

```

1  <?php
2  /*
3   Projet: SOS INFOBOBO
4   Description: Page d'affichage des informations personnelles du réparateur.
5   Auteur: Borel-Jaquet Jonathan
6   Version: 1.0
7   Date: Mai 2019
8   */
9  require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 $about = AboutManager::GetAboutInformation();
11 ?>
12 <!doctype html>
13 <html lang="fr">
14
15 <head>
16     <title>Accueil</title>
17     <?php include "inc/header.php" ?>
18 </head>
19
20 <body>
21     <?php include "inc/navBar.php"; ?>
22     <div class="container-fluid">
23         <div class="row justify-content-center">
24             <div class="section col-md-4 col-lg-10 border justify-content-
25                 center border-primary rounded mt-4 p-4">
26                 <h1>À propos de moi</h1>
27                 <hr />
28                 <table>
29                     <tr>
30                         <td>
31                             <h6>Numéro de téléphone : </h6>
32                             <td><?= $about->phoneNumber ?></td>
33                     </tr>
34                     <tr>
35                         <td>
36                             <h6>Email : </h6>
37                             <td>
38                             <td>
39                                 <?= $about->email ?>
40                             </td>
41                     </tr>
42                     <tr>
43                         <td>
44                             <h6>Tarif : </h6>
45                             <td>
46                             <td>
47                                 <?= $about->price ?>
48                             </td>
49                     </tr>
50                     <tr>
51                         <td>
52                             <h6>Description : </h6>
53                             <td>
54                             <td class="text-justify">
55                                 <?= $about->description ?>
56                             </td>
57                     </tr>
58                 <?php
59                 if ($isLoggedIn == true) :

```

```
60     ?>
61     <tr>
62         <td>
63             <a class="btn btn-primary" href=?=
PAGE_ADMIN_ABOUT ?> role="button">Modifier</a>
64         </td>
65     </tr>
66     <?php endif; ?>
67 </table>
68 </div>
69 </div>
70 </div>
71 </body>
72
73 </html>
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Page de création d'une demande de réparation informatique pour
le client.
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 if (isset($_POST["btnSendRequest"])) {
11     if (!empty($_POST["contactFirstName"]) && !empty($_POST["contactSecondName"])
12         && !empty($_POST["contactEmail"]) && !empty($_POST["contactPhoneNumber"]) && !
13         empty($_POST["contactDescription"])) {
14         $contactFirstName = filter_input(INPUT_POST, "contactFirstName",
15             FILTER_SANITIZE_STRING);
16         $contactSecondName = filter_input(INPUT_POST, "contactSecondName",
17             FILTER_SANITIZE_STRING);
18         $contactEmail = filter_input(INPUT_POST, "contactEmail",
19             FILTER_SANITIZE_STRING);
20         $contactPhoneNumber = filter_input(INPUT_POST, "contactPhoneNumber",
21             FILTER_SANITIZE_STRING);
22         $contactDescription = filter_input(INPUT_POST, "contactDescription",
23             FILTER_SANITIZE_STRING);
24         if (RequestManager::AddRequest($contactFirstName, $contactSecondName,
25             $contactEmail, $contactPhoneNumber, $contactDescription)) {
26             MailerManager::SendMailToRepairer($contactFirstName,
27                 $contactSecondName, $contactEmail, $contactPhoneNumber, $contactDescription);
28             StyleManager::ShowAlert(ALERT_TYPE_SUCCESS, "Votre demande de
29                 réparation informatique a été envoyée au réparateur");
30         } else {
31             StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Demande invalide");
32         }
33     } else {
34         StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Tous les champs n'ont pas
35             été remplis");
36     }
37 }
38 ?>
39 <!doctype html>
40 <html lang="fr">
41
42 <head>
43     <title>Contact</title>
44     <?php include "inc/header.php" ?>
45 </head>
46
47 <body>
48     <?php include "inc/navBar.php"; ?>
49     <div class="container-fluid">
50         <div class="row justify-content-center">
51             <div class="section col-md-4 col-lg-10 border justify-content-
52                 center border-primary rounded mt-4 p-4">
53                 <div class="row justify-content-center">
54                     <h4>Faire une demande de réparation</h4>
55                     <hr />
56                 </div>
57                 <div class="row">
58                     <div class="col-md-4 col-lg-4">
59                         <form action="#" method="POST" class="w-100">

```

```

48         <div class="form-group">
49             <label for="contactFirstName">Nom</label>
50             <input type="text" value="<?= ((isset($_POST["
contactFirstName"])) ? $_POST["contactFirstName"] : "") ?>" name="
contactFirstName" class="form-control" id="contactFirstName">
51         </div>
52         <div class="form-group">
53             <label for="contactSecondName">Prénom</label>
54             <input type="text" value="<?= ((isset($_POST["
contactSecondName"])) ? $_POST["contactSecondName"] : "") ?>" name="
contactSecondName" class="form-control" id="contactSecondName">
55         </div>
56         <div class="form-group">
57             <label for="contactEmail">Email</label>
58             <input type="email" value="<?= ((isset($_POST["
contactEmail"])) ? $_POST["contactEmail"] : "") ?>" name="contactEmail" class="
form-control" id="contactEmail">
59         </div>
60         <div class="form-group">
61             <label for="contactPhoneNumber">Numéro de
téléphone</label>
62             <input type="text" value="<?= ((isset($_POST["
contactPhoneNumber"])) ? $_POST["contactPhoneNumber"] : "") ?>" name="
contactPhoneNumber" class="form-control" id="contactPhoneNumber">
63         </div>
64     </div>
65 </div>
66 <div class="row justify-content-center">
67     <div class="col-12">
68         <div class="form-group">
69             <label for="contactDescription">Description du
problème</label>
70             <textarea class="form-control" name="
contactDescription" id="contactDescription" rows="10"><?= ((isset($_POST["
contactDescription"])) ? $_POST["contactDescription"] : "") ?></textarea>
71         </div>
72     </div>
73 </div>
74 <button type="submit" name="btnSendRequest" class="btn btn-
primary">Envoyer</button>
75 <small id="emailHelp" class="form-text text-muted">Tous les
champs sont obligatoires.</small>
76 </form>
77 </div>
78 </div>
79 </div>
80 </body>
81
82 </html>

```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Page de création d'avis pour le client.
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 $arrOpinion = OpinionManager::GetOpinionValidate();
11 if (isset($_POST["btnSendOpinion"])) {
12     if (!empty($_POST["opinionComment"])) {
13         $comment = filter_input(INPUT_POST, "opinionComment",
14 FILTER_SANITIZE_STRING);
15         if (OpinionManager::AddOpinion($comment)) {
16             StyleManager::ShowAlert(ALERT_TYPE_SUCCESS, "Votre avis est en
17 attente de validation");
18         } else {
19             StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Avis invalide");
20         }
21     } else {
22         StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Le champ commentaire n'a
23 pas été rempli");
24     }
25 }
26
27 <?>
28 <!doctype html>
29 <html lang="fr">
30
31 <head>
32     <title>Avis</title>
33     <?php include "inc/header.php" ?>
34 </head>
35
36 <body>
37     <?php include "inc/navBar.php"; ?>
38     <div class="container-fluid">
39         <div class="row justify-content-center">
40             <div class="section col-md-4 col-lg-10 border justify-content-
41 center border-primary rounded mt-4 p-4">
42                 <div class="row justify-content-center">
43                     <h4>Votre avis m'intéresse</h4>
44                     <hr />
45                     <form class="col-12 m-0 p-0" action="#" method="post">
46                         <div class="form-group">
47                             <label for="opinionComment">Commentaire :</label>
48                             <textarea class="form-control" name="opinionComment
49 " rows="5" id="opinionComment"></textarea>
50                         </div>
51                         <button type="submit" name="btnSendOpinion" class="btn
52 btn-primary mb-3">Envoyer</button>
53                     </form>
54                 </div>
55                 <div class="row justify-content-center mb-2">
56                     <h4>Les avis précédents</h4>
57                     <hr />
58                 </div>
59             </div>
60         </div>
61     </div>
62
63 <?php
64 if (!empty($arrOpinion)) {
65     foreach ($arrOpinion as $opinion) :

```

```
55     ?>
56     <div class="row justify-content-center">
57         <div class="col-12 media border-bottom border-dark
mb-2">
58             <div class="media-body">
59                 <p class="text-justify mb-1">
60                     <?= htmlspecialchars_decode($opinion->
comment) ?>
61                 </p>
62             <div class="row justify-content-end">
63                 <div class="col-md-4 col-lg-2">
64                     <h5><?= StyleManager::
SqlDateToWritten($opinion->date); ?></h5>
65                 </div>
66             </div>
67         </div>
68     </div>
69 </div>
70 <?php
71     endforeach;
72     } else {
73         echo "<h6>Aucun avis validé</h6>";
74     }
75     ?>
76 </div>
77 </div>
78 </div>
79 </body>
80
81 </html>
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Page de connexion du réparateur
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 if (isset($_POST["btnConnection"])) {
11     if (!empty($_POST["pseudoLogin"]) && !empty($_POST["pwdLogin"])) {
12         $pseudo = filter_input(INPUT_POST, "pseudoLogin",
13 FILTER_SANITIZE_STRING);
14         $pwd = filter_input(INPUT_POST, "pwdLogin", FILTER_SANITIZE_STRING);
15         if (UserManager::Connection($pseudo, $pwd)) {
16             $_SESSION["isLoggedIn"] = true;
17             header("location:" . PAGE_ABOUT);
18         } else {
19             StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Mot de passe ou pseudo
incorrect");
20         }
21     } else {
22         StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Tous les champs n'ont pas
été remplis");
23     }
24 }
25 <!doctype html>
26 <html lang="fr">
27
28 <head>
29     <title>Connexion</title>
30     <?php include "inc/header.php" ?>
31 </head>
32
33 <body>
34     <?php include "inc/navBar.php"; ?>
35     <div class="container-fluid">
36         <div class="row justify-content-center">
37             <div class="section col-md-4 col-lg-4 border justify-content-center
border-primary rounded mt-4 p-4">
38                 <div class="row justify-content-center">
39                     <form action="#" method="post" style="width:100%;">
40                         <h1 class="text-center">Connexion</h1>
41                         <hr />
42                         <div class="form-group">
43                             <label for="pseudoLogin">Pseudo</label>
44                             <input type="text" value="<?=(isset($_POST["
pseudoLogin"]) ? $_POST["pseudoLogin"] : "") ?>" class="form-control" id="
pseudoLogin" name="pseudoLogin" placeholder="Entrez votre pseudo">
45                         </div>
46                         <div class="form-group">
47                             <label for="pwdLogin">Mot de passe</label>
48                             <input type="password" class="form-control" id="
pwdLogin" name="pwdLogin" placeholder="Entrez votre mot de passe">
49                         </div>
50                         <button type="submit" name="btnConnection" class="btn
btn-primary">Connexion</button>
51                     </form>
52                 </div>

```

