

Rapport projet SIR

Université de Rennes 1
ISTIC - Master 1 MIAGE

2022-2023

Binôme
Eljeddi Yosser 1A
Cyriaque Toussou 1A

Enseignant
Adrien Le Roch
Olivier Barais

Sommaire

I. Présentation et contexte du projet

1. Analyse conceptuelle et fonctionnelle

1.1 Diagramme de cas d'utilisation

1.2 Diagramme de classe

2. Choix de technologies et langages utilisés

3. Présentation de l'architecture du Back End

4. Présentation de l'architecture du Front End

II. Conclusion

I. Présentation et contexte du projet

Le présent projet consiste en la mise en place d'un gestionnaire de tickets dans le cadre de validation des acquis durant le cours de SIR notamment :

- L'implémentation du mapping relationnel en JPA;
- L'implémentation des DAO, DTO;
- L'implémentation d'API rest (JaxRs);
- L'implémentation d'une application Single page pour interagir avec l'API.

Dans ce rapport, nous présenterons l'analyse fonctionnelle, et technique ainsi que les technologies utilisées.

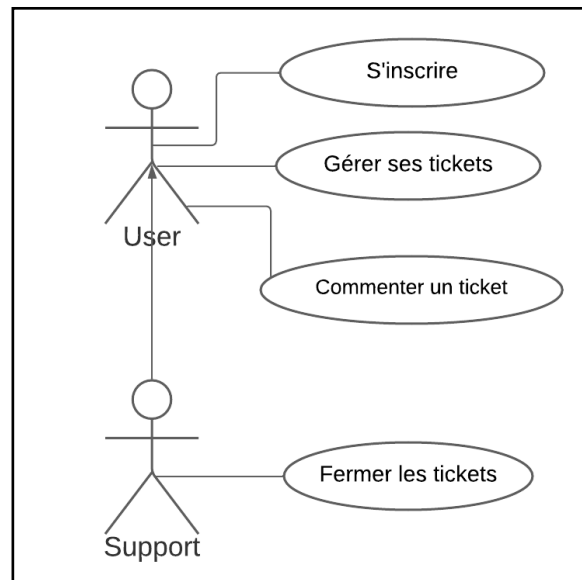
Afin de pouvoir atteindre cet objectif et répondre aux contraintes imposées, le projet a été scindé en plusieurs parties qu'on va essayer de vous présenter à travers ce rapport et vous lister nos choix d'outils pour la réalisation du projet et enfin quelques liens pour le test de notre API et de la base de données.

