**DOCUMENT TECHNIQUE**

Système de sauvetage et de redistribution alimentaire

**Auteur :**Teki Azanfack Ivana

**SOMMAIRE**

Introduction……………………………………………………………………………………………………………………p1

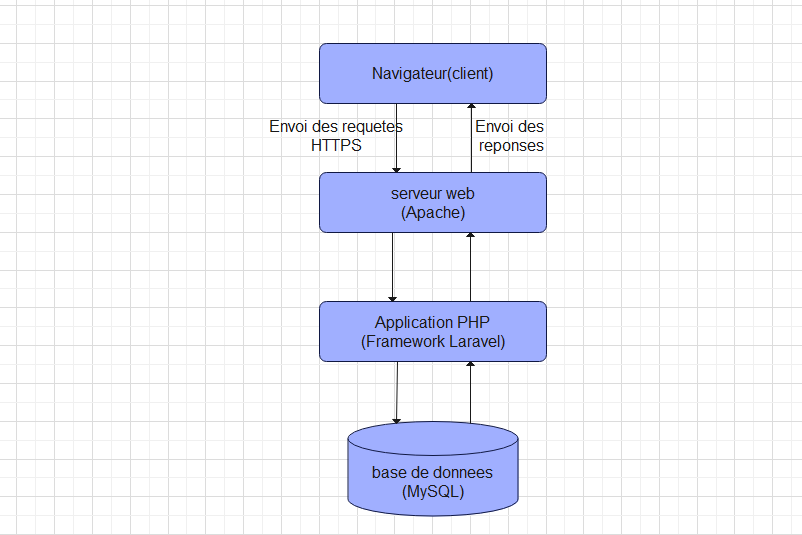
1. Architecture technique du système……………………………………………………………………p1-2
2. Technologies et outils utilisés……………………………………………………………………………..p1-2
3. Structure de la base de données…………………………………………………………………………p1-2
4. Configurations…………………………………………………………………………………………………….p1-2

Conclusion………………………………………………………………………………………………………………………p1

**Introduction**

Ce document décrit l’architecture technique et les spécifications techniques du site FoodHelper, une plateforme dédiée au sauvetage et à la redistribution des invendus alimentaires. Le site vise à faciliter la mise en relation entre donateurs (commerces, restaurants, producteurs) et associations bénéficiaires, en proposant une interface de gestion des dons, une visualisation des dons disponibles et un système de notifications. Dans la suite du présent document nous allons donc présenter l’architecture technique de notre système, les technologies que nous allons utiliser, la structure de la base de données ainsi que les configuration de notre système pour ainsi mieux documenter les développeurs, architectes système et administrateurs techniques impliqués dans le projet.

1. **Architecture technique du système**



L’architecture utilisée pour concevoir notre application est l’architecture 3 tiers. Dans cette dernière, chaque couche ou niveau ou tier effectue une tâche (un service) spécialisée. Ainsi, une couche peut utiliser les services de la couche qui lui est inferieure afin de fournir son propre service. Le premier niveau (un navigateur web dans notre cas) est l'interface utilisateur et la couche de présentation de l'application, où l'utilisateur final interagit avec l'application ; Sa principale fonction est d'afficher des informations à l'attention de l'utilisateur et d'en collecter de ce dernier.  Le niveau Application, également appelé niveau logique ou niveau intermédiaire, est le cœur de l'application. Dans ce niveau, les informations collectées dans le niveau Présentation sont traitées, par exemple pour notre système le navigateur envoi des requêtes HTTPS a notre serveur d’application (dans notre cas c’est l’application même) qui les traite et lui renvoi des réponses. Le niveau Données, parfois appelé niveau Base de données ou Système dorsal, est l'endroit où les informations traitées par l'application sont stockées et gérées ; Dans notre cas, le serveur de base de données (MYSQL) d’assure du traitement des données de notre application.

