



Université de Bretagne Occidentale

Faculté des Sciences et Techniques

**Master 2 Développement Logiciel des
Systèmes d'Informations**

Framework JEE Backend Webservice

« Plateforme des Jeux STEAM »

Réalisé par :

**ALOULOU Mouad
MANCOUR BILLAH Ayoub
SAFOUANE Zineb
SALHI Anas**

Encadré par :

LEMARECHAL Yannick

Année Universitaire 2022-2023

Table des matières

I. Introduction.....	3
II. Besoin et Objectifs.....	3
1) Besoin.....	3
2) Objectifs.....	3
III. Gestion du Projet.....	4
3) Équipe.....	4
4) Outils de gestion.....	4
a. Eclipse.....	4
b. Apache Tomcat.....	4
c. JSON Viewer.....	5
a. JEE:.....	5
b. Spring.....	6
c. RESTful Web Service.....	6
d. JUnit.....	6
e. Maven.....	6
IV. Développement technique.....	7
1) Architecture.....	7
2) Interfaces.....	9
a. Service 1 ImageRevolver (Gestion des Images).....	9
b. Service 2 MenuBuilder buildMenu (Gestion des Menus).....	10
c. Service 2 MenuBuilder rechercherJeux (Gestion des Jeux).....	12
3) Les différentes fonctions.....	13
4) Base de données.....	14
V. Bilan du Projet.....	15

I. Introduction

Dans le cadre de l'année en cours en Master 2 Développement IOgiciel des Systèmes d'Informations à l'Université de Bretagne Occidentale – Faculté des Sciences et Techniques, il nous est proposé un projet permettant de mettre en pratique nos connaissances et nos compétences en développement web et précisément en JEE à travers des étapes précises suivies à chaque fois par notre professeur Mr. LEMARECHAL Yannick ayant pour finalité le développement d'une plateforme de jeux vidéos STEAM. Ayant une passion pour le développement informatique, on a saisi l'opportunité d'exploiter cet intérêt commun pour soumettre l'ébauche d'un projet personnel innovant et acquérir par la suite toutes les compétences et les qualités nécessaires.

II. Besoin et Objectifs

1) Besoin

Pour pouvoir établir ce travail, on doit absolument connaître toutes les règles basiques de l'informatique, ainsi que des frameworks tel que JEE et Spring. Dotés d'un savoir-faire ainsi qu'une motivation éblouissante, on ne ratera pas cette occasion parfaite pour faire briller notre côté talentueux.

2) Objectifs

L'objectif du projet consiste à développer une plateforme de jeux vidéos rassemblant à Steam en partie Backend. Cette plate-forme permettra de réaliser les opérations suivantes :

- Afficher les différentes images de notre base de données sous format 64Byte ainsi que leurs noms.
- Lever une exception de type SteamException en cas d'erreur.
- Afficher deux menus différents en se basant sur le choix de l'utilisateur.
- Publier une liste de jeux vidéos selon les critères de l'utilisateur.

En effet, ce projet permet à l'utilisateur de naviguer dans le site web et chercher les jeux qui l'intéressent en fonction de leurs besoins. Ainsi que la possibilité de choisir d'autres critères (par nom, prix etc...).

III. Gestion du Projet

3) Équipe

Durant ce projet, on a travaillé en équipe de quatre personnes (ALOULOU Mouad, MANCOUR BILLAH Ayoub, SAFOUANE Zineb et SALHI Anas). Ce travail nous a permis de renforcer les différentes compétences acquises tout au long de notre parcours universitaire. Échanger et discuter les idées nous a permis de mieux développer le Webservice tout en conservant son cadre général.

