******

***Un clone Whatsapp avec MONGO, EXPRESS, REACT, NODE (MERN Stack)***

**Réalisé par :**

*Sami KAHLOUL & Imen RAHAL*

**Caractéristiques de l’application**

* Fonctionnalité de discussion de base.
* Indicateurs de saisie.
* Aperçu du message.
* Envoyez des pièces jointes.
* Envoyez des messages vocaux. (FRAIS)

**Identification des acteurs de l’application**

**Utilisateur**

C'est la personne qui peut jouer le rôle des deux acteurs précédents de plus de sa prise en charge de la gestion des messages et la gestion des contacts.

**Spécification des besoins fonctionnels**

Notre application doit permettre les actions suivantes pour l’acteur utilisateur :

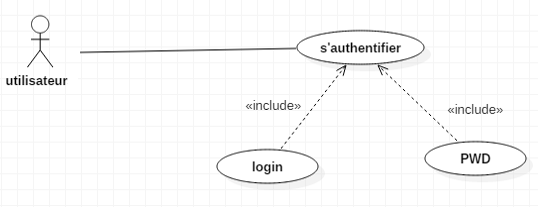
|  |  |
| --- | --- |
| **Acteur** | **Cas d’utilisation** |
| **Utilisateur** | \* s’authentifier  \*gérer les messages  \*\*envoyer des messages  \*\*consulter des messages  \*\* envoyer vocal  \*\*envoyer des photos  \*gérer les contacts  \*\*ajouter un contact  \*\*supprimer un contact  \*\* consulter liste des contacts |

***Tableau 1 : Acteur et leurs cas d’utilisations***

**Analyse du cas d’utilisation « s’authentifier »**

|  |  |
| --- | --- |
| **Titre** | S’authentifier |
| **But** | Authentification et autorisation d’accès |
| **Pré-conditions** | Vérification des conditions d’authentification |
| **Résumé** | L’acteur introduit son login et mot de passe pour accéder au système |
| **Acteur** | Utilisateur |
| **Post-conditions** | Aller à l’interface demandée |
| **Exceptions** | Pas d’authentification |

***Tableau 2 : cas d’utilisation « s’authentifier »***

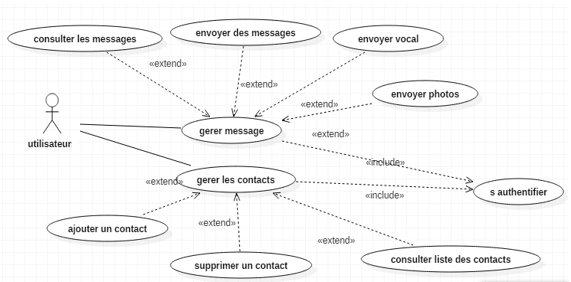


***Figure 1 : diagramme de cas d’utilisation « s’authentifier »***

**Analyse de cas d’utilisation « acteur utilisateur»**

|  |  |
| --- | --- |
| **Titre** | Utilisateur |
| **But** | Gestion des messages  Gestion des contacts |
| **Pré-conditions** | Vérification des conditions d’accès à l’interface demandé |
| **Résumé** | l’acteur, après l’authentification, accède l’interface et choisit l’interface correspondante |
| **Acteur** | Utilisateur |
| **Post-conditions** | Aller à l’interface demandé |
| **Exceptions** | Pas d’accès à l’interface demandé |

***Tableau 3 : cas d’utilisation pour l’acteur utilisateur***

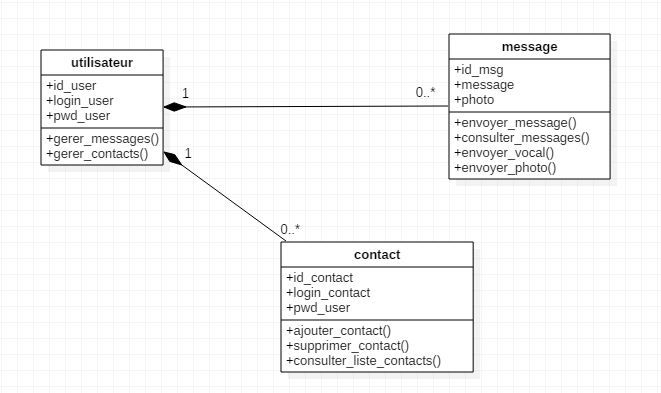


***Figure 2 : diagramme des cas d’utilisations pour l’acteur utilisateur***

**Diagramme des classes**

Les diagrammes de classes expriment de manière générale la structure statique d’un système, en termes de classes et de relations entre elles. De même qu’une classe décrit un ensemble d’objets, une association décrit un ensemble de liens ; les objets sont des instances de classes et les liens sont des instances de relations.

Et voici notre diagramme des classes :



***Figure 3 : diagramme des classes***

**Environnement de travail**

* Nous utilisons des crochets React comme useEffect, useReducer, useContext, useState et plus encore.
* Pour un chat en temps réel, nous utilisons socket.io.
* Pour le backend, nous utilisons Nodejs/Express pour créer notre API.
* Nous allons stocker nos discussions dans Mongo.
* Et pour l'activité des utilisateurs, nous utilisons Redis.
* Pour la conception, nous utilisons StarUML.