1. **Technologies et Outils utilisés**

* **Frontend** :
  + Technologies : HTML5, CSS3 et JavaScript.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour écrire les pages web. C’est donc lui qui va nous permettre de définir les différents contenus de nos pages. | Le css(cascading style sheet) est le langage de style utilisé pour définir l’apparence visuelle de l’application en définissant les couleurs, la police, la mise en page et le responsive de l’application pour la rendre agréable à la vue. | JavaScript est un langage de programmation utilisé par les développeurs pour concevoir des sites web interactifs.il permet d'améliorer l'expérience utilisateur d'un site web, de l'affichage d'animations au développement de contenus interactifs. |

* + Framework : Tailwindcss

|  |  |
| --- | --- |
|  | . Tailwind CSS est un framework CSS open source. La fonctionnalité principale de cette bibliothèque est qu’elle fournit une liste de classes CSS « utilitaires » pouvant être utilisés pour ajouter un style à chaque élément en les mélangeant et en les agençant |

* **Backend** :
  + Technologie : PHP
  + Framework : Laravel

|  |  |
| --- | --- |
|  |  |
| Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur web. | Laravel est un framework web open-source expressif et intuitif écrit en PHP respectant le principe MVC(modèle-vue-contrôleur)r et entièrement développé en programmation orientée objet avec ses codes sources hébergées sur GitHub. |

* **Base de données** :
  + Type : Relationnelle (MySQL)

|  |  |
| --- | --- |
|  | MySQL est un système de gestion de bases de données relationnelles qui permet de créer et administrer une base de données et sur lequel on peut effectuer des requêtes SQL. |

* **Serveur Web** :
  + Utilisation d’Apache pour la gestion des requêtes HTTPS.

|  |  |
| --- | --- |
|  | Le serveur Web qui traite les requêtes et diffuse les ressources et le contenu Web via http. Apache est un serveur web open-source et multiplateforme populaire qui est, selon les chiffres, le serveur web le plus populaire qui existe |

* **Gestion du code source et collaboration :** GitHub

|  |  |
| --- | --- |
|  | Est une plateforme de gestion de code source et de collaboration utilisée pour le contrôle de version et l’hébergement des projets. GitHub facilite le suivi des modifications et la collaboration entre les développeurs. |

* **Environnement de développement** : Visual Studio Code

|  |  |
| --- | --- |
|  | VSCode est un éditeur de code open source offrant de multiples outils intégrés pour le développement, la gestion de projets, le débogage et offrant une plénitude d’extensions. |

* **Modélisation** : EdrawMax

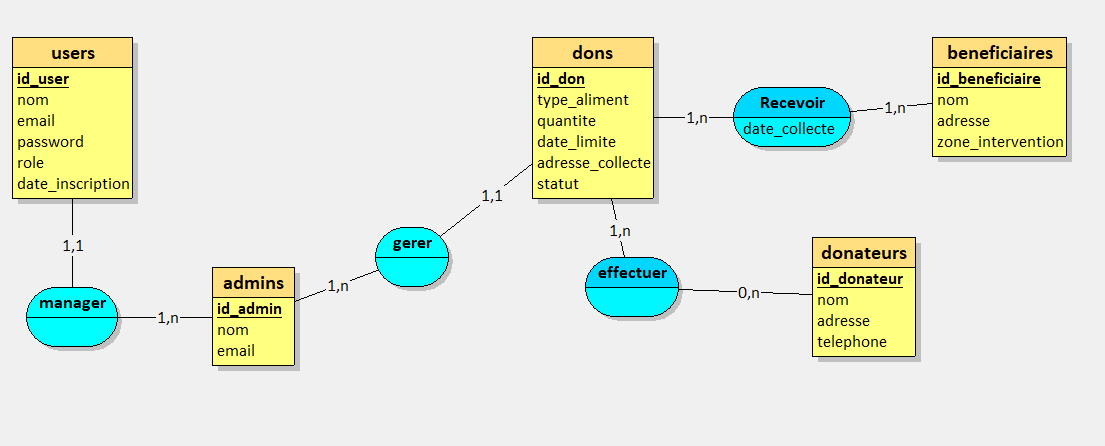
|  |  |
| --- | --- |
|  | EdrawMax est un outil de création de diagrammes polyvalent qui permet de créer plus de 280 types de diagrammes tels que des organigrammes, des diagrammes de flux, des diagrammes UML et bien d’autres. |

