



Projet JEE

Création d'une application de Chat online

Réalisé par :

DERKAOUI Ahmed-Amine

EL HASSANI Youness

Sous l'encadrement de :

M. MAHMOUD EL HAMLAOUI

Année universitaire : 2019-2020

Remerciements

Nous tenons tout d'abord à remercier Dieu, qui nous a donné la force, la perspicacité et la patience pour accomplir ce travail.

Nous tenons sincèrement à remercier notre professeur et encadrant :

M. MAHMOUD EL HAMLAOUI

pour le temps qu'il nous a consacré, ses conseils, ses encouragements, sa disponibilité et ses précieux conseils durant le cours de « Ingénierie du Web ».

Nos vifs remerciements vont également à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Membres du groupe

Table des figures

2.1	Diagramme de cas d'utilisation	10
2.2	Diagramme de classes	11
2.3	Diagramme de sequence : Authentification	12
2.4	Diagramme de sequence : Authentification room	12
2.5	Diagramme de sequence : Envoyer message	13
2.6	Diagramme d'activité	14
3.1	Maquette : Login	16
3.2	Maquette : Sing-in	16
3.3	Maquette : Accueil	17
3.4	Maquette : Interface Room	17
3.5	Logo : JEE	18
3.6	Logo : Eclipse	18
3.7	Logo : HTML/CSS/JS/Bootstrap	19
3.8	Logo : phpMyAdmin	19
3.9	Visual Studio Code	20
3.10	Docker	20
3.11	Page : Accueil	21
3.12	Page : Accueil	22
3.13	Page : Sign-in	22
3.14	Page : Login	23
3.15	Rooms	24
3.16	Page : Login Room	24

3.17 Page : Mes Rooms	25
3.18 Page : Modifier Room	25
3.19 Page : Chat	26
3.20 Page : Envoi des fichiers	26
3.21 Construire conteneur	27
3.22 Déploiement du conteneur	27
3.23	28

Table des matières

Introduction générale	6
1 Description et analyse du projet	7
1.1 Description du sujet	8
1.2 Les règles de gestion	8
1.2.1 Gestion des utilisateurs	8
1.2.2 Gestion des rooms	8
2 Conception du projet	9
2.1 Diagramme de cas d'utilisation	10
2.2 Diagramme de classes	11
2.3 Diagramme de sequence	12
2.4 Diagramme d'activité	13
3 Réalisation du projet	15
3.1 Maquettes :	16
3.2 Outils informatiques	18
3.3 Interfaces	21
3.4 Déploiement docker :	27
3.5 Test unitaire :	28
Conclusion	28
Bibliographie	30

Introduction générale

Internet a connu la montée en puissance des réseaux sociaux, devenus pour certains de véritables médias sociaux, qui permettent aux internautes et aux professionnels de créer une page profil et de partager des informations, photos et vidéos avec leur réseau. Des espaces de partage qui se distinguent par leurs utilités multiples.

Dans ce sens, nous avons choisi de travailler sur une application de chat dédiée à l'usage professionnel qui présente un espace d'échange facile à utiliser et surtout adapté au contenu pouvant être partagé dans ce contexte :

- Le premier chapitre « Description et analyse du projet » donne une vue globale sur le projet. La première partie présente une description du sujet, la deuxième partie décrit les règles de gestion de l'application.
- Le deuxième chapitre « Conception et réalisation du projet » présente les différents modèles conceptuels et organisationnels.
- Le troisième chapitre « Réalisation » présente les différentes interfaces réalisées.

1

1. Le lien vers github : https://github.com/ENSIAS-MEH/2019_DW_Pr11

Chapitre 1

Description et analyse du projet

1.1 Description du sujet

1.2 Les règles de gestion

1.2.1 Gestion des utilisateurs

- Chaque utilisateur doit disposer d'un compte pour pouvoir se connecter et visualiser les "rooms" disponibles.
- Pour s'inscrire, l'utilisateur doit fournir toutes les informations demandées et vérifier son adresse mail.
- Chaque utilisateur peut modifier ses informations personnelles ainsi que son mot de passe.

1.2.2 Gestion des rooms

- Chaque room est visible par tous les utilisateurs entre sa date de début et sa date de fin.
- Chaque room est accessible par tout utilisateur à condition qu'il saisisse le mot de passe correcte.
- Chaque room est modifiable uniquement par son créateur.
- Chaque room est supprimée avec tous ses messages une fois sa date de fin est dépassée.

Chapitre 2

Conception du projet

2.1 Diagramme de cas d'utilisation

Ce diagramme a pour but de donner une vision globale du comportement fonctionnel de notre application.