4) Outils de gestion

➤ Programmes de réalisation :

a. Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java. Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, reporting... Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.



b. Apache Tomcat

Apache Tomcat ou simplement Tomcat est un serveur d'applications, plus précisément un conteneur web libre de servlets et JSP. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process5, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.



c. JSON Viewer



JSON Viewer

Sélection

★★★★★ 994

Outils de développement

1000 000+ utilisateurs

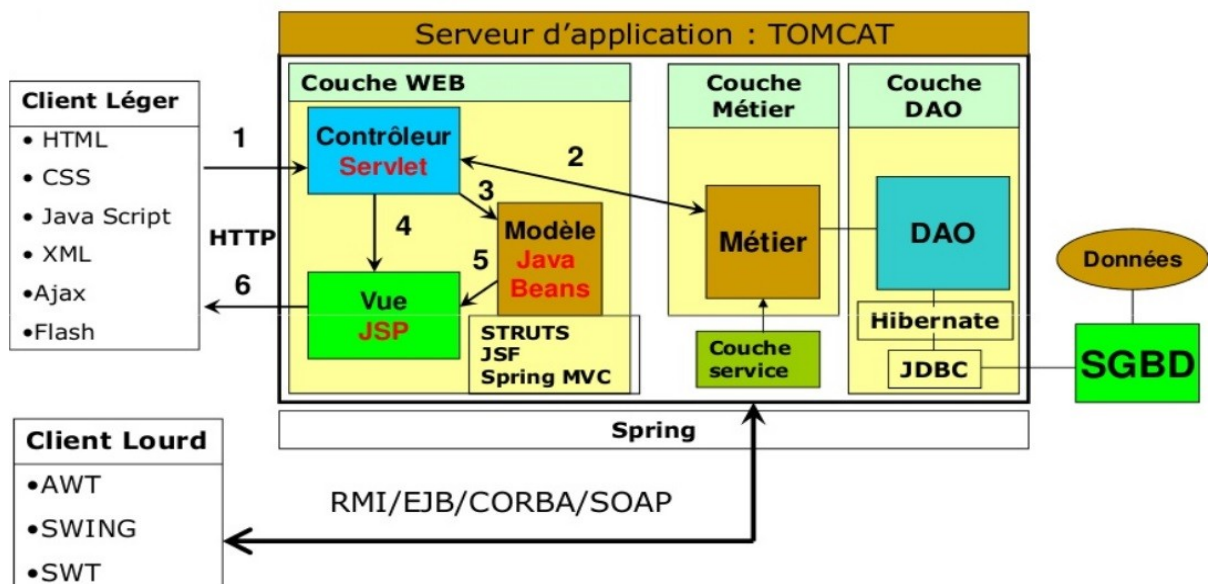
Le surligneur JSON/JSONP le plus beau et le plus personnalisable que vos yeux aient jamais vu. Il s'agit d'une extension Chrome pour l'impression JSON et JSONP.

➤ Langages de programmation et frameworks:

a. JEE:

J2EE est l'acronyme de Java 2 Entreprise Edition. Cette édition est dédiée à la réalisation d'applications pour entreprises. J2EE est basé sur J2SE (Java 2 Standard Edition) qui contient les API de base de Java. Depuis sa version 5, J2EE est renommée Java EE (Enterprise Edition).

Architecture J2EE



b. Spring

En informatique, Spring est un framework open source pour construire et définir l'infrastructure d'une application Java3, dont il facilite le développement et les tests.

Spring s'appuie principalement sur l'intégration de trois concepts clés : 1- L'inversion de contrôle est assurée de deux façons différentes : la recherche de dépendances et l'injection de dépendances. 2- La programmation orientée aspect. 3- Une couche d'abstraction.



c. RESTful Web Service

RESTful Web Service est un Web Service qui utilise la structure REST. REST a été largement utilisé, remplaçant les Web Services basés sur SOAP et WSDL. Les RESTful Web Services sont légers (lightweigh), faciles à étendre et à entretenir.



d. JUnit

JUnit est un framework de test unitaire pour le langage de programmation Java, créé par Kent Beck et Erich Gamma.

JUnit définit deux types de fichiers de tests. Les TestCase (cas de test) sont des classes contenant un certain nombre de méthodes de tests. Un TestCase sert généralement à tester le bon fonctionnement d'une classe. Une TestSuite permet d'exécuter un certain nombre de TestCase déjà définis.



e. Maven

Apache Maven (couramment appelé Maven) est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel. Maven est géré par l'organisation Apache Software Foundation. L'outil était précédemment une branche de l'organisation Jakarta Project.