1. **Structure de la base de donnees**

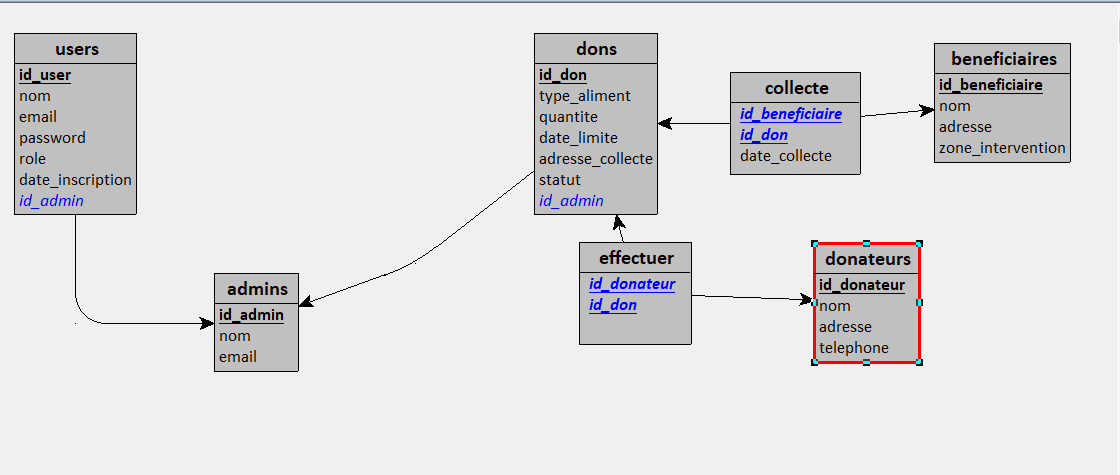
* **Tables Principales**
* **Utilisateurs**  
  users (id\_user, nom, email, mot\_de\_passe, rôle, date\_inscription)
* **Donateurs**  
  donateurs (id\_donateur, id\_user, nom\_entreprise, adresse, téléphone)
* **Beneficiaires**  
  beneficiaires (id\_beneficiaire, id\_user, nom, adresse, zone\_intervention)
* **Dons**  
  dons (id\_don, id\_donateur, type\_aliment, quantité, date\_limite, adresse\_collecte, statut)
* **Administrateurs**

admins (id\_admin, nom, email)

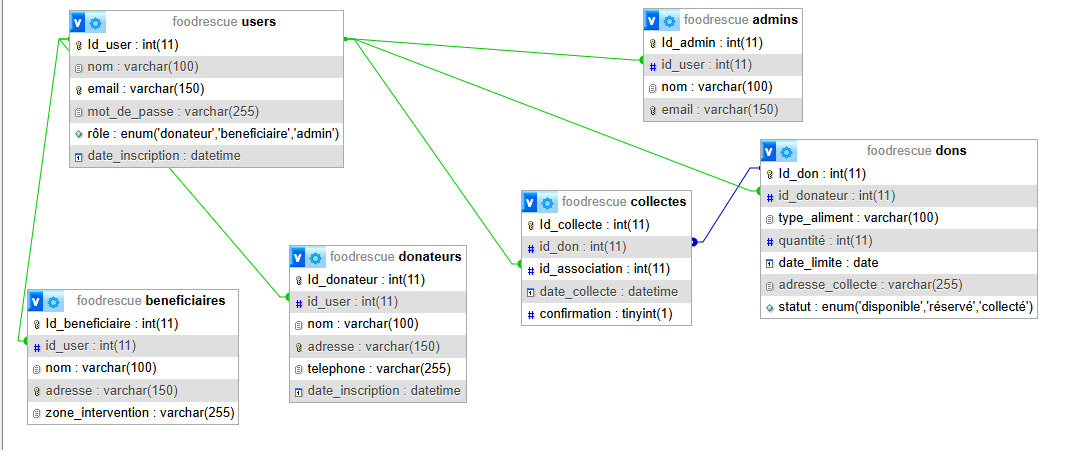
* **Collectes et Réservations**  
  collectes (id, id\_don, id\_association, date\_collecte, confirmation)
* **Modèle conceptuel de données (MCD)**