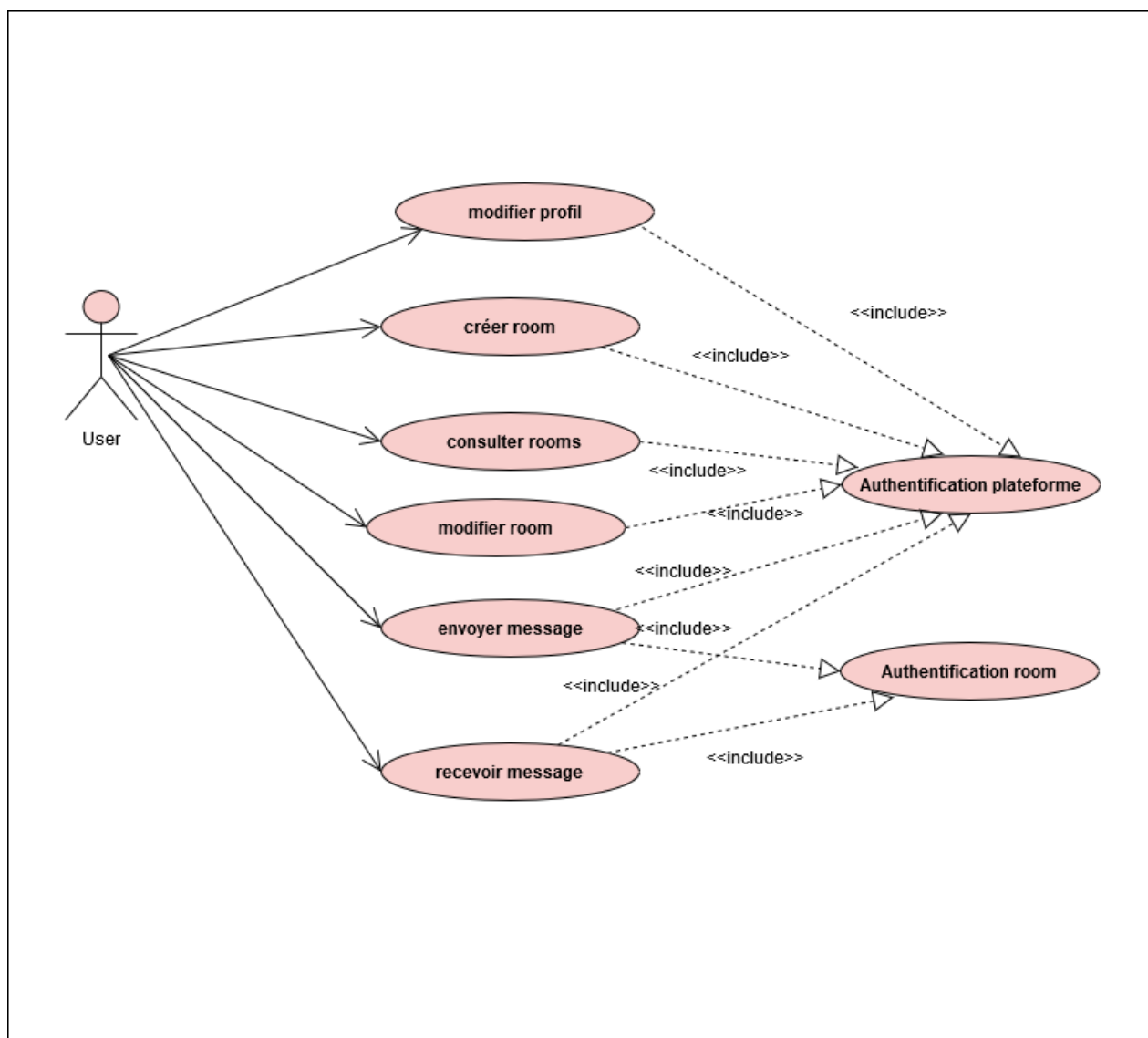


FIGURE 2.1 – Diagramme de cas d'utilisation

2.2 Diagramme de classes

Dans le but de représenter les classes intervenant dans le système, le diagramme de classes est une représentation statique des éléments qui composent un système et de leurs relations, nous élaborons le diagramme de classes suivant :

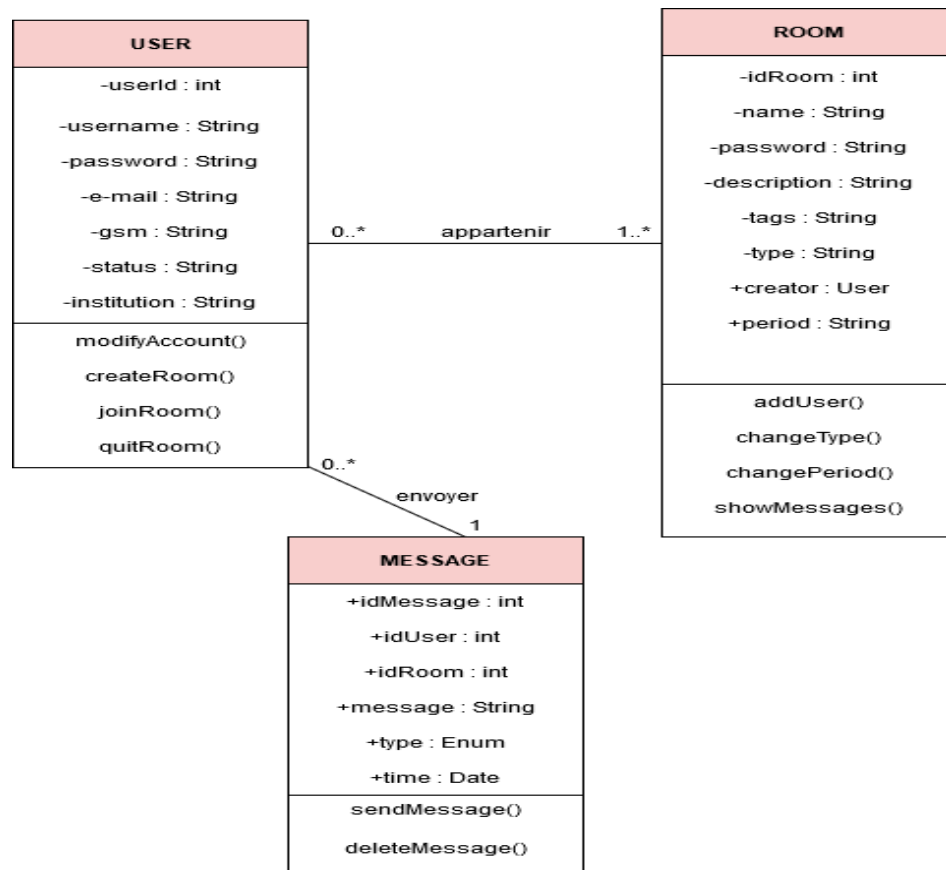


FIGURE 2.2 – Diagramme de classes

2.3 Diagramme de sequence

Le diagramme de séquence suivant décrit comment les éléments du système interagissent entre eux et avec les acteurs.

