Titre professionnel Développeur-se logiciel - Niveau III

Dossier de projet

Sos Suicide Phenix

JuLIEN FRERET

Résumé de projet

Site pour SOS Suicide Phénix le Havre

Mon projet professionnel ce fait pour l'association SOS suicide situé au Havre.

N'ayant aucun site leur étant dédié je leur ai proposé de faire mon projet pour eux afin qu'il puisse bénéficier de visibilité.

Le site internet aura comme fonctionnalité de pouvoir se connecter via une page utilisateur qui ne sera pas visible par l'utilisateur mais seulement accessible pour les bénévoles. On ne veux pas que l'utilisateur soit perturbé par des données qui lui ai pas concerné car cela serais plus de la pollution visuelle qu'autre chose.

Depuis le compte administrateur que seule la présidente de l'association détiendra, il sera possible d'ajouter des utilisateurs en fonction des bénévoles, de supprimer des utilisateurs ou de modifier des utilisateurs. Seul l’utilisateur ADMIN ne pourra pas être supprimer.

Une fois connecté sur le site, l'idée est de faire véhiculer leurs actualités, donc il y aura une rubrique actualité où il sera possible d'ajouter / modifier / supprimer une actualité et ceux si disponible depuis n'importe quel compte. L'idée est de faire passer des événements / actualités pour que les utilisateurs concerné puisse voir et suivre l’événement / actualité à sa guise.

Table des matières

[1.Cahier des charges 3](#_Toc1727650)

[1.1 Descriptif de la demande 3](#_Toc1727651)

[1.2 Architecture technique 3](#_Toc1727652)

[1.3 Spécifications fonctionnelles 4](#_Toc1727653)

[a) Description 4](#_Toc1727654)

[b) Utilisation 4](#_Toc1727655)

[1.4 Spécifications techniques 5](#_Toc1727656)

[a) Langages utilisés 5](#_Toc1727657)

[b) Bibliothèques et frameworks utilisés 5](#_Toc1727658)

[c) Outils utilisés 6](#_Toc1727659)

[d) Sécurité des systèmes d’information : Recommandations et bonnes pratiques 6](#_Toc1727660)

[2. Compétences du référentiel couvertes 7](#_Toc1727661)

[2.1 Tableau récapitulatif 7](#_Toc1727662)

[2.2 Maquetter une application 8](#_Toc1727663)

[2.3 Réaliser une interface utilisateur web statique et adaptable 8](#_Toc1727664)

[2.4 Développer une interface utilisateur web dynamique 9](#_Toc1727665)

[2.5 Développer une interface utilisateur avec une solution de gestion de contenu 10](#_Toc1727666)

[2.6 Créer une base de données 10](#_Toc1727667)

[2.7 Développer des composants d’accès aux données 12](#_Toc1727668)

[2.8 Développer la partie back-end d’une application web ou web mobile 13](#_Toc1727669)

[3. Réalisation 16](#_Toc1727670)

[3.1 Création d’une actualité 16](#_Toc1727671)

[3.2 Affichage d’une actualité 17](#_Toc1727672)

[3.3 Modification d’une actualité 19](#_Toc1727673)

[3.4 Suppression d’une actualité 22](#_Toc1727674)

[4. Anglais de rigueur : Une communauté importante orientée vers le partage 23](#_Toc1727675)

[5. Conclusion 23](#_Toc1727676)

# 1.Cahier des charges

## 1.1 Descriptif

L’association Sos suicide phénix du havre n’ayant pas de site internet leur étant dédié je me suis proposé pour leur en créer un. Pour des raisons évidentes de visibilité, d’accessibilité et de confort.

L’idée est de pouvoir créer des actualités de pouvoir les modifier ou supprimer a ça guise.

## 1.2 Architecture technique

L’application doit être accessible sur tous les types de navigateurs. Elle est principalement destinée à être déployée sur un serveur WAMP.

Le développement utilisera des langages orientés WEB : HTML, CSS, JavaScript et PHP. Il pourra également contenir un FRAMEWORK tel que Bootstrap.

L’application sera responsive et structuré d’après l’architecture de développement Modèle-Vue-Contrôleur.

Les données seront stockées sur une base de données MySQL.

## 1.3 Spécifications fonctionnelles

### a) Description

Un utilisateur (utilisateurs) est défini par :

* Un nom
* Un prénom
* Une adresse mail
* Un numéro de téléphone
* Un mot de passe
* Un Rôle (ex : admin)

Une actualité (actualite) est défini par :

* Un titre
* Une description
* Une image (facultative)

Un Role (role) est défini par :

* Un rôle

### b) Utilisation

L’utilisation de l’application respecte une hiérarchie de 2 niveaux :

* Présidente de l’association (Admin)
* Les bénévoles (user)

Création de d’actualité

Le site permet d’ajouter une actualité, de pouvoir la voir, la modifier et la supprimer.