```
53         </div>
54     </div>
55 </div>
56 </body>
57
58 </html>
```



```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Page de modification des informations personnelles du réparateur
5      .
6      Auteur: Borel-Jaquet Jonathan
7      Version: 1.0
8      Date: Mai 2019
9      */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
11 UserManager::VerificateRoleUser();
12 if (isset($_POST["btnUpdateAbout"])) {
13     if (!empty($_POST["newAboutPhoneNumber"]) && !empty($_POST["newAboutEmail"])
14         && !empty($_POST["newAboutPrice"]) && !empty($_POST["newAboutDescription"])) {
15         $newPhoneNumber = filter_input(INPUT_POST, "newAboutPhoneNumber",
16 FILTER_SANITIZE_STRING);
17         $newEmail = filter_input(INPUT_POST, "newAboutEmail",
18 FILTER_SANITIZE_STRING);
19         $newPrice = filter_input(INPUT_POST, "newAboutPrice",
20 FILTER_SANITIZE_STRING);
21         $newDescription = filter_input(INPUT_POST, "newAboutDescription",
22 FILTER_SANITIZE_STRING);
23         if (AboutManager::UpdateAboutInformation($newPhoneNumber, $newEmail,
24 $newPrice, $newDescription)) {
25             StyleManager::ShowAlert(ALERT_TYPE_SUCCESS, "Informations bien
26 modifiées");
27         } else {
28             StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Problème lors de la
29 modification des informations");
30         }
31     } else {
32         StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Tous les champs n'ont pas
33 été remplis");
34     }
35 }
36 $about = AboutManager::GetAboutInformation();
37 ?>
38 <!doctype html>
39 <html lang="fr">
40
41 <head>
42     <title>Modification d'informations</title>
43     <?php include "inc/header.php" ?>
44 </head>
45
46 <body>
47     <?php include "inc/navBar.php"; ?>
48     <div class="container-fluid">
49         <div class="row justify-content-center">
50             <div class="section col-md-4 col-lg-4 border justify-content-center
51 border-primary rounded mt-4 p-4">
52                 <div class="row justify-content-center">
53                     <h1>Modification de mes informations</h1>
54                     <hr />
55                 </div>
56                 <div class="row justify-content-center">
57                     <form action="#" method="POST" class="w-100">
58                         <div class="form-group">
59                             <label for="newAboutPhoneNumber">Numéro de
60 téléphone</label>

```

```
49         <input type="text" name="newAboutPhoneNumber"
class="form-control" id="newAboutPhoneNumber" value="<?= $about->phoneNumber
?>">
50     </div>
51     <div class="form-group">
52         <label for="newAboutEmail">Email</label>
53         <input type="email" name="newAboutEmail" class="
form-control" id="newAboutEmail" value="<?= $about->email ?>">
54     </div>
55     <div class="form-group">
56         <label for="newAboutPrice">Price</label>
57         <input type="text" name="newAboutPrice" class="
form-control" id="newAboutPrice" value="<?= $about->price ?>">
58     </div>
59     <div class="form-group">
60         <label for="newAboutDescription">Description</
label>
61         <textarea class="form-control" name="
newAboutDescription" id="newAboutDescription" rows="4"><?= $about->description
?></textarea>
62     </div>
63     <button type="submit" name="btnUpdateAbout" class="btn
btn-primary">Modifier</button>
64     <small id="emailHelp" class="form-text text-muted">
Tous les champs sont obligatoires.</small>
65 </form>
66 </div>
67 </div>
68 </div>
69 </div>
70 </body>
71
72 </html>
```

```

1 <?php
2 /*
3     Projet: SOS INFOBOBO
4     Description: Page d'administration des demandes de réparation informatique du
      réparateur.
5     Auteur: Borel-Jaquet Jonathan
6     Version: 1.0
7     Date: Mai 2019
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 UserManager::VerificateRoleUser();
11 if (isset($_GET["status"]) && isset($_GET["id_request"])) {
12     $status = filter_input(INPUT_GET, "status", FILTER_SANITIZE_STRING);
13     $id_request = filter_input(INPUT_GET, "id_request",
      FILTER_SANITIZE_NUMBER_INT);
14     $arrRequestClientById = RequestManager::GetRequestById($id_request);
15     if (RequestManager::UpdateRequestStatusById($id_request, $status)) {
16         MailerManager::SendMailToClient($arrConstStatus[$status],
      $arrRequestClientById[CLIENT], $arrRequestClientById[REQUEST]);
17         StyleManager::ShowAlert(ALERT_TYPE_SUCCESS, "Statut de la demande bien
      modifié");
18     } else {
19         StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Problème lors du changement
      de statut de la demande");
20     }
21 }
22 $arrRequest = RequestManager::GetAllRequest();
23 ?>
24 <!doctype html>
25 <html lang="fr">
26
27 <head>
28     <title>Administration des demandes</title>
29     <?php include "inc/header.php" ?>
30 </head>
31
32 <body>
33     <?php include "inc/navBar.php"; ?>
34     <div class="container-fluid">
35         <div class="row justify-content-center">
36             <div class="section col-md-4 col-lg-10 border justify-content-
      center border-primary rounded mt-4 p-4">
37                 <div class="row justify-content-center">
38                     <h4>Listes des demandes de réparation</h4>
39                     <hr />
40                 </div>
41                 <?php
42                 if (!empty($arrRequest)) {
43                     foreach ($arrRequest as $arrRequestClient) :
44                         ?>
45                         <div class="row">
46                             <div class="col-md-12">
47                                 <div class="table-responsive">
48                                     <table class="table table-dark rounded
      border-primary">
49                                         <thead>
50                                             <tr>
51                                                 <th style="width: 15%" scope="
      col">Nom</th>
52                                                 <th style="width: 15%" scope="

```

```

52 col">Prénom</th>
53                                     <th style="width: 15%" scope="
54 col">Email</th>
55                                     <th style="width: 15%" scope="
56 col">Téléphone</th>
57                                     <th style="width: 15%" scope="
58 col">Statut</th>
59                                     <th style="width: 15%" scope="
60 col"></th>
61                                     </tr>
62                                     </thead>
63                                     <tbody>
64                                     <tr>
65                                     <td><?= $arrRequestClient [
66 CLIENT]->firstName ?></td>
67                                     <td><?= $arrRequestClient [
68 CLIENT]->secondName ?></td>
69                                     <td><?= $arrRequestClient [
70 CLIENT]->email ?></td>
71                                     <td><?= $arrRequestClient [
72 CLIENT]->phoneNumber ?></td>
73                                     <td style="color:<?=
74 StyleManager::ColorStatus($arrRequestClient[REQUEST]->status) ?>;"><?=
75 $arrConstStatus[$arrRequestClient[REQUEST]->status] ?></td>
76                                     </tr>
77                                     <tr>
78                                     <td colspan="5">
79                                     <textarea rows="5" class="
80 form-control" readonly><?= $arrRequestClient[REQUEST]->description ?></
81 textarea>
82                                     </td>
83                                     <td>
84                                     <div class="btn-group btn-
85 group-rounded w-100">
86                                     <button type="button"
87 class="btn btn-primary btn-xs w-100" data-toggle="dropdown" aria-haspopup="
88 true" aria-expanded="false" style="border-radius:3px;">
89                                     <span class="caret
90 ">Modifier</span>
91                                     </button>
92                                     <div class="dropdown-
93 menu w-100" aria-labelledby="dropdownMenuButton">
94                                     <a class="dropdown
95 -item" href="adminContact.php?status=<?= STATUS_REFUSED ?>&id_request=<?=
96 $arrRequestClient[REQUEST]->id_request ?>">Refusée</a>
97                                     <a class="dropdown
98 -item" href="adminContact.php?status=<?= STATUS_PROCESSED ?>&id_request=<?=
99 $arrRequestClient[REQUEST]->id_request ?>">Traitée</a>
100                                     </div>
101                                     </div>
102                                     </td>
103                                     </tr>
104                                     </tbody>
105                                     </table>
106                                     </div>
107                                     </div>
108                                     </div>
109                                     <?php
110                                     endforeach;
111                                     } else {

```

```
91         echo "<h6>Aucune demande de réparation</h6>";
92     }
93     ?>
94     </div>
95 </div>
96 </body>
97
98 </html>
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Page d'administration pour la validation ou le refus d'un avis.
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 UserManager::VerificateRoleUser();
11 if (isset($_GET["type"]) && isset($_GET["id_opinion"])) {
12     $type = filter_input(INPUT_GET, "type", FILTER_SANITIZE_STRING);
13     $id_opinion = filter_input(INPUT_GET, "id_opinion",
14 FILTER_SANITIZE_NUMBER_INT);
15     if ($type == "accept") {
16         if (OpinionManager::ValidateOpinionById($id_opinion)) {
17             StyleManager::ShowAlert(ALERT_TYPE_SUCCESS, "Avis validé");
18         } else {
19             StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Problème lors de la
20 validation de l'avis");
21         }
22     } else if ($type == "refuse") {
23         if (OpinionManager::DeleteOpinionById($id_opinion)) {
24             StyleManager::ShowAlert(ALERT_TYPE_SUCCESS, "Avis refusé");
25         } else {
26             StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Problème lors du refus
27 de l'avis");
28         }
29     }
30 }
31 $arrOpinion = OpinionManager::GetOpinionNotValidate();
32 ?>
33 <!doctype html>
34 <html lang="fr">
35 <head>
36     <title>Validation d'avis</title>
37     <?php include "inc/header.php" ?>
38 </head>
39 <body>
40     <?php include "inc/navBar.php"; ?>
41     <div class="container-fluid">
42         <div class="row justify-content-center">
43             <div class="section col-md-4 col-lg-10 border justify-content-
44 center border-primary rounded mt-4 p-4">
45                 <div class="row justify-content-center mb-2">
46                     <h4>Avis en attente de validation</h4>
47                     <hr />
48                 </div>
49                 <?php
50                 if (!empty($arrOpinion)) {
51                     foreach ($arrOpinion as $opinion) :
52                         ?>
53                         <div class="row justify-content-center">
54                             <div class="col-12 media border-bottom border-dark
55 mb-2">
56                                 <div class="media-body">
57                                     <p class="text-justify mb-1">
58                                         <?= htmlspecialchars_decode($opinion->

```

```
55 comment) ?>
56                                     </p>
57                                     <div class="row justify-content-end">
58                                         <div class="col-md-4 col-lg-2">
59                                             <h5><?= StyleManager::
SqlDateToWritten($opinion->date) ?></h5>
60                                         </div>
61                                         <div class="col-md-4 col-lg-2">
62                                             <a class="btn btn-success w-100"
href="adminOpinion.php?type=accept&id_opinion=<?= $opinion->id_opinion ?>"
role="button">Valider</a>
63                                             <a class="btn btn-danger w-100"
href="adminOpinion.php?type=refuse&id_opinion=<?= $opinion->id_opinion ?>"
role="button">Refuser</a>
64                                         </div>
65                                     </div>
66                                 </div>
67                            </div>
68                    </div>
69                <?php
                    endforeach;
70                } else {
71                    echo "<h6>Aucun avis en attente de validation</h6>";
72                }
73            }
74        ?>
75    </div>
76 </div>
77 </div>
78 </body>
79
80 </html>
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Page d'administration des rendez-vous du réparateur.
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 UserManager::VerificateRoleUser();
11 if (isset($_GET["saveEvent"])) {
12     if (!empty($_GET["id_requestSelected"]) && !empty($_GET["eventDateStart"])
13     && !empty($_GET["eventDateEnd"])) {
14         $id_request = filter_input(INPUT_GET, "id_requestSelected",
15 FILTER_SANITIZE_STRING);
16         $eventDateStart = filter_input(INPUT_GET, "eventDateStart",
17 FILTER_SANITIZE_STRING);
18         $eventDateEnd = filter_input(INPUT_GET, "eventDateEnd",
19 FILTER_SANITIZE_STRING);
20         $arrRequestClientById = RequestManager::GetRequestById($id_request);
21         if (EventManager::AddEvent($id_request, $eventDateStart, $eventDateEnd)
22 ) {
23             StyleManager::ShowAlert(ALERT_TYPE_SUCCESS, "Rendez-vous créé");
24             MailerManager::SendMailToClient($arrConstStatus[STATUS_IN_PROGRESS]
25 , $arrRequestClientById[CLIENT], $arrRequestClientById[REQUEST]);
26         } else {
27             StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Problème lors de l'
28 insertion du rendez-vous");
29         }
30     } else {
31         StyleManager::ShowAlert(ALERT_TYPE_FAILED, "Tous les éléments
32 nécessaires n'ont pas été sélectionnés");
33     }
34 }
35 $arrOpenRequest = RequestManager::GetOpenRequest();
36 ?>
37 <!doctype html>
38 <html lang="fr">
39
40 <head>
41     <!-- Plugins css pour un affichage correct des différents plugins utilisés
42     -->
43     <link href='lib/fullcalendar-4.1.0/packages/core/main.css' rel='stylesheet'
44     />
45     <link href='lib/fullcalendar-4.1.0/packages/daygrid/main.css' rel='
46 stylesheet' />
47     <link href='lib/fullcalendar-4.1.0/packages/timegrid/main.css' rel='
48 stylesheet' />
49     <link href='lib/fullcalendar-4.1.0/packages/list/main.css' rel='stylesheet'
50     />
51     <!-- Plugins JS pour le bon fonctionnement des différents plugins utilisés
52     -->
53     <script src='lib/fullcalendar-4.1.0/packages/core/main.js'></script>
54     <script src='lib/fullcalendar-4.1.0/packages/daygrid/main.js'></script>
55     <script src='lib/fullcalendar-4.1.0/packages/timegrid/main.js'></script>
56     <script src='lib/fullcalendar-4.1.0/packages/list/main.js'></script>
57     <!-- Plugin pour sélectionner une date dans le calendrier et pour mettre
58     le calendrier en francais -->
59     <script src='lib/fullcalendar-4.1.0/packages/interaction/main.js'></script>
60     <script src='lib/fullcalendar-4.1.0/packages/core/locales/fr-ch.js'></

```



