

```

1 <?php
2 /*
3 *      Author           :   Fujise Thomas.
4 *      Project          :   ProjectTPI.
5 *      Page             :   Index.
6 *      Brief            :   Home page.
7 *      Date             :   02.06.2020.
8 */
9
10 /*Requirements */
11 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FUserManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FSessionManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/view/
    FItineraryView.php';
14 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
15 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FCodeManager.php';
16
17 if (session_status() == PHP_SESSION_NONE) {
18     session_start();
19 }
20 if(FSessionManager::GetItineraryFilter() != false){
21     $itineraries = FItineraryManager::GetInstance()->GetAllWithFilter(
        FSessionManager::GetItineraryFilter());
22 }
23 else{
24     $itineraries = FItineraryManager::GetInstance()->GetAll();
25 }
26 ?>
27 <!DOCTYPE html>
28 <html lang="FR" dir="ltr">
29 <head>
30     <meta charset="utf-8">
31     <meta name="viewport" content="width=device-width, initial-scale
        =1, shrink-to-fit=no">
32     <meta name="description" content="">
33     <meta name="author" content="">
34     <title>Home</title>
35     <!-- Font Awesome icons (free version)-->
36     <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
        /5.13.0/js/all.min.js" crossorigin="anonymous"></script>
37     <!-- Core theme CSS (includes Bootstrap)-->
38     <link href="./assets/css/styles.css" rel="stylesheet">
39     <!-- Fonts CSS-->
40     <link rel="stylesheet" href="./assets/css/heading.css">
41     <link rel="stylesheet" href="./assets/css/body.css">
42     <!-- Maquest CSS and JS Load-->
43     <link type="text/css" rel="stylesheet" href="https://api.mqcdn.com
        /sdk/mapquest-js/v1.3.2/mapquest.css"/>
44     <script src="https://api.mqcdn.com/sdk/mapquest-js/v1.3.2/mapquest
        .js"></script>
45
46 </head>
47 <body id="page-top">

```

```

48 <?php
49
50 //Show the good navbar depends if logged and if user is admin
51 if (FUserManager::getInstance()->isLoggedIn()) {
52     if (FUserManager::getInstance()->isAllowed(FSessionManager::
53         getUserLogged())) {
54         include "../inc/navbar/navbarAdmin.php";
55     }
56     else {
57         include "../inc/navbar/navbarLogged.php";
58     }
59 }
60 else {
61     include "../inc/navbar/navbarNotLogged.php";
62 }
63 ?>
64 <header class="masthead bg-primary text-white text-center">
65     <div class="container d-flex align-items-center flex-column">
66         <!-- Masthead Heading-->
67         <h1 class="masthead-heading mb-0">TRAVLER</h1>
68         <!-- Icon Divider-->
69         <div class="divider-custom divider-light">
70             <div class="divider-custom-line"></div>
71             <div class="divider-custom-icon"><i class="fas fa-star"></i></div>
72             <div class="divider-custom-line"></div>
73         </div>
74         <!-- Masthead Subheading-->
75         <p class="pre-wrap masthead-subheading font-weight-light mb-0">
76             <i>YOUR ITINERARY FINDER</i> </p>
77     </header>
78     <section class="page-section portfolio" id="portfolio">
79         <div class="container">
80             <!-- Portfolio Section Heading-->
81             <div class="text-center">
82                 <h2 class="page-section-heading text-secondary mb-0 d-
83                 inline-block">ITINERARY</h2>
84             </div>
85             <!-- Icon Divider-->
86             <div class="divider-custom">
87                 <div class="divider-custom-line"></div>
88                 <div class="divider-custom-icon"><i class="fas fa-star"></i></div>
89             </div>
90             <div class="container">
91                 <form name="itineraryFilter">
92                     <div class="form-group">
93                         <div class="row">
94                             <div class="col">
95                                 <label for="countryFilter">Country :</
96                                 label>
97                                 <select name="countryFilter" id="
98                                 countryFilter" size="1" class="custom-select">
99                                     <option id="None" selected>None</option>

```

```

98         <?php
99             $countryManager = new FCodeManager();
100             foreach ($countryManager::GetInstance
101             ()->getAllCountry() as $country) {
102                 echo "<option id=". $country->Code
103                 . "> " . $country->Name . "</option>";
104             }
105         ?>
106     </select>
107 </div>
108 <div class="col">
109     <label for="rateFilter">Rate Min:</label>
110     <select name="rateFilter" id="rateFilter" size
111     ="1" class="custom-select">
112         <?php
113         for ($i = 0;$i<=10;$i++) {
114             echo '<option id="'. $i .' ">' . $i .
115             '</option>';
116         }
117         ?>
118     </select>
119 </div>
120 <div class="col">
121     <div class="row">
122         <label for="distanceMinFilter">
123         Distance Min (KM) :</label>
124         <input class="form-control" id="
125         distanceMinFilter" type="number" value="0" data-decimals="0" min="1
126         " max="999" step="1"/>
127     </div>
128     <div class="row">
129         <label for="distanceMaxFilter">
130         Distance Max (KM) :</label>
131         <input class="form-control" id="
132         distanceMaxFilter" type="number" value="0" data-decimals="0" min="1
133         " max="999" step="1"/>
134     </div>
135 </div>
136 <div class="col">
137     <div class="row">
138         <label for="durationMinFilter">
139         Duration Min (H) :</label>
140         <input type="time" id="
141         durationMinFilter" class="form-control" placeholder="Select time">
142     </div>
143     <div class="row">
144         <label for="durationMaxFilter">Duration
145         Max (H) :</label>
146         <input type="time" id="
147         durationMaxFilter" class="form-control" placeholder="Select time">
148     </div>
149 </div>
150 <div class="row">
151     <div class="col">
152         <button type="submit" id="