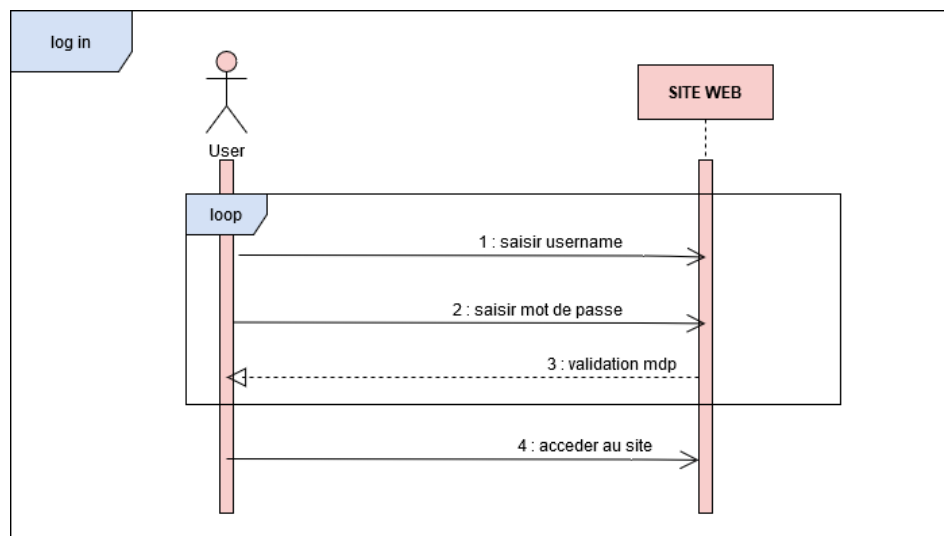


FIGURE 2.3 – Diagramme de sequence : Authentification

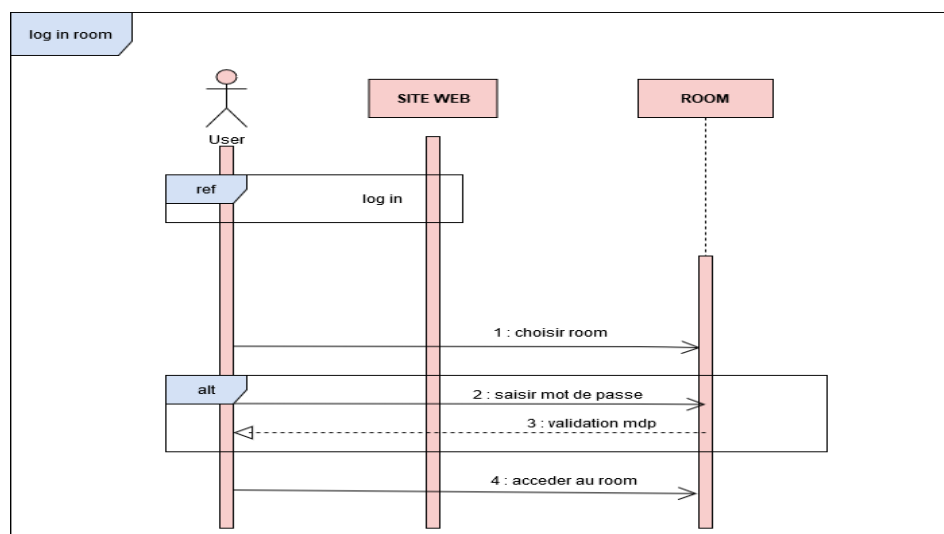


FIGURE 2.4 – Diagramme de sequence : Authentification room

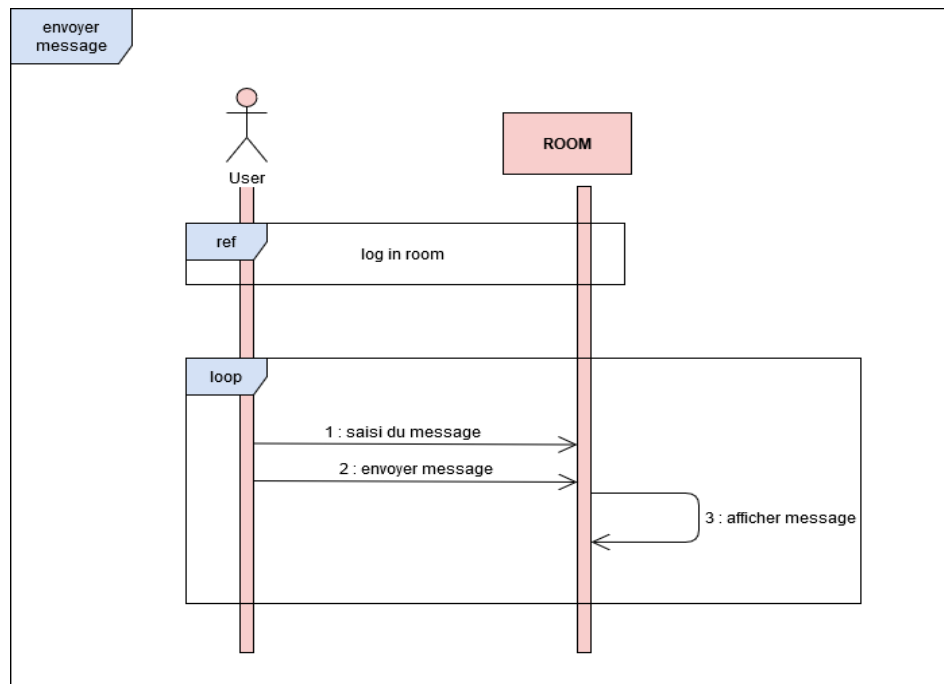


FIGURE 2.5 – Diagramme de sequence : Envoyer message

2.4 Diagramme d'activité

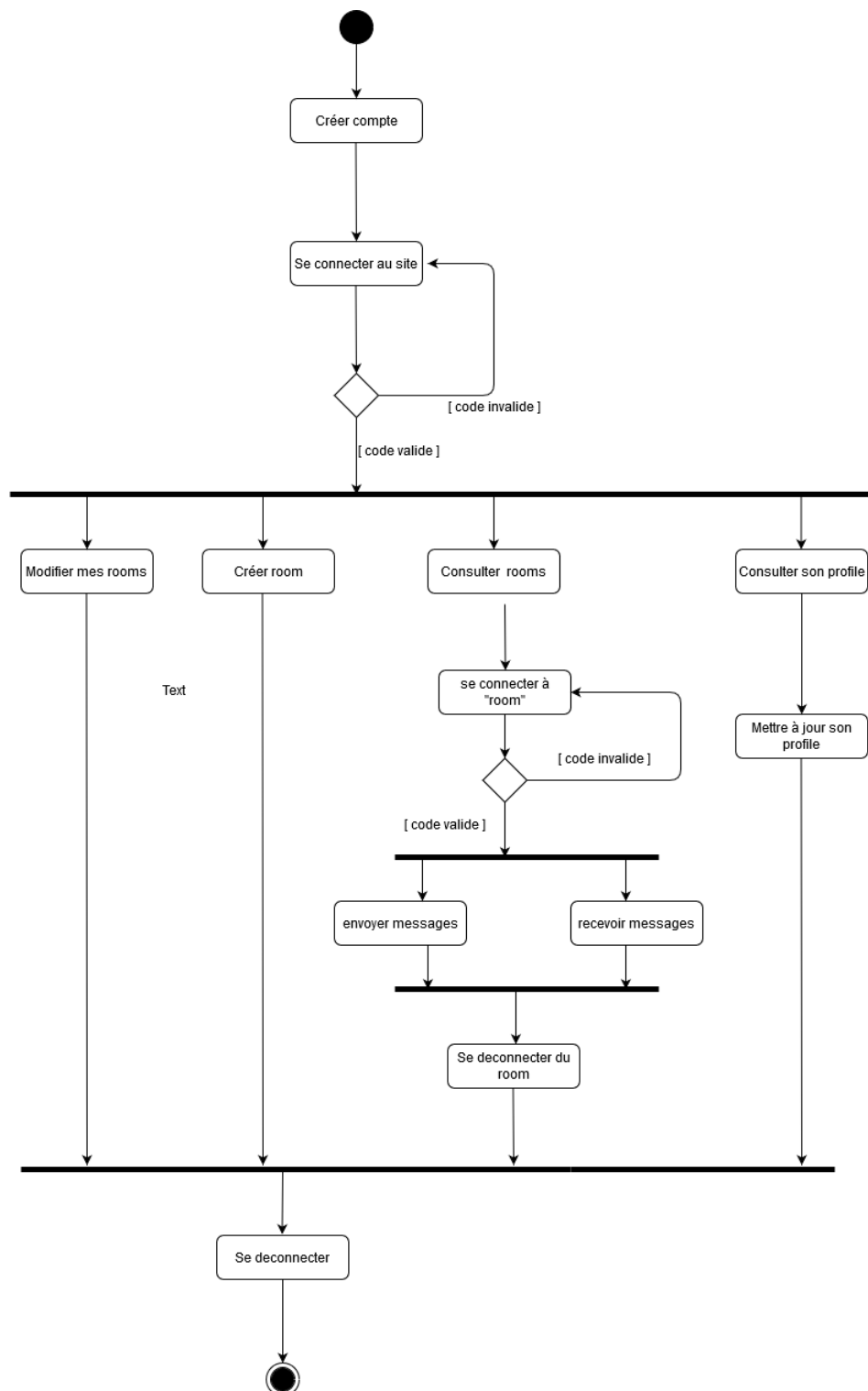


FIGURE 2.6 – Diagramme d'activité

Chapitre 3

Réalisation du projet

3.1 Maquettes :

Login : cette maquette login qui contient deux champ et un bouton

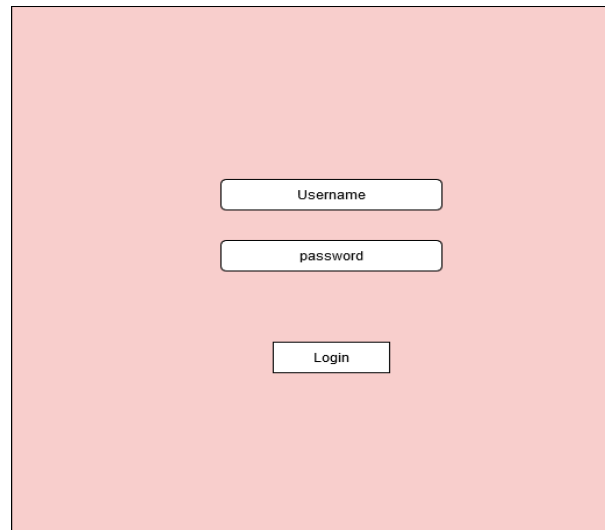
A mockup of a login form. It consists of a light red rectangular background. Centered on this background are three white rectangular input fields with black borders. The first field is labeled 'Username', the second 'password', and the third, which is slightly smaller and positioned below the others, is labeled 'Login'.

FIGURE 3.1 – Maquette : Login

Sign-in : cette maquette contient quatre champ de saisie et un bouton