1. Analyse conceptuelle et fonctionnelle

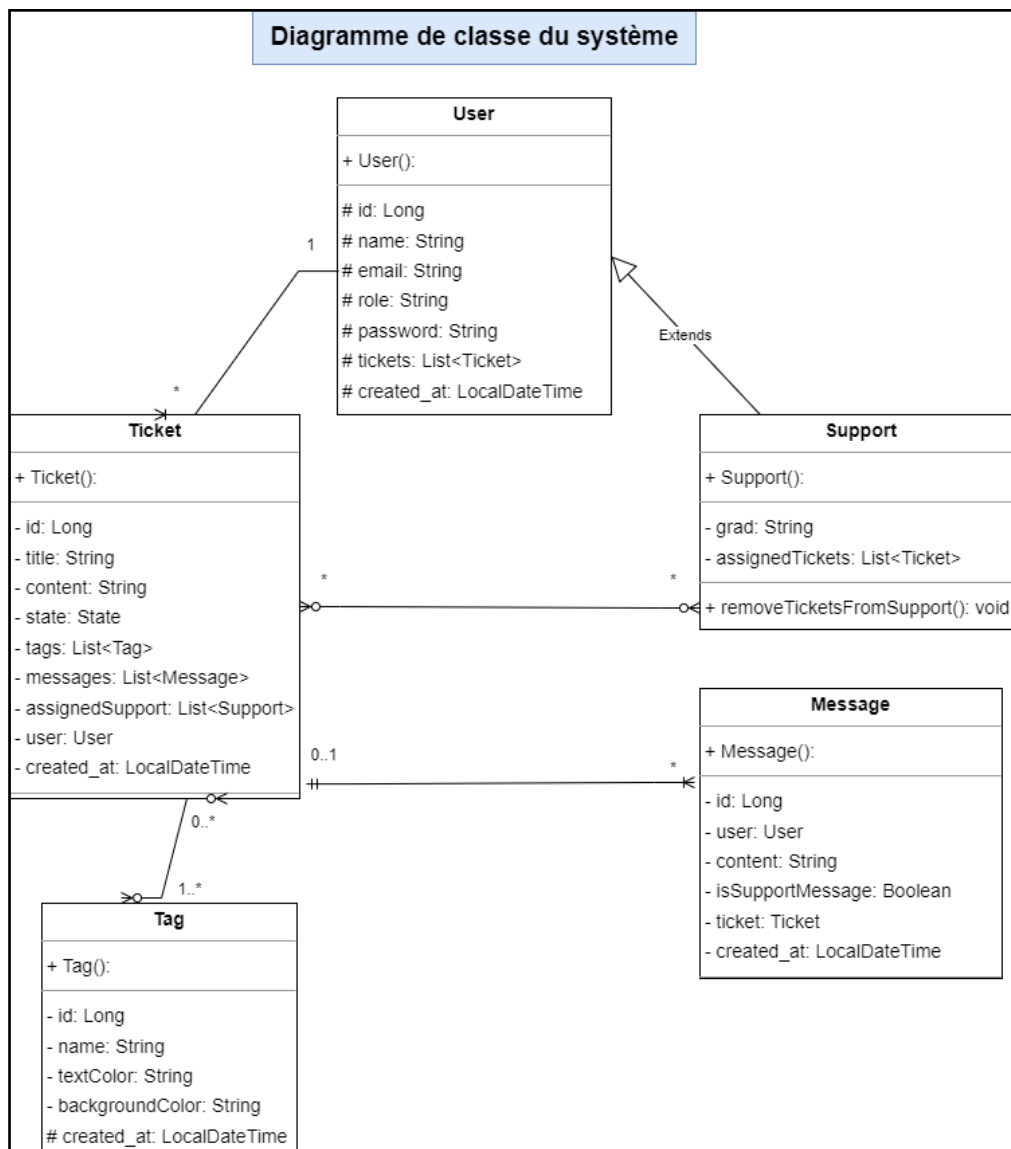
Explication des différentes fonctionnalités :

Acteur	Fonctionnalité
User et Support	S'inscrire/Se connecter Créer/Modifier/Supprimer ses tickets Commenter un ticket
Support	-Fermer n'importe quels tickets

1.1 Diagramme de cas d'utilisation (DCU)



1.2 Diagramme de classe



2. Choix de technologies et langages utilisés

Pour le Back End :

