

Projet Festival



BLAGNAC



Documentation Administrateur V2

Rédigé le 20/09/2021 par LLORET Léo
Dernière mise à jour : le 21/01/2022

Projet de PTUT : Questionnaire

Réalisé par :

- ETCHEVERRIA Maxime
- LAFONTANT Yanis
- LE DEVEDEC Maxime
- LLORET Léo
- SERVANT Geoffrey

Sommaire

Documentation Administrateur V2	1
Sommaire	2
Introduction	3
Installation du logiciel	3
Prérequis	3
Installation	4
Vérification	4
Connexions	5
Pour l'Application mobile	5
Fonctionnement du logiciel	6
Diagramme de classe	6
Diagramme des cas d'utilisation	7
Code	8
Base de données	8
Structure de la base	8
Code de génération de la base	9
Application mobile	10
Architecture	10
Questionnaire Flutter	10
Connexion Base de Données	11
Page web	11
Architecture	11
Requêtes	13
HTML & PHP	13

I. Introduction

Ce document est la documentation administrateur officielle de la partie Questionnaire.

L'application permet de gérer des questionnaires et comprend deux rôles : administrateur et bénévole. Ainsi, un administrateur peut créer, modifier et supprimer des questions et des réponses et visualiser les statistiques générées par celui-ci, au travers d'une page web. Quant aux bénévoles, ils peuvent consulter les questions et y répondre. Toutes les données seront stockées dans une base de données.

Pour cela l'application disposera d'une interface graphique en Dart développée avec Flutter et Android Studio qui sera reliée à une base de données Mysql reliée elle-même à une page web en HTML, CSS et PHP.

II. Installation du logiciel

A. Prérequis

Pour le bon fonctionnement de l'application il est recommandé d'avoir les réglages suivants :

- Avoir télécharger le build et le transférer sur un smartphone (Android uniquement).

Lien du GitHub : <https://github.com/geoffrey66280/FlutterappFestival>

- **Profil** : disposer d'un identifiant et d'un mot de passe afin de se connecter à l'application ou à la page web (administrateur seulement) en tant que bénévole ou administrateur.
- **Logiciels Utilisés et Versions** :
 - Android Studio v4.2.2
 - Flutter v2.8.0
 - WinSCP v5.19.5
 - MySQL v8.0.26
 - PhpStorm v2021.3
 - VsCode v1.63.2

Pour le serveur :

- Ubuntu server 20.4.3
- Apache 2.4.41
- Php 7.4

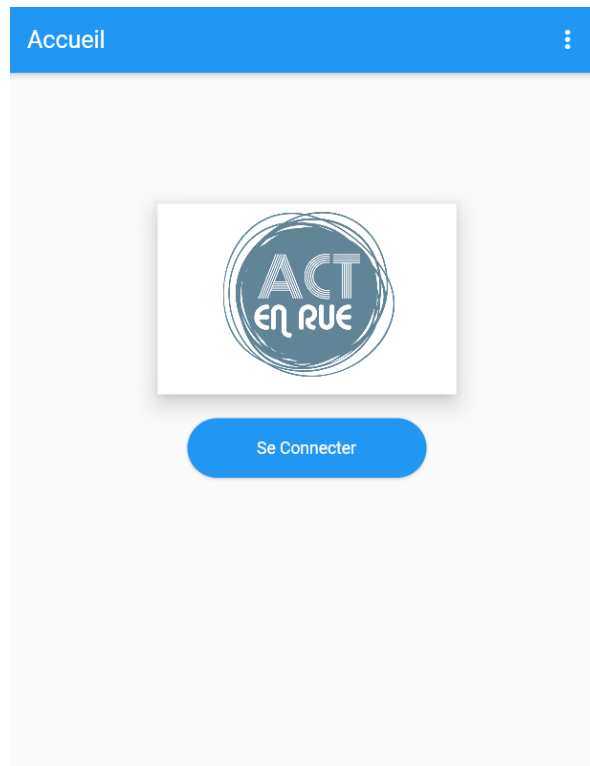
B. Installation

Depuis Github, aller dans les "releases" et télécharger le fichier .apk, une fois téléchargé, transférez-le dans un smartphone. Dans votre smartphone, chercher dans vos dossiers le fichier .apk et cliquez pour l'installer. Une fois installé, l'application est prête à l'emploi.

C. Vérification

Pour s'assurer que l'installation s'est bien réalisée, une icône devrait apparaître sur votre smartphone, quand vous la lancez, vous devriez atterrir sur la page d'accueil.

