

Documentation Technique AlertMNS

- I) Introduction
- II) L'architecture
- III) L'arborescence
- IV) Fonctionnalités
- V) Structure de la base de données
- VI) Jeu d'essai
- VII) Charte graphique
 - a) Les couleurs
 - b) Logo
 - c) La typographie
 - d) Les icônes
- VIII) Gestion de projet
- IX) Intégration et déploiement du site
- X) Sécurité
 - a) Authentification et autorisations
 - b) Protection contre les attaques

I - Introduction

Notre plateforme a été conçue pour permettre aux entreprises et aux particuliers de pouvoir communiquer facilement et efficacement, que ce soit pour des conversations individuelles ou en groupe. Notre application est là pour faciliter les échanges entre les collaborateurs.

Notre application web de messagerie offre une variété de fonctionnalités pour répondre aux besoins de communication de l'entreprise.

Cette documentation technique fournit des informations détaillées sur l'architecture, les fonctionnalités et les technologies utilisées dans notre application web de messagerie.

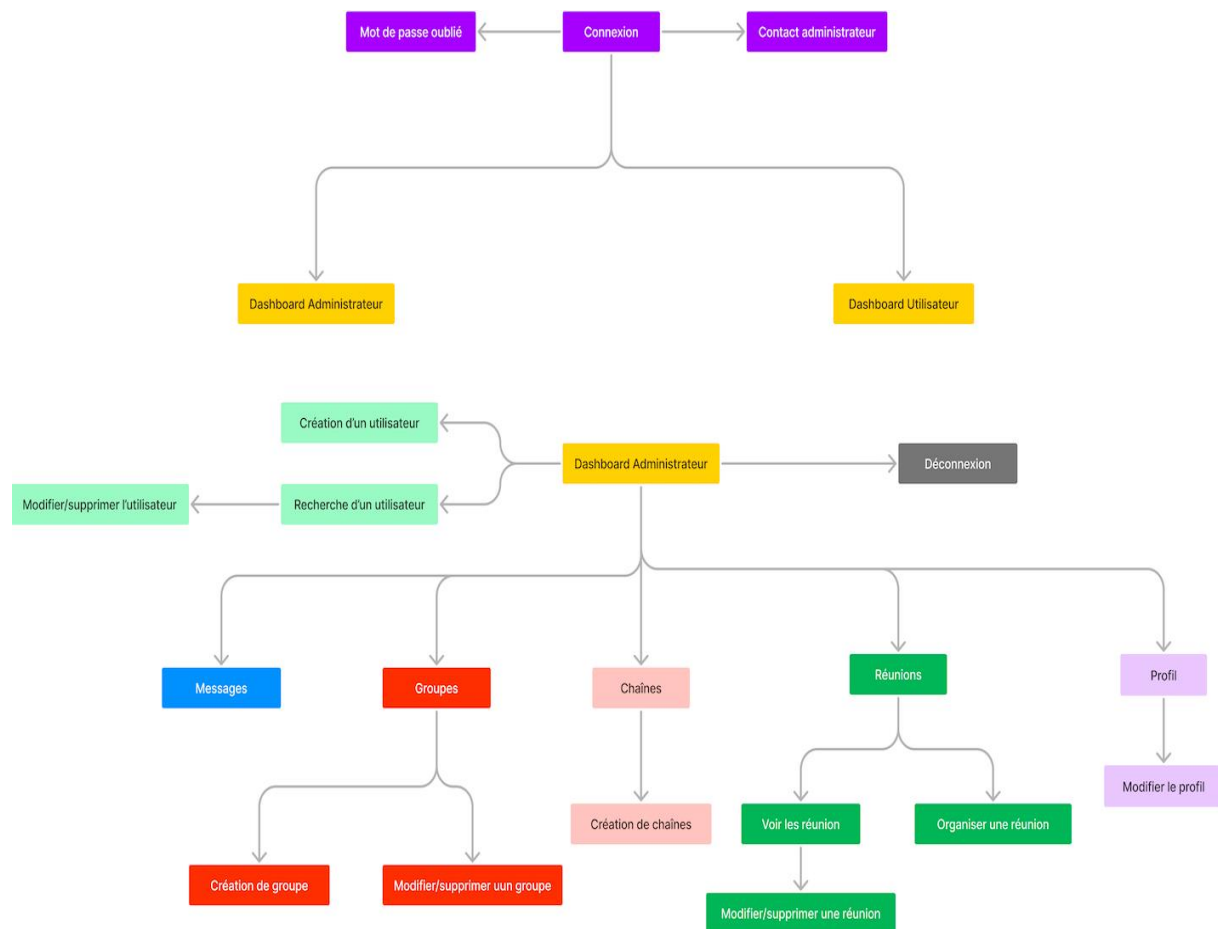
II - L'architecture