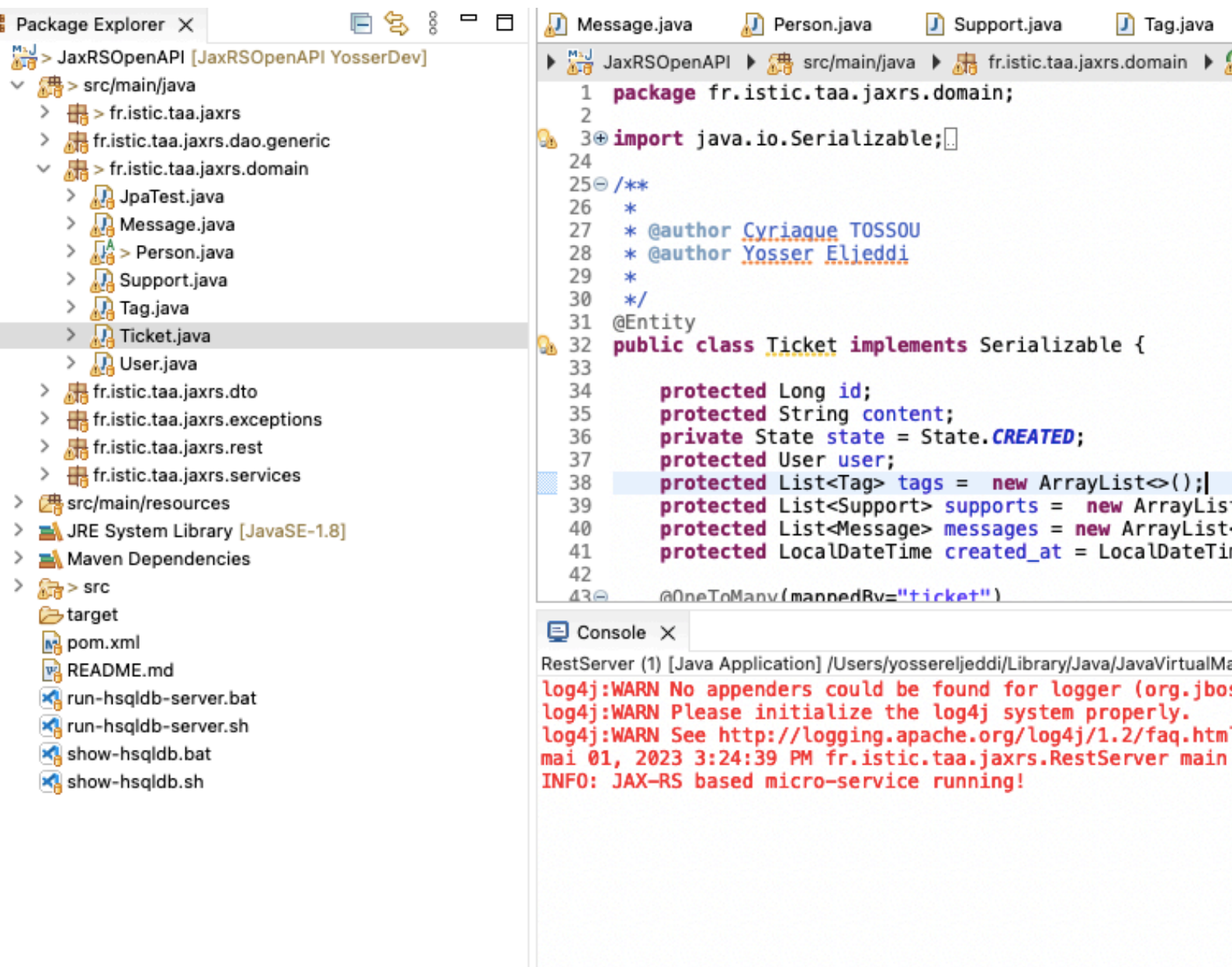
Choix technologique	Commentaires
Langage : Java 8	
Mapping relationnel : JPA	
API Rest avec JaxRs	
Hibernate , JPQL	Pour la gestion des persistances Création des requêtes personnalisées en JPQL
Maven	Pour la gestion des dépendances
MySql	Pour la gestion de la base de données
OpenAPI	Pour la description et la documentation de l'API
Jakarta Validation	Les données soumises à l'API sont validées avec Jakarta validation afin de retourner des messages d'erreurs personnalisés (en cas d'erreur).
Provider	Pour gérer les problèmes de CORS en ajoutant des entêtes HTTP

Pour le Front End :

Choix technologique	Commentaires
Langage : TypeScript	
Framework : Angular	Nous avons utilisé le framework Angular avec l'architecture MVVN
Design Bootstrap, SCSS, CASS , FLEX	Pour la gestion des styles
Librairie : PrimeNG	Pour utiliser les données NG pour certaines interfaces

3. Présentation de l'architecture du Back End

Le projet backend (API REST) est développé suivant une architecture multi couches reparties dans des packages différents. Un aperçu des packages disponibles est le suivant :



fr.istic.taa.jaxrs.dao.generic : Ce package contient les DAO (Data Accès Object) permettant de communiquer avec la base de données (faire les requêtes).

fr.istic.taa.jaxrs.domain : Ce package contient le model metier avec le mapping relationnel implémenté grâce àJPA.