Page d'accueil

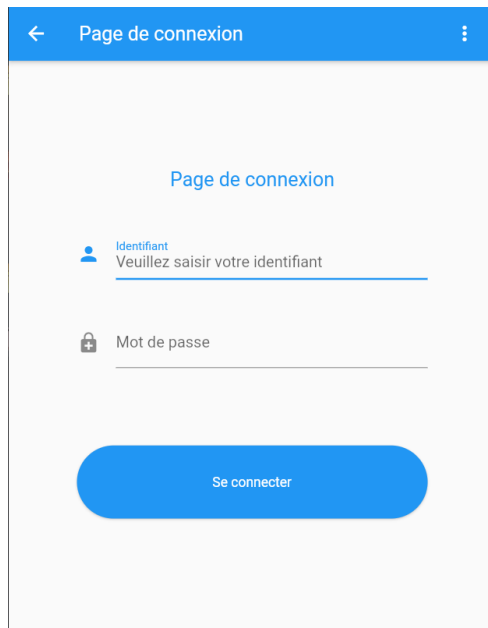


III. Connexions

A. Pour l'Application mobile

Une fois l'application installée sur votre smartphone et lancer celle-ci, cette page s'affichera.

Page de connexion :

The screenshot shows a mobile application interface for a login page. At the top, there is a blue header bar with a back arrow on the left, the text "Page de connexion" in the center, and a menu icon on the right. Below the header, the page has a light gray background. In the center, the text "Page de connexion" is displayed in blue. There are two input fields: the first is labeled "Identifiant" with a person icon and the text "Veuillez saisir votre identifiant"; the second is labeled "Mot de passe" with a lock icon. Below these fields is a large blue button with the text "Se connecter".

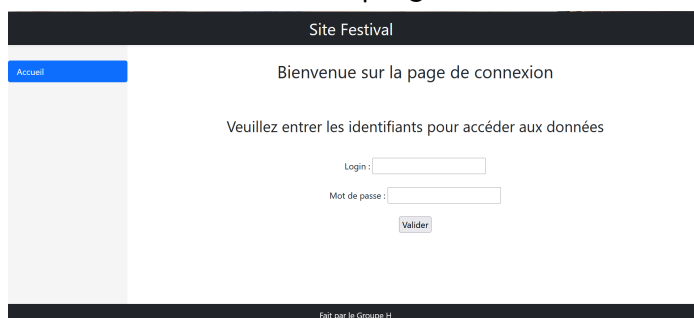
Pour vous connecter, utilisez l'identifiant "geoff" et le mot de passe "admin" puis appuyer sur le bouton "Se connecter".

B. Pour le site web

Pour vous accéder au site web copier le lien suivant :

<https://questionnaires.iut-blagnac.fr/>

Vous atterrirez sur la page suivante :

The screenshot shows a web application interface for a login page. At the top, there is a dark gray header bar with the text "Site Festival". Below the header, there is a blue navigation bar with the text "Accueil". The main content area has a light gray background. In the center, the text "Bienvenue sur la page de connexion" is displayed. Below this, the text "Veuillez entrer les identifiants pour accéder aux données" is shown. There are two input fields: the first is labeled "Login:" and the second is labeled "Mot de passe:". Below these fields is a button labeled "Valider". At the bottom of the page, there is a dark gray footer bar with the text "Fait par le Groupe H".

Pour vous connecter, entrez l'identifiant "admin" et le mot de passe "admin" puis cliquez sur le bouton "Valider".

C. Pour la base de donnée

L'adresse pour se connecter à la base de donnée en SQL est :