L'architecture de notre application web repose sur une approche moderne à **trois niveaux**, comprenant le **frontend**, le **backend** et la **base de données**. Cette architecture permet une séparation claire des responsabilités et offre une évolutivité et une flexibilité accrues. Nous n'avons utilisé aucun framework lors de la réalisation de l'application.

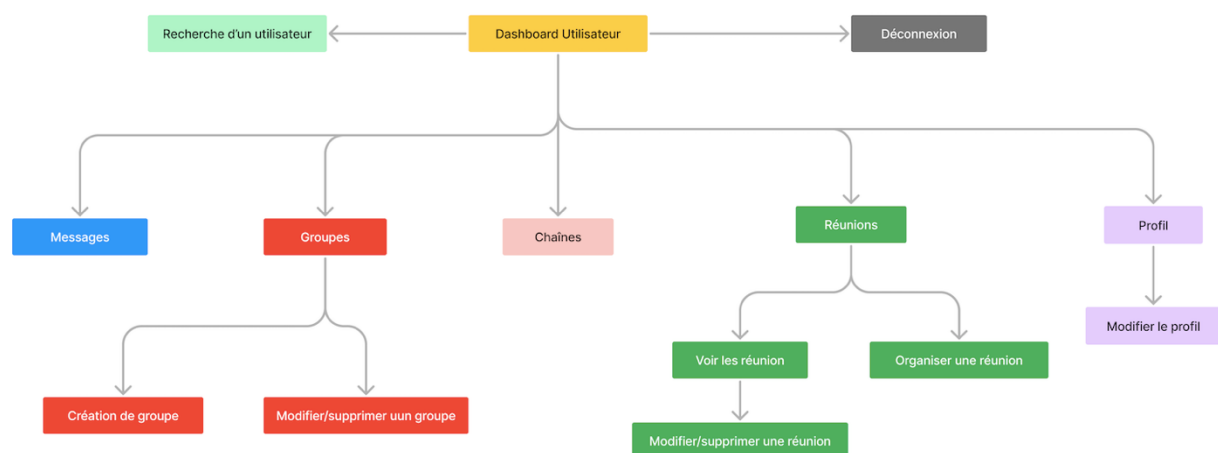
1. **Frontend** : Le frontend de notre application est développé en utilisant les technologies **HTML5**, **CSS3** et **JavaScript**.
2. **Backend** : Nous avons utilisé le PHP comme langage de programmation pour la partie Backend, mais aussi des requêtes **JSON** et **AJAX** afin de pouvoir rendre notre application plus dynamique.
3. **Base de données** : Pour la gestion des données, nous utilisons une base de données relationnelle **MySQL**.
MySQL est un système de gestion de base de données fiable et performant, adapté aux applications web à fort trafic. Il nous permet de stocker et d'accéder efficacement aux données de l'application.

Cette architecture en trois niveaux permet la réutilisabilité du code et facilite la maintenance et l'évolution de l'application. Elle offre également une extensibilité, ce qui nous permet d'ajouter facilement de nouvelles fonctionnalités à l'application à l'avenir.