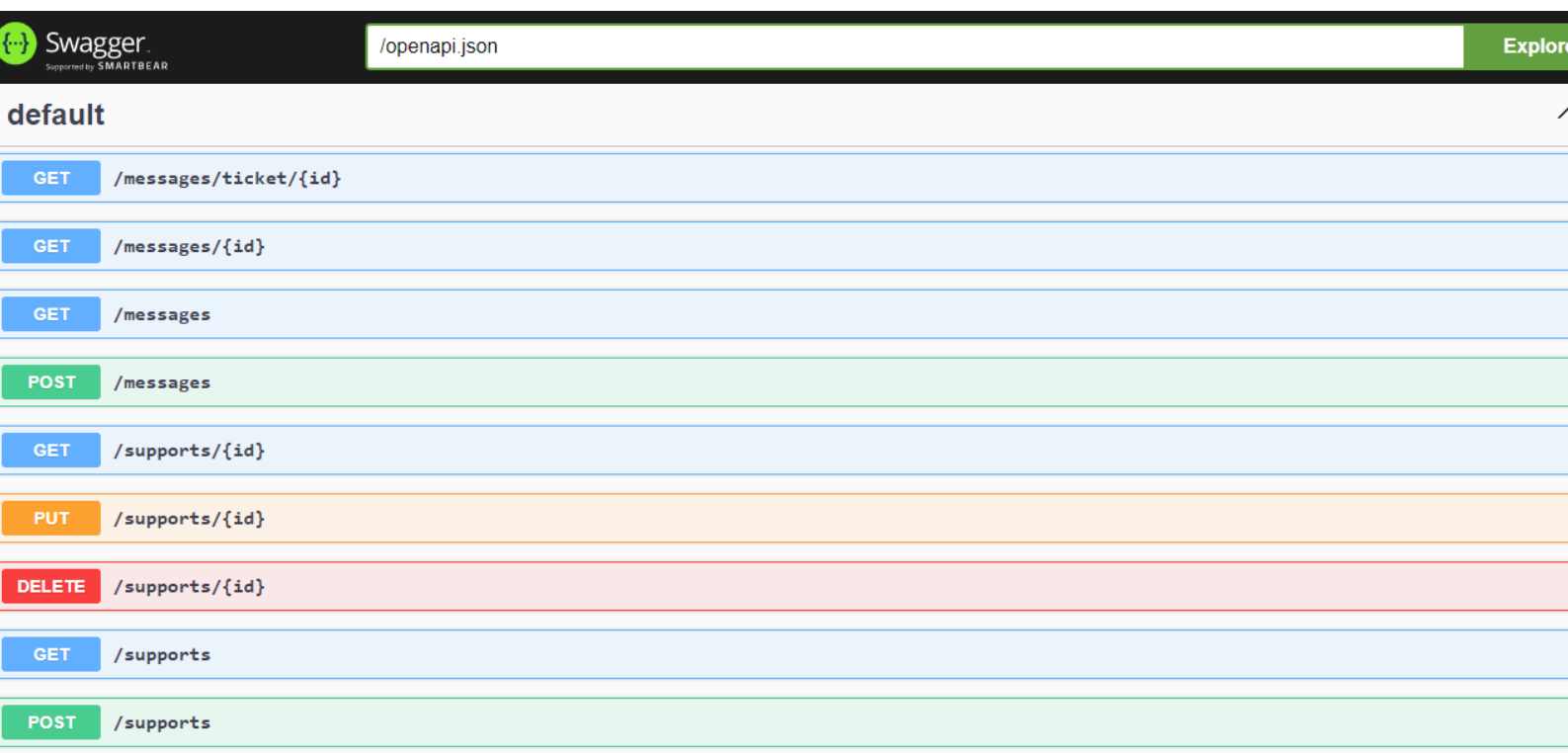
fr.istic.taa.jaxrs.dto : Ce package contient les classes DTO (Data Transfert Object) ayant permis le reformatages des données envoyées par l'API.

fr.istic.taa.jaxrs.rest : Ce package contient les différentes ressources et leur endPoint fournis par l'API.

fr.istic.taa.jaxrs.services : Ce package contient les différents services de traitement intermédiaire entre les ressources et notre model metier. Nous y avons implémenté notamment le service de validation des données et de personnalisation des messages d'erreur (DefaultValidator.java).

fr.istic.taa.jaxrs.filter : Ce package contient le filter de gestion des problèmes de CORS.