IV. Développement technique

Pour ce qui est technique, on a commencé par structurer le projet en créant une application web dynamique sous eclipse. Ensuite on a créé nos différents packages pour chaque couche (DAO, Business, Bean et Service). Après on s'est mis vers la création de chaque classe et son interface (Image, Menu et Jeu) avec une classe de type Bean Jeu pour pouvoir l'instancier.

On a choisi comme nom de notre plateforme «STEAM», ce choix reflète des choix techniques fournies au cahier des charges.

1) Architecture

Pour l'architecture du projet, on l'a divisé en trois 4 packages : Bean, Business, DAO et Service. Chaque package et partie sont responsables d'une action.

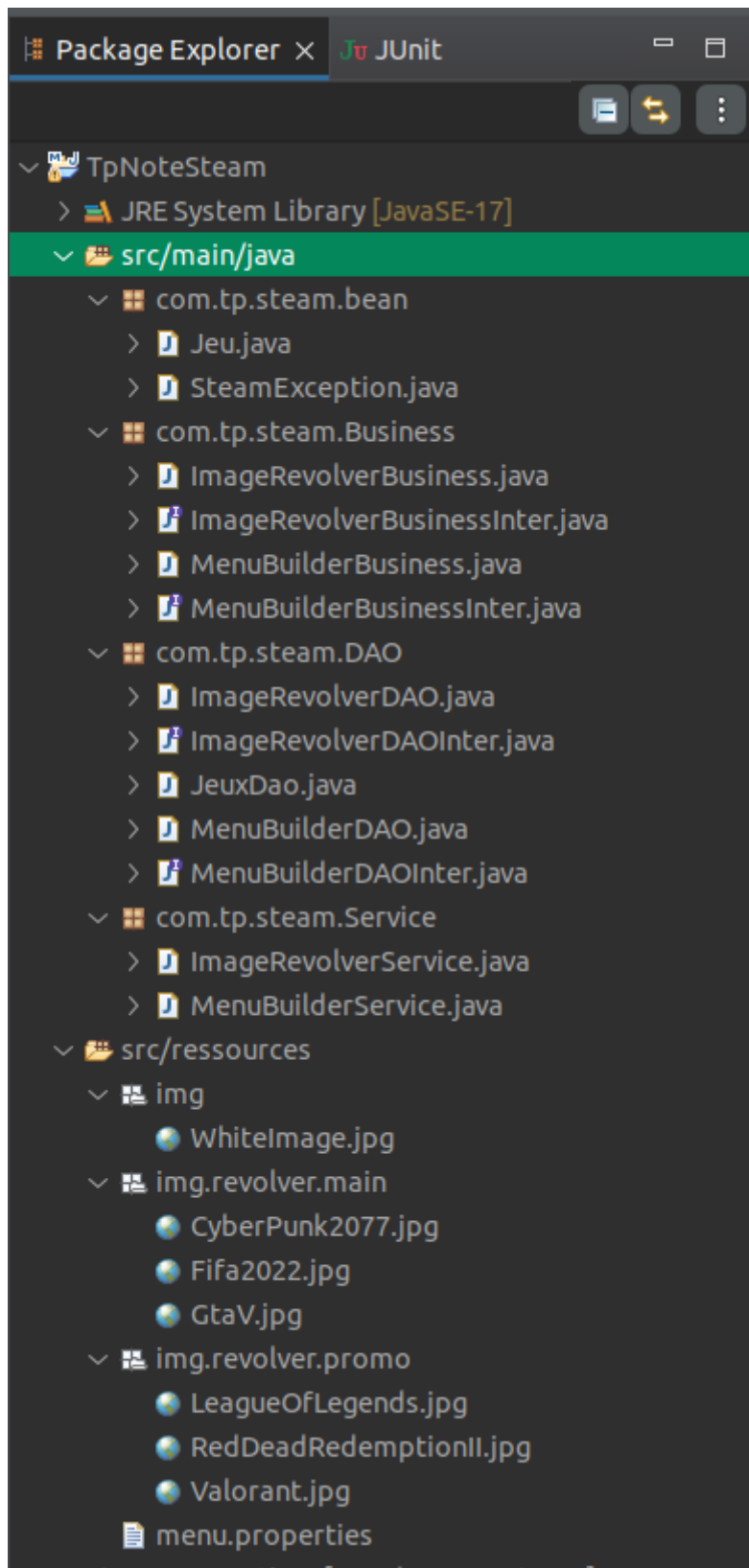
Bean consiste à instancier la classe JEU et définir ses getters et setters ainsi que son constructeur.