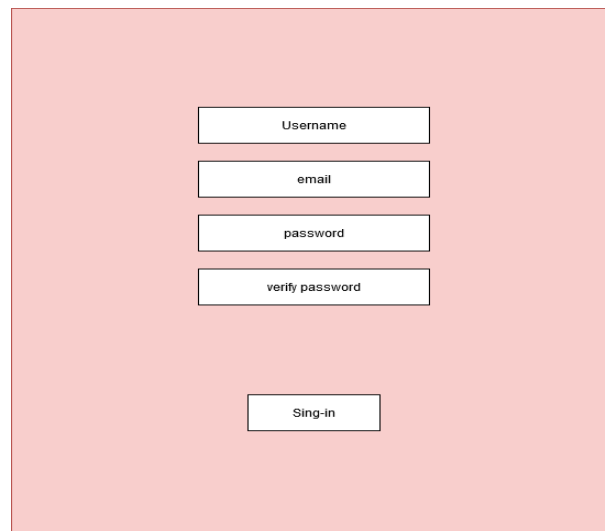
A mockup of a sign-in form. It consists of a light red rectangular background. Centered on this background are four white rectangular input fields with black borders, stacked vertically. The labels for these fields are 'Username', 'email', 'password', and 'verify password'. Below these four fields is a single white rectangular button with a black border, labeled 'Sing-in'.

FIGURE 3.2 – Maquette : Sing-in

Aceuil : Cette maquette correspond à l'accueil après l'authentification

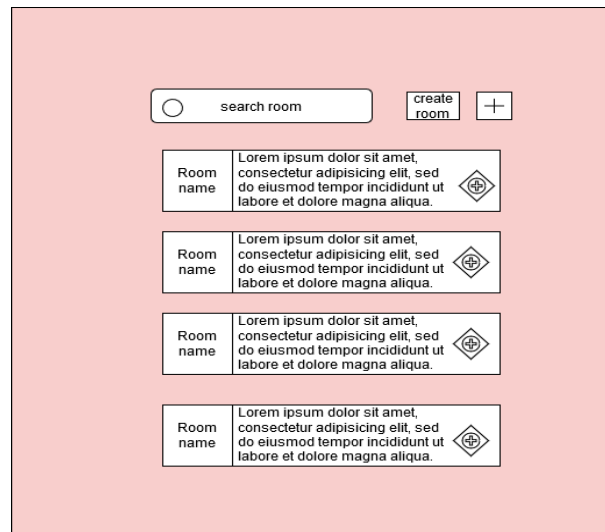


FIGURE 3.3 – Maquette : Accueil

Room : Cette maquette correspond à l'interface de discussion

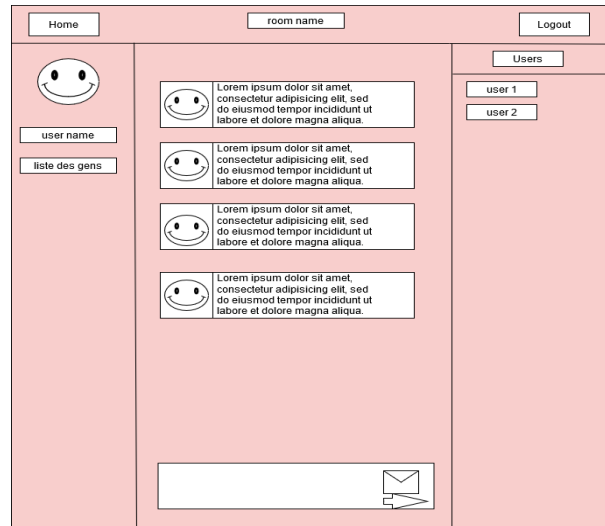


FIGURE 3.4 – Maquette : Interface Room

3.2 Outils informatiques

JEE : JEE est l'acronyme de Java 2 Entreprise Edition. Cette édition est dédiée à la réalisation d'applications pour entreprises. JEE est basé sur J2SE (Java 2 Standard Edition) qui contient les API de base de Java.