Affichages des informations

Chaque utilisateur pourra consulter leurs informations renseignées lors de la création leur compte.

Et seul l’admin pourra supprimer un utilisateur.

Modification des informations du compte utilisateur

Selon les différents profils, certaines modifications seront réalisables de manière autonome. Ex. Changement de nom, de prénom, date de naissance … Néanmoins, un changement de profil ne pourra être effectué que par un responsable (Admin).

Suppression du compte utilisateur

L’application permettra de supprimer des utilisateurs mais uniquement via un compte admin. Le but est de contrôler les comptes utilisateurs.

Création / Affichage / Modification / Suppression d’une actualite

Seul la Président de l’association (Admin) aura la main sur la suppression des articles.

## 1.4 Spécifications techniques

### Langages utilisés

|  |  |
| --- | --- |
| Appellation | Description |
| HTML 5 | HTML5 (HyperText Markup Language 5) est la dernière révision majeure du HTML. C’est un format de données conçu pour représenter les pages web. C’est un langage de balisage permettant de l’hypertexte. |
| CSS 3 | Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. |
| PHP 7 | PHP: Hypertext Preprocessor, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. |
| JavaScript | JavaScript est un langage événementiel de programmation de scripts, principalement employé dans les pages web interactives. |
| MySQL | Langage de requête fonctionnant avec le système de gestion des bases de données relationnelles. Il permet la génération de la base de donnée, la lecture et la modification de ses tables. |

### Bibliothèques et frameworks utilisés

|  |  |
| --- | --- |
| Appellation | Description |
| Bootstrap | Bootstrap est un framework front-end . Son système de grille permet de rendre aisément une application web responsive. |

### Outils utilisés

|  |  |
| --- | --- |
| Appellation | Description |
| VS Code | Vs code est un environnement de développement |
| Git | Git est un logiciel de gestion de versions décentralisé. Il permet de travailler avec des services d’hébergement spécifiques tel que GitHub et BitBucket. |
| phpMyAdmin | phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP. |
| JMerise | Logiciel d’aide à la conception des bases de données (Modèle Conceptuel de Données) qui permet la génération d’un Modèle Logique de Données et d’un script SQL pour plus de facilité. |

### Sécurité des systèmes d’information : Recommandations et bonnes pratiques

* Description de la veille techno = Alertes ANSSI
* Bonnes pratiques sur votre site/appli web, savoir les retrouver, les comprendre et les appliquer grâce aux docs de ANSSI : ex. <https://www.ssi.gouv.fr/entreprise/bonnes-pratiques/>
* Ne pas hésiter à parler de : <https://openclassrooms.com/fr/courses/2091901-protegez-vous-efficacement-contre-les-failles-web/2680162-nayez-jamais-confiance>
* Intégration de regex
* Utilisation de PDO qui permet des requête SQL plus sécurisées.
* Htmlspecialchars, qui permet d’éviter l’injection de script.

# 2. Compétences du référentiel couvertes

## 2.1 Tableau récapitulatif



Légendes :

|  |
| --- |
| Compétences couvertes |
| Compétences non couvertes |

## 2.2 Maquetter une application

Une fois le cahier des charges défini, j’ai commencé mon projet par l’élaboration d’une maquette qui me permettait d’avoir un visuel des différents éléments et des différentes fonctions de l’application. Le but de mon maquettage était de matérialiser les différentes idées et avoir un fil conducteur tant sur le plan visuel que conceptuel.

J’ai alors utilisé le logiciel « Adobe xD » qui permet des créer rapidement une interface et de créer une rapide navigations de la future application.

Version maquettée sous Adobe xD



Version de l’interface dans l’application web



## 2.3 Réaliser une interface utilisateur web statique et adaptable

Afin de gagner de la mobilité, j’ai fait en sorte que mon application soit responsive en m’appuyant sur un framework appelé Bootstrap. Il m’a permis d’utiliser un système de GRID pour une meilleure structure ainsi que des classes spéciales pour que mon application puisse s’adapter parfaitement aux différents supports et tailles d’écran.