III - L'arborescence



Arborescence de l'utilisateur



IV - Fonctionnalités

Voici un aperçu des principales caractéristiques :

1. Un **Dashboard** qui regroupe toutes les actions que l'utilisateur pourra effectuer lors de l'utilisation de l'application. Il existe à ce jour 3 rôles différents :
 - Le rôle **administrateur** : Il possède tous les droits, peut créer et ajouter des utilisateurs, il peut ajouter ou supprimer des rôles...
 - Le rôle **utilisateur** : Il possède moins de liberté que l'administrateur. Il ne peut pas créer d'utilisateur par exemple.
 - Le rôle **formateur** : Il possède le même rôle qu'un utilisateur. Nous avons décidé de choisir ce nom de rôle car notre application de messagerie est dédiée à l'entreprise Metz Numeric School, qui est un centre de formation.

L'administrateur pourra décider de supprimer ou d'ajouter des rôles différents selon les besoins de l'entreprise.

2. **Messagerie instantanée** : Il est possible d'envoyer des messages texte en temps réel à vos contacts, leur permettant de recevoir instantanément vos messages. Que ce soit pour des discussions informelles ou professionnelles, vous pouvez échanger des messages rapidement et efficacement.

3. **Conversations de groupe** : Notre application prend également en charge les conversations de groupe, vous permettant de créer des discussions collectives avec plusieurs participants. Vous pouvez ainsi collaborer avec vos collègues, planifier des événements ou simplement discuter avec un groupe d'amis.

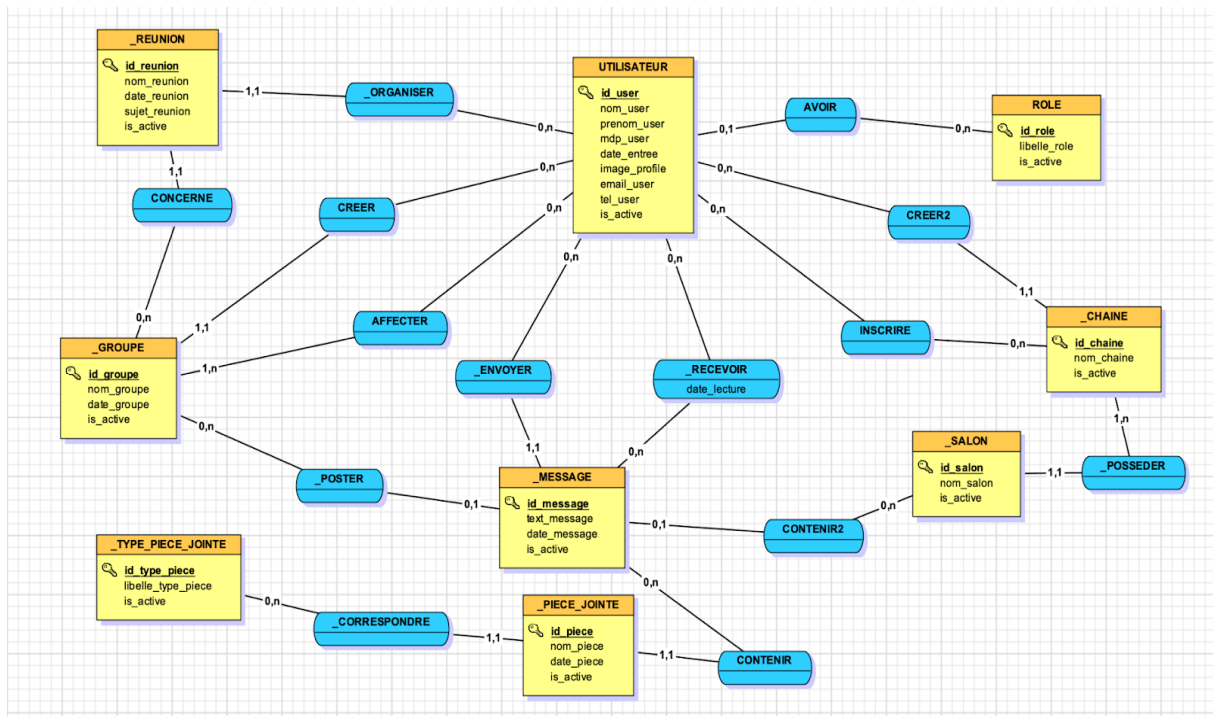
4. **Chaînes** : Similaire à Discord, nous avons inclus dans notre application la possibilité de créer des chaînes. Les chaînes regroupent plusieurs utilisateurs qui pourront discuter de sujets divers liés aux thèmes de la chaîne, s'informer sur la vie communautaire de l'entreprise, etc.