```

45 <script>
46     <title>Calendrier</title>
47     <?php include "inc/header.php"; ?>
48 </head>
49
50 <body>
51     <?php include "inc/navBar.php"; ?>
52     <div class="container-fluid">
53         <div class="row justify-content-center">
54             <div class="section col-sm-10 col-lg-10 col-md-10 border justify-
content-center border-primary rounded mt-4 p-4">
55                 <div class="row justify-content-center mb-2">
56                     <h4>Planification de rendez-vous</h4>
57                     <hr />
58                 </div>
59                 <div class="row">
60                     <div class="col-lg-7 col-md-12 col-sm-12">
61                         <div id='calendar' style="padding: 15px; border: solid
#007bff 1px; border-radius: 5px;"></div>
62                     </div>
63                     <div class="col-lg-5 col-md-12 col-sm-12" style="
overflow: scroll; height: 650px">
64                         <form method="GET" action="#">
65                             <input type="hidden" name="id_requestSelected" id=
"id_requestSelected">
66                             <div class="form-group">
67                                 <label for="eventDateStart">Date de
récupération</label>
68                                 <input name="eventDateStart" type="text" class
="form-control" id="eventDateStart" readonly>
69                             </div>
70                             <div class="form-group">
71                                 <label for="eventDateEnd">Date de reddition</
label>
72                                 <input name="eventDateEnd" type="text" class="
form-control" id="eventDateEnd" readonly>
73                             </div>
74                             <button type="submit" name="saveEvent" class="btn
btn-primary mb-2">Planifier</button>
75                         </form>
76                         <?php foreach ($arrOpenRequest as $request) : ?>
77                             <div class="table-responsive">
78                                 <table class="tableRequest table table-dark
rounded border-primary ">
79                                     <thead>
80                                         <tr>
81                                             <th style="width: 10%" scope="col"
>Id</th>
82                                             <th style="width: 10%" scope="col"
>Nom</th>
83                                             <th style="width: 10%" scope="col"
>Prénom</th>
84                                             <th style="width: 70%" scope="col"
>Description</th>
85                                         </tr>
86                                     </thead>
87                                     <tbody>
88                                         <tr>
89                                             <td><?= $request[REQUEST]->
id_request ?></td>

```

```

90         <td><?= $request[CLIENT]->
    firstName ?></td>
91         <td><?= $request[CLIENT]->
    secondName ?></td>
92         <td> <textarea rows="5" class="
    form-control" readonly><?= $request[REQUEST]->description ?></textarea></td>
93     </tr>
94 </tbody>
95 </table>
96 </div>
97 <?php endforeach; ?>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 <script>
104     //Fonction Javascript de sélection de la demande de réparation
    informatique
105     $(document).ready(function() {
106         $(".tableRequest").click(function() {
107             $(".tableRequest").css("background-color", "#212529");
108             $(".tableRequest").css("border", "none");
109             $(this).css("background-color", "#2c3e50");
110             $(this).css("border", "2px solid #007bff ");
111             var tableData = $(this).find("td").map(function() {
112                 return $(this).text();
113             });
114             document.getElementById("id_requestSelected").value =
    tableData[0];
115         });
116     });
117 </script>
118 <script>
119     //Liste d'événements en format JSON à afficher dans le calendrier (
    propriétés events)
120     var data = <?= EventManager::GetAllEventFormatJSON() ?>
121 </script>
122 </body>
123 <script>
124     //Script de configuration et d'affichage FullCalendar
125     var calendarEl = document.getElementById('calendar');
126     var calendar = new FullCalendar.Calendar(calendarEl, {
127         timezone: "local",
128         plugins: ['interaction', 'dayGrid', 'timeGrid', 'list'],
129         header: {
130             left: 'dayGridMonth,dayGridWeek,timeGridDay,list',
131             center: 'title',
132             right: 'prevYear,prev,next,nextYear'
133         },
134         defaultView: 'dayGridMonth',
135         selectable: true,
136         locale: 'fr-ch',
137         select: function(info) {
138
139             document.getElementById("eventDateStart").value = info.startStr;
140             document.getElementById("eventDateEnd").value = info.endStr;
141         },
142         events: data,
143         eventColor: '#2C3E50',

```

```
144         eventTextColor: '#FFFFFF'  
145     });  
146     calendar.render();  
147 </script>  
148  
149 </html>
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Script de déconnexion d'un utilisateur connecté.
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 session_start();
10 session_destroy();
11 header("location: connection.php");
12
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Page d'administration des statistiques de réparation effectuée
                    du réparateur.
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 UserManger::VerificateRoleUser();
11 if (isset($_POST["btnShowStat"])) {
12     $year = filter_input(INPUT_POST, "year", FILTER_SANITIZE_STRING);
13     $arrMonthStat = RequestManager::GetProcessedRequestOrderByMonthAndYear(
        $year);
14 } else {
15     $arrMonthStat = RequestManager::GetProcessedRequestOrderByMonthAndYear(date
        ('Y'));
16 }
17 ?>
18 <!doctype html>
19 <html lang="fr">
20
21 <head>
22     <title>Statistiques</title>
23     <?php include "inc/header.php" ?>
24 </head>
25
26 <body>
27     <?php include "inc/navBar.php"; ?>
28     <div class="container-fluid">
29         <div class="row justify-content-center">
30             <div class="section col-md-4 col-lg-10 border justify-content-
                center border-primary rounded mt-4 p-4">
31                 <div class="row justify-content-center mb-2">
32                     <h4>Statistiques des réparations effectuées</h4>
33                     <hr />
34                 </div>
35                 <div class="row mb-4">
36                     <div class="col-md-4">
37                         <form method="POST" action="#">
38                             <div class="form-group">
39                                 <label for="numericUpDownYear">Année :</label>
40                                 <input type="number" class="form-control" id="
                numericUpDownYear" name="year" value="<?= ((isset($_POST["btnShowStat"])) ? $year
                : date('Y'))" ?>">
41                             </div>
42                             <button type="submit" name="btnShowStat" class="btn
                btn-primary">Rechercher</button>
43                         </form>
44                     </div>
45                 </div>
46                 <div class="row">
47                     <div class="col-md-12">
48                         <?php if (!empty($arrMonthStat)) { ?>
49                             <div class="table-responsive">
50                                 <table class="table table-dark rounded border-
                primary">
51                                     <thead>
52                                     <tr>

```

```
53         <th scope="col">Mois</th>
54         <th scope="col">Nombre de
    réparation effectuée</th>
55     </tr>
56 </thead>
57 <tbody>
58     <?php foreach ($arrMonthStat as
    $request) : ?>
59         <tr>
60             <td><?= $arrMonth[$request["
    month"]] ?></td>
61             <td><?= $request["nbRequest"]
    ?></td>
62         </tr>
63     <?php endforeach; ?>
64 </tbody>
65 </table>
66 </div>
67 <?php
68     } else {
69         echo "<h6>Aucune réparation effectuée</h6>";
70     }
71     ?>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </body>
78
79 </html>
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe permettant la connexion à la base de données en utilisant
                    un singleton.
5      Auteur: Aigroz Dominique
6      Version: 1.0
7      Date: Mai 2019
8  */
9  /**
10     * @remark Mettre le bon chemin d'accès à votre fichier contenant les constantes
11     */
12     require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/config/conparam.php'
13     ;
14     /**
15     * Retourne un objet PDO connecté à la base de données
16     * @return \PDO
17     */
18     class Database
19     {
20         private static $pdoInstance;
21         /**
22          * @brief Class Constructor - Créer une nouvelle connexion à la database
23          si la connexion n'existe pas
24          * On la met en privé pour que personne puisse créer une nouvelle
25          instance via ' = new Database();'
26          */
27         private function __construct()
28         { }
29         /**
30          * @brief Comme le constructeur, on rend __clone privé pour que personne
31          ne puisse cloner l'instance
32          */
33         private function __clone()
34         { }
35         /**
36          * @brief Retourne l'instance de la Database ou créer une connexion
37          initiale
38          * @return $objInstance;
39          */
40         public static function getInstance()
41         {
42             if (!self::$pdoInstance) {
43                 try {
44                     $dsn = EDB_DBTYPE . ':host=' . EDB_HOST . ';port=' . EDB_PORT .
45                     ';dbname=' . EDB_DBNAME;
46                     self::$pdoInstance = new PDO($dsn, EDB_USER, EDB_PASS, array('
47                     charset' => 'utf8'));
48                     self::$pdoInstance->setAttribute(PDO::ATTR_ERRMODE, PDO::
49                     ERRMODE_EXCEPTION);
50                 } catch (PDOException $e) {
51                     echo "EDatabase Error: " . $e->getMessage();
52                 }
53             }
54             return self::$pdoInstance;
55         }
56         # end method
57         /**
58          * @brief Passes on any static calls to this class onto the singleton PDO

```

```
51 instance
52     * @param $chrMethod    The method to call
53     * @param $arrArguments  The method's parameters
54     * @return $mix         The method's return value
55     */
56     final public static function __callStatic($chrMethod, $arrArguments)
57     {
58         $pdo = self::getInstance();
59         return call_user_func_array(array($pdo, $chrMethod), $arrArguments);
60     }
61     # end method
62 }
63
```



```
1  /*
2   Projet: SOS INFOBOBO
3   Description: Unique fichier css de base pour tout mon site WEB
4   Auteur: Borel-Jaquet Jonathan
5   Version: 1.0
6   Date: Mai 2019
7  */
8  *{
9      font-family: 'Open Sans', sans-serif;
10 }
11 body{
12     background-color: #272727;
13 }
14 .section{
15     background-color: #E0E0E0;
16 }
17 hr{
18     width: 100%;
19     color: black;
20     height: 1px;
21     background-color:black;
22 }
23 textarea{
24     resize: none;
25 }
26 /*
27 CSS pour FullCalendar
28 */
29 #calendar {
30     background-color: #bfbfbf;
31     color: black;
32 }
33
34 .fc-highlight {
35     background-color: #595959;
36 }
37
38 .fc-unthemed td.fc-today {
39     background-color: grey;
40 }
```