```

```

141     itineraryFilterCancel" class="form-control btn btn-outline-danger"
142     name="itineraryFilter" >Cancel Filter</button>
143         </div>
144         <div class="col">
145             <button type="submit" id="itineraryFilter
146             " class="form-control btn btn-outline-primary" name="
147             itineraryFilter" >Filter</button>
148         </div>
149     </div>
150 </form>
151 </div>
152 <!-- Portfolio Grid Items-->
153 <div class="row justify-content-center">
154     <!-- Portfolio Items-->
155     <?php echo FItnineraryView::DisplayItineraries($itineraries
156 );?>
157 </div>
158 </div>
159 </section>
160
161 <footer class="footer text-center">
162     <div class="container">
163         <div class="row">
164             <!-- Footer Location-->
165             <div class="col-lg-4 mb-5 mb-lg-0">
166                 <h4 class="mb-4">LOCATION</h4>
167                 <p class="pre-wrap lead mb-0">CFPT Informatique
168 Chemin Gérard-De-Ternier 10
169 1213 Lancy</p>
170             </div>
171             <!-- Footer Middle Section-->
172             <div class="col-lg-4 mb-5 mb-lg-0">
173
174                 </div>
175                 <!-- Footer About Text-->
176                 <div class="col-lg-4">
177                     <h4 class="mb-4">ABOUT TRAVLER</h4>
178                     <p class="pre-wrap lead mb-0">Travler is a project
179 made by Thomas Fujise for his CFC work in the CFPT</p>
180                 </div>
181             </div>
182         </div>
183 </footer>
184 <!-- Copyright Section-->
185 <section class="copyright py-4 text-center text-white">
186     <div class="container"><small class="pre-wrap">Copyright
187     Travler 2020</small></div>
188 </section>
189 <!-- Scroll to Top Button (Only visible on small and extra-small
190 screen sizes)-->
191 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-
192 scroll-trigger d-block text-center text-white rounded" href="#page-
193 top"><i class="fa fa-chevron-up"></i></a></div>
194 <!-- Bootstrap core JS-->
195 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
196 jquery.min.js"></script>

```

```

187 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.bundle.min.js"></script>
188 <!-- Third party plugin JS-->
189 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
    /1.4.1/jquery.easing.min.js"></script>
190 <!-- Core theme JS-->
191 <script src="./assets/js/script.js"></script>
192 <!-- Include -->
193 <!-- Display maps on itineraries card with mapquest-->
194 <script src="./assets/js/home.js"></script>
195 </body>
196 </html>

```

Listing 1: index.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjectTPI.
5 *      Page             :  Login.
6 *      Brief            :  Login page.
7 *      Date             :  02.06.2020.
8 */
9
10 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FUserManager.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/model/
    FUser.php';
12
13 if (session_status() == PHP_SESSION_NONE) {
14     session_start();
15 }
16 ?>
17 <!DOCTYPE html>
18 <html lang="FR" dir="ltr">
19 <head>
20     <meta charset="utf-8">
21     <meta name="viewport" content="width=device-width, initial-scale
        =1, shrink-to-fit=no">
22     <meta name="description" content="">
23     <meta name="author" content="">
24     <title>Login</title>
25     <!-- Font Awesome icons (free version)-->
26     <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
        /5.13.0/js/all.min.js" crossorigin="anonymous"></script>
27     <!-- Core theme CSS (includes Bootstrap)-->
28     <link href="./assets/css/styles.css" rel="stylesheet">
29     <!-- Fonts CSS-->
30     <link rel="stylesheet" href="./assets/css/heading.css">
31     <link rel="stylesheet" href="./assets/css/body.css">
32 </head>
33 <body>
34 <?php
35     //Show the good navbar depends if logged and if user is admin
36     if (FUserManager::getInstance()->isLoggedIn()) {
37         if (FUserManager::getInstance()->isAllowed($_SESSION['
            userLogged'])) {
38             include "./inc/navbar/navbarAdmin.php";

```

```

39     }
40     else {
41         include "../inc/navbar/navbarLogged.php";
42     }
43 }
44 else {
45     include "../inc/navbar/navbarNotLogged.php";
46 }
47 ?>
48 <section class="page-section mb-0" id="formLogin">
49     <div class="container">
50         <div class="row justify-content-center mt-4">
51             <div class="col-md-12">
52                 <div class="card">
53                     <div class="card-header">Login</div>
54                     <div class="card-body">
55                         <form name="login">
56                             <div class="form-group">
57                                 <div class="row">
58                                     <div class="col">
59                                         <label for="username">Username :</
60                                         label>
61                                             <input type="text" class="form
62                                             -control" id="username" placeholder="Username" name="username"
63                                             required>
64                                             <p id="errorLogin" class="
65                                             errormsg">Wrong username or password</p>
66                                             <p id="errorActivation" class="
67                                             "errormsg">Account not activate</p>
68                                             <p id="errorBlocked" class="
69                                             errormsg">Account blocked</p>
70                                         </div>
71                                         <div class="col">
72                                             <label for="password">Password :</
73                                             label>
74                                             <input type="password" class="
75                                             form-control" id="password" placeholder="*****" name="password"
76                                             required>
77                                             <p id="errorParam" class="
78                                             errormsg">Error with the server</p>
79                                         </div>
80                                     </div>
81                                 </div>
82                                 <div class="form-group">
83                                     <label for="">&nbsp;</label>
84                                     <button type="submit" id="login" class="
85                                     form-control btn btn-outline-primary" name="formLogin">Sign in</
86                                     button>
87                                     <a href="../register.php">Not register yet
88                                     ?</a>
89                                 </div>
90                             </div>
91                         </form>
92                     </div>
93                 </div>
94             </div>
95         </div>
96     </div>
97 </section>

```