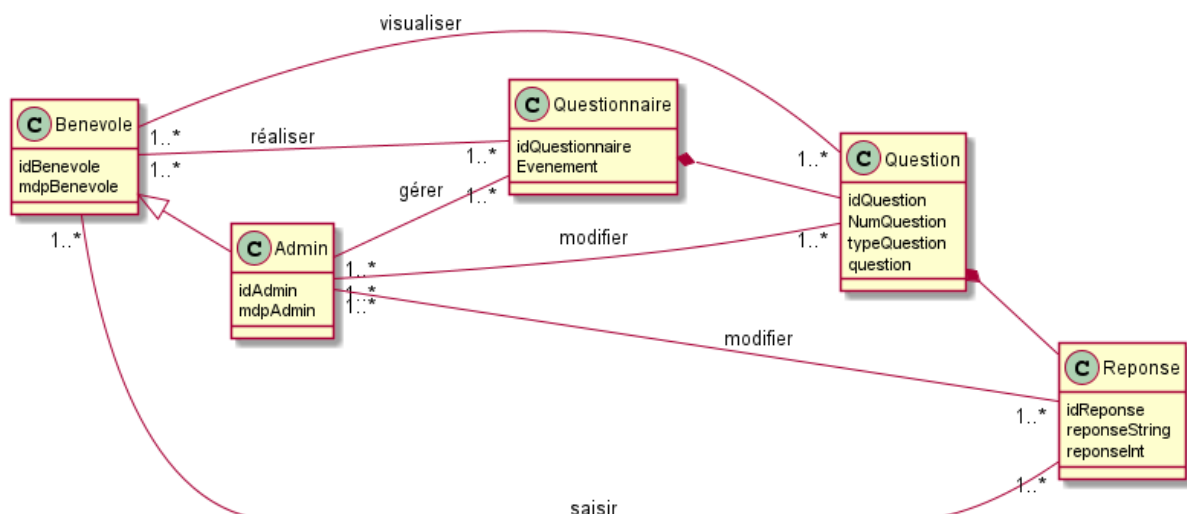
<http://193.54.227.216/phpmyadmin/>

La base dédiée à ce projet est : user_ptut, l'identifiant de connexion est : user_ptut et le mot de passe est : ZQSVEAbwsxpB7eth .



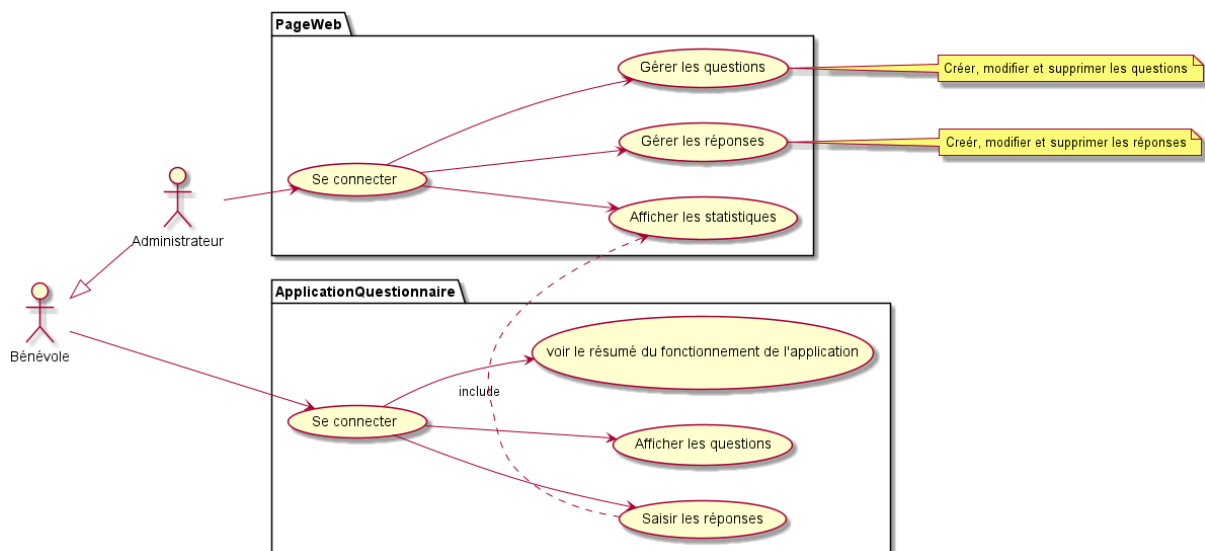
IV. Fonctionnement du logiciel

A. Diagramme de classe



Un administrateur possède un identifiant et mot de passe ainsi, il peut se connecter pour gérer un questionnaire qui est composé d'un identifiant et d'un nom. Un administrateur hérite des droits d'un bénévole qui avec un identifiant et un mot de passe peut visualiser une question et saisir une réponse. Un questionnaire est composé de questions disposant d'un identifiant, d'un numéro et d'un intitulé. Une question est quant à elle composée d'une réponse qui possède un identifiant, un numéro ou un intitulé.

B. Diagramme des cas d'utilisation



Cette application possède 2 rôles :

- Administrateur qui, depuis une page web, à la possibilité de créer un questionnaire en créant des questions et des réponses et qui, par la suite, à la possibilité de gérer ses questions et réponses soit de modifier celles-ci ou de les supprimer. De plus, il a la possibilité d'afficher les statistiques générées à partir des réponses et hérite des droits d'un bénévole.
- Bénévole qui, depuis l'application, peut accéder au questionnaire et visualiser les questions puis saisir la réponse. Il peut aussi voir le résumé du fonctionnement de l'application.