Aperçue des End Point disponible :



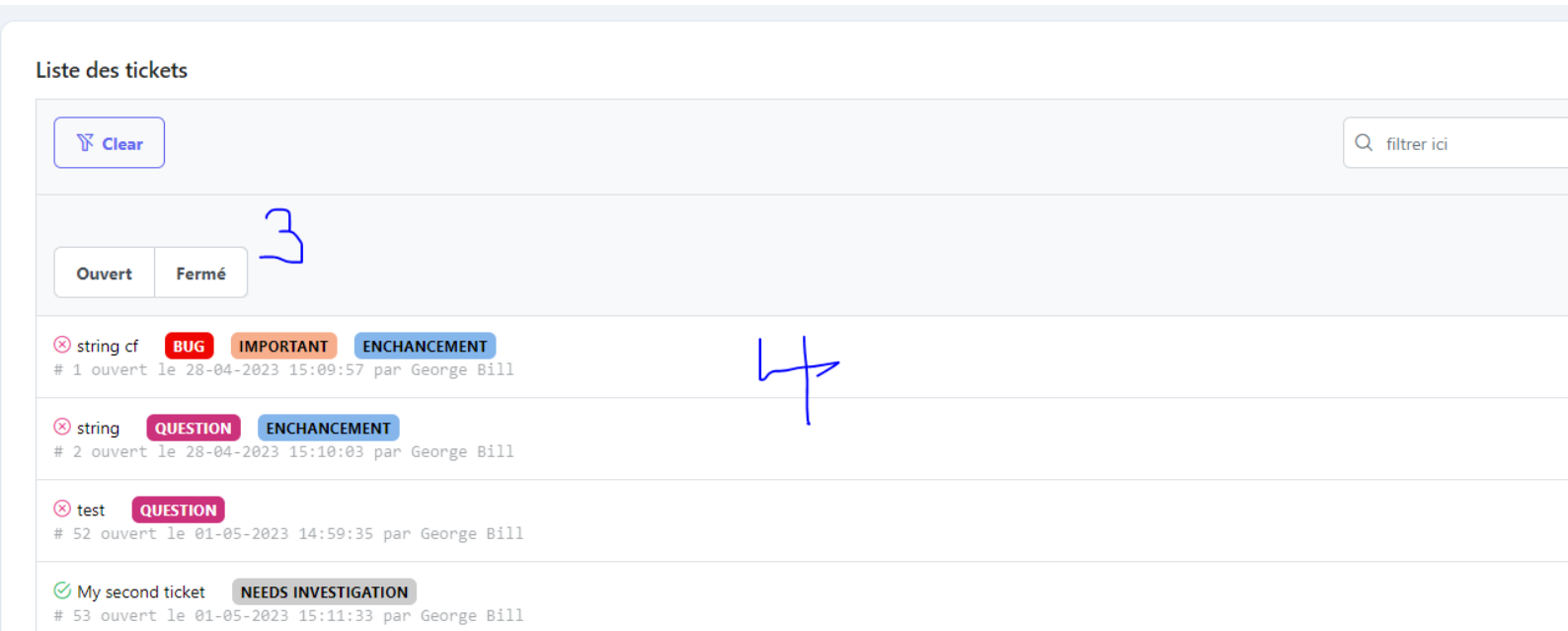
The image shows a Swagger UI interface for an API. At the top, the Swagger logo is visible on the left, and a search bar contains the text "/openapi.json". To the right of the search bar is a green button labeled "Explore". Below the search bar, the word "default" is displayed. The main area of the interface lists several API endpoints, each with a colored button indicating the HTTP method (GET, POST, PUT, DELETE) and the endpoint path. The endpoints are: /messages/ticket/{id} (GET), /messages/{id} (GET), /messages (GET), /messages (POST), /supports/{id} (GET), /supports/{id} (PUT), /supports/{id} (DELETE), /supports (GET), and /supports (POST).

Method	Endpoint
GET	/messages/ticket/{id}
GET	/messages/{id}
GET	/messages
POST	/messages
GET	/supports/{id}
PUT	/supports/{id}
DELETE	/supports/{id}
GET	/supports
POST	/supports

GET	/api	▼
GET	/api/{path}	▼
GET	/tags	▼
POST	/tags	▼
GET	/tags/{id}	▼
PUT	/tags/{id}	▼
DELETE	/tags/{id}	▼
GET	/tickets	▼
POST	/tickets	▼
GET	/tickets/{id}	▼
PUT	/tickets/{id}	▼
PUT	/tickets/toggleState/{id}	
PUT	/tickets/addTags/{id}	
PUT	/tickets/assign_support/{id}	
GET	/users/{id}	
PUT	/users/{id}	
DELETE	/users/{id}	
GET	/users	
POST	/users	
POST	/users/login	

4. Présentation de l'architecture du Front End

Page Liste des tickets :



C'est la page d'accueil qui s'affiche dès qu'on démarre l'application, elle est accessible par tout le monde.

