances dans le développement web notamment dans la manipulation du Framework Java Spring boot et ses différentes fonctionnalités et de découvrir de nouvelles notions et approches de travail comme les micro-services, les messages brokers, docker et l'un des Framework JS « Angular ».

Qwicklabs :

En voulant faire une capture d'écran j'ai constaté que mon compte est bloqué.



Lien de l'application :

Vous trouverez sur git hub l'application complète « les service et le frontend ».

<https://github.com/Gharbimohamed/gestion-club-sportif>

Lien du docker :

https://hub.docker.com/r/mohamedgharbi/projetweb/tags?page=1&ordering=last_updated

Le manuel d'utilisation :

Pour le manuel d'utilisation vous trouverez les instructions à suivre dans le readme dans le git hub.