```
1 <!--
2   Projet: SOS INFOBOBO
3   Description: Fichier à inclure dans tous les fichiers, correspond à toutes
   les balises d'header HTML
4   Auteur: Borel-Jaquet Jonathan
5   Version: 1.0
6   Date: Mai 2019
7 -->
8 <!-- Balise meta d'encondage -->
9 <meta charset="utf-8">
10 <!-- Balise meta permettant au site d'être affiché correctement sur d'autres
   dispositif -->
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <!-- FONTAWESOME CDN -->
13 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/css/
   all.css" integrity="sha384-oS3vJWv+0UjzBfQzYUhtDYW+Pj2ysiDjXpsK1OYPAYjqT085Qq/
   1cq5FLXAZQ7Ay" crossorigin="anonymous">
14 <!-- BOOTSTRAP JavaScript CDN (jQuery, Popper.js, Plugins JavaScript de
   Bootstrap) -->
15 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="
   sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
   crossorigin="anonymous"></script>
16 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper
   .min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/
   18WvCWPiPm49" crossorigin="anonymous"></script>
17 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.
   min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/
   JmZQ5stwEULTy" crossorigin="anonymous"></script>
18 <!-- BOOTSTRAP CSS CDN -->
19 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1
   /css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/
   j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
20 <!-- POLICE D'ECRITURE GOOGLE -->
21 <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,800&display=
   swap" rel="stylesheet">
22 <!-- FEUILLE DE STYLE PERSONNELLE -->
23 <link rel="stylesheet" type="text/css" href="css/style.css">
```

```

1  <?php
2  /*
3   Projet: SOS INFOBOBO
4   Description: Fichier à inclure dans tous les fichiers, correspond à la barre
   de navigation du site.
5   Auteur: Borel-Jaquet Jonathan
6   Version: 1.0
7   Date: Mai 2019
8   */
9
10 if (isset($_SESSION["isLoggedIn"]) == true && $_SESSION["isLoggedIn"] == true) {
11     $isLoggedIn = true;
12 } else {
13     $isLoggedIn = false;
14 }
15 ?>
16 <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
17     
18     <button class="navbar-toggler" type="button" data-toggle="collapse" data-
19 target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
20 expanded="false" aria-label="Toggle navigation">
21         <span class="navbar-toggler-icon"></span>
22     </button>
23     <div class="collapse navbar-collapse" id="navbarSupportedContent">
24         <ul class="navbar-nav">
25             <li class="nav-item <?= (strpos($_SERVER['PHP_SELF'], PAGE_ABOUT)) ?
26 "active" : ""; ?>">
27                 <a class="nav-link" href=<?= PAGE_ABOUT ?>><span class="fa fa-
28 home"></span> Accueil</a>
29             </li>
30             <li class="nav-item <?= (strpos($_SERVER['PHP_SELF'], PAGE_CONTACT))
31 ? "active" : ""; ?>">
32                 <a class="nav-link" href=<?= PAGE_CONTACT ?>>Contact</a>
33             </li>
34             <li class="nav-item <?= (strpos($_SERVER['PHP_SELF'], PAGE_OPINION))
35 ? "active" : ""; ?>">
36                 <a class="nav-link" href=<?= PAGE_OPINION ?>>Avis</a>
37             </li>
38             <li>
39                 <?php if ($isLoggedIn == true) : ?>
40                     <div class="dropdown">
41                         <button class="btn btn-primary dropdown-toggle" type="
42 button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
43 expanded="false">
44                             <span class="fa fa-user"></span> Administration
45                         </button>
46                         <div class="dropdown-menu" aria-labelledby="
47 dropdownMenuButton">
48                             <a class="dropdown-item <?= (strpos($_SERVER['
49 PHP_SELF'], PAGE_ADMIN_CALENDAR)) ? "active" : ""; ?>" href=<?=
50 PAGE_ADMIN_CALENDAR ?>>Calendrier <span class="fa fa-calendar"></span></a>
51                             <a class="dropdown-item <?= (strpos($_SERVER['
52 PHP_SELF'], PAGE_ADMIN_CONTACT)) ? "active" : ""; ?>" href=<?=
53 PAGE_ADMIN_CONTACT ?>>Demande</a>
54                             <a class="dropdown-item <?= (strpos($_SERVER['
55 PHP_SELF'], PAGE_ADMIN_OPINION)) ? "active" : ""; ?>" href=<?=
56 PAGE_ADMIN_OPINION ?>>Avis</a>
57                             <a class="dropdown-item <?= (strpos($_SERVER['
58 PHP_SELF'], PAGE_ADMIN_STATISTIC)) ? "active" : ""; ?>" href=<?=
59 PAGE_ADMIN_STATISTIC ?>>Statistique</a>

```

```
43         </div>
44     </div>
45     <?php endif; ?>
46 </li>
47 <li class="nav-item <?= (strpos($_SERVER['PHP_SELF'],
PAGE_CONNECTION)) ? "active" : ""; ?>">
48     <?= ($isLoggedIn == false) ? '<a class="nav-link" href="' .
PAGE_CONNECTION . '"><span class="fas fa-sign-in-alt"></span> Connexion</a>' :
    '<a class="nav-link" href="disconnection.php"><span class="fas fa-sign-out-
alt"></span> Deconnexion</a>'; ?>
49 </li>
50 </ul>
51 </div>
52 </nav>
```

```
1 <?php
2 /*
3     Projet: SOS INFOBOBO
4     Description: Fichier à inclure dans tous les fichiers, inclus
5                  le démarrage de la session,
6                  la classe de connexion à la base de données,
7                  les classe conteneur,
8                  les classes manager,
9                  La configuration de Swift Mailer,
10                 Les constantes.
11
12     Auteur: Borel-Jaquet Jonathan
13     Version: 1.0
14     Date: Mai 2019
15 */
16
17 //Démarrage de la session
18 session_start();
19
20 //Classe de connexion à la base de données
21 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/db/database.php';
22
23 //Classe conteneur
24 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/class/Client.php';
25 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/class/Opinion.php';
26 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/class/About.php';
27 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/class/Request.php';
28
29 //Classe Manager
30 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/manager/UserManager.
31 php';
32 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/manager/
33 OpinionManager.php';
34 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/manager/AboutManager
35 .php';
36 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/manager/
37 ClientManager.php';
38 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/manager/
39 RequestManager.php';
40 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/manager/StyleManager
41 .php';
42 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/manager/EventManager
43 .php';
44 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/manager/
45 MailerManager.php';
46
47 //Swift Mailer
48 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/config/mailparam.php
49 ';
50 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/lib/swiftmailer5/lib
51 /swift_required.php';
52
53 //Constantes
54
55 //Index pour tableau d'objet CLIENT-REQUEST
56 const CLIENT = 0;
57 const REQUEST = 1;
58
59 //Statut d'une demande de réparation informatique
60 const STATUS_OPEN = "OUVERTE";
```

```
51 const STATUS_IN_PROGRESS = "ENCOURS";
52 const STATUS_PROCESSED = "TRAITEE";
53 const STATUS_REFUSED = "REFUSEE";
54
55 //Lien de toutes les vues du site
56 const PAGE_ABOUT = "index.php";
57 const PAGE_CONTACT = "contact.php";
58 const PAGE_OPINION = "opinion.php";
59 const PAGE_CONNECTION = "connection.php";
60 const PAGE_ADMIN_ABOUT = "aboutUpdate.php";
61 const PAGE_ADMIN_OPINION = "adminOpinion.php";
62 const PAGE_ADMIN_CONTACT = "adminContact.php";
63 const PAGE_ADMIN_STATISTIC = "adminStatistic.php";
64 const PAGE_ADMIN_CALENDAR = "adminCalendar.php";
65
66 //Type d'alerte Bootstrap
67 const ALERT_TYPE_SUCCESS = "success";
68 const ALERT_TYPE_FAILED = "danger";
69
70 //Tableau permettant la traduction BDD->Lisible des statuts de demande de
    réparation informatique
71 $arrConstStatus = array(
72     STATUS_OPEN => "Ouverte",
73     STATUS_IN_PROGRESS => "En cours",
74     STATUS_PROCESSED => "Traitée",
75     STATUS_REFUSED => "Refusée"
76 );
77 //Tableau des mois pour l'affichage des statistiques
78 $arrMonth = array(
79     1 => "Janvier",
80     2 => "Février",
81     3 => "Mars",
82     4 => "Avril",
83     5 => "Mai",
84     6 => "Juin",
85     7 => "Juillet",
86     8 => "Août",
87     9 => "Septembre",
88     10 => "Octobre",
89     11 => "Novembre",
90     12 => "Décembre",
91 );
92
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe conteneur en référence à la table "infos_dynamiques".
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  /**
10   * @brief    Objet About
11   * @remark  Cet objet est utilisé comme conteneur en référence avec la table "
12   infos_dynamiques".
13   *
14   *      Exemple d'utilisation 1
15   *      $a = new About();
16   *      $a->id_about = 1
17   *      $a->phoneNumber = "077 421 34 20";
18   *      $a->email = "sos@infobobo.ch";
19   *      $a->price = "50 - 100 CHF";
20   *      $a->description = "Bonjour, je m'appelle Thierry Borel-Jaquet et je
21   suis la solut..";
22   *
23   *      Exemple d'utilisation 2
24   *      $a = new About(1,"077 421 34 20","sos@infobobo.ch","50 - 100 CHF","
25   Bonjour, je m'appelle Thierry Borel-Jaquet et je suis la solut..");
26   */
27 class About
28 {
29     /**
30      * @brief  Le Constructor appelé au moment de la création de l'objet ex: new
31      About();
32      * @param InId_about      l'identifiant des informations dynamiques. (Optionel)
33      Default 0
34      * @param InPhoneNumber    Le numéro de téléphone du réparateur. (Optionel)
35      Default ""
36      * @param InEmail          L'email du réparateur. (Optionel) Default ""
37      * @param InPrice           Le prix du potentiel service. (Optionel) Default ""
38      * @param InDescription     La description du réparateur et de ses services.
39      (Optionel) Default ""
40      */
41     public function __construct($InId_about = 0, $InPhoneNumber = "", $InEmail = "
42     ", $InPrice = "", $InDescription = "")
43     {
44         $this->id_about = $InId_about;
45         $this->phoneNumber = $InPhoneNumber;
46         $this->email = $InEmail;
47         $this->price = $InPrice;
48         $this->description = $InDescription;
49     }
50     /**
51      * @var int l'identifiant des informations dynamiques
52      */
53     public $id_about;
54     /**
55      * @var string Le numéro de téléphone du réparateur
56      */
57     public $phoneNumber;
58     /**

```

```
53     * @var string L'email du réparateur
54     */
55     public $email;
56     /**
57     * @var string Le prix du potentiel service
58     */
59     public $price;
60     /**
61     * @var string La description du réparateur et de ses services
62     */
63     public $description;
64 }
65
```



```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe conteneur en référence à la table "clients".
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  /**
10     * @brief    Objet Client
11     * @remark  Cet objet est utilisé comme conteneur en référence avec la table
clients.
12     *
13     *      Exemple d'utilisation 1
14     *      $c = new Client();
15     *      $c->id_client = 2;
16     *      $c->firstName = "Coptère";
17     *      $c->secondName = "Ellie";
18     *      $c->email = "elli.coptere@gmail.com";
19     *      $c->phoneNumber = "077 421 34 20";
20     *
21     *
22     *      Exemple d'utilisation 2
23     *      $c = new Client(2,"Coptère","Ellie","elli.coptere@gmail.com","077
421 34 20");
24     */
25  class Client
26  {
27      /**
28       * @brief    Le Constructor appelé au moment de la création de l'objet ex:
new Client();
29       * @param InId_client    L'identifiant du client. (Optionel) Default 0
30       * @param InFirstName    Le nom du client. (Optionel) Default ""
31       * @param InSecondName   Le prénom du client. (Optionel) Default ""
32       * @param InEmail        L'email du client. (Optionel) Default ""
33       * @param InPhoneNumber  Le numéro de téléphone du client. (Optionel) Default
""
34       */
35       public function __construct($InId_client = 0, $InFirstName = "",
$InSecondName = "", $InEmail = "", $InPhoneNumber = "")
36       {
37           $this->id_client = $InId_client;
38           $this->firstName = $InFirstName;
39           $this->secondName = $InSecondName;
40           $this->email = $InEmail;
41           $this->phoneNumber = $InPhoneNumber;
42       }
43       /**
44       * @var int L'identifiant du client
45       */
46       public $id_client;
47       /**
48       * @var string Le nom du client
49       */
50       public $firstName;
51       /**
52       * @var string Le prénom du client
53       */
54       public $secondName;
55       /**

```