Ex : Vue en grande résolution

**

Vue sur la résolution d’un mobile

*Une image contenant texte

Description générée automatiquement*

## 2.4 Développer une interface utilisateur web dynamique

* Utilisation des modules JS et JQuery du FrameWorks  Bootstrap

## 2.5 Développer une interface utilisateur avec une solution de gestion de contenu

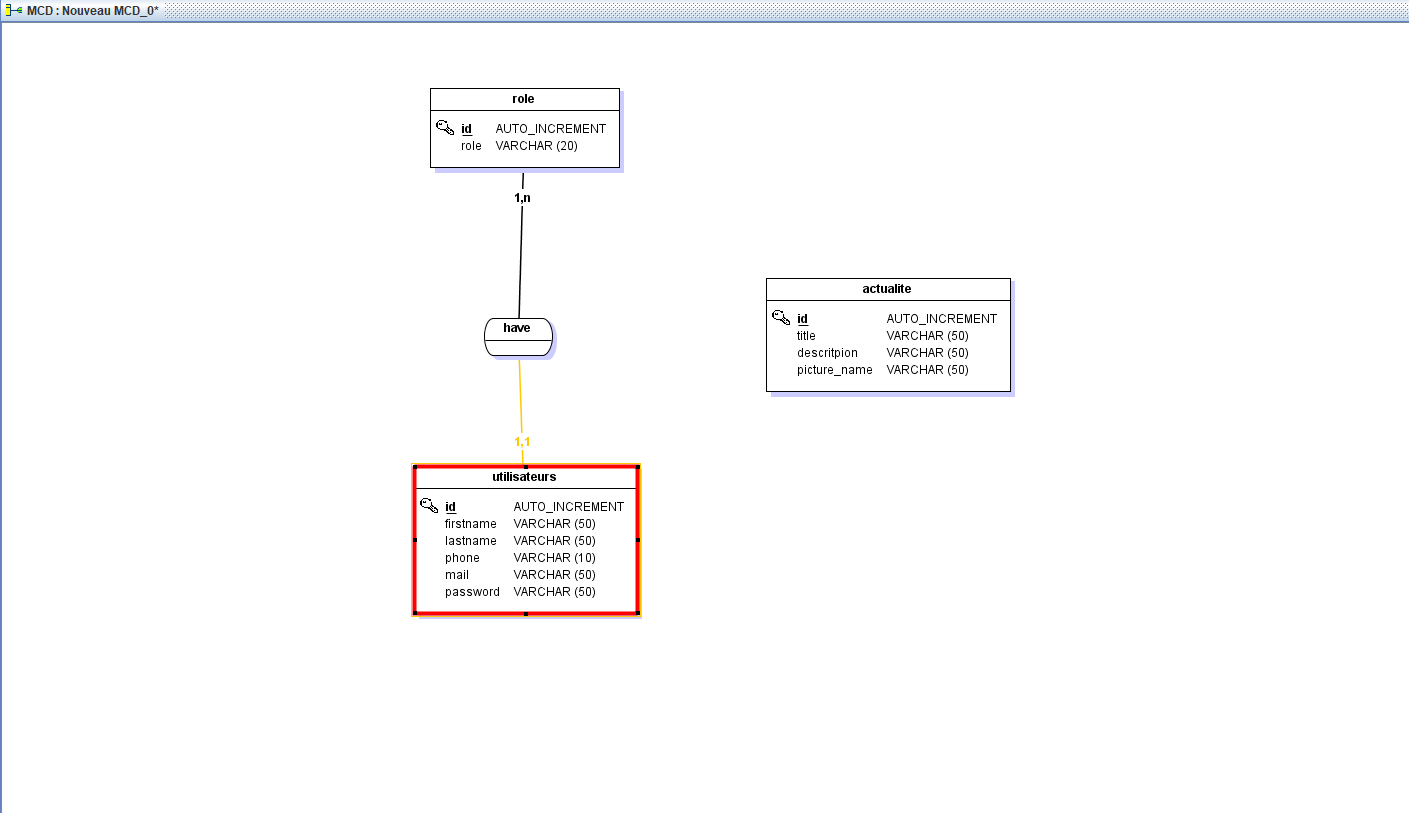
Ayant opté pour une architecture Modèle-Vue-Contrôleur (MVC), une interface utilisateur était indispensable pour afficher les données de mes modèles via mes différents contrôleurs. Ce type d’architecture permet également de faciliter la Programmation Orientée Objet (POO).

## 2.6 Créer une base de données

Une fois les différentes fonctionnalités définies et le maquettage terminé, il a été primordiale d’établir la base de données car c’est elle qui stockera toutes les données de l’applications.