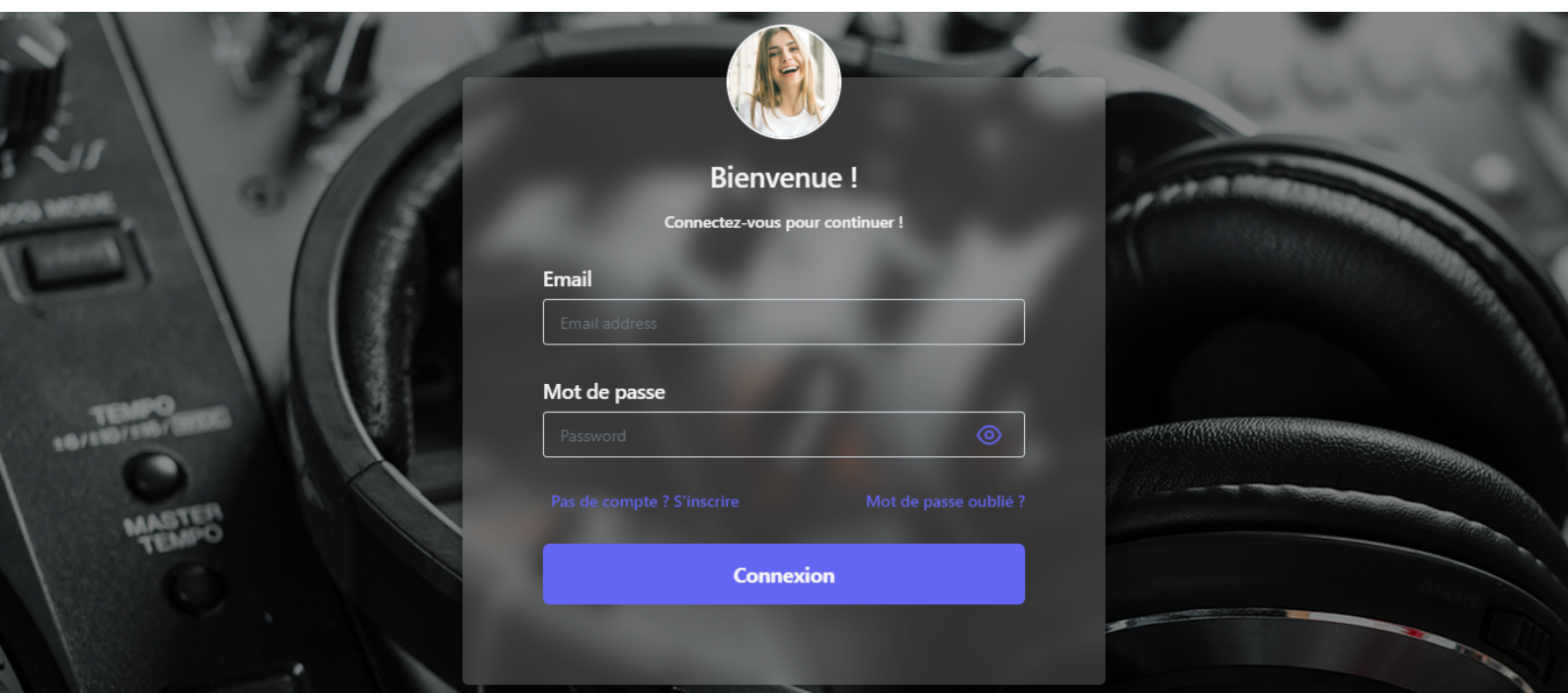
1 : Cliquer sur ce bouton pou se connecter

2: Bouton d'accès à la liste de tous les tickets.

3: Bouton pour filtrer les tickets par leurs status (Fermé/Ouvert).

4: Cliquer sur la ligne du tickets pour voir les détails du tickets et ses commentaires.