5. **Création de salons** : Dans ces chaînes il est possible de créer des "salons" qui y seront associés. Cela permet de mieux organiser les chaînes ainsi que les différents sujets relatifs à cette chaîne. (Ex : Création d'une chaîne "Metz Numeric School", avec un salon "Général", "Informations pratiques"...)

6. **Organisation de réunions** : L'application AlertMNS offre la possibilité d'organiser des réunions entre collaborateurs. Vous pouvez y choisir : Le nom de la réunion, la date, ainsi que les participants.

7. **Profil personnalisé** : Vous pouvez personnaliser votre profil en ajoutant une image de profil personnalisée. Lorsque vous récupérez les identifiants de votre compte grâce à l'administrateur, une fois connecté(e), vous pouvez vous diriger vers votre profil, et y modifier votre prénom, nom, ainsi que l'adresse e-mail.

V) Structure de la base de données



Notre application est composée de 9 tables :

- Une table **utilisateur** qui contient les colonnes : nom_utilisateur, prenom_utilisateur, mdp_utilisateur, date_entree, image_profile, email_utilisateur, tel_utilisateur
- Une table **rôle**
- Une table **chaîne**
- Une table **réunion**
- Une table **groupe**
- Une table **message**
- Une table **pièce jointe**
- Une table **type pièce jointe**
- Une table **salon**

VI) Jeu d'essai

Pour générer un jeu d'essai pour notre base de données, nous avons utilisé le site **Generatedata**.

Generatedata.com est un site web qui permet de générer des données aléatoires dans différents formats, tels que CSV, JSON, SQL, XML, etc. On peut y définir les types de données souhaités, les champs à inclure, et les éventuelles contraintes.

Le site **generatedata.com** propose une large gamme de types de données prédéfinis, tels que les **noms**, les **adresses**, les **numéros de téléphone**, les **adresses email**, les **dates**, les **nombre**s, les **codes postaux**, etc.

Une fois les paramètres configurés, nous pouvons choisir le nombre de lignes de données que l'on souhaite générer. Les données générées peuvent ensuite être exportées dans le format de notre choix, nous avons choisi le format SQL afin de pouvoir importer ces données dans notre base de données dans phpMyAdmin.

The screenshot shows the Generatedata.com website interface. At the top, there's a navigation bar with the site name and a 'New Data Set' button. Below this, a table lists the data types and column names for the new set:

#	Data Type	Column Name	Examples
1	Names	nom_utilisateur	Smith (surname)
2	Names	prenom_utilisateur	Alex (any gender)
3	Random Number of W...	mdp_utilisateur	No examples available.
4	Date	date	06/05/2023

Below the table, there are options to generate a specific number of records (7 to 14) and a format code (yLLdd). A 'Generate' button is visible. A modal window titled 'Generate' is open, showing the number of records generated (30) and a 'Data generated.' message. A 'DOWNLOAD' button is also present. The bottom part of the screenshot shows the generated SQL code for creating a table and inserting data.