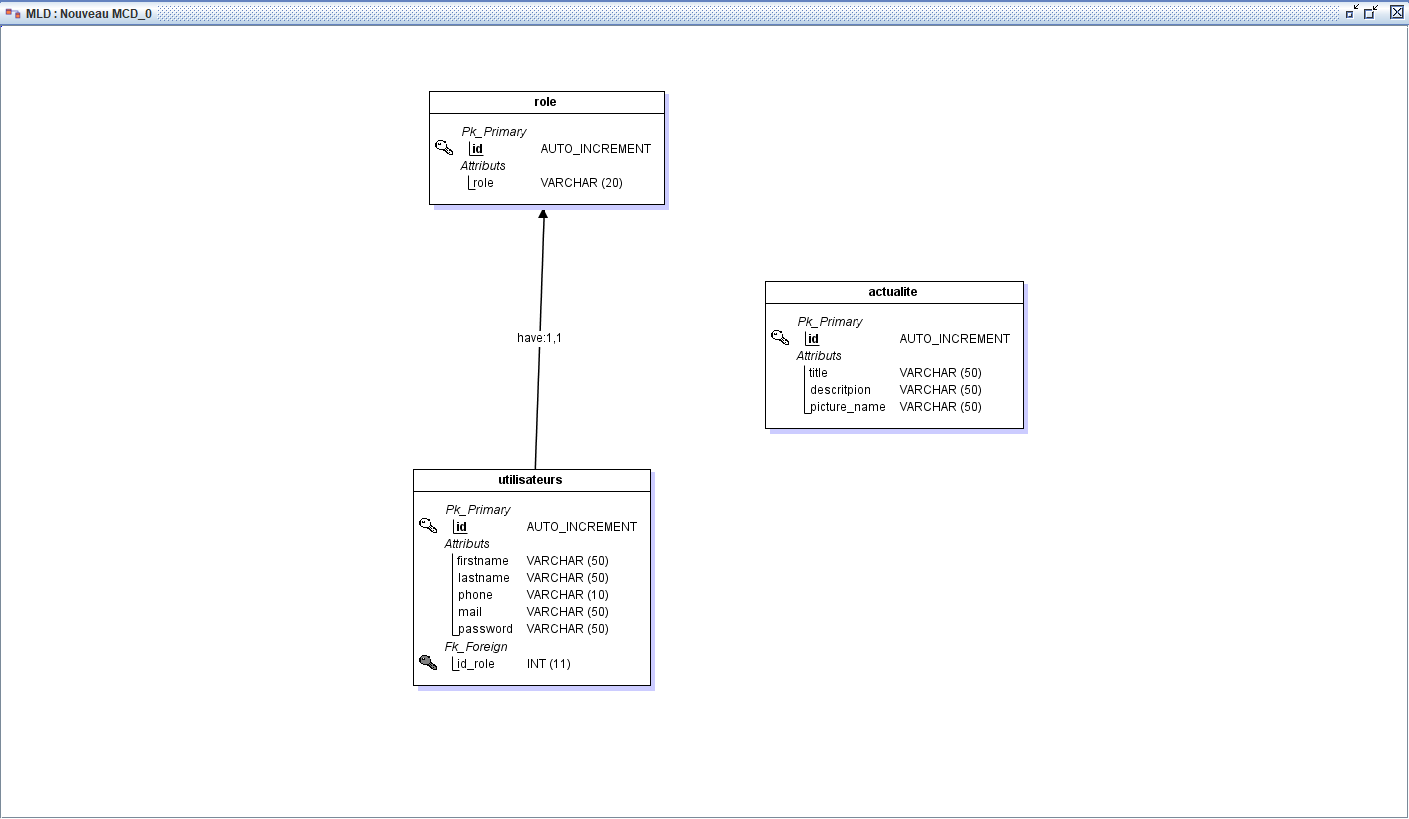
J’ai donc utilisé le logiciel JMerise pour modéliser mes entités, mes champs ainsi que mes cardinalités. Dans un premier temps, j’ai obtenu un MCD.

MCD :

**

Puis dans un second temps, une fois mon MCD modélisé, j’ai utilisé une fonctionnalité de JMerise pour générer automatiquement mon MLD : Modèle Logique de Données. L’outil se base sur les clefs primaires et les cardinalités renseignées pour créer les clefs étrangères dans les différentes tables.

MLD :



## 2.7 Développer des composants d’accès aux données

Dans une architecture MVC, chaque modèle représente une table de la base de données. Ils doivent pouvoir s’y connecter à chaque exécution d’une méthode afin d’accéder aux différentes données.