```
56     * @var string L'email du client
57     */
58     public $email;
59     /**
60     * @var string Le numéro de téléphone du client
61     */
62     public $phoneNumber;
63 }
64
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe conteneur en référence à la table "avis".
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  /**
10     * @brief    Objet Opinion
11     * @remark  Cet objet est utilisé comme conteneur en référence avec la table
avis.
12     *
13     *      Exemple d'utilisation 1
14     *      $o = new Opinion();
15     *      $o->id_opinion = 2;
16     *      $o->date = "27/03/19";
17     *      $o->comment = "Super réparation de Monsieur Borel-Jaquet, au top !"
;
18     *      $o->is_validate = FALSE;
19     *      $o->id_repairer = 1;
20     *
21     *
22     *      Exemple d'utilisation 2
23     *      $o = new Opinion(2, "27/03/19", "Super réparation de Monsieur Borel
-Jaquet, au top !",FALSE,1);
24     */
25  class Opinion
26  {
27      /**
28       * @brief    Le Constructor appelé au moment de la création de l'objet ex:
new Opinion();
29       * @param InId_opinion      L'identifiant de l'avis. (Optionel) Default 0
30       * @param InDate            La date de l'avis. (Optionel) Default ""
31       * @param InComment        Le commentaire soumis par le client. (
Optionel) Default ""
32       * @param InIs_validate    Variable indiquant si l'avis a été validé
ou non. (Optionel) Default FALSE
33       * @param InId_repairer    L'identifiant du réparateur. (Optionel)
Default 0
34       */
35       public function __construct($InId_opinion = 0, $InDate = null, $InComment =
"", $InIs_validate = FALSE, $InId_repairer = 0)
36       {
37           $this->id_opinion = $InId_opinion;
38           $this->date = $InDate;
39           $this->comment = $InComment;
40           $this->is_validate = $InIs_validate;
41           $this->id_repairer = $InId_repairer;
42       }
43       /**
44       * @var int L'identifiant de l'avis
45       */
46       public $id_opinion;
47       /**
48       * @var string La date de l'avis
49       */
50       public $date;
51       /**
52       * @var string Le commentaire soumis par le client

```

```
53     */
54     public $comment;
55     /**
56      * @var bool Variable indiquant si l'avis a été validé ou non
57      */
58     public $is_validate;
59     /**
60      * @var int L'identifiant du réparateur
61      */
62     public $id_reparer;
63 }
64
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe conteneur en référence à la table "demandes".
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  /**
10     * @brief    Objet Request
11     * @remark  Cet objet est utilisé comme conteneur en référence avec la table
demandes.
12     *
13     *      Exemple d'utilisation 1
14     *      $r = new Request();
15     *      $r->id_request = 2;
16     *      $r->id_client = 4;
17     *      $r->description = "Bonjour, mon ordinateur à fréquemment des écrans
bleus, est-ce dans vos compétences ?";
18     *      $r->status = "OUVERTE";
19     *
20     *
21     *      Exemple d'utilisation 2
22     *      $r = new Request(1, 4, "Bonjour, mon ordinateur à...", "OUVERTE");
23     */
24  class Request
25  {
26      /**
27       * @brief    Le Constructor appelé au moment de la création de l'objet ex:
new Request();
28       * @param InId_request      L'identifiant de la demande. (Optionel)
Default 0
29       * @param InId_client      L'identifiant du client. (Optionel) Default
0
30       * @param InDescription      La description du problème informatique. (
Optionel) Default ""
31       * @param InStatus          Le statut de la demande. (Optionel) Default
""
32       */
33       public function __construct($InId_request = 0, $InId_client = 0,
$InDescription = "", $InStatus = "")
34       {
35           $this->id_request = $InId_request;
36           $this->id_client = $InId_client;
37           $this->description = $InDescription;
38           $this->status = $InStatus;
39       }
40       /**
41       * @var int L'identifiant de la demande
42       */
43       public $id_request;
44       /**
45       * @var int l'identifiant du client
46       */
47       public $id_client;
48       /**
49       * @var string La description du problème informatique
50       */
51       public $description;
52       /**

```

```
53     * @var string Le statut de la demande
54     */
55     public $status;
56 }
57
```

```
1 <?php
2 /*
3     Projet: SOS INFOBOBO
4     Description: Fichier de test de la fonction Connection du manager UserManager
5     Auteur: Borel-Jaquet Jonathan
6     Version: 1.0
7     Date: Mai 2019
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test de connexion d'un utilisateur au site web réussit avec les bonnes
    informations.<br/>";
11 if (UserManager::Connection("infobobo", "qwert")) {
12     echo "Le login a bien fonctionné.<br/>";
13 }
14 echo "<br/>";
15 echo "Test de connexion d'un utilisateur au site web échoué avec les mauvaises
    informations.<br/>";
16 if (!UserManager::Connection("infoboboFALSE", "qwert")) {
17     echo "Login ou mot de passe erroné.<br/>";
18 }
19
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction GetAboutInformation du manager
   AboutManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test de récupération et d'affichage du seul objet About créé en référence
    avec la table 'infos_dynamiques'.<br>";
11 echo "Résultat :<br/>";
12 $aboutInformation = AboutManager::GetAboutInformation();
13 var_dump($aboutInformation);
14
```



```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction UpdateAboutInformation du manager
   AboutManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test de modification des informations personnelles du réparateur en
   référence avec la table 'infos_dynamiques'.<br/>";
11 echo "Résultat :<br/>";
12 if (AboutManager::UpdateAboutInformation("077 456 23 32", "sos@infobobo.ch", "
   50 - 100 CHF", "Bonjour je m'appelle Thierry et je suis réparatrice
   informatique !")) {
13     echo "L'update a bien fonctionné.";
14 } else {
15     echo "L'update n'a pas fonctionné.";
16 }
17
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction AddEvent du manager EventManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test d'ajout d'un événement dans la table 'evenement'.<br/>";
11 echo "Résultat :<br/>";
12 if (EventManager::AddEvent(24, "2019-05-10", "2019-05-12")) {
13     echo "L'insert a bien fonctionné.";
14 } else {
15     echo "L'insert n'a pas fonctionné.";
16 }
17
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction GetAllEventFormatJSON du manager
   EventManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 echo "Test de récupération de tous les événements en référence à la table '
   evenement' en format JSON.<br/>";
11 echo "Résultat :<br/>";
12 $eventJSON = EventManager::GetAllEventFormatJSON();
13 echo $eventJSON;
14
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction ColorStatus du manager
5  StyleManager
6  Auteur: Borel-Jaquet Jonathan
7  Version: 1.0
8  Date: Mai 2019
9  */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
11 echo "Test de récupération de la bonne couleur assigné à un statut.<br/>";
12 echo "Statut 'En cours'<br/>";
13 echo StyleManager::ColorStatus(STATUS_IN_PROGRESS) . "<br/>";
14 echo "Statut 'Traitee'<br/>";
15 echo StyleManager::ColorStatus(STATUS_PROCESSED) . "<br/>";
16 echo "Statut 'Refusee'<br/>";
17 echo StyleManager::ColorStatus(STATUS_REFUSED) . "<br/>";
18 echo "Statut 'Ouverte'<br/>";
19 echo StyleManager::ColorStatus(STATUS_OPEN);
20
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction SqlDateToWritten du manager
   StyleManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test d'affichage d'une date de base de données au bon format.<br/>";
11 echo "Résultat :<br/>";
12 echo "2019-05-30 devenue " . StyleManager::SqlDateToWritten("2019-05-30");
13
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction AddClient du manager
5  ClientManager
6  Auteur: Borel-Jaquet Jonathan
7  Version: 1.0
8  Date: Mai 2019
9  */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
11 echo "Test d'ajout d'un client dans la table 'clients' et retourne son
12   identifiant.<br/>";
13 echo "Résultat :<br/>";
14 echo ClientManager::AddClient("Borel", "Jonathan", "jonathan.brljq@eduge.ch", "
15   077 421 39 90");
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction SendMailToClient du manager
5  MailerManager
6  Auteur: Borel-Jaquet Jonathan
7  Version: 1.0
8  Date: Mai 2019
9  */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
11 $c = new Client();
12 $c->firstName = "Alfieri";
13 $c->secondName = "Elena";
14 $c->email = "muq47833@cndps.com";
15 $c->phoneNumber = "079 324 32 12";
16 $d = new Request();
17 $d->description = "Mon ordinateur ne fonctionne plus, pouvez-vous m'aider ?";
18 echo "Test d'envoi d'un email au client contenant les informations de sa
19 demande de réparation informatique
20 ainsi que son nouveau statut.<br/>";
21 echo "Résultat :<br/>";
22 if (MailerManager::SendMailToClient(STATUS_IN_PROGRESS, $c, $d)) {
23     echo "L'envoi du mail a bien fonctionné.";
24 } else {
25     echo "L'envoi du mail n'a pas bien fonctionné.";
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction SendMailToRepairer du manager
5  MailerManager
6  Auteur: Borel-Jaquet Jonathan
7  Version: 1.0
8  Date: Mai 2019
9  */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
11 $contactFirstName = "Alfieri";
12 $contactSecondName = "Elena";
13 $contactEmail = "elena.alfieri@gmail.com";
14 $contactPhoneNumber = "079 324 32 12";
15 $contactDescription = "Mon ordinateur ne fonctionne plus, pouvez-vous m'aider ?";
16 echo "Test d'envoi d'un email au réparateur contenant les informations de la
17 demande de réparation informatique d'un client.<br/>";
18 echo "Résultat :<br/>";
19 if (MailerManager::SendMailToRepairer($contactFirstName, $contactSecondName,
20 $contactEmail, $contactPhoneNumber, $contactDescription)) {
21     echo "L'envoi du mail a bien fonctionné.";
22 } else {
23     echo "L'envoi du mail n'a pas bien fonctionné.";
24 }
```



```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction AddOpinion du manager
5  OpinionManager
6  Auteur: Borel-Jaquet Jonathan
7  Version: 1.0
8  Date: Mai 2019
9  */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
11 echo "Test d'ajout d'un avis dans la table 'avis'.<br/>";
12 echo "Résultat :<br/>";
13 if (OpinionManager::AddOpinion("Le service était parfait, très satisfait")) {
14     echo "L'insert a bien fonctionné.";
15 } else {
16     echo "L'insert n'a pas fonctionné.";
17 }
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction DeleteOpinionById du manager
5  OpinionManager
6  Auteur: Borel-Jaquet Jonathan
7  Version: 1.0
8  Date: Mai 2019
9  */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
11 echo "Test de suppression d'un avis grâce à son identifiant dans la table 'avis'
12   <br/>";
13 echo "Résultat :<br/>";
14 if (OpinionManager::DeleteOpinionById(29)) {
15     echo "La suppression a bien fonctionné.";
16 } else {
17     echo "La suppression n'a pas fonctionné.";
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction GetOpinionValidate du manager
   OpinionManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test de récupération et d'affichage de tous les avis validés dans des
   objets Opinion créés en référence avec la table 'avis'.<br/>";
11 echo "Résultat :<br/>";
12 $arrOpinion = OpinionManager::GetOpinionValidate();
13 var_dump($arrOpinion);
14
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction ValidateOpinionById du manager
OpinionManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 echo "Test de validation d'un avis grâce à son identifiant dans la table 'avis'
    '<br/>";
11 echo "Résultat :<br/>";
12 if (OpinionManager::ValidateOpinionById(31)) {
13     echo "La validation a bien fonctionné.";
14 } else {
15     echo "La validation n'a pas fonctionné.";
16 }
17
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction GetOpinionNotValidate du manager
   OpinionManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test de récupération et d'affichage de tous les avis non validés dans des
   objets Opinion créés en référence avec la table 'avis'.<br/>";
11 echo "Résultat :<br/>";
12 $arrOpinion = OpinionManager::GetOpinionNotValidate();
13 var_dump($arrOpinion);
14
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction AddRequest du manager
   RequestManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test d'ajout d'une demande de réparation informatique dans la table '
   evenement'.<br/>";
11 echo "Résultat :<br/>";
12 if (RequestManager::AddRequest("Borel", "Jonathan", "jonathan.brljq@eduge.ch", "
   077 421 39 90", "Mon ordinateur ne fonctionne plus, pouvez-vous m'aider ?")) {
13     echo "L'insert a bien fonctionné.";
14 } else {
15     echo "L'insert n'a pas fonctionné.";
16 }
17
```