V. Code

A. Base de données

1) Structure de la base

Table user

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	login 🔑	varchar(15)	utf8mb4_0900_ai_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 2	password	varchar(15)	utf8mb4_0900_ai_ci		Non	Aucun(e)			Modifier Supprimer Plus

La table User permet d'enregistrer les identifiants et les mots de passe des utilisateurs de l'application et du site web, soit, ceux des bénévoles et de l'administrateur.

Table reponseRepondant

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	idResponseRepondant 🔑	mediumint			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	idQuestion 🗣️	mediumint			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	repNum	int			Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/> 4	repString	text	utf8mb4_0900_ai_ci		Oui				Modifier Supprimer Plus
<input type="checkbox"/> 5	idRepondant 🗣️	mediumint			Non	Aucun(e)			Modifier Supprimer Plus

Cette table sert à enregistrer les réponses saisies par les bénévoles. Elles contiennent l'identifiant de la question associée et le nombre répondant à la question ou le texte saisi selon le type de la réponse.

Table repondant

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	idRepondant 🔑	mediumint			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	idEvenement 🗣️	mediumint			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	idQuestion	mediumint			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 4	dateHeure	datetime			Non	CURRENT_TIMESTAMP		DEFAULT_GENERATED	Modifier Supprimer Plus

La table repondant permet d'enregistrer le festivalier qui a répondu au questionnaire. Il est alors rattaché à un événement, ici le festival du rire et on enregistre l'heure où il a répondu aux questions.

Table questionnaire

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	idQuestion	mediumint			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	idEvenement	mediumint			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	numero	int			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 4	libelle	varchar(25)	utf8mb4_0900_ai_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 5	type	enum('slide', 'check', 'combo', 'string', 'int')	utf8mb4_0900_ai_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 6	min	int			Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/> 7	max	int			Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/> 8	listeReps	text	utf8mb4_0900_ai_ci		Oui				Modifier Supprimer Plus

Cette table permet d'enregistrer les questions d'un questionnaire associé, qui est attaché à un événement (ici, la Mekanik du rire). Elle comprend le numéro de la question, son libellé, le type de celle-ci (texte, nombre, checkbox, liste ...), le nombre minimum et maximum entre le quel est compris la réponse et la liste des réponses.

Table questionnaireEvenement

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	idEvenement	mediumint			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	nom	varchar(15)	utf8mb4_0900_ai_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	dateD	date			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 4	dateF	date			Non	Aucun(e)			Modifier Supprimer Plus

Cette table permet de créer plusieurs événements, comme par exemple celui du festival de la Mekanik du rire en enregistrant son nom, la date de début et celle de fin.

2) Code de génération de la base

Le code se trouve dans le fichier [codeBD.sql](#).

B. Application mobile

1) Architecture

Voici le rôle de chaque fichiers .dart de l'application mobile :

- `a_propos.dart` : donne des informations sur l'application notamment les droits
- `accueil.dart` : accueil de l'application avec un bouton qui redirige vers la page seconnecter
- `login.dart` : page qui contient les méthodes de récupération et d'envoi HTTP (get; post)
- `main` : permet de lancer l'application et d'afficher l'accueil
- `methodes.dart` : méthodes servant à créer quelques éléments du questionnaire comme un conteneur de questionnaire par exemple
- `questionnaire.dart` : questionnaire de l'application regroupant des questions sous forme de checkbox, int, chaîne de caractère, sidebar...
- `seconnecter.dart` : page de connexion de l'application ou l'on doit s'identifier à l'aide de user et mdp provenant de la base de données
- `bouton_widget.dart` : bouton créée qui est celui de validation et d'annulation du questionnaire
- `utils.dart` : ébauche de classe qui nous servira à implémenter quelques méthodes de dynamisation de l'application

2) Questionnaire Flutter

```
// Widget de la note du festival (nombre)
int indexTop = 0;

get labels => ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10'];

Widget buildToggle() => CupertinoSwitch(
  value: retouroupas,
  onChanged: (value) => setState(() => retouroupas = value),
);

@override
Widget build(BuildContext context) {
  String titl = 'Questionnaire';
  final labels = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10'];
  final double min = 0;
  final double max = 10;
  final divisions = labels.length;

  return Scaffold(
    appBar: AppBar(
      title: Text("Questionnaire"),
      actions: <Widget>[
        Padding(
          padding: EdgeInsets.only(right: 20.0),
          child: GestureDetector(
            onTap: () {
              Navigator.push(context,
                MaterialPageRoute(builder: (context) => const Apropos()));
            }
          )
        )
      ]
    )
  );
}
```

Morceau de Code responsable du widget permettant de mettre une note au festival.