J’ai donc créé une classe spécifique regroupant tous les éléments nécessaires pour se connecter à la base de données.

|  |
| --- |
| Une image contenant texte  Description générée automatiquement |

La classe « database » sera commune à toutes les autres classes nécessitant une connexion à la base de données, ex : « class Actualite »

|  |
| --- |
| Une image contenant texte  Description générée automatiquement |

## 2.8 Développer la partie back-end d’une application web ou web mobile

Comme indiqué précédemment, dans une architecture MVC, un Modèle correspond à une table de la base de données. Dans notre application, les Modèles utilisent la classe « database » pour se connecter et disposent de diverses méthodes pour créer, lire, modifier et supprimer les données respectives.

Pour pouvoir les utiliser, les Modèles sont directement instanciés dans les Contrôleurs des différentes Vue.

En effet, les Vues permettent d’afficher des données et interagir avec l’utilisateur. Pour cela, chaque Vue dispose d’un contrôleur respectif.

Ex : Affichage d’une actualité selon les données de la table actualite.

**Vue** : On inclut le Contrôleur via un « require »

|  |
| --- |
| Une image contenant texte  Description générée automatiquement  Une image contenant texte  Description générée automatiquement |

**Contrôleur** : J’instancie un nouvel objet à l’aide du Modèle respectif et j’appelle les méthodes, ici je l’appelle la méthode « showThisActualite($id). »

|  |
| --- |
| Une image contenant texte  Description générée automatiquement |

**Modèle** : Connexion à la base de données via la classe « database » + création d’une méthode pour effectuer une requête dans la table « actualite »

|  |
| --- |
| Une image contenant texte  Description générée automatiquement  *Une image contenant texte  Description générée automatiquement* |

# 3. Réalisation

## 3.1 Création d’une actualité

Création d’une Vue avec un formulaire pour recueillir les différentes informations



Création d’un contrôleur pour effectuer l’enregistrement des inputs en effectuant des contrôles à de conditions et de regex

|  |
| --- |
| Une image contenant texte  Description générée automatiquement |

Dans le modèle de la table « actualite », je crée une méthode pour ajouter une actualité une fois toutes les informations de validées

|  |
| --- |
| Models :  Une image contenant texte  Description générée automatiquement  Controller :  Une image contenant texte  Description générée automatiquement |
|  |

## 3.2 Affichage d’une actualité

Dans la Vue correspondante, je crée une liste pour pouvoir afficher toutes les actualités de créés, l’utilisateur aura également la possibilité d’afficher une actualité en particulier

Une image contenant texte

Description générée automatiquement

J’ai créé une méthode pour récupérer l’id de l’actualité afin d’afficher uniquement l’actualité sélectionné

|  |
| --- |
| Model :  Une image contenant texte  Description générée automatiquement  Controller :    Views :  Une image contenant texte  Description générée automatiquement |

## 3.3 Modification d’une actualité

Dans la Vue respective, je crée un formulaire pour afficher les données préalablement renseignées et je permets à l’utilisateur de modifier celles-ci pour qu’il puisse effectuer une modification / mise à jour.

Une image contenant texte

Description générée automatiquement

Je crée un contrôleur pour enregistrer les différentes interactions avec l’utilisateur (Modification, envoi de formulaire, etc. …) et je m’assure que les informations respectent les prérequis (ex. Regex) avant d’utiliser une méthode pour mettre à jour la table respective (les mêmes sécurités que pour la création d’une actualitée ).

|  |
| --- |
| Une image contenant texte  Description générée automatiquement  Une image contenant texte  Description générée automatiquement |

Dans le modèle correspondant, je crée une méthode pour mettre à jour selon l’id de l’article.

|  |
| --- |
| Une image contenant texte  Description générée automatiquement |

## 3.4 Suppression d’une actualité

Pour effectuer la suppression d’une actualité, j’ai créer une méthode qui cible l’article grâce à son id dans la variable $thisActu.

Une image contenant texte

Description générée automatiquement

Dans le contrôleur de la Vue, je récupère l’id dans la variable et j’exécute ma méthode

|  |
| --- |
| Une image contenant texte  Description générée automatiquement |

# 4. Anglais de rigueur : Une communauté importante orientée vers le partage

* Description de la situation de travail nécessitant une recherche : FRONT / BACK
* Extrait du site Anglophone et la traduction faite par vos soins



Traduction : Utilisez flex-column pour définir une direction verticale ou .flex-column-reverse pour inverser l’ordre.

# 5. Conclusion

A travers ce projet, j’ai pu me rendre compte que de développer une application demande du temps et un très fort investissement que ce soit de la conception, à la réalisation et aux séances de débogage.

Je vous remercie de m’avoir lu et écouté.

Merci.