```

83 </div>
84     </div>
85 </section>
86 <footer class="footer text-center">
87     <div class="container">
88         <div class="row">
89             <!-- Footer Location-->
90             <div class="col-lg-4 mb-5 mb-lg-0">
91                 <h4 class="mb-4">LOCATION</h4>
92                 <p class="pre-wrap lead mb-0">CFPT Informatique
93 Chemin Gérard-De-Ternier 10
94 1213 Lancy</p>
95             </div>
96             <!-- Footer Middle Section-->
97             <div class="col-lg-4 mb-5 mb-lg-0">
98
99             </div>
100            <!-- Footer About Text-->
101            <div class="col-lg-4">
102                <h4 class="mb-4">ABOUT TRAVLER</h4>
103                <p class="pre-wrap lead mb-0">Travler is a project
made by Thomas Fujise for his CFC work in the CFPT</p>
104            </div>
105        </div>
106    </div>
107 </footer>
108 <!-- Copyright Section-->
109 <section class="copyright py-4 text-center text-white">
110     <div class="container"><small class="pre-wrap">Copyright
Travler 2020</small></div>
111 </section>
112 <!-- Scroll to Top Button (Only visible on small and extra-small
screen sizes)-->
113 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-
scroll-trigger d-block text-center text-white rounded" href="#page-
top"><i class="fa fa-chevron-up"></i></a></div>
114 <!-- Bootstrap core JS-->
115 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
jquery.min.js"></script>
116 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
bootstrap.bundle.min.js"></script>
117 <!-- Third party plugin JS-->
118 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
/1.4.1/jquery.easing.min.js"></script>
119 <!-- Core theme JS-->
120 <script src="./assets/js/script.js"></script>
121 <!-- Include constants-->
122 <script src="./constants/constants.js"></script>
123 <!-- Ajax call to send forms field -->
124 <script src="./assets/js/login.js"></script>
125 </body>
126 </html>

```

Listing 2: login.php

```

1 <?php
2 /*
3 *      Author      :  Fujise Thomas.

```

```

4 *      Project      : ProjectTPI.
5 *      Page        : Register.
6 *      Brief       : Register page.
7 *      Date        : 02.06.2020.
8 */
9
10 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FDatabaseManager.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FCodeManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FUserManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FSessionManager.php';
14
15 if (session_status() == PHP_SESSION_NONE) {
16     session_start();
17 }
18 ?>
19 <!DOCTYPE html>
20 <html lang="FR" dir="ltr">
21 <head>
22     <meta charset="utf-8">
23     <meta name="viewport" content="width=device-width, initial-scale
        =1, shrink-to-fit=no">
24     <meta name="description" content="">
25     <meta name="author" content="">
26     <title>Register</title>
27     <!-- Font Awesome icons (free version)-->
28     <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
        /5.13.0/js/all.min.js" crossorigin="anonymous"></script>
29     <!-- Core theme CSS (includes Bootstrap)-->
30     <link href="./assets/css/styles.css" rel="stylesheet">
31     <!-- Fonts CSS-->
32     <link rel="stylesheet" href="./assets/css/heading.css">
33     <link rel="stylesheet" href="./assets/css/body.css">
34 </head>
35 <body>
36 <?php
37
38     //Show the good navbar depends if logged and if user is admin
39     if (FUserManager::getInstance()->isLoggedIn()) {
40         if (FUserManager::getInstance()->isAllowed(FSessionManager::
            getUserLogged())) {
41             include "./inc/navbar/navbarAdmin.php";
42         }
43         else {
44             include "./inc/navbar/navbarLogged.php";
45         }
46     }
47     else {
48         include "./inc/navbar/navbarNotLogged.php";
49     }
50
51 ?>
52 <section class="page-section mb-0" id="formRegister">
53     <div class="container">

```



```

54     <div class="row justify-content-center mt-4">
55         <div class="col-md-12">
56             <div class="card">
57                 <div class="card-header">Register as new user</div>
58                 <div class="card-body">
59                     <form enctype="multipart/form-data">
60                         <div class="form-group">
61                             <div class="row">
62                                 <div class="col">
63                                     <label for="nickname">Nickname :</
64                                     label>
65                                     <input type="text" class="form-
66                                     control" id="nicknameUser" placeholder="Nickname" name="
67                                     nicknameUser" required>
68                                     <p id="errorNickname" class="
69                                     errormsg">This nickname has been already registered</p>
70                                     </div>
71                                     </div>
72                                     <div class="form-group">
73                                         <div class="row">
74                                             <div class="col">
75                                                 <label for="emailUser">Email :</
76                                                 label>
77                                                 <input type="email" class="form-
78                                                 control" id="emailUser" placeholder="Email" name="emailUser"
79                                                 required>
80                                                 <p id="errorEmail" class="errormsg
81                                                 ">This email has been already registered</p>
82                                                 </div>
83                                                 </div>
84                                                 <div class="form-group">
85                                                     <div class="row">
86                                                         <div class="col">
87                                                             <label for="password">Password :</
88                                                             label>
89                                                             <input type="password" class="form
90                                                             -control" id="password" placeholder="*****" name="password"
91                                                             required>
92                                                             </div>
93                                                             <div class="col">
94                                                                 <label for="verifyPassword">
95                                                                    Confirm Password :</label>
96                                                                    <input type="password" class="form
97                                                                    -control" id="verifyPassword" placeholder="*****" name="
98                                                                    verifyPassword" required>
99                                                                    </div>
100                                                                </div>
101                                                                <p id="errorParam" class="errormsg">
102                                                                    Register fail, please try again</p>
103                                                                <p id="errorDifferentPassword" class="
104                                                                errormsg">Both password don't match</p>
105                                                                <p id="errorPasswordMatch" class="errormsg
106                                                                ">Your password don't match the requirements</p>
107                                                                </div>

```