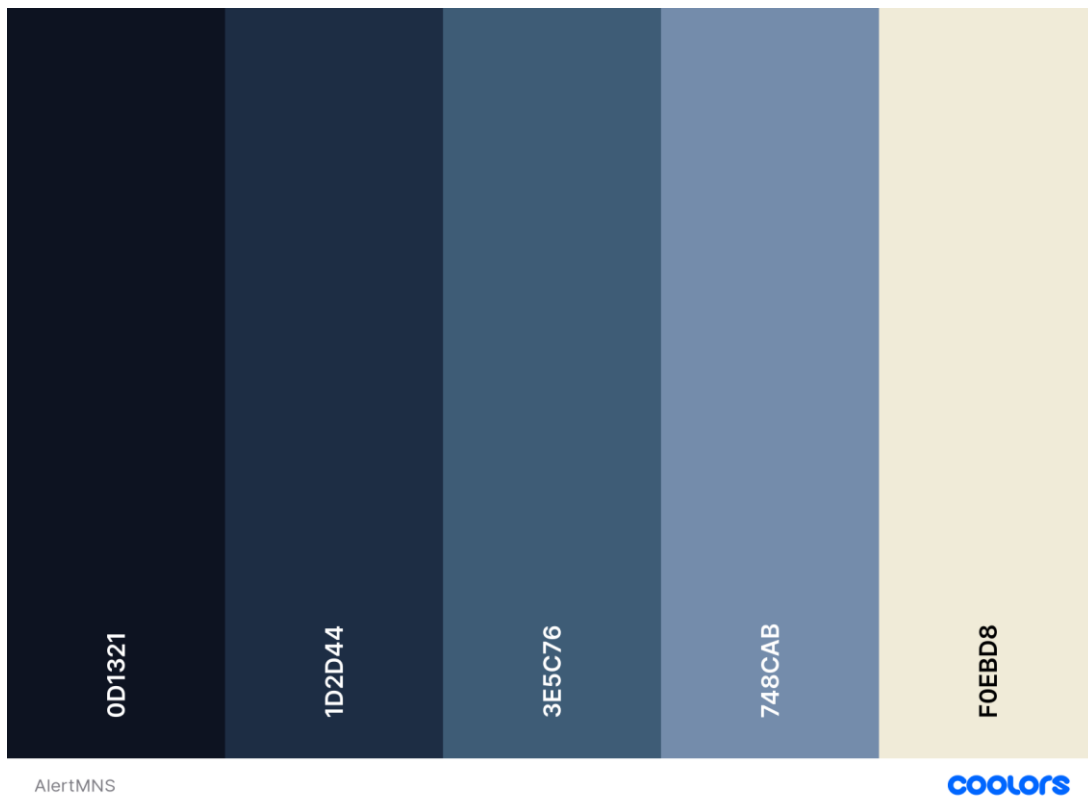
```
CREATE TABLE `myTable` (
  `nom_utilisateur` varchar(255) default NULL,
  `prenom_utilisateur` varchar(255) default NULL,
  `mdp_utilisateur` TEXT default NULL,
  `date` varchar(255),
  `email_utilisateur` varchar(255) default NULL,
  `tel_utilisateur` varchar(100) default NULL,
  `id_role` varchar(255) default NULL,
  `is_active` varchar(255) default NULL,
  PRIMARY KEY (`id`)
) AUTO_INCREMENT=1;

INSERT INTO `myTable` (`nom_utilisateur`,`prenom_utilisateur`,`mdp_utilisateur`,`date`,`email_utilisateur`,`tel_utilisateur`,`id_role`,`is_active`)
VALUES
('Torres','September','lobortis risus. In mi pede, nonummy ut, molestie in, tempus eu','2024/05/20','magna.a@google.ca','07 54 65 80 95','0','0'),
('Martin','Jillian','ridiculus mus. Donec dignissim magna a tortor. Nunc commodo auctor velit. Aliquam nisl','2022/09/12','ante.dictum.mi@yahoo.ca','09 19 29 47 72','1','1'),
('Mullins','Bradley','torquent per conubia nostra, per inceptos hymenaeos. Mauris ut','2023/02/16','dolore@cloud.com','08 57 14 58 45','0','1'),
('Bishop','Athena','magnis dis parturient montes, nascetur ridiculus mus. Proin vel','2022/09/13','convallis.est@hotmail.ca','04 47 38 15 58','1','0'),
('Long','Cairo','lorem. Donec elementum, lorem ut aliquam iaculis, lacus','2022/11/08','luctus.curabitur@aol.com','04 50 55 96 93','0','0');
```