FIGURE 3.5 – Logo : JEE

Eclipse JEE : 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.



FIGURE 3.6 – Logo : Eclipse

HTML/CSS/JS/Bootstrap : Sont des outils Front-end



FIGURE 3.7 – Logo : HTML/CSS/JS/Bootstrap

PhpMyAdmin : Est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL.



FIGURE 3.8 – Logo : phpMyAdmin

Visual Studio Code : Est un éditeur de code extensible développé par Microsoft pour Windows, Linux.

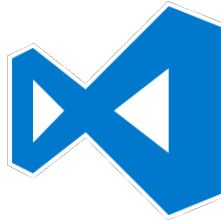


FIGURE 3.9 – Visual Studio Code

Docker : Docker fournit une API pour interagir avec le démon Docker (appelée API du moteur Docker) ... L'API du moteur du Docker est une API RESTful accessible par un client HTTP tel que wget ou curl.



FIGURE 3.10 – Docker

JSTL : JavaServer Pages Standard Tag Library est un composant de la plate-forme JEE de développement. Elle étend la spécification JSP en ajoutant une bibliothèque de balises pour les tâches courantes, comme le travail sur des fichiers XML, l'exécution conditionnelle, les boucles et l'internationalisation

3.3 Interfaces

Accueil : Elle s'agit de la page d'accueil du site visible par tous les navigateurs et qui leur permet d'avoir une idée sur les fonctionnalités de la plateforme.

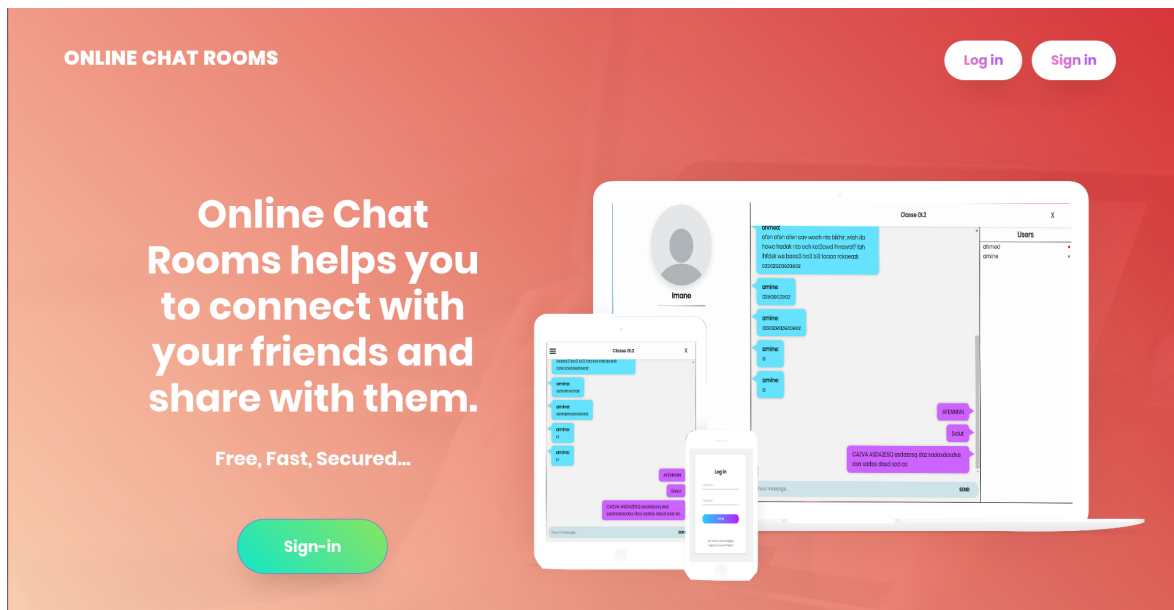


FIGURE 3.11 – Page : Accueil

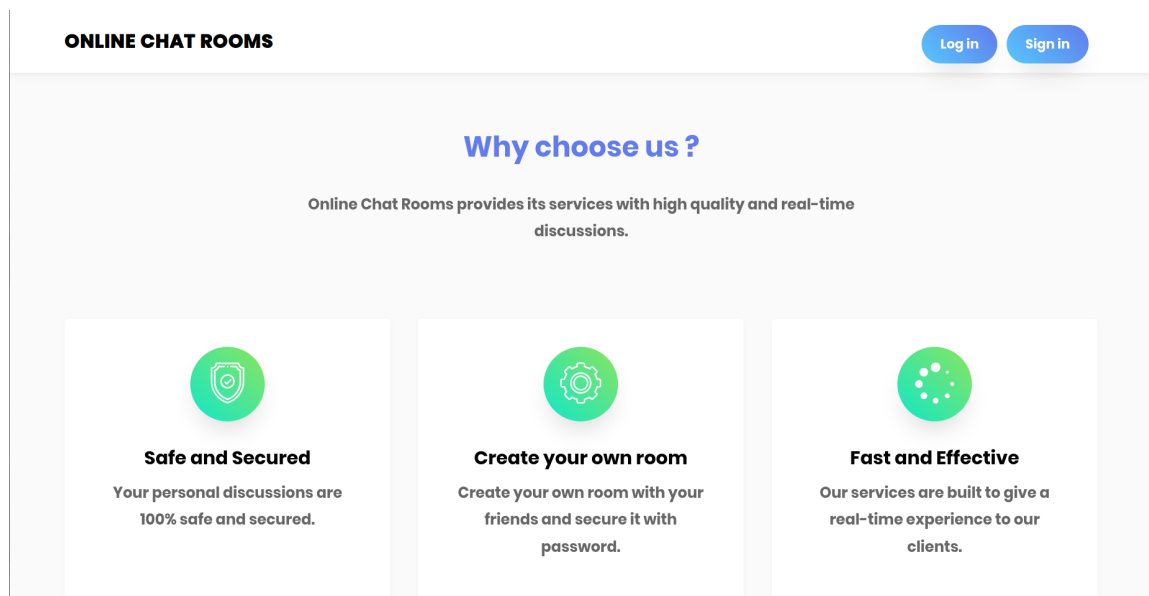


FIGURE 3.12 – Page : Accueil

Sign-in : L'inscription au plateforme.