```

94         <div class="form-group">
95             <label for="">&nbsp;</label>
96             <button type="submit" class="form-
control btn btn-outline-primary" id="registerUser" name="
registerUser">Register</button>
97             <a href="./login.php">Already register
? Log in</a>
98         </div>
99     </form>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </section><footer class="footer text-center">
108     <div class="container">
109         <div class="row">
110             <!-- Footer Location-->
111             <div class="col-lg-4 mb-5 mb-lg-0">
112                 <h4 class="mb-4">LOCATION</h4>
113                 <p class="pre-wrap lead mb-0">CFPT Informatique
114 Chemin Gérard-De-Ternier 10
115 1213 Lancy</p>
116             </div>
117             <!-- Footer Middle Section-->
118             <div class="col-lg-4 mb-5 mb-lg-0">
119
120             </div>
121             <!-- Footer About Text-->
122             <div class="col-lg-4">
123                 <h4 class="mb-4">ABOUT TRAVLER</h4>
124                 <p class="pre-wrap lead mb-0">Travler is a project
made by Thomas Fujise for his CFC work in the CFPT</p>
125             </div>
126         </div>
127     </div>
128 </footer>
129 <!-- Copyright Section-->
130 <section class="copyright py-4 text-center text-white">
131     <div class="container"><small class="pre-wrap">Copyright
Travler 2020</small></div>
132 </section>
133 <!-- Scroll to Top Button (Only visible on small and extra-small
screen sizes)-->
134 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-
scroll-trigger d-block text-center text-white rounded" href="#page-
top"><i class="fa fa-chevron-up"></i></a></div>
135 <!-- Bootstrap core JS-->
136 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
jquery.min.js"></script>
137 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
bootstrap.bundle.min.js"></script>
138 <!-- Third party plugin JS-->
139 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
/1.4.1/jquery.easing.min.js"></script>

```

```

140 <!-- Core theme JS-->
141 <script src="./assets/js/script.js"></script>
142 <!-- Include constants-->
143 <script src="./constants/constants.js"></script>
144 <!-- Ajax call to send forms field -->
145 <script src="./assets/js/register.js"></script>
146 </body>
147 </html>

```

Listing 3: register.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjectTPI.
5 *      Page             :  Index.
6 *      Brief            :  Home page.
7 *      Date             :  02.06.2020.
8 */
9
10 /*Requirements */
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FUserManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FSessionManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/view/
    FITineraryView.php';
14 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
15
16 if (session_status() == PHP_SESSION_NONE) {
17     session_start();
18 }
19 ?>
20 <!DOCTYPE html>
21 <html lang="FR" dir="ltr">
22 <head>
23     <meta charset="utf-8">
24     <meta name="viewport" content="width=device-width, initial-scale
        =1, shrink-to-fit=no">
25     <meta name="description" content="">
26     <meta name="author" content="">
27     <title>Home</title>
28     <!-- Font Awesome icons (free version)-->
29     <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
        /5.13.0/js/all.min.js" crossorigin="anonymous"></script>
30     <!-- Core theme CSS (includes Bootstrap)-->
31     <link href="./assets/css/styles.css" rel="stylesheet">
32     <!-- Fonts CSS-->
33     <link rel="stylesheet" href="./assets/css/heading.css">
34     <link rel="stylesheet" href="./assets/css/body.css">
35     <!-- Maquest CSS and JS Load-->
36     <link type="text/css" rel="stylesheet" href="https://api.mqcdn.com
        /sdk/mapquest-js/v1.3.2/mapquest.css"/>
37     <script src="https://api.mqcdn.com/sdk/mapquest-js/v1.3.2/mapquest
        .js"></script>
38
39 </head>

```

```

40 <body id="page-top">
41 <?php
42
43 //Show the good navbar depends if logged and if user is admin
44 if (FUserManager::getInstance()->isLogged()) {
45     if (FUserManager::getInstance()->isAllowed(FSessionManager::
46     getUserLogged())) {
47         include "../inc/navbar/navbarAdmin.php";
48     }
49     else {
50         include "../inc/navbar/navbarLogged.php";
51     }
52 }
53 else {
54     include "../inc/navbar/navbarNotLogged.php";
55 }
56 ?>
57 <header class="masthead bg-primary text-white text-center">
58     <div class="container d-flex align-items-center flex-column">
59         <!-- Masthead Heading-->
60         <h1 class="masthead-heading mb-0">TRAVLER</h1>
61         <!-- Icon Divider-->
62         <div class="divider-custom divider-light">
63             <div class="divider-custom-line"></div>
64             <div class="divider-custom-icon"><i class="fas fa-star"></i></div>
65             <div class="divider-custom-line"></div>
66         </div>
67         <!-- Masthead Subheading-->
68         <p class="pre-wrap masthead-subheading font-weight-light mb-0">
69             <i>YOUR ITINERARY FINDER</i> </p>
70     </div>
71 </header>
72 <section class="page-section portfolio" id="portfolio">
73     <div class="container">
74         <!-- Portfolio Section Heading-->
75         <div class="text-center">
76             <h2 class="page-section-heading text-secondary mb-0 d-
77             inline-block">YOUR ITINERARY</h2>
78         </div>
79         <!-- Icon Divider-->
80         <div class="divider-custom">
81             <div class="divider-custom-line"></div>
82             <div class="divider-custom-icon"><i class="fas fa-star"></i></div>
83             <div class="divider-custom-line"></div>
84         </div>
85         <!-- Portfolio Grid Items-->
86         <div class="row justify-content-center">
87             <!-- Portfolio Items-->
88             <?php echo FItineraryView::DisplayItineraries(
89             FItineraryManager::GetInstance()->GetAllByUserId(FSessionManager::
90             getUserLogged()->Id));?>
91         </div>
92     </div>
93 </section>