VII) Charte graphique

a. Les couleurs

Nous avons utilisé le site <https://coolors.co/> afin de définir les codes couleurs de notre application. Nous avons opté pour un thème sombre.



a. Logo



b. La typographie

Pour l'ensemble de notre application, nous avons uniquement utiliser la police **Roboto**.

Roboto est une famille de polices typographiques développée par Google. Elle est largement utilisée dans les projets web et d'interface utilisateur en raison de sa lisibilité et de sa polyvalence. Roboto a été spécialement conçue pour être utilisée sur les appareils mobiles, mais elle est également adaptée à d'autres supports, tels que les sites web, les applications et les documents imprimés.

c. Les icônes

Pour les icônes nous avons décidé d'utiliser une bibliothèque gratuite en ligne : **Font Awesome**. (<https://fontawesome.com/>).

Font Awesome est une bibliothèque d'icônes très populaire utilisée par de nombreux développeurs et designers web. Elle fournit une vaste collection d'icônes prêtes à être intégrées dans des projets web, ce qui permet d'améliorer facilement l'aspect visuel et l'expérience utilisateur.

VIII) Gestion de projet

Afin de faciliter l'organisation et la répartition des tâches pour notre application web, nous avons utilisé le logiciel de gestion de projet Trello. Trello s'est avéré être un choix judicieux, car il nous a permis de rester organisé, de collaborer facilement avec l'équipe et de suivre nos progrès tout au long du processus de développement.

Nous avons donc créé un tableau Trello dédié à notre projet d'application web. Nous avons utilisé des listes pour représenter différentes étapes clés du processus de développement, telles que la planification, la conception, le développement, les tests et le déploiement. Chaque étape avait sa propre liste, ce qui nous a permis de visualiser clairement les tâches à accomplir et de suivre leur progression.

L'utilisation de Trello pour gérer notre projet d'application web s'est avérée extrêmement bénéfique.

IX) Intégration et déploiement du site

L'application est hébergée sur un serveur OVH qui est un fournisseur d'hébergement proposant des serveurs dédiés ou des machines virtuelles.

Nous utilisons le système d'exploitation Debian de Linux car il offre une grande flexibilité.

Pour accéder au serveur à distance, il était nécessaire d'utiliser le protocole sécurisé SSH (Secure Shell). Ce qui nous permet de gérer l'environnement de déploiement.

Le serveur Web utilisé est NGINX. Il s'agit d'un serveur très performant, stable et capable de gérer de nombreuses connexions simultanées.

X) Sécurité

a. Authentification et autorisations

Pour assurer la protection de l'application, nous avons mis en place un système d'authentification. Pour se connecter, l'utilisateur doit posséder une adresse mail et un mot de passe. Le compte utilisateur est créé par l'administrateur qui fournit les identifiants à l'utilisateur. Les mots de passe sont bien évidemment chiffrés dans la base de données.

Lors de la connexion, l'utilisateur est dirigé soit vers la dashboard administrateur, soit vers la dashboard utilisateur, selon son rôle. L'administrateur a tous les droits. L'utilisateur, quant à lui, ne peut pas gérer les utilisateurs, il ne peut pas gérer les chaînes non plus mais il peut créer des groupes privés.

b. Protection contre les attaques

Notre application est protégée contre les attaques. Nous l'avons protégé contre les injections SQL et avons corrigé les failles XSS. Nous avons fait en sorte qu'il ne soit pas possible de contourner l'authentification en mettant en place des vérifications lors de la connexion. Nous avons mis en place des filtres pour éviter la vulnérabilité lors de l'envoi d'information dans la base de données.