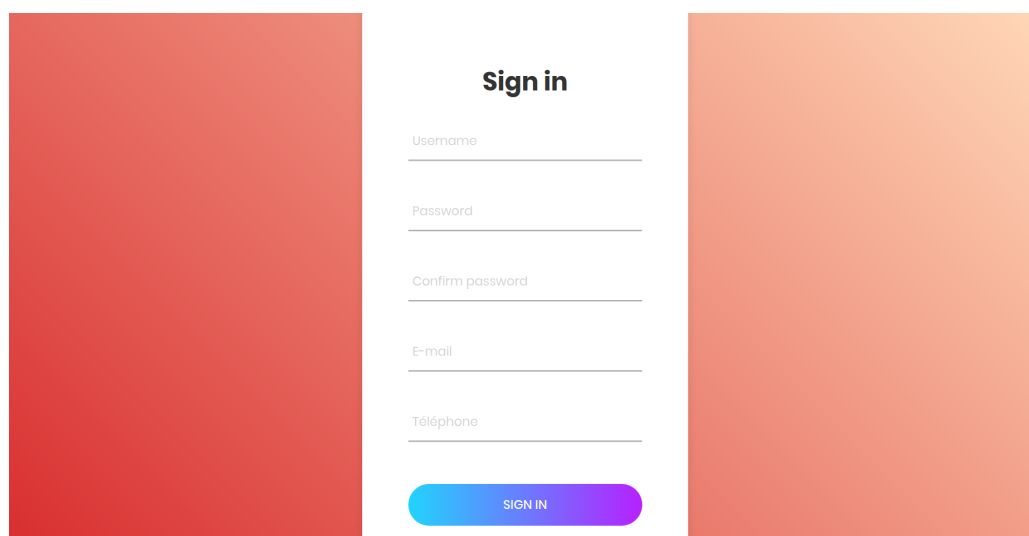


FIGURE 3.13 – Page : Sign-in

Login : L'authentification au plateforme.

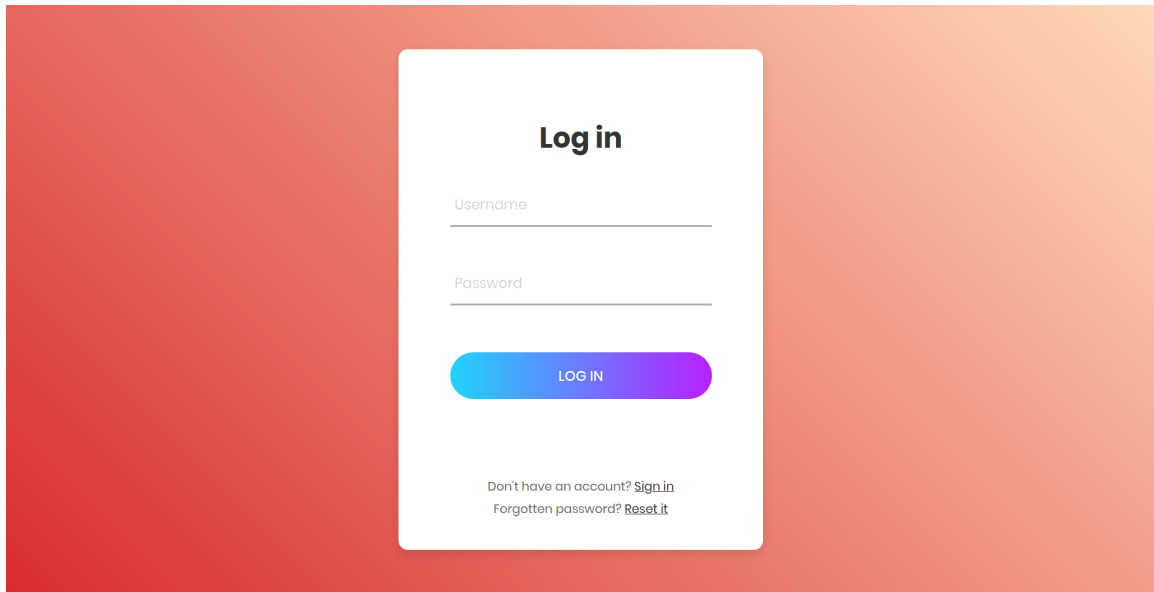


FIGURE 3.14 – Page : Login

Rooms et RoomLogin : Cette page permet à tout utilisateur connecté de voir les “Rooms” créés par tous les utilisateurs de la plateforme et de se connecter à l’une d’entre elles s’il connaît le mot de passe.