```

```

90 <footer class="footer text-center">
91     <div class="container">
92         <div class="row">
93             <!-- Footer Location-->
94             <div class="col-lg-4 mb-5 mb-lg-0">
95                 <h4 class="mb-4">LOCATION</h4>
96                 <p class="pre-wrap lead mb-0">CFPT Informatique
97 Chemin Gérard-De-Ternier 10
98 1213 Lancy</p>
99             </div>
100            <!-- Footer Middle Section-->
101            <div class="col-lg-4 mb-5 mb-lg-0">
102
103            </div>
104            <!-- Footer About Text-->
105            <div class="col-lg-4">
106                <h4 class="mb-4">ABOUT TRAVLER</h4>
107                <p class="pre-wrap lead mb-0">Travler is a project
made by Thomas Fujise for his CFC work in the CFPT</p>
108            </div>
109        </div>
110    </div>
111 </footer>
112 <!-- Copyright Section-->
113 <section class="copyright py-4 text-center text-white">
114     <div class="container"><small class="pre-wrap">Copyright
Travler 2020</small></div>
115 </section>
116 <!-- Scroll to Top Button (Only visible on small and extra-small
screen sizes)-->
117 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-
scroll-trigger d-block text-center text-white rounded" href="#page-
top"><i class="fa fa-chevron-up"></i></a></div>
118 <!-- Bootstrap core JS-->
119 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
jquery.min.js"></script>
120 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
bootstrap.bundle.min.js"></script>
121 <!-- Third party plugin JS-->
122 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
/1.4.1/jquery.easing.min.js"></script>
123 <!-- Core theme JS-->
124 <script src="./assets/js/script.js"></script>
125 <!-- Display maps on itineraries card with mapquest-->
126 <script src="./assets/js/home.js"></script>
127 </body>
128 </html>

```

Listing 4: myItineraries.php

```

1 <?php
2 /*
3 *     Auteur           :   Fujise Thomas.
4 *     Projet          :   game.
5 *     Page            :   Administration.
6 *     Description     :   Page d'administration.
7 *     Date début projet :   04.09.2019.
8

```

```

9 */
10 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FSessionManager.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FUserManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/view/
    FAdminView.php';
13
14 if (session_status() == PHP_SESSION_NONE) {
15     session_start();
16 }
17 ?>
18 <!DOCTYPE html>
19 <html lang="FR" dir="ltr">
20 <head>
21     <meta charset="utf-8">
22     <meta name="viewport" content="width=device-width, initial-scale
        =1, shrink-to-fit=no">
23     <meta name="description" content="">
24     <meta name="author" content="">
25     <title>User Management</title>
26     <!-- Font Awesome icons (free version)-->
27     <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
        /5.13.0/js/all.min.js" crossorigin="anonymous"></script>
28     <!-- Core theme CSS (includes Bootstrap)-->
29     <link href="./assets/css/styles.css" rel="stylesheet">
30     <!-- Fonts CSS-->
31     <link rel="stylesheet" href="./assets/css/heading.css">
32     <link rel="stylesheet" href="./assets/css/body.css">
33 </head>
34 <body id="page-top">
35 <?php
36
37     //Show the good navbar depends if logged and if user is admin
38     if (FUserManager::getInstance()->isLoggedIn()) {
39         if (FUserManager::getInstance()->isAllowed(FSessionManager::
            getUserLogged())) {
40             include "./inc/navbar/navbarAdmin.php";
41         }
42         else {
43             include "./inc/navbar/navbarLogged.php";
44         }
45     }
46     else {
47         include "./inc/navbar/navbarNotLogged.php";
48     }
49
50 ?>
51 <div class="masthead text-center">
52 <div class="text-center">
53     <h2 class="page-section-heading text-secondary mb-0 d-
        inline-block">USER MANAGEMENT</h2>
54 </div>
55 <!-- Icon Divider-->
56 <div class="divider-custom">
57     <div class="divider-custom-line"></div>
58     <div class="divider-custom-icon"><i class="fas fa-star"></

```

```

i></div>
59     <div class="divider-custom-line"></div>
60     </div>
61 </div>
62
63 <section class="page-section" id="panelAdmin">
64
65     <table class="table">
66         <thead>
67             <tr>
68                 <th scope="col">Nickname</th>
69                 <th scope="col">Email</th>
70                 <th scope="col">Name</th>
71                 <th scope="col">Surname</th>
72                 <th scope="col">Country</th>
73                 <th scope="col">Status</th>
74                 <th scope="col">Action</th>
75             </tr>
76         </thead>
77         <? = FAdminView::DisplayUsers(FUserManager::GetInstance()->
GetAll()) ?>
78     </table>
79     <div class="modal fade" id="modalDisable" tabindex="-1" role="
dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
80     <div class="modal-dialog" role="document">
81         <div class="modal-content">
82             <div class="modal-header">
83                 <h5 class="modal-title" id="exampleModalLabel">-</h5>
84                 <button type="button" class="close" data-dismiss="
modal" aria-label="Close">
85                     <span aria-hidden="true">&times;</span>
86                 </button>
87             </div>
88             <div class="modal-body">
89                 -
90             </div>
91             <div class="modal-footer">
92                 <form action="" method="post">
93                     <button type="button" class="btn btn-danger" data-
dismiss="modal">Cancel</button>
94                     <input id="btnModal" class="btn btn-success" type=
"submit" name="" value="Disable">
95                     <input id="hiddenModal" type="hidden" name=""
value="">
96                 </form>
97             </div>
98         </div>
99     </div>
100 </section>
101 <footer class="footer text-center">
102     <div class="container">
103         <div class="row">
104             <!-- Footer Location-->
105             <div class="col-lg-4 mb-5 mb-lg-0">
106                 <h4 class="mb-4">LOCATION</h4>
107                 <p class="pre-wrap lead mb-0">CFPT Informatique
108 Chemin Gérard-De-Ternier 10

```

```

109 1213 Lancy</p>
110     </div>
111     <!-- Footer Middle Section-->
112     <div class="col-lg-4 mb-5 mb-lg-0">
113
114     </div>
115     <!-- Footer About Text-->
116     <div class="col-lg-4">
117         <h4 class="mb-4">ABOUT TRAVLER</h4>
118         <p class="pre-wrap lead mb-0">Travler is a project
made by Thomas Fujise for his CFC work in the CFPT</p>
119     </div>
120 </div>
121 </div>
122 </footer>
123 <!-- Copyright Section-->
124 <section class="copyright py-4 text-center text-white">
125     <div class="container"><small class="pre-wrap">Copyright
Travler 2020</small></div>
126 </section>
127 <!-- Scroll to Top Button (Only visible on small and extra-small
screen sizes)-->
128 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-
scroll-trigger d-block text-center text-white rounded" href="#page-
top"><i class="fa fa-chevron-up"></i></a></div>
129 <!-- Bootstrap core JS-->
130 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
jquery.min.js"></script>
131 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
bootstrap.bundle.min.js"></script>
132 <!-- Third party plugin JS-->
133 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
/1.4.1/jquery.easing.min.js"></script>
134 <!-- Core theme JS-->
135 <script src="./assets/js/script.js"></script>
136 <script src="./constants/constants.js"></script>
137 <!-- Include -->
138 <script src="./assets/js/administration.js"></script>
139 </body>
140 </html>