3) Connexion Base de Données

```
<?php
try {
    $conn = new PDO('mysql:host=localhost;dbname=user_ptut','user_ptut','ZQSVEAbwsxpB7eth');
    $conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

    } catch(PDOException $e){
    echo $e->getMessage();
    die('pas de connexion');
    }
?>
```

Ce code permet à l'application de se connecter à la base de données.

C.Site web

1) Architecture

Voici le rôle de fichiers et répertoire du site web.

- Répertoire image : sert à stocker les images utilisé pour le site web
- Répertoire include :
 - [bootstrap.min.css](#) : bootstrap utilisé pour le site web
 - [styles.css](#) : mise en page du site web
 - [connect.css](#) : mise en page de la connection
 - [footer.php](#) : bandeau en bas du site
 - [header.php](#) : bandeau en haut du site
 - [menus.php](#) : bar de navigation à gauche du site

- Racine du serveur :
 - [AjouterEvenement.php](#) : Un questionnaire qui permet d'ajouter un évènement dans la base de données
 - [BDConnexion.php](#) : le système de connexion du site vers la base de données
 - [Credit.php](#) : La page de Crédit
 - [export.php](#) : La page de téléchargement du fichier format csv
 - [exportDonnees.php](#) : la fonction qui écrit les données dans le fichier
 - [FormConnection.php](#) : ancienne page de connexion
 - [function.php](#) : Un fichier qui stocke toute nos fonction utilisé
 - [getData.php](#) : Permet à l'application de savoir si le mdp saisie est correcte
 - [getQuestionnaire.php](#) : la page qui envoie les données du serveur à l'application
 - [index.php](#) : Page d'accueil du site web
 - [Questionnaire.php](#) : Formulaire qui permet de créer une question et l'ajouter dans la base de donnée
 - [rechercher.php](#) : Permet de visualiser la base de donnée sous forme de tableau
 - [TraitConnection.php](#) : Traitement du questionnaire de connexion pour accéder au site
 - [Utilisateur.php](#) : Permet de créer un utilisateur dans la base de donnée

2) Requetes

```
<?php
include('BDConnexion.php');
$req = 'SELECT * FROM Question';
$stmt = $conn->prepare($req);
$stmt->execute();

$tabQ = [];
$tabQ = ['idQuestion', 'idEvenement', 'numero', 'libelle', 'type', 'min', 'max', 'listeReps'];
$tabQ = ['', '', '', '', ''];

while($res = $stmt->fetch(PDO::FETCH_ASSOC)){

    $tabQ[] = [$res['idQuestion'], $res['idEvenement'], $res['numero'], $res['libelle'],
    $res['type'], $res['min'], $res['max'], $res['listeReps']];

}

foreach($tabQ as $ligne){

    if($ligne[1] != ""){

        echo '[' . $ligne[1].";". $ligne[2].";". $ligne[3].";". $ligne[4].";". $ligne[5].";". $ligne[6].";". $ligne[7].";". $ligne[8].']';

    }

}

?>
```

Requête pour récupérer le questionnaire stocké dans la base de données.

3) HTML & PHP

```
<!--Geoffrey-->
<div class="col">
  <div class="card mb-4 rounded-3 shadow-sm" style="border-radius:25px;">
    <div class="card-header py-3 text-white bg-primary border-primary">
      <center><h2 class="my-0 fw-normal">Geoffrey Servant</h2></center>
    </div>
    <div class="card-body">
      <h3 style="color:black">Chef projet <BR> Application</h3>
      <ul class="list-unstyled mt-3 mb-4">
        <li>
          
          <p style="color:black"> <BR> Développeur de l'application mobile sur Flutter<p>
        </li>
      </ul>
      <a href="mailto:geoffrey_servant@hotmail.fr"><button type="button" class="w-100 btn btn-lg btn-primary">Contactez moi !
    </div>
  </div>
</div>
```

Morceau de code PHP contenu dans les crédits affichant le profil du chef de projet.

4) CSS

```
1  body,html{
2    height: 100%;
3    padding-top: 30px;
4  }
5
6  #sidebar {
7    width: inherit;
8    min-width: 220px;
9    max-width: 220px;
10   background-color: #f5f5f5;
11   float: left;
12   height:100%;
13   position:relative;
14   overflow-y:auto;
15   overflow-x:hidden;
16 }
17 #main {
18   height:100%;
19   overflow:auto;
20 }
21 }
22
23 table{
24   width:90%;
25   border-collapse:collapse;
26 }
```

Fragment de code CSS permettant l'affichage de l'apparence du site web.