Business gère l'implémentation de la couche Service.

DAO manipule tout ce qui est base de données (Fichiers dans notre cas).

Et enfin Service consiste à démarrer le serveur et exécuter le programme côté serveur.

Vous trouverez ci-après une capture d'écran de l'architecture de notre application :



2) Interfaces

a. Service 1 ImageRevolver (Gestion des Images)

Ce service permet de choisir les images depuis notre dossier ressources (BDD) selon un type de paramètre donné par l'utilisateur.

1- Si l'utilisateur choisit le paramètre MAIN :

[illegible]

2- Si l'utilisateur choisit le paramètre PROMO:

[illegible]

3- Si l'utilisateur choisit la paramètre OTHER, on lève une exception de type SteamException avec un message 'Actuellement non supporté':

← → ↺ ⓘ localhost:8081/TpNoteSteam/steam/imageRevolver/OTHER

État HTTP 500 – Erreur interne du serveur

Type Rapport d'exception

message Request processing failed; nested exception is com.tp.steam.bean.SteamException: Actuellement Non supporter

description Le serveur a rencontré une erreur interne qui l'a empêché de satisfaire la requête.

exception

```
org.springframework.web.util.NestedServletException: Request processing failed; nested exception is com.tp.steam.bean.SteamException: Actuellement Non supporter
org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:982)
org.springframework.web.servlet.FrameworkServlet.doGet(FrameworkServlet.java:861)
javax.servlet.http.HttpServlet.service(HttpServlet.java:670)
org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:846)
javax.servlet.http.HttpServlet.service(HttpServlet.java:779)
org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
```

cause mère

```
com.tp.steam.bean.SteamException: Actuellement Non supporter
com.tp.steam.Business.ImageRevolverBusiness.loadImages(ImageRevolverBusiness.java:41)
com.tp.steam.Service.ImageRevolverService.loadImages(ImageRevolverService.java:24)
java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
java.base/java.lang.reflect.Method.invoke(Method.java:568)
org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:220)
org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:134)
org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:116)
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:827)
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:738)
org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:85)
org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:963)
org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:897)
org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:970)
org.springframework.web.servlet.FrameworkServlet.doGet(FrameworkServlet.java:861)
javax.servlet.http.HttpServlet.service(HttpServlet.java:670)
org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:846)
javax.servlet.http.HttpServlet.service(HttpServlet.java:779)
org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
```

note La trace complète de la cause mère de cette erreur est disponible dans les fichiers journaux de ce serveur.

Apache Tomcat/9.0.68

b. Service 2 MenuBuilder buildMenu (Gestion des Menus)

Ce service permet à l'utilisateur de choisir quel menu afficher selon un paramètre donné en entrée (MAIN, TAGS ou GENRE).

1- Si l'utilisateur choisit le paramètre MAIN :

← → ↺ ⓘ localhost:8081/TpNoteSteam/steam/MenuBuilder/MAIN

```
{
  "Feature Items ": [],
  "Games": [
    "Free to play",
    "Action",
    "Adventure"
  ],
  "Software": [
    "Utilities",
    "Design"
  ],
  "Demos": [],
  "News": []
}
```

2- Si l'utilisateur choisit le paramètre TAGS:

```
← → ↻ ⓘ localhost:8081/TpNoteSteam/steam/MenuBuilder/TAGS

▼ {
  "Great Soundtrack": [],
  "Platformer": [],
  "Single Player": [],
  "Sci-fi": [],
  "Difficult": []
}
```

3- Si l'utilisateur choisit le paramètre GENRE:

```
← → ↻ ⓘ localhost:8081/TpNoteSteam/steam/MenuBuilder/GENRE

▼ {
  "Free to play": [],
  "Early Access": [],
  "Action": [],
  "Adventure": [],
  "Casual": [],
  "Indie": [],
  "MMO": [],
  "Racing": [],
  "RPG": [],
  "Simulation": [],
  "Sports": [],
  "Strategy": []
}
```

c. Service 2 MenuBuilder rechercherJeux (Gestion des Jeux)

Ce service nous permet de rechercher des jeux selon un critère donné par l'utilisateur, il peut s'agir de la date, le prix, le nom, l'OS etc.

1- Si l'utilisateur choisit de chercher les jeux par système d'exploitation LINUX:

```
localhost:8081/TpNoteSteam/steam//rechercherJeux/Linux

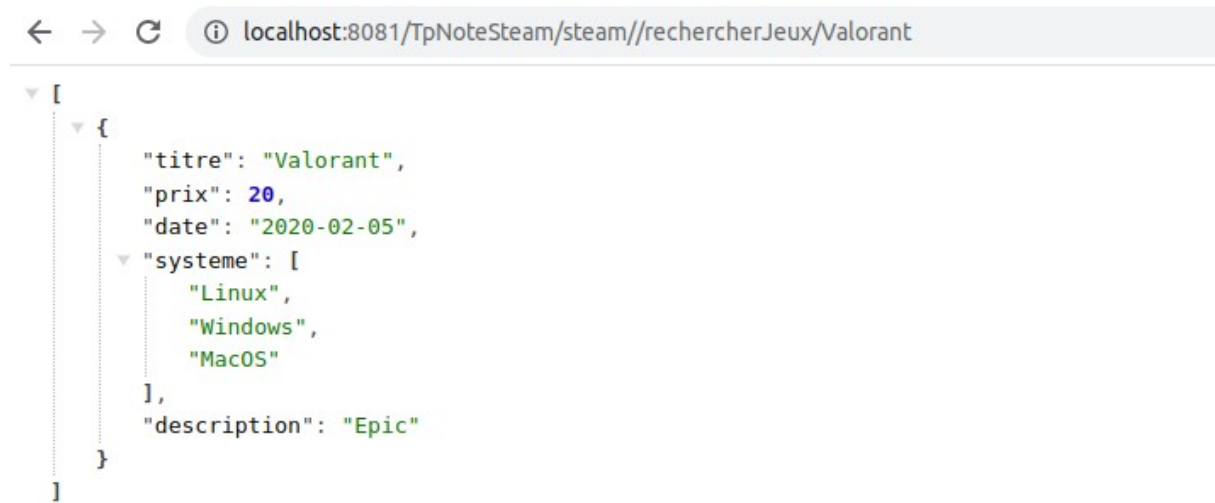
[
  {
    "titre": "Far Cry",
    "prix": 20,
    "date": "2019-08-21",
    "systeme": [
      "Linux",
      "Windows",
      "MacOS"
    ],
    "description": "Gold Version"
  },
  {
    "titre": "Battlefield",
    "prix": 40,
    "date": "2020-04-11",
    "systeme": [
      "Linux",
      "Windows",
      "MacOS"
    ],
    "description": "Premium"
  },
  {
    "titre": "Fifa",
    "prix": 30,
    "date": "2018-08-21",
    "systeme": [
      "Linux",
      "Windows",
      "MacOS"
    ],
    "description": "Promotion"
  },
  {
    "titre": "Valorant",
    "prix": 20,
    "date": "2020-02-05",
    "systeme": [
      "Linux",
      "Windows",
      "MacOS"
    ],
    "description": "Epic"
  }
]
```

2- Si l'utilisateur choisit de chercher les jeux par prix (20.0\$) :

```
localhost:8081/TpNoteSteam/steam//rechercherJeux/20

[
  {
    "titre": "Far Cry",
    "prix": 20,
    "date": "2019-08-21",
    "systeme": [
      "Linux",
      "Windows",
      "MacOS"
    ],
    "description": "Gold Version"
  },
  {
    "titre": "Valorant",
    "prix": 20,
    "date": "2020-02-05",
    "systeme": [
      "Linux",
      "Windows",
      "MacOS"
    ],
    "description": "Epic"
  }
]
```

3- Si l'utilisateur choisit de chercher les jeux par nom (Valorant):



3) Les différentes fonctions

Fonction loadImages

```
/**
 * <b>Fonction loadImages</b>
 *
 * @param TypeRevolver
 *   Un string passé en paramètre par l'utilisateur en fonction de son besoin de l'image
 * @return Une liste Map<String,String>
 * @throws SteamException
 *   Element non fourni actuellement
 */
@Override
public Map<String, String> loadImages(String TypeRevolver) throws SteamException {
    Map<String, byte[]> images=new HashMap<String,byte[]>();
    Map<String,String> images64=new HashMap<String,String>();
    if(TypeRevolver.equals("MAIN")) {
        images=dao.loadImagesMain();
        for (Map.Entry<String,byte[]> entry : images.entrySet())
        {
            images64.put(entry.getKey(), getImageData(entry.getValue()));
        }

    }else if (TypeRevolver.equals("PROMO")) {
        images=dao.loadImagesPromo();
        for (Map.Entry<String,byte[]> entry : images.entrySet())
        {
            images64.put(entry.getKey(), getImageData(entry.getValue()));
        }

    }else if (TypeRevolver.equals("OTHER")){
        throw new SteamException("Actuellement Non supporter");
    }

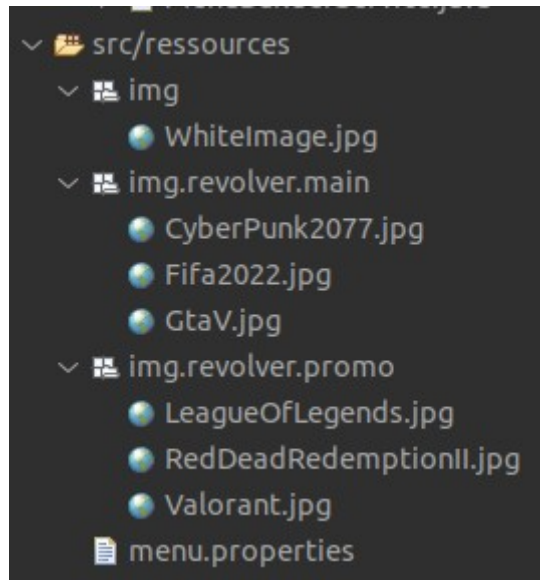
    return images64;
}
```


4) Base de données

Dans notre cas on a travaillé avec des fichiers locaux tout en respectant les consignes du cahier des charges où on nous a demandé de ne pas créer une base de données.

Ci-après l'architecture de notre dossier ressources ainsi que la fonction de chargement des images.

Architecture :



Fonction de chargement des images :

```
/**
 * <b>Fonction loadImagesMain</b>
 *
 * @return Une liste HashMap<String, byte[]> qui permet de stocker le nom d'image et sa conversion en byte 64 du dossier MAIN
 * @throws IOException
 * Fichier qui n'est pas convertit en 64bytes
 */
@Override
public Map<String, byte[]> loadImagesMain() {

    Map<String, byte[]> mainImages= new HashMap<String, byte[]>();
    File dir = new File(classLoader.getResource(mainPath).getFile());
    File[] images = dir.listFiles();
    if (images.length > 0) {
        for (File image : images) {
            try {
                byte[] bytes = Files.readAllBytes(image.toPath());
                mainImages.put(image.getName(), bytes);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    else
    {
        File image = new File(classLoader.getResource(whiteImage).getFile());
        try {
            mainImages.put(image.getName(), Files.readAllBytes(image.toPath()));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    return mainImages;
}
```

V. Bilan du Projet

En travaillant en équipe et en s'entraïdant on a pu réaliser ce travail en groupe en partageant les tâches pour mettre en pratique les compétences acquises pendant cette UE. Nous avons pu concevoir des services web en respectant les normes du sujet donné.

Nous avons pu acquérir, lors de notre projet, certaines techniques qui nous ont servi d'approfondir nos connaissances dans le développement informatique.

Comme perspective, une amélioration de notre application reste toutefois envisageable telle qu'une partie front, une partie test et la dynamisation des jeux à l'aide des APIs.

Pour conclure, ce fût une expérience enrichissante sur tous les points qu'elles soient théoriques, pratiques ou même relationnels. Ce projet nous a été très profitable, et nous espérons, à travers ce modeste travail, avoir été à la hauteur de vos attentes cher Professeur.