```

Listing 5: administration.php

```

1 <?php
2 /*
3 *      Author      :   Fujise Thomas.
4 *      Project     :   ProjectTPI.
5 *      Page        :   CreateItinerary.
6 *      Brief       :   Page for create an itinerary.
7 *      Date        :   03.06.2020.
8 */
9
10 /*Requirements */
11 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
manager/FCodeManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
manager/FSessionManager.php';

```



```

13 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
14
15 if (session_status() == PHP_SESSION_NONE) {
16     session_start();
17 }
18 //If user isn't logged in
19 if(FSessionManager::getUserLogged() == null){
20     header("Location: index.php");
21     exit();
22 }
23 ?>
24 <!DOCTYPE html>
25 <html lang="FR" dir="ltr">
26 <head>
27     <meta charset="utf-8">
28     <meta name="viewport" content="width=device-width, initial-scale
        =1, shrink-to-fit=no">
29     <meta name="description" content="">
30     <meta name="author" content="">
31     <title>Create your itinerary</title>
32     <!-- Font Awesome icons (free version)-->
33     <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
        /5.13.0/js/all.min.js" crossorigin="anonymous"></script>
34     <!-- Core theme CSS (includes Bootstrap)-->
35     <link href="./assets/css/styles.css" rel="stylesheet">
36     <!-- Fonts CSS-->
37     <link rel="stylesheet" href="./assets/css/heading.css">
38     <link rel="stylesheet" href="./assets/css/body.css">
39     <!-- Maquest CSS and JS Load-->
40     <link type="text/css" rel="stylesheet" href="https://api.mqcdn.com
        /sdk/mapquest-js/v1.3.2/mapquest.css"/>
41     <script src="https://api.mqcdn.com/sdk/mapquest-js/v1.3.2/mapquest
        .js"></script>
42
43 </head>
44 <body id="page-top">
45 <?php
46
47     //Show the good navbar depends if logged and if user is admin
48     if (FUserManager::getInstance()->isLoggedIn()) {
49         if (FUserManager::getInstance()->isAllowed(FSessionManager::
            getUserLogged())) {
50             include "./inc/navbar/navbarAdmin.php";
51         }
52         else {
53             include "./inc/navbar/navbarLogged.php";
54         }
55     }
56     else {
57         include "./inc/navbar/navbarNotLogged.php";
58     }
59
60 ?>
61 <canvas id="canvasPreview" style="display:none"></canvas>
62
63 <section class="page-section mb-0">

```

```

64 <div class="container">
65 <div class="row justify-content-center mt-4">
66 <div class="col-md-12">
67 <div class="card">
68 <div class="card-header">Create your itinerary</div>
69 <div class="card-body">
70 <form enctype="multipart/form-data">
71 <div class="form-group">
72 <div class="row">
73 <div class="col">
74 <label for="title">Title :</label>
75 <input type="text" class="form
-control" id="title" placeholder="Title" name="title" required>
76 <p id="errorTitle" class="
errorMsg">Title already used</p>
77 </div>
78 </div>
79 </div>
80 <div class="form-group">
81 <div class="row">
82 <div class="col">
83 <label for="startCountry">Country
: </label>
84 <select name="startcountry" id="
startCountry" size="1" class="custom-select">
85 <?php
86 $countryManager = new
FCodeManager();
87 foreach ($countryManager::
GetInstance()->getAllCountry() as $country) {
88 echo "<option id=" .
$country->Code . "> " . $country->Name . "</option>";
89 }
90 ?>
91 </select>
92 </div>
93 </div>
94 </div>
95 <div class="form-group">
96 <div class="row">
97 <div class="col">
98 <label for="description">
Description :</label>
99 <textarea class="form-control" id=
"description" rows="3"></textarea>
100 </div>
101 </div>
102 </div>
103 <div class="form-group">
104 <div class="md-form md-outline">
105 <label for="duration">Duration (Hours:
Minutes):</label>
106 <input type="time" id="duration" class="
form-control" placeholder="Select time">
107 </div>
108 </div>
109 </div>

```

```

110         <div class="row">
111             <div class="col">
112                 <label for="distance">Distance (KM
113             ) :</label>
114                 <input class="form-control" id="
115                 distance" type="number" value="100" data-decimals="1" min="1" max="
116                 999" step="0.1"/>
117                 <p id="errorDistance" class="
118                 errormsg">Distance not valid</p>
119             </div>
120         </div>
121         <div class="form-group">
122             <div class="row">
123                 <div class="col">
124                     <label for="mapItinerary">Place
125                     your itinerary :</label>
126                     <div id="mapItinerary" class="img-
127                     fluid rounded mb-5" style="width: 100%; height: 630px;"></div>
128                 </div>
129             </div>
130         </div>
131         <div class="form-group">
132             <label for="itineraryImages">Images :</
133             label>
134             <input type="file" class="form-control -
135             file" name="itineraryImages" id="itineraryImages" multiple>
136             </div>
137             <div class="form-group">
138                 <label for="">&nbsp;</label>
139                 <button type="submit" id="createItinerary
140                 " class="form-control btn btn-outline-primary" name="
141                 createItinerary" >Create</button>
142                 <button type="button" id="cancel" class="
143                 form-control btn btn-outline-danger">Cancel</button>
144                 <p id="errorCreate" class="errormsg">
145                 Itinerary creation fail</p>
146             </div>
147         </div>
148     </form>
149 </div>
150 </div>
151 </div>
152 </div>
153 </section>
154
155 <footer class="footer text-center">
156 <div class="container">
157     <div class="row">
158         <!-- Footer Location-->
159         <div class="col-lg-4 mb-5 mb-lg-0">
160             <h4 class="mb-4">LOCATION</h4>
161             <p class="pre-wrap lead mb-0">CFPT Informatique
162             Chemin Gérard-De-Ternier 10