```
1 <?php
2 /*
3     Projet: SOS INFOBOBO
4     Description: Fichier de test de la fonction GetAllRequest du manager
5     RequestManager
6     Auteur: Borel-Jaquet Jonathan
7     Version: 1.0
8     Date: Mai 2019
9 */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
11 echo "Test de récupération de toutes les demandes de réparation informatique
12     dans un tableau contenant des tableaux d'objet Client et Request.<br/>";
13 echo "Résultat :<br/>";
14 $arrRequest = RequestManager::GetAllRequest();
15 var_dump($arrRequest);
16
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction GetOpenRequest du manager
RequestManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
10 echo "Test de récupération de toutes les demandes de réparation informatique de
    type 'Ouverte' dans un tableau contenant des tableaux d'objet Client et
    Request.<br/>";
11 echo "Résultat :<br/>";
12 $arrRequest = RequestManager::GetOpenRequest();
13 var_dump($arrRequest);
14
```



```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction GetRequestById du manager
   RequestManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 echo "Test de récupération d'une demande de réparation informatique grâce à son
   identifiant dans un tableau contenant un objet Client et Request.<br/>";
11 echo "Résultat :<br/>";
12 $arrRequest = RequestManager::GetRequestById(27);
13 var_dump($arrRequest);
14
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de test de la fonction UpdateRequestStatusById du
   manager RequestManager
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'] . '/SOSInfobobo/www/inc/inc.all.php';
10 echo "Test de modification du statut d'une demande de réparation informatique
   grâce à son identifiant.<br/>";
11 echo "Résultat :<br/>";
12 if (RequestManager::UpdateRequestStatusById(24, "REFUSEE")) {
13     echo "L'update a bien fonctionné.";
14 } else {
15     echo "L'update n'a pas fonctionné.";
16 }
17
```

```
1 <?php
2 /*
3     Projet: SOS INFOBOBO
4     Description: Fichier de test de la fonction
5     GetProcessedRequestOrderByMonthAndYear du manager RequestManager
6     Auteur: Borel-Jaquet Jonathan
7     Version: 1.0
8     Date: Mai 2019
9 */
10 require_once $_SERVER['DOCUMENT_ROOT'] . '/SosInfobobo/www/inc/inc.all.php';
11 echo "Test de récupération des statistiques triée par mois et par année choisie
12     en parametre de statut 'TRAITEE'.<br/>";
13 echo "Résultat :<br/>";
14 $arrRequestProcessed = RequestManager::GetProcessedRequestOrderByMonthAndYear("
15     2019");
16 var_dump($arrRequestProcessed);
17
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de constante pour la connexion à la base de données
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 define('EDB_DBTYPE', "mysql");
10 define('EDB_DBNAME', "bj_tpi_bd");
11 define('EDB_HOST', "127.0.0.1");
12 define('EDB_PORT', "3306");
13 define('EDB_USER', "BOREL_JAQUET_TPI");
14 define('EDB_PASS', "Super2019");
15
```

```
1 <?php
2 /*
3  Projet: SOS INFOBOBO
4  Description: Fichier de constante pour la connexion au serveur smtp de google
5  Auteur: Borel-Jaquet Jonathan
6  Version: 1.0
7  Date: Mai 2019
8  */
9 define('EMAIL_SERVER', "smtp.gmail.com");
10 define('EMAIL_PORT', 465);
11 define('EMAIL_TRANS', "ssl");
12 define('EMAIL_USERNAME', "infoboboTPI@gmail.com");
13 define('EMAIL_PASSWORD', "Super2018");
14
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe UserManager contenant une fonction permettant de
                    connecter l'utilisateur au site web.
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  class UserManager
10 {
11     /**
12      * Valide la connexion d'un réparateur au site web.
13      *
14      * @param string $pseudo Le pseudo du réparateur
15      * @param string $pwd Le mot de passe du réparateur
16      *
17      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
18      * @return bool Retourne TRUE si le pseudo et le mot de passe sont corrects
19      *              ou FALSE si le mot de passe ou le pseudo est incorrect
20      */
21     public static function Connection($pseudo, $pwd)
22     {
23         $sql = "SELECT mdp
24                 FROM reparateur
25                 WHERE pseudo=:pseudo and mdp=:mdp";
26         $salt = "TPIBJ";
27         $passwordShal = sha1($salt . $pwd);
28         try {
29             $stmt = Database::prepare($sql);
30             $stmt->bindParam(':pseudo', $pseudo, PDO::PARAM_STR);
31             $stmt->bindParam(':mdp', $passwordShal, PDO::PARAM_STR);
32             $stmt->execute();
33             $result = $stmt->fetch(PDO::FETCH_ASSOC);
34             if ($result == null) {
35                 return FALSE;
36             }
37             if ($passwordShal == $result["mdp"]) {
38                 return TRUE;
39             } else {
40                 return FALSE;
41             }
42         } catch (Exception $e) {
43             return FALSE;
44         }
45     }
46     /**
47      * Vérifie si l'utilisateur est connecté ou non.
48      *
49      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
50      * @return bool Redirige l'utilisateur vers la page d'accueil si l'
                    utilisateur n'est pas connecté
51      */
52     public static function VerificateRoleUser()
53     {
54         if (!isset($_SESSION["isLogged"]) && $_SESSION["isLogged"] != true) {
55             header("location: index.php");
56         }
57     }
58 }

```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe AboutManager contenant les fonctions en rapport avec les
                    informations personnelles du réparateur,
5                  plus précisément de la table "infos_dynamiques".
6      Auteur: Borel-Jaquet Jonathan
7      Version: 1.0
8      Date: Mai 2019
9  */
10 class AboutManager
11 {
12     /**
13      * Récupère les informations personnelles du réparateur dans un objet About
14      *
15      * @throws bool Retourne false s'il y a un problème
16      * @author Jonathan Borel-Jaquet <jonathan.brlij@eduge.ch>
17      * @return About $a Retourne un objet About ou FALSE s'il y a un problème
18      */
19     public static function GetAboutInformation()
20     {
21         $sql = "SELECT id_infos_dynamiques,telephone,email,tarif,description
22                 FROM bj_tpi_bd.infos_dynamiques;";
23         try {
24             $stmt = Database::prepare($sql);
25             $stmt->execute();
26             $about = $stmt->fetch(PDO::FETCH_ASSOC);
27             $a = new About();
28             $a->id_about = intval($about["id_infos_dynamiques"]);
29             $a->phoneNumber = $about["telephone"];
30             $a->email = $about["email"];
31             $a->price = $about["tarif"];
32             $a->description = $about["description"];
33
34             return $a;
35         } catch (Exception $e) {
36             return FALSE;
37         }
38     }
39     /**
40      * Modifie les informations personnelles du réparateur.
41      *
42      * @param string $phoneNumber Le numéro de téléphone du réparateur
43      * @param string $email L'email du réparateur
44      * @param string $price Le tarif de la réparation
45      * @param string $description La description du réparateur
46      *
47      * @author Jonathan Borel-Jaquet <jonathan.brlij@eduge.ch>
48      * @return bool Retourne TRUE ou FALSE s'il y a un problème
49      */
50     public static function UpdateAboutInformation($phoneNumber, $email, $price,
51         $description)
52     {
53         $sql = "UPDATE `bj_tpi_bd`.`infos_dynamiques`
54                 SET `telephone`=:phoneNumber, `email` = :email, `tarif` = :
55 price, `description` = :description
56                 WHERE id_infos_dynamiques=1";
57         try {
58             $stmt = Database::prepare($sql);

```

```
57         $stmt->bindParam(':phoneNumber', $phoneNumber, PDO::PARAM_STR);
58         $stmt->bindParam(':email', $email, PDO::PARAM_STR);
59         $stmt->bindParam(':price', $price, PDO::PARAM_STR);
60         $stmt->bindParam(':description', $description, PDO::PARAM_STR);
61         $stmt->execute();
62         return TRUE;
63     } catch (Exception $e) {
64         return FALSE;
65     }
66 }
67 }
68
```



```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe EventManager contenant les fonctions en rapport avec les
                    événements créés par le réparateur,
5                  plus précisément de la table "evenement".
6      Auteur: Borel-Jaquet Jonathan
7      Version: 1.0
8      Date: Mai 2019
9  */
10 class EventManager
11 {
12     /**
13      * Enregistre un événement et modifie le statut d'une demande dans la base
        de données.
14     *
15     * @param string $id_request L'identifiant de la demande
16     * @param string $dateEventStart La date de début de l'événement
17     * @param string $dateEventEnd La date de fin de l'événement
18     *
19     * @author Jonathan Borel-Jaquet <jonathan.brlijq@eduge.ch>
20     * @return bool Retourne TRUE ou FALSE s'il y a un problème
21     */
22     public static function AddEvent($id_request, $dateEventStart, $dateEventEnd
23 )
24     {
25         $sql = "INSERT INTO `bj_tpi_bd`.`evenement` (`id_demande`, `date_debut`
26         `, `date_fin`, `id_reparateur`)
27             VALUES (:id_request, :dateEventStart, :dateEventEnd, 1)";
28         try {
29             $pdo = Database::getInstance();
30             $stmt = Database::prepare($sql);
31             $pdo->beginTransaction();
32             RequestManager::UpdateRequestStatusById($id_request,
33 STATUS_IN_PROGRESS);
34             $stmt->bindParam(':id_request', $id_request, PDO::PARAM_INT);
35             $stmt->bindParam(':dateEventStart', $dateEventStart, PDO::PARAM_STR
36 );
37             $stmt->bindParam(':dateEventEnd', $dateEventEnd, PDO::PARAM_STR);
38             $stmt->execute();
39             $pdo->commit();
40             return TRUE;
41         } catch (Exception $e) {
42             $pdo->rollBack();
43             return FALSE;
44         }
45     }
46     /**
47     * Récupère tous les événements en format JSON destiné au calendrier
        JavaScript FullCalendar.
48     *
49     * @throws bool Retourne FALSE s'il y a un problème
50     * @author Jonathan Borel-Jaquet <jonathan.brlijq@eduge.ch>
51     * @return string Retourne une chaîne de caractères contenant la
        représentation JSON des événements
52     * ou FALSE s'il y a un problème
53     */
54     public static function GetAllEventFormatJSON()
55     {
56         $sql = "SELECT e.date_debut,e.date_fin,c.nom,c.prenom

```

```
53         FROM bj_tpi_bd.evenement as e,bj_tpi_bd.demandes as d,  
bj_tpi_bd.clients as c  
54         WHERE d.id_demande=e.id_demande and c.id_client = d.id_client"  
;  
55     $arrAllEvent = array();  
56     try {  
57         $stmt = Database::prepare($sql);  
58         $stmt->execute();  
59         $result = $stmt->fetchAll(PDO::FETCH_ASSOC);  
60         foreach ($result as $event) {  
61             $arrEvent = array(  
62                 "title" => $event["nom"] . " " . $event["prenom"],  
63                 "start" => $event["date_debut"],  
64                 "end" => $event["date_fin"]  
65             );  
66             array_push($arrAllEvent, $arrEvent);  
67         }  
68  
69         return json_encode($arrAllEvent);  
70     } catch (Exception $e) {  
71         return FALSE;  
72     }  
73 }  
74 }  
75
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe StyleManager contenant les fonctions permettant de rendre
                    le site plus esthétique et lisible.
5      Auteur: Borel-Jaquet Jonathan
6      Version: 1.0
7      Date: Mai 2019
8  */
9  class StyleManager
10 {
11     /**
12      * Retourne la bonne couleur pour un statut.
13      *
14      * @param string $status Le statut choisi
15      *
16      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
17      * @return string $color Retourne la couleur du statut
18      */
19     public static function ColorStatus($status)
20     {
21         $color = "";
22         if ($status == STATUS_IN_PROGRESS) {
23             $color = "yellow";
24         }
25         if ($status == STATUS_OPEN) {
26             $color = "white";
27         }
28         if ($status == STATUS_PROCESSED) {
29             $color = "green";
30         }
31         if ($status == STATUS_REFUSED) {
32             $color = "red";
33         }
34         return $color;
35     }
36     /**
37      * Retourne le bon format d'une date récupérée en base de données.
38      *
39      * @param string $date La date à modifier
40      *
41      * @author Projet ProVélo École Entreprise
42      * @return string Retourne la date en format => 10 septembre 2019
43      */
44     public static function SqlDateToWritten($date)
45     {
46
47         $arrMonth = array("janvier", "février", "mars", "avril", "mai", "juin",
48 "juillet", "août", "septembre", "octobre", "novembre", "décembre");
49         //On transforme les string en int pour retirer le 0 à l'avant (05 -> 5)
50         $year = (int)explode('-', $date)[0];
51         $month = (int)explode('-', $date)[1] - 1;
52         $day = (int)explode('-', $date)[2];
53         $day .= ($day < 2);
54         return "$day $arrMonth[$month] $year";
55     }
56     /**
57      * Affiche un message d'erreur ou de confirmation avec Bootstrap.
58      *
59      * @param string $type Le type de message à afficher [success/danger]

```

```
59      * @param string $message Le message à afficher
60      *
61      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
62      * @return string une balise div d'alert avec Bootstrap
63      */
64      public static function showAlert($type, $message)
65      {
66          switch ($type) {
67              case ALERT_TYPE_SUCCESS:
68                  echo "<div class='alert alert-success mb-0' role='alert'>" .
$message . "</div>";
69                  break;
70              case ALERT_TYPE_FAILED:
71                  echo "<div class='alert alert-danger mb-0' role='alert'>" .
$message . "</div>";
72                  break;
73              default:
74                  return false;
75          }
76      }
77 }
78
```

```
1 <?php
2 /*
3     Projet: SOS INFOBOBO
4     Description: Classe ClientManager contenant les fonctions en rapport avec les
5                 informations personnelles du client,
6                 plus précisément de la table "clients".
7     Auteur: Borel-Jaquet Jonathan
8     Version: 1.0
9     Date: Mai 2019
10 */
11 class ClientManager
12 {
13     /**
14      * Enregistre les informations d'un client dans la base de données.
15      *
16      * @param string $firstName Le nom du client
17      * @param string $secondName Le prénom du client
18      * @param string $email L'email du client
19      * @param string $phoneNumber Le numéro de téléphone du client
20      *
21      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
22      * @return string $last_id Retourne l'id du client précédemment créé
23      */
24     public static function AddClient($firstName, $secondName, $email, $phoneNumber
25 )
26     {
27         $sql = "INSERT INTO `bj_tpi_bd`.`clients` (`nom`, `prenom`, `email`, `
28 telephone`)
29             VALUES (:firstName, :secondName, :email, :phoneNumber) ";
30         $pdo = Database::getInstance();
31         $stmt = Database::prepare($sql);
32         $stmt->bindParam(':firstName', $firstName, PDO::PARAM_STR);
33         $stmt->bindParam(':secondName', $secondName, PDO::PARAM_STR);
34         $stmt->bindParam(':email', $email, PDO::PARAM_STR);
35         $stmt->bindParam(':phoneNumber', $phoneNumber, PDO::PARAM_STR);
36         $stmt->execute();
37         $last_id = $pdo->lastInsertId();
38         return $last_id;
39     }
40 }
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe MailerManager contenant les fonctions permettant d'
envoyer un email
5          en utilisant la librairie externe Swift Mailer.
6      Auteur: Borel-Jaquet Jonathan
7      Version: 1.0
8      Date: Mai 2019
9  */
10 class MailerManager
11 {
12     /**
13      * Envoie un email au réparateur décrivant les informations de la demande
du client en utilisant la librairie Swift Mailer
14      *
15      * @param string $contactFirstName Le nom du client
16      * @param string $contactSecondName Le prénom du client
17      * @param string $contactEmail L'email du client
18      * @param string $contactPhoneNumber Le numéro de téléphone du client
19      * @param string $contactDescription La description du problème
informatique du client
20      *
21      * @throws Swift_TransportException
22      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
23      * @return bool Retourne TRUE si l'email a bien été envoyé
24      */
25     public static function SendMailToRepairer($contactFirstName,
$contactSecondName, $contactEmail, $contactPhoneNumber, $contactDescription)
26     {
27         $transport = Swift_SmtpTransport::newInstance(EMAIL_SERVER, EMAIL_PORT,
EMAIL_TRANS)
28             ->setUsername(EMAIL_USERNAME)
29             ->setPassword(EMAIL_PASSWORD);
30         try {
31             // Création d'une nouvelle instance de mail en utilisant le
transport créé précédemment
32             $mailer = Swift_Mailer::newInstance($transport);
33             // Création du message
34             $message = Swift_Message::newInstance();
35             // Le sujet du message
36             $message->setSubject("[SOS INFOBOBO] Nouvelle demande de réparation
reçu");
37             // Qui envoie le message
38             $message->setFrom(array('infoboboTPI@gmail.ch' => 'SOS Infobobo
réparation informatique'));
39             // A qui on envoie le message
40             $message->setTo(array(EMAIL_USERNAME));
41             //Création du message html à envoyer
42             $body =
43                 "<html>" .
44                 " <head></head>" .
45                 " <body>" .
46                 " <p>Une nouvelle demande de réparation informatique vient d'
être créée.</p>" .
47                 " <h4>Details de la demande de réparation informatique :</h4>"
.
48                 "<table border='1'>" .
49                 "<tr>" .
50                 "<td>Nom :</td>" .

```

```

51         "<td>" . $contactFirstName . "</td>" .
52         "</tr>" .
53         "<tr>" .
54         "<td>Prénom :</td>" .
55         "<td>" . $contactSecondName . "</td>" .
56         "</tr>" .
57         "<tr>" .
58         "<td>Email :</td>" .
59         "<td>" . $contactEmail . "</td>" .
60         "</tr>" .
61         "<tr>" .
62         "<td>Numéro de téléphone :</td>" .
63         "<td>" . $contactPhoneNumber . "</td>" .
64         "</tr>" .
65         "<tr>" .
66         "<td>Description du problème :</td>" .
67         "<td>" . $contactDescription . "</td>" .
68         "</tr>" .
69         "</table>" .
70         " </body>" .
71         "</html>";
72         //Asignation du message et tu type à envoyer
73         $message->setBody($body, 'text/html');
74         //Envoie du mail
75         $mailer->send($message);
76         return true;
77     } catch (Swift_TransportException $e) {
78         echo "Problème d'envoi de message: " . $e->getMessage();
79         exit();
80     }
81 }
82 /**
83  * Envoie un email au client décrivant les informations de sa demande et
le nouveau statut
84  * attribué à celle-ci en utilisant la librairie Swift Mailer.
85  *
86  * @param string $newStatus Le nouveau statut de la demande
87  * @param string $client Un objet Client
88  * @param string $request Un objet Request
89  *
90  * @throws Swift_TransportException
91  * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
92  * @return bool Retourne TRUE si l'email a bien été envoyé
93  */
94 public static function SendMailToClient($newStatus, $client, $request)
95 {
96     $transport = Swift_SmtpTransport::newInstance(EMAIL_SERVER, EMAIL_PORT
, EMAIL_TRANS)
97     ->setUsername(EMAIL_USERNAME)
98     ->setPassword(EMAIL_PASSWORD);
99     try {
100         // On crée un nouvelle instance de mail en utilisant le transport
créé précédemment
101         $mailer = Swift_Mailer::newInstance($transport);
102         // Création du message
103         $message = Swift_Message::newInstance();
104         // Le sujet du message
105         $message->setSubject("[SOS INFOBOBO] Changement de statut");
106         // Qui envoie le message
107         $message->setFrom(array('infoboboTPI@gmail.ch' => 'SOS Infobobo

```

```

107 réparation informatique'));
108     // A qui on envoie le message
109     $message->setTo(array($client->email));
110
111     //Création du message html à envoyer
112     $body =
113         "<html>" .
114         " <head></head>" .
115         " <body>" .
116         " <p>Bonjour,</p>" .
117         " <p>Le statut de votre demande de réparation informatique a
été modifié en : " . $newStatus . " </p>" .
118         " <h4>Details de la demande de réparation informatique :</h4>"
119         " .
120         "<table border='1'>" .
121         "<tr>" .
122         "<td>Nom :</td>" .
123         "<td>" . $client->email . "</td>" .
124         "</tr>" .
125         "<tr>" .
126         "<td>Prénom :</td>" .
127         "<td>" . $client->firstName . "</td>" .
128         "</tr>" .
129         "<tr>" .
130         "<td>Email :</td>" .
131         "<td>" . $client->email . "</td>" .
132         "</tr>" .
133         "<tr>" .
134         "<td>Numéro de téléphone :</td>" .
135         "<td>" . $client->phoneNumber . "</td>" .
136         "</tr>" .
137         "<tr>" .
138         "<td>Description du problème :</td>" .
139         "<td>" . $request->description . "</td>" .
140         "</tr>" .
141         "</table>" .
142         " </body>" .
143         "</html>";
144     //Asignation du message et tu type à envoyer
145     $message->setBody($body, 'text/html');
146     //Envoie du mail
147     $mailer->send($message);
148     return true;
149 } catch (Swift_TransportException $e) {
150     echo "Problème d'envoi de message: " . $e->getMessage();
151     exit();
152 }
153 }
154

```



```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe OpinionManager contenant les fonctions en rapport avec
5                  les avis créés par les clients,
6                  plus précisément de la table "avis".
7      Auteur: Borel-Jaquet Jonathan
8      Version: 1.0
9      Date: Mai 2019
10 */
11 class OpinionManager
12 {
13     /**
14      * Enregistre un avis non validé dans la base de données.
15      *
16      * @param string $comment Le commentaire de l'avis
17      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
18      * @return bool Retourne TRUE ou FALSE s'il y a un problème
19      */
20     public static function AddOpinion($comment)
21     {
22         $sql = "INSERT INTO `bj_tpi_bd`.`avis` (`date`, `comment`, `est_valide`
23             , `id_reparateur`)
24             VALUES (:date, :comment, '0', '1')";
25         $date = date("Y-m-d");
26         try {
27             $stmt = Database::prepare($sql);
28
29             $stmt->bindParam(':date', $date, PDO::PARAM_STR);
30             $stmt->bindParam(':comment', $comment, PDO::PARAM_STR);
31             $stmt->execute();
32             return TRUE;
33         } catch (Exception $e) {
34             throw $e;
35             return FALSE;
36         }
37     }
38     /**
39      * Récupère tous les avis validés de la base de données.
40      *
41      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
42      * @return array[Opinion] $arrOpinion Retourne un tableau d'objet Opinion
43      ou FALSE s'il y a un problème
44      */
45     public static function GetOpinionValidate()
46     {
47         $sql = "SELECT id_avis,date,comment,est_valide,id_reparateur
48             FROM bj_tpi_bd.avis
49             WHERE est_valide = 1
50             ORDER BY date DESC";
51         $arrOpinion = array();
52         try {
53             $stmt = Database::prepare($sql);
54             $stmt->execute();
55             $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
56             foreach ($result as $opinion) {
57                 $o = new Opinion();
58                 $o->id_opinion = intval($opinion["id_avis"]);
59                 $o->date = $opinion["date"];

```

```

58         $o->comment = $opinion["comment"];
59         $o->is_validate = boolval($opinion["est_valide"]);
60         $o->id_repairer = intval($opinion["id_reparateur"]);
61         array_push($arrOpinion, $o);
62     }
63     return $arrOpinion;
64 } catch (Exception $e) {
65     return FALSE;
66 }
67 }
68 /**
69  * Récupère tous les avis non validés de la base de données.
70  *
71  * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
72  * @return array[Opinion] $arrOpinion Retourne un tableau d'objet Opinion
73  * ou FALSE s'il y a un problème
74  */
75 public static function GetOpinionNotValidate()
76 {
77     $sql = "SELECT id_avis,date,comment,est_valide,id_reparateur
78           FROM bj_tpi_bd.avis
79           WHERE est_valide = 0
80           ORDER BY date ASC";
81     $arrOpinion = array();
82     try {
83         $stmt = Database::prepare($sql);
84         $stmt->execute();
85         $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
86         foreach ($result as $opinion) {
87             $o = new Opinion();
88             $o->id_opinion = intval($opinion["id_avis"]);
89             $o->date = $opinion["date"];
90             $o->comment = $opinion["comment"];
91             $o->is_validate = boolval($opinion["est_valide"]);
92             $o->id_repairer = intval($opinion["id_reparateur"]);
93             array_push($arrOpinion, $o);
94         }
95         return $arrOpinion;
96     } catch (Exception $e) {
97         return FALSE;
98     }
99 }
100 /**
101  * Valide un avis dans la base de données en utilisant son identifiant.
102  *
103  * @param string $id_opinion L'identifiant de l'avis
104  *
105  * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
106  * @return bool Retourne TRUE ou FALSE s'il y a un problème
107  */
108 public static function ValidateOpinionById($id_opinion)
109 {
110     $sql = "UPDATE `bj_tpi_bd`.`avis`
111           SET `est_valide` = '1'
112           WHERE (`id_avis` = :id_opinion)";
113     try {
114         $stmt = Database::prepare($sql);
115         $stmt->bindParam(':id_opinion', $id_opinion, PDO::PARAM_STR);
116         $stmt->execute();
117         return TRUE;

```

```
117     } catch (Exception $e) {
118         return FALSE;
119     }
120 }
121 /**
122  * Supprime un avis dans la base de données en utilisant son identifiant.
123  *
124  * @param string $id_opinion L'identifiant de l'avis
125  *
126  * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
127  * @return bool Retourne TRUE ou FALSE s'il y a un problème
128  */
129 public static function DeleteOpinionById($id_opinion)
130 {
131     $sql = "DELETE
132           FROM `bj_tpi_bd`.`avis`
133           WHERE (`id_avis` = :id_opinion)";
134     try {
135         $stmt = Database::prepare($sql);
136         $stmt->bindParam(':id_opinion', $id_opinion, PDO::PARAM_STR);
137         $stmt->execute();
138         return TRUE;
139     } catch (Exception $e) {
140         return FALSE;
141     }
142 }
143 }
144
```

```

1  <?php
2  /*
3      Projet: SOS INFOBOBO
4      Description: Classe RequestManager contenant les fonctions en rapport avec
5                  les demandes de réparation
6                  informatiques créées par les clients, plus précisément de la
7                  table "demandes".
8      Auteur: Borel-Jaquet Jonathan
9      Version: 1.0
10     Date: Mai 2019
11 */
12
13 class RequestManager
14 {
15     /**
16      * Enregistre une demande de réparation informatique et le client l'ayant
17      * créé dans la base de données.
18      *
19      * @param string $firstName Le nom du client
20      * @param string $secondName Le prénom du client
21      * @param string $email L'email du client
22      * @param string $phoneNumber Le numéro de téléphone du client
23      * @param string $description La description du problème informatique du
24      * client
25      *
26      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
27      * @return bool Retourne TRUE ou FALSE s'il y a un problème
28      */
29     public static function AddRequest($firstName, $secondName, $email,
30                                     $phoneNumber, $description)
31     {
32         $sql = "INSERT INTO `bj_tpi_bd`.`demandes` (`id_client`, `description`
33             , `statut`)
34             VALUES (:id_client, :description, :statut);";
35         $status = STATUS_OPEN;
36
37         try {
38             $pdo = Database::getInstance();
39             $stmt = Database::prepare($sql);
40             $pdo->beginTransaction();
41             $id_client = ClientManager::AddClient($firstName, $secondName,
42                 $email, $phoneNumber);
43             $stmt->bindParam(':id_client', $id_client, PDO::PARAM_INT);
44             $stmt->bindParam(':description', $description, PDO::PARAM_STR);
45             $stmt->bindParam(':statut', $status, PDO::PARAM_STR);
46             $stmt->execute();
47             $pdo->commit();
48             return TRUE;
49         } catch (Exception $e) {
50             $pdo->rollBack();
51             return FALSE;
52         }
53     }
54
55     /**
56      * Récupère toutes les demandes de réparation informatique de la base de
57      * données.
58      *
59      * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
60      * @return array[Client[],Request[]] $arrRequest Retourne un tableau
61      * contenant des tableaux d'objet Client et Request
62      *
63      * ou FALSE s'il y a un problème

```

```

52     */
53     public static function GetAllRequest()
54     {
55         $sql = "SELECT clients.id_client,demandes.id_demande, nom, prenom,
email,telephone,description,statut
56             FROM clients, demandes
57             WHERE clients.id_client = demandes.id_client
58             ORDER BY statut ASC, id_demande DESC";
59         $arrRequest = array();
60         try {
61             $stmt = Database::prepare($sql);
62             $stmt->execute();
63             $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
64             foreach ($result as $requestClient) {
65                 $arrClientRequest = array();
66                 $c = new Client();
67                 $c->id_client = intval($requestClient["id_client"]);
68                 $c->firstName = $requestClient["nom"];
69                 $c->secondName = $requestClient["prenom"];
70                 $c->email = $requestClient["email"];
71                 $c->phoneNumber = $requestClient["telephone"];
72                 array_push($arrClientRequest, $c);
73                 $r = new Request();
74                 $r->id_request = intval($requestClient["id_demande"]);
75                 $r->id_client = intval($requestClient["id_client"]);
76                 $r->description = $requestClient["description"];
77                 $r->status = $requestClient["statut"];
78                 array_push($arrClientRequest, $r);
79                 array_push($arrRequest, $arrClientRequest);
80             }
81             return $arrRequest;
82         } catch (Exception $e) {
83             return FALSE;
84         }
85     }
86     /**
87      * Récupère une demande de réparation informatique avec les informations
du client l'ayant créé
88      * de la base de données grâce à son identifiant.
89      *
90      * @param string $id_request L'identifiant de la demande
91      *
92      * @author Jonathan Borel-Jaquet <jonathan.brlij@eduge.ch>
93      * @return array[Client,Request] Retourne un tableau contenant un objet
Client et Request
94      *
95      * ou FALSE s'il y a un problème
96      */
97     public static function GetRequestById($id_request)
98     {
99         $sql = "SELECT clients.id_client,demandes.id_demande, nom, prenom,
email,telephone,description,statut
100             FROM clients, demandes
101             WHERE clients.id_client = demandes.id_client and demandes.
id_demande =:id_request ";
102         $arrClientRequest = array();
103         try {
104             $stmt = Database::prepare($sql);
105             $stmt->bindParam(':id_request', $id_request, PDO::PARAM_STR);
106             $stmt->execute();
107             $result = $stmt->fetch(PDO::FETCH_ASSOC);

```

```

107         $c = new Client();
108         $c->id_client = intval($result["id_client"]);
109         $c->firstName = $result["nom"];
110         $c->secondName = $result["prenom"];
111         $c->email = $result["email"];
112         $c->phoneNumber = $result["telephone"];
113         array_push($arrClientRequest, $c);
114         $r = new Request();
115         $r->id_request = intval($result["id_demande"]);
116         $r->id_client = intval($result["id_client"]);
117         $r->description = $result["description"];
118         $r->status = $result["statut"];
119         array_push($arrClientRequest, $r);
120         return $arrClientRequest;
121     } catch (Exception $e) {
122         return FALSE;
123     }
124 }
125 /**
126  * Modifie le statut d'une demande de réparation informatique grâce à son
    identifiant.
127  *
128  * @param string $id_request L'identifiant de la demande
129  * @param string $status Le nouveau statut de la demande
130  *
131  * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
132  * @return bool Retourne TRUE ou FALSE s'il y a un problème
133  */
134 public static function UpdateRequestStatusById($id_request, $status)
135 {
136     $sql = "UPDATE `bj_tpi_bd`.`demandes`
137           SET `statut` = :status
138           WHERE (`id_demande` = :id_request)";
139     try {
140         $stmt = Database::prepare($sql);
141         $stmt->bindParam(':id_request', $id_request, PDO::PARAM_STR);
142         $stmt->bindParam(':status', $status, PDO::PARAM_STR);
143         $stmt->execute();
144         return TRUE;
145     } catch (Exception $e) {
146         return FALSE;
147     }
148 }
149 /**
150  * Récupère toutes les demandes de réparation informatique de statut "
    Ouverte" de la base de données.
151  *
152  * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
153  * @return array[Client,Request] $arrRequest Retourne un tableau d'objet
    Client et Request
154  *
155  * ou FALSE s'il y a un problème
156  */
157 public static function GetOpenRequest ()
158 {
159     $sql = "SELECT clients.id_client,demandes.id_demande, nom, prenom,
160           email,telephone,description,statut
161           FROM clients, demandes
162           WHERE clients.id_client = demandes.id_client
163           AND demandes.statut = 'OUVERTE' ";
164     $arrRequest = array();

```

```

163     try {
164         $stmt = Database::prepare($sql);
165         $stmt->execute();
166         $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
167         foreach ($result as $requestClient) {
168             $arrClientRequest = array();
169             $c = new Client();
170             $c->id_client = intval($requestClient["id_client"]);
171             $c->firstName = $requestClient["nom"];
172             $c->secondName = $requestClient["prenom"];
173             $c->email = $requestClient["email"];
174             $c->phoneNumber = $requestClient["telephone"];
175             array_push($arrClientRequest, $c);
176
177             $r = new Request();
178             $r->id_request = intval($requestClient["id_demande"]);
179             $r->id_client = intval($requestClient["id_client"]);
180             $r->description = $requestClient["description"];
181             $r->status = $requestClient["statut"];
182             array_push($arrClientRequest, $r);
183             array_push($arrRequest, $arrClientRequest);
184         }
185         return $arrRequest;
186     } catch (Exception $e) {
187         return FALSE;
188     }
189 }
190 /**
191  * Récupère toutes les demandes de réparation informatique de statut "
192  * Traitée" de la base de données triée par mois et par année.
193  *
194  * @param string $year L'année choisie
195  *
196  * @author Jonathan Borel-Jaquet <jonathan.brljq@eduge.ch>
197  * @return array $result Retourne un tableau associatif représentant le
198  * nombre de réparation
199  * effectué par mois dans une année ou FALSE s'il y
200  * a un problème
201  */
202 public static function GetProcessedRequestOrderByMonthAndYear($year)
203 {
204     $sql = "SELECT MONTH( e.date_fin ) AS month, YEAR( e.date_fin ) AS
205     year, COUNT( * ) as nbRequest
206     FROM bj_tpi_bd.demandes as d, bj_tpi_bd.evenement as e
207     WHERE d.id_demande = e.id_demande and d.statut ='TRAITEE' and YEAR( e.
208     date_fin ) =:year
209     GROUP BY year, month
210     ORDER BY month ASC";
211     try {
212         $stmt = Database::prepare($sql);
213         $stmt->bindParam(':year', $year, PDO::PARAM_INT);
214         $stmt->execute();
215         $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
216         return $result;
217     } catch (Exception $e) {
218         return FALSE;
219     }
220 }

```