Page de connexion :



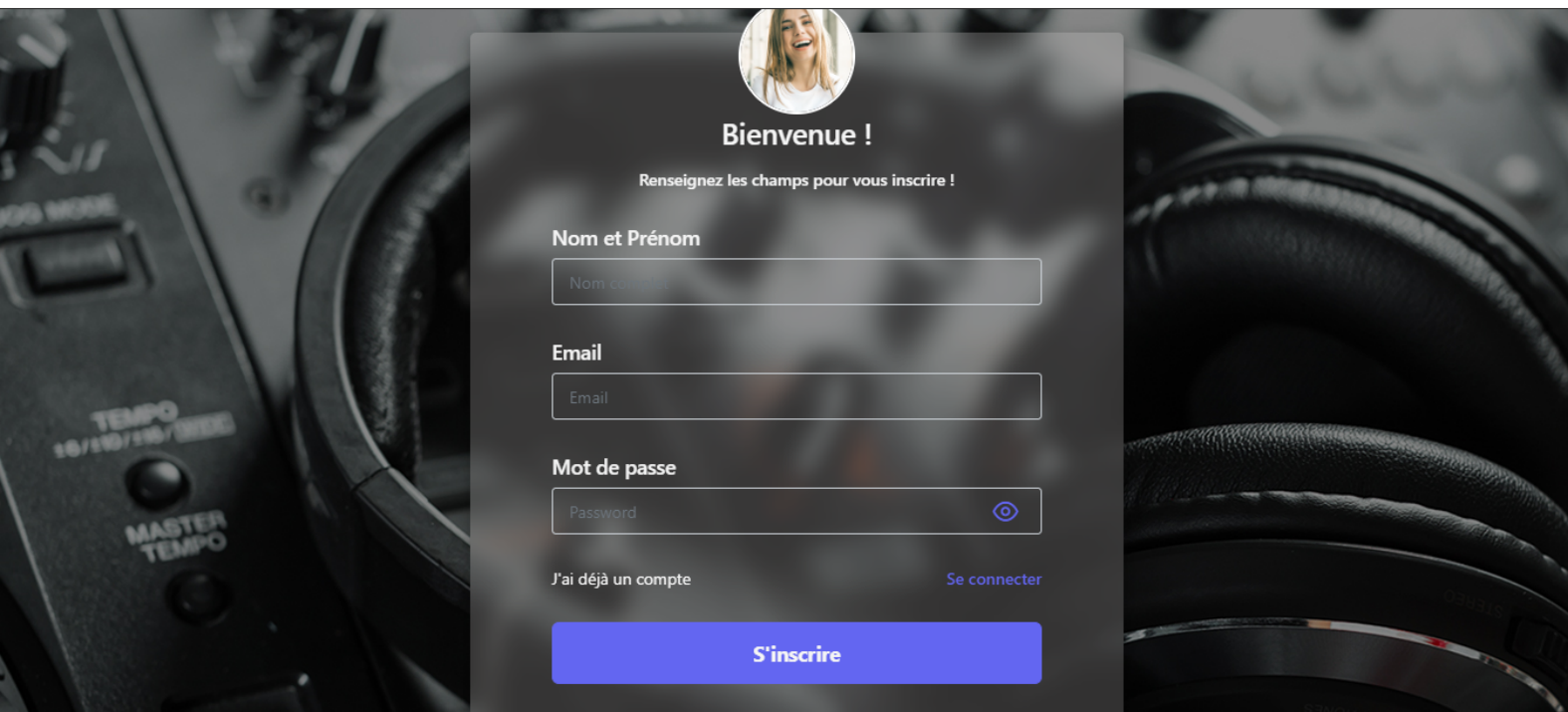
C'est la page qui nous permet de se connecter en tant que User simple ou Support.

NB: Le système est livré avec deux comptes d'accès par défaut :

-User (Login : user@gmail.com, Mot de passe : User@2023)

-Support (Login : support@gmail.com, Mot de passe : Support@2023)

Page d'inscription :




Bienvenue !

Renseignez les champs pour vous inscrire !

Nom et Prénom

Email

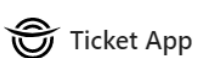
Mot de passe





 

[J'ai déjà un compte](#) [Se connecter](#)


S'inscrire

Page de session en tant que User :





 Ticket  Utilisateur  Bienvenue, Yosser ELJEDDI(User!) 