```

```

155 1213 Lancy</p>
156     </div>
157     <!-- Footer Middle Section-->
158     <div class="col-lg-4 mb-5 mb-lg-0">
159
160     </div>
161     <!-- Footer About Text-->
162     <div class="col-lg-4">
163         <h4 class="mb-4">ABOUT TRAVLER</h4>
164         <p class="pre-wrap lead mb-0">Travler is a project
165         made by Thomas Fujise for his CFC work in the CFPT</p>
166     </div>
167 </div>
168 </footer>
169 <!-- Copyright Section-->
170 <section class="copyright py-4 text-center text-white">
171     <div class="container"><small class="pre-wrap">Copyright
172     Travler 2020</small></div>
173 </section>
174 <!-- Scroll to Top Button (Only visible on small and extra-small
175     screen sizes)-->
176 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-
177     scroll-trigger d-block text-center text-white rounded" href="#page-
178     top"><i class="fa fa-chevron-up"></i></a></div>
179 <!-- Bootstrap core JS-->
180 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
181     jquery.min.js"></script>
182 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
183     bootstrap.bundle.min.js"></script>
184 <!-- Third party plugin JS-->
185 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
186     /1.4.1/jquery.easing.min.js"></script>
187 <!-- Core theme JS-->
188 <script src="./assets/js/script.js"></script>
189 <!-- Include JS-->
190 <script src="./constants/constants.js"></script>
191 <!-- Display maps on itineraries card with mapquest-->
192 <script src="./assets/js/createItinerary.js"></script>
193 </body>
194 </html>

```

Listing 6: createItinerary.php

```

1 <?php
2
3 /*
4 *      Author           :   Fujise Thomas.
5 *      Project          :   ProjectTPI.
6 *      Page             :   help.
7 *      Brief            :   User guide.
8 *      Date             :   09.06.2020.
9 */
10
11 /**
12 * @brief Help user to use the website
13 */

```

```

14
15 require_once $_SERVER['DOCUMENT_ROOT'] . '/ProjectTPI/src/www/inc/inc
    .all/inc.all.php';
16
17 if (session_status() == PHP_SESSION_NONE) {
18     session_start();
19 }
20 ?>
21 <!doctype html>
22 <html lang="en">
23
24 <head>
25     <meta charset="utf-8">
26     <meta name="viewport" content="width=device-width, initial-scale
    =1, shrink-to-fit=no">
27     <meta name="description" content="">
28     <meta name="author" content="">
29     <title>Help</title>
30     <!-- Font Awesome icons (free version)-->
31     <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
    /5.13.0/js/all.min.js" crossorigin="anonymous"></script>
32     <!-- Core theme CSS (includes Bootstrap)-->
33     <link href="./assets/css/styles.css" rel="stylesheet">
34     <!-- Fonts CSS-->
35     <link rel="stylesheet" href="./assets/css/heading.css">
36     <link rel="stylesheet" href="./assets/css/body.css">
37
38 </head>
39 <body id="page-top">
40 <?php
41
42     //Show the good navbar depends if logged and if user is admin
43     if (FUserManager::getInstance()->isLogged()) {
44         if (FUserManager::getInstance()->isAllowed(FSessionManager::
    getUserLogged())) {
45             include "./inc/navbar/navbarAdmin.php";
46         }
47         else {
48             include "./inc/navbar/navbarLogged.php";
49         }
50     }
51     else {
52         include "./inc/navbar/navbarNotLogged.php";
53     }
54
55 ?>
56 <header class="masthead bg-primary text-white text-center">
57     <div class="container d-flex align-items-center flex-column">
58         <!-- Masthead Heading-->
59         <h1 class="masthead-heading mb-0">TRAVLER</h1>
60         <!-- Icon Divider-->
61         <div class="divider-custom divider-light">
62             <div class="divider-custom-line"></div>
63             <div class="divider-custom-icon"><i class="fas fa-star"></
    i></div>
64             <div class="divider-custom-line"></div>
65         </div>

```

```

66         <!-- Masthead Subheading-->
67         <p class="pre-wrap masthead-subheading font-weight-light mb-0"
68         >' <i>YOUR ITINERARY FINDER</i> ' </p>
69     </div>
70 </header>
71 <section>
72     <div class="container">
73         <div class="text-center">
74             <h2 class="page-section-heading text-secondary mb-0 d-
75             inline-block">Manuel utilisateur</h2>
76         </div>
77         <!-- Icon Divider-->
78         <div class="divider-custom">
79             <div class="divider-custom-line"></div>
80             <div class="divider-custom-icon"><i class="fas fa-star"></
81             i></div>
82             <div class="divider-custom-line"></div>
83         </div>
84         <div class="list-group mb-5">
85             <a href="#filter" class="list-group-item list-group-item-
86             action">Filtres</a>
87             <a href="#loginRegister" class="list-group-item list-group
88             -item-action">Connexion / Inscription</a>
89             <a href="#ratingComment" class="list-group-item list-group
90             -item-action">Commentaire et notation d'un itinéraire</a>
91             <a href="#createItinerary" class="list-group-item list-
92             group-item-action">Création d'un itinéraire</a>
93             <a href="#updateItinerary" class="list-group-item list-
94             group-item-action">Modification d'un itinéraire</a>
95             <a href="#profile" class="list-group-item list-group-item-
96             action">Page de profil</a>
97         </div>
98         <h2 id="filter">Filtres</h2>
99         <p>
100             Un formulaire est disponible en haut de <a href="./index.
101             php" target="_blank">la page d'accueil</a> et de <a href="./
102             myItineraries.php" target="_blank">la page "Mes itinéraires"</a>
103             afin de pouvoir filtrer les itinéraires. Lorsque l'on applique un
104             filtre, il est appliqué sur tous les itinéraires affichés sur la
105             page d'accueil. Il est également possible d'appliquer plusieurs
106             filtres en même temps, un itinéraire sera affiché s'il remplit tous
107             les critères du filtre.
108         </p>
109         <p>Il est possible de filtrer les itinéraires par :
110             <ul>
111                 <li>Pays</li>
112                 <li>Note</li>
113                 <li>Distance Min</li>
114                 <li>Distance Max</li>
115                 <li>Durée Min</li>
116                 <li>Durée Max</li>
117             </ul>
118         </p>
119         <p>Pour utiliser les filtres il vous suffit d'entrer les
120             valeurs désirés dans les champs désignés et d'appuyer sur le bouton
121             "Filter" pour appliquer le filtre.</p>
122         <p>Pour annuler le filtre qui est appliqué, vous devez appuyer