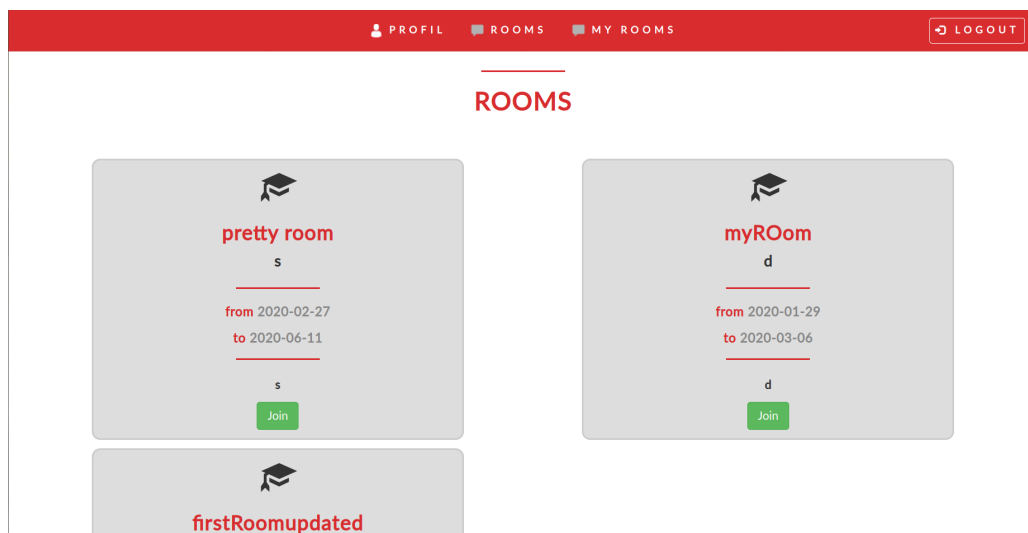


FIGURE 3.15 – Rooms

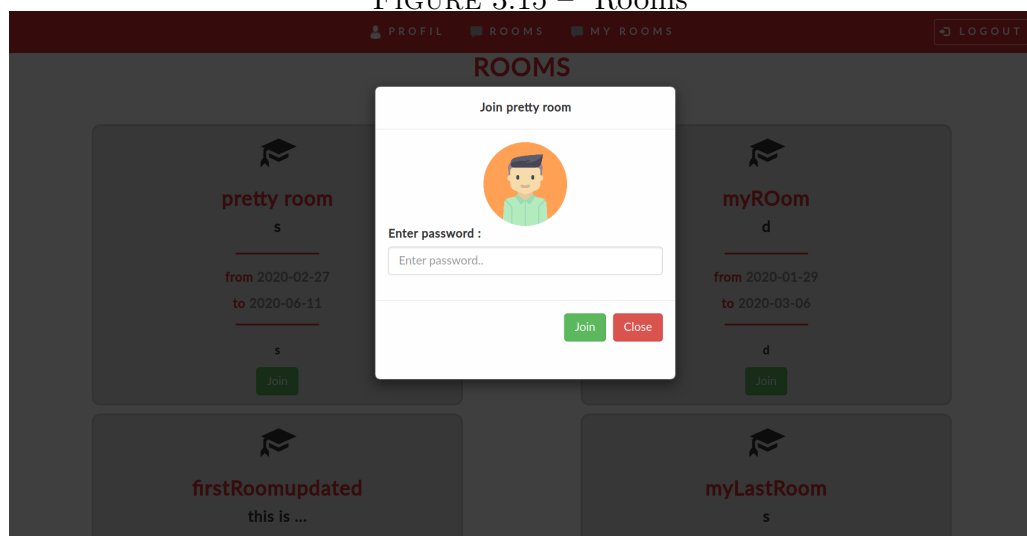


FIGURE 3.16 – Page : Login Room

Mes Rooms : Cette page présente les Rooms créés par l'utilisateur connecté et lui permet éventuellement d'y apporter des modifications.

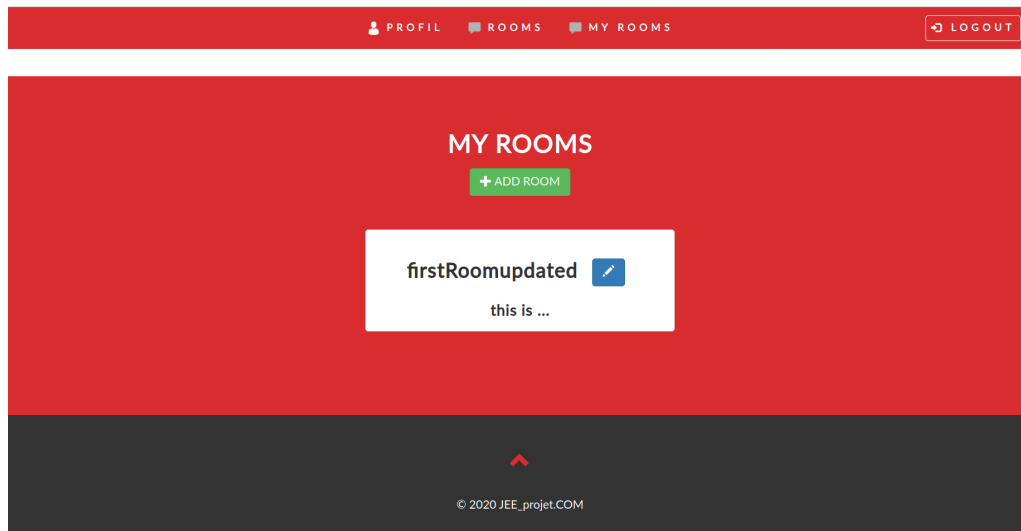


FIGURE 3.17 – Page : Mes Rooms

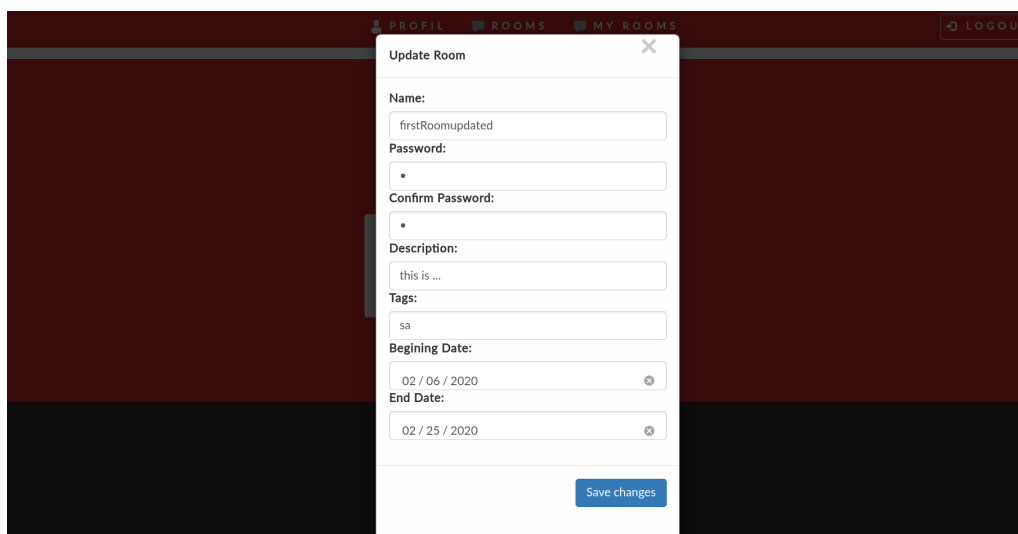


FIGURE 3.18 – Page : Modifier Room

Chat : Cette page permet aux utilisateurs connectés au Room de visualiser et d'envoyer des messages et des fichiers, ainsi de voir les utilisateurs connectés.