Liste des tickets


 Clear


filtrer ici


Ouvert Fermé

 Nouveau ticket 

 string cf **BUG** **IMPORTANT** **ENCHANCEMENT**
1 ouvert le 28-04-2023 15:09:57 par George Bill

 string **QUESTION** **ENCHANCEMENT**
2 ouvert le 28-04-2023 15:10:03 par George Bill

 test **QUESTION**
52 ouvert le 01-05-2023 14:59:35 par George Bill

 My second ticket **NEEDS INVESTIGATION**
53 ouvert le 01-05-2023 15:11:33 par George Bill

- 1: Le nom et le rôle de l'utilisateur connecté
- 2: Bouton d'accès à la création d'un nouveau ticket (On ne peut créer de ticket que lorsqu'on est connecté).

Page de création et modification d'un ticket :

Editer ticket

Titre du ticket

string cf

Description

string updateds zrhg

Ajouter des tags

bug important enchancement

Quitter

Valider

Dans cette page, vous pouvez Créer/Modifier un ticket en renseignant son titre, sa description ou ses tags.

Page détails d'un ticket :

Yosser Test

Ouvert Yosser ELJEDDI a ouvert ce ticket le 03-05-2023 14:32:48 · 4 commentaire(s)

'Yosser ELJEDDI a commenté le ...'

Description du ticket

4 commentaires ajouté(s) à ce ticket !

George Bill a commenté le 03-05-2023 14:37:12

Vous devez supprimer ce ticket

George Bill a commenté le 03-05-2023 14:39:56

Tags ajoutés

QUESTION IMPORTANT

- 1 : Ici s'affichent les tags assignés à un ticket.
- 2: Les détails du tickets (Le status, la date de création du ticket, l'utilisateur ayant créé, le nombre de commentaires)
- 3: La liste des commentaires

Page commentaire :

Cyriaque Tossou a commenté le 03-05-2023 14:43:25 •

Virez moi ce utilisateurs

Ajouter votre commentaire ici...

Your Message

Retour

Commenter

C'est la page dans laquelle vous pouvez saisir votre commentaire.

Page gestion d'un ticket :

string cf

Fermé

George Bill a ouvert ce ticket le 28-04-2023 15:09:57 • 4 commentaire(s)

1

2

3

Reouvrir

Editer

+ ticket

'George Bill a commenté le ... •'

string updateds zrhg

4 commentaires ajouté(s) à ce ticket !

George Bill a commenté le 29-04-2023 01:59:01 •

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece

Tags ajoutés


BUG

IMPORTANT

ENCHANCEMENT

- 1: Pour Ouvrir/Fermer un ticket
- 2: Pour modifier un ticket
- 3: Pour ajouter un nouveau ticket

Page Liste des utilisateurs :

 Ticket App









TicketUtilisateurBienvenue, Name(Support)!

→

Liste des utilisateurs

Clear

filter ici

Nom et Prénom	Email	Rôle	Date de création	Actions
George Bill	user1@gmail.com	USER	28-04-2023 15:06:57	 
Name	test@gmail.com	SUPPORT	28-04-2023 15:07:26	 
Abdel Orekan	abdel@gmail.com	USER	02-05-2023 03:20:07	 
Test 2	test2@gmail.com	USER	02-05-2023 03:45:47	 

Dans cette page affiche la liste de tous les utilisateurs du système. Elle est accessible par le User et le Support par contre seul le Support peut modifier et supprimer un utilisateur.

II. Conclusion

L'objectif de ce projet était d'utiliser les technologies étudiées en cours de SIR afin de développer une API REST en Java qui serait consommée par un frontend (Angular, VueJs ou React).

Nous vous avons apporté à travers ce rapport en tandem avec les codes sources présents sur le repo **GitHub** de notre projet la solution implémentée.

La réalisation de ce projet nous a permis de mettre en pratique des les éléments clés appris en cours de SIR notamment :

- le mapping avec JPA
- la mise en place d'une API REST
- la résolution des problèmes de CORS
- le fonctionnement des frameworks (Angular, VueJs, React)
- et bien d'autres choses.

Un grand merci à : **M. Olivier Barais et M. Adrien le Roch** pour les précieux savoirs transmis et l'accompagnement tout au long de ce projet.

Liens repo :

<https://github.com/cyrDominicytos/IssueManagerFront.git>

<https://github.com/cyrDominicytos/JaxRSOpenAPI.git>