```

```

    sur le bouton "Cancel filter" et l'affichage des itinéraires sera
    remit par défaut</p>
105     <h2 id="loginRegister">Connexion / Inscription</h2>
106     <p>
107         Pour se connecter, il faut se rendre sur <a href="login.
            php" target="_blank">la page de connexion</a> et renseigner les
            champs du formulaire de connexion. Il faut ensuite appuyer sur le
            bouton "Sign in" afin de se connecter à l'application.
108         </br>
109         Si vous ne possédez pas encore de compte, il faut cliquer
            sur le bouton "Register" de la barre de navigation. Vous arriverez
            sur <a href="./register.php" target="_blank">la page d'inscription
            </a>.
110         </br>Il vous faudra alors renseigner les champs :
111         <ul>
112             <li><b>Nickname</b>: pour votre pseudo qui doit être
                unique</li>
113             <li><b>Email</b>: pour votre adresse email</li>
114             <li><b>Password</b> : pour votre mot de passe</li>
115             <li><b>Confirm Password</b> : pour la confirmation de
                votre mot de passe</li>
116         </ul>
117         </br>
118         Une fois les champs renseigné et le bouton "Register"
            pressé, vous recevrez un mail d'activation à l'email renseigné. Il
            vous faudra alors cliquer sur le lien reçu par mail pour confirmer
            votre compte et pouvoir vous connecter.
119         </p>
120         <h2 id="ratingComment">Commentaire et notation d'un itinéraire
            </h2>
121         <p>
122             Lorsque l'on clique sur le titre d'un itinéraire dans <a
                href="./index.php" target="_blank">la page d'accueil</a>, on se
                retrouve sur la page de détails d'un itinéraire.
123             </br>
124             Sur cette page, si l'utilisateur visionne un itinéraire
                dont il n'est pas propriétaire, il peut noter et commenter l'itiné
                raire.
125             </br>
126             </p>
127             <p>
128                 Pour noter un itinéraire, vous devez entrez la note que
                vous souhaitez attribuer dans le champ "Rate" en dessous de la map
                ou est affiché l'itinéraire (<b>Attention: vous ne pouvez noter qu'
                une note par itinéraire</b>) et appuyer sur le bouton "Rate".
129                 Une fois le bouton appuyé, la note est ajoutée à la
                moyenne de l'itinéraire.
130                 </p>
131                 <p>
132                     Pour commenter un itinéraire, vous devez renseigner le
                        champs "Comment", en dessous du champs de notation, avec le
                        commentaire que vous voulez publier. Il vous suffira de cliquer sur
                        le bouton "Comment" pour publier le commentaire. Le propriétaire
                        de l'itinéraire recevra un mail lorsque votre commentaire sera post
                        é
133                     </p>
134                     <h2 id="createItinerary">Création d'un itinéraire</h2>

```

```

135     <p>
136         Pour créer un itinéraire, il faut posséder un compte et être
        connecté avec. En cliquant sur le bouton "Create itinerary" sur
        la barre de navigation vous arriverez sur <a href="createItinerary
        .php" target="_blank">la page de création</a>, tous les champs sont
        obligatoires sauf les éventuelles photos que vous pouvez ajouter.
137     </p>
138     <p>
139         Les différents champs doivent être renseigné pour la création
        de votre itinéraire :
140         <ul>
141             <li>
142                 <b>Titre</b> : Le titre de votre itinéraire (<b>
        Attention</b>: il doit être unique)
143             </li>
144             <li>
145                 <b>Pays</b> : Sélectionner le pays de départ de
        votre itinéraire
146             </li>
147             <li>
148                 <b>Description</b> : Donner une description a
        votre itinéraire
149             </li>
150             <li>
151                 <b>Durée</b> : La durée de votre itinéraire.
152             </li>
153             <li>
154                 <b>Distance</b> : La distance totale de l'itiné
        raire.
155             </li>
156             <li>
157                 <b>Placer les points de votre itinéraire sur la
        carte</b>
158             </li>
159         </ul>
160     </p>
161     <p>
162         Une fois les champs ci-dessus renseigné, vous pouvez
        cliquer sur le bouton "Create" en-dessous et votre itinéraire sera
        créé si le titre que vous avez saisi n'est pas déjà utilisé.
163     </p>
164     <h2 id="updateItinerary">Modification d'un itinéraire</h2>
165     <p>
166         Pour pouvoir modifier un itinéraire, il faut aller sur la
        page de détail d'un itinéraire que vous avez créé (Vous pouvez voir
        les itinéraires que vous avez créé sur <a href="./myItineraries.
        php" target="_blank">la page "Mes itinéraires"</a>).
167         Une fois sur la page détail d'un itinéraire, vous pourrez
        modifier les différentes informations de votre itinéraire en
        changeant le champs et en cliquant sur le bouton "Save infos".
168         Une fois le bouton "Save infos" cliqué les informations
        sont sauvