

Cahier Technique

Projet de fin d'année Web1 P2019

Groupe #15

1Table des matières

1Table des matières	1
2 Introduction	3
2.1 Objectifs	3
2.2 Périmètre	3
2.3 Définitions, acronymes, glossaire	3
2.4 Références	3
2.5Vue d'ensemble	3
3 Description d'ensemble	3
3.1Choix techniques	3
3.1.1 Base de données	3
3.1.2 Solution back-end	3
3.1.3 Solution front-end	3
3.2 Dépendances	3
4 Exigences spécifiques	3
4.1Cas d'utilisations	3
4.2Exigences supplémentaires	3
5 Base de données	3
5.1 Définition des entités	3

5.2 Modélisation	3
5.3 Projection de volumétrie	3
6 Architecture technique	3
6.1 Classes	3
6.2 Diagramme de classes	3
6.3 Interfaces externes	3
7 Sécurité	4
7.1 Étude des risques	4
7.2 Solutions	4
8 Installation et déploiement	4
8.1 Installation	4
8.2 Déploiement	4
9 Plan de reprise d'activité	4

2 Introduction

2.1 Objectifs

Nous avons pour objectif de faire connaître Yalla. La communication est donc la priorité. Mais aussi la refonte en terme de design et en terme de technologie. Nous voulons donner un nouveau souffle à Yalla. Permettre à Yalla d'être plus visible, c'est aussi contribuer à l'amélioration de vie des enfants et plus généralement des réfugiés Syriens au Liban.

2.2 Périmètre

Nous avons en tête de donnée une base de donnée solide avec Mysql au site Yalla.com

Accompagné de laravel vue. HTML5, CSS3, Bootstrap 3.3, et JS pour le front-end.

Les fonctionnalités sont nombreuses et nécessaires. Nous allons faire en sorte que chaque utilisateur ait une expérience du site qui lui ait propre. Particulièrement pour les parrains.

Nous allons donc mettre en place un espace parrains/marraines avec un historique des parrainages de l'utilisateur et un suivie de ce que son parrainage à permis d'apporter au parrainé, en vue de l'installation du système de parrainage par la société Yalla courant 2017.

Le soutiens par le don souffre cruellement d'une interface UX, nous préférons mettre en place une interface qui permet de faire un don directement sur le site et de confirmer son don sur le site-lui-même : un service de paiement sécurisé est donc nécessaire pour accompagner cette démarche.

Nous allons avoir recours à une API Twitters pour laisser les utilisateurs de compte twitter, laisser des commentaires de don ou d'actualités. pour cela nous envisageons de mettre en place un nouvel #don4Yalla & #YallaActualité

Un panier pour lister tous les dons qu'on a eu lors de la navigation.

Nous allons aussi essayer de traduire le site (Anglais, Arabe)

Nous aurons aussi pour fonction une newsletter avec envoie des dernières actualités.

éléments intégrer	Fonctionnalités
parrainage	système de parrainage
commentaire et # twitter	API Twitter
langues	traduction en anglais et arabe
historique	listing de tous les dons
paiement par l'interface d'une api	stripe
envoie de mail	newsletter
paiement par sms & email	inscription paytweak

2.3 Définitions, acronymes, glossaire

- Ajoutées
- Supprimées
- Modifiées
- Listées
- Visualisées
- Liées entre elles (si hiérarchique)
- Affectées à un contenu
- Stylisées

2.4 Références

2.5 Vue d'ensemble

3 Description d'ensemble

3.1 Choix techniques

3.1.1 Base de données

Nous allons opter pour une base de données gratuite pour l'association Yalla. Mysql est sans hésiter un bon choix pour notre base de donnée tant la prise en main pour les héritiers du site sera facile. Par ailleurs toutes les fonctionnalités que nous voulons exécutées sont disponibles sur Mysql.

3.1.2 Solution back-end

Pour le Back-end, nous nous penchons sur le framework **Laravel** qui est un framework web open-source écrit entièrement en php en programmation orientée objet

3.1.3 Solution front-end

Le HTML5, CSS3, Sass, Javascript native, vue.js, en plus de plugins. Et bootstrap seront d'excellentes technologies pour la mise en place du site et le front-end du site. Nous pensons que ces technologie suffisent largement pour éditer un site conforme et navigable.

3.2 Dépendances

Nous allons utiliser comme dépendances après installation de composer, de axios, cross-env, laravel-mix, lodash, vue.

4 Exigences spécifiques

4.1 Cas d'utilisations

4.2 Exigences supplémentaires

5 Base de données

5.1 Définition des entités

5.2 Modélisation

5.3 Projection de volumétrie

6 Architecture technique

6.1 Classes

Article

Administrateur

Utilisateur

6.2 Diagramme de classes

Utilisateur
Nom
Mot de passe

Administrateur
Nom
Mot de passe

Articles
Titre
Meta description
Meta robot
Slug
résumé
Corps
Image titre
Média
SEO

6.3 Interfaces externes

stripe

7 Sécurité

7.1 Étude des risques

Nous n'avons pas le choix d'utiliser un système de paiement en ligne directement sur le site et sécuriser à 100% les données bancaires.

Risque d'intrusion dans le code php avec la méthode GET, et la barre d'url est visible.

En cas de panne de serveur.

7.2 Solutions

Pour les paiements, nous allons opter pour un paiement avec une entreprise externe qui gère les paiement en ligne avec sms et email.

Pour palier au difficultés suite à un serveur qui plante, nous envisageons de sauvegarder à chaque actualisation des informations du site sur un compte github dédié. Ce compte github sera nourrit en fonction des ajouts et des demandes que fait Yalla à ses développeurs.

8 Installation et déploiement

8.1 Installation

8.2 Déploiement

Le déploiement s'effectue avec leur nom de domaine : www.yalla-enfants.com
on garde le même nom de domaine.

9 Plan de reprise d'activité

Pour la reprise d'activité on prévoit de faire des sauvegardes à chaque actualisation de la base de donnée du site et aussi de laisser une sauvegarde sur le gitHub. On passera l'ensemble du code à travers github, aux développeurs de l'association Yalla.

Charte Graphique