****

* **Modèle logique de données (MLD)**

****

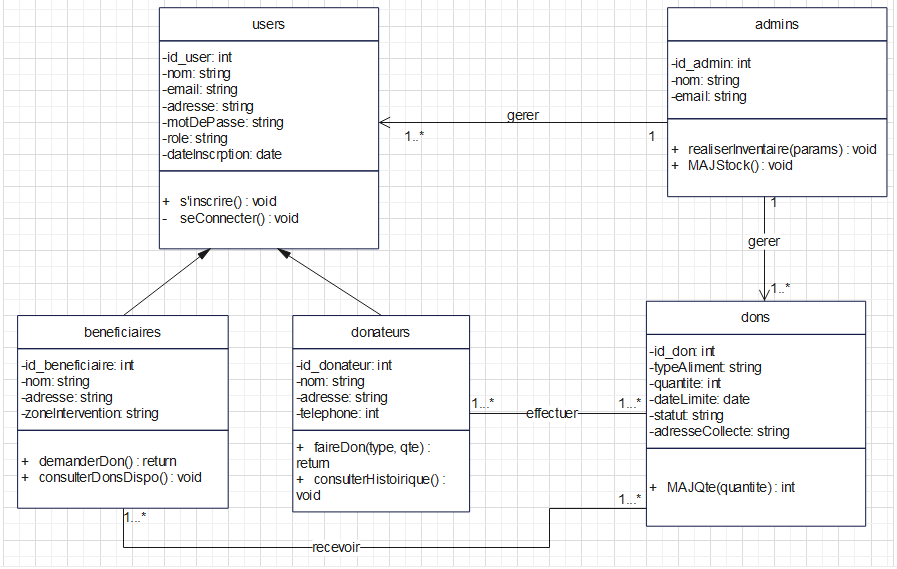
* **Schéma relationnel de la base de données**

****

1. **Diagrammes**

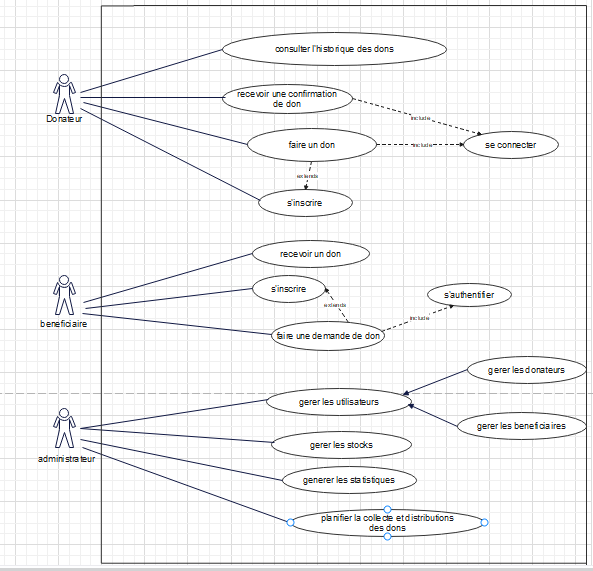
* **Diagramme de classes**

Il présente les différentes classes de notre système ainsi que les interactions entre ces dernières. Pour le cas de notre système on distingue les classes admins pour les administrateurs du site, dons pour les différents dons effectues, donateurs et bénéficiaires qui héritent toutes les deux de la classe users présentant les différents utilisateurs du site FOODHELPER.

****

* **Diagramme de cas d’utilisation**

Ils sont utilisés pour une représentation du comportement fonctionnelle d’un système et décrivent les fonctions générales et la portée de ce dernier ; Dans le cas du site FOODHELPER on distingue plusieurs cas d’utilisations pour les différents acteurs de notre système qui sont les admins les donateurs et les bénéficiaires.

****

**Conclusion**

Cette documentation technique offre une vue d’ensemble détaillée de l’architecture et des choix techniques du site FoodHelper. Elle constitue une référence pour l’équipe de développement et les administrateurs système, afin d’assurer le développement, le déploiement, la maintenance et l’évolution de la plateforme dans des conditions optimales.