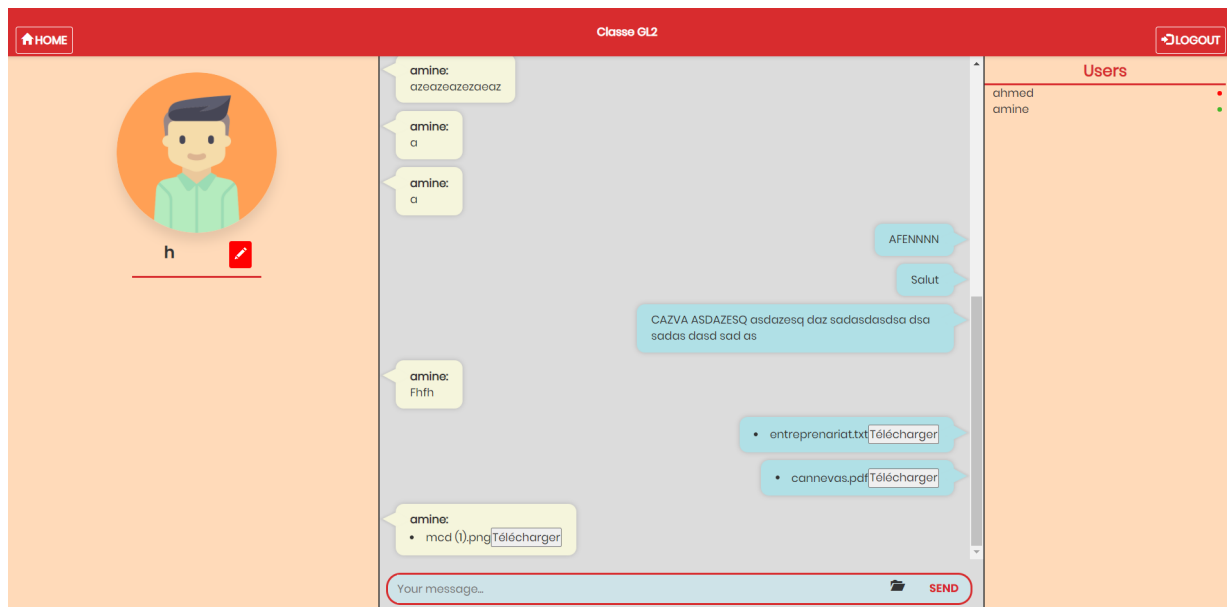


FIGURE 3.19 – Page : Chat

L'utilisateur peut envoyer des fichiers en cliquant le bouton près du bouton "SEND".

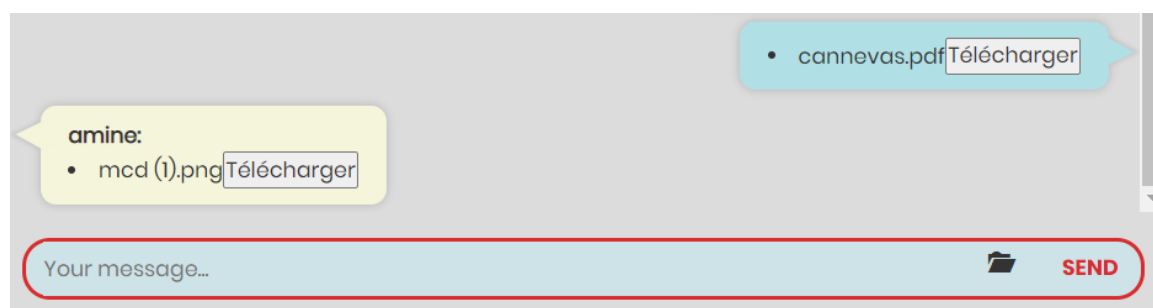


FIGURE 3.20 – Page : Envoi des fichiers

3.4 Déploiement docker :

Dockerfile : Le fichier dockerfile pour déployer le projet sur une image TomEE contenant Tomcat + Java EE.

```
FROM tomee:latest
ADD SiteChat.war /usr/local/tomee/webapps/
EXPOSE 8081
CMD ["catalina.sh", "run"]
```

a

effectuer un test unitaire sur le comportement du "Login" avec le framework open
source JUNIT
aption Dockerfile

Création conteneur : On construit un conteneur pour l'image précédente.

```
C:\Users\Ahmed\Desktop\JEE>docker build -t sitechat .
```

FIGURE 3.21 – Construire conteneur

Création conteneur : On déploie le conteneur sur le port 8081 de la machine.

```
C:\Users\Ahmed\Desktop\JEE>docker run -p 8081:8081 sitechat
```

FIGURE 3.22 – Déploiement du conteneur

3.5 Test unitaire :

Les tests unitaires sont destinés à tester une unité du logiciel. Afin d'être vraiment unitaires, ils doivent être en totale isolation pour ne tester qu'une classe et qu'une méthode à la fois.

Tester le bon comportement de l'application c'est bien détecter les éventuelles régressions de ce dernier. On a effectué un test unitaire sur le comportement du **"Login"** avec le framework open source JUnit



The screenshot displays an IDE window with two tabs: `TestLogin.java` and `LoginDao.java`. The `TestLogin.java` file contains the following code:

```
1 package Tests;
2 import junit.framework.*;
3 import org.junit.Test;
4
5 import com.JeeProjetDao.LoginDao;
6 public class TestLogin extends TestCase {
7     public TestLogin(String method) {
8         super(method);
9     }
10
11     @Test
12     public final void testLogin() {
13         String login = "ahmed"; String password = "ahmed";
14         assertTrue("Login succeeded", LoginDao.check(login, password) != -1);
15
16         String login2 = "ahmed"; String password2 = "amine";
17         assertFalse("Login not succeeded", LoginDao.check(login2, password2) != -1);
18     }
19 }
20 }
```

Below the code editor, the IDE's status bar shows the execution results:

- Console: Finished after 0,403 seconds
- JUnit Results: Runs: 1/1, Errors: 0, Failures: 0
- Test Run: Tests.TestLogin [Runner: JUnit 4] (0,154 s)
- Buttons: Failure Trace

FIGURE 3.23 –

Conclusion

En guise de conclusion, ce travail fait la synthèse des différentes étapes que nous avons suivies pour développer une application web du chat .

Nous nous sommes intéressés aux différentes réflexes et outils pour intégrer cette solution dans l'univers de l'ENSIAS. Premièrement, nous avons vu une conception et une analyse basées sur des bonnes pratiques. Ensuite, nous avons mis en oeuvre notre application sous JEE.

Ce travail nous a permis de solidifier nos compétences techniques et gagner de la confiance en soi lors du développement en JEE et en Front-end.

Bibliographie

[1] Cours :

<https://openclassrooms.com/fr/courses/626954-creez-votre-application-web-avec-java>

[2] Un questionnaire sur Stackoverflow :

<https://stackoverflow.com/questions/15055968/java-ee-simple-login>.

[3] Editeur de texte Latex en ligne :

<https://www.overleaf.com>.

[4] Editeur diagrammes en ligne :

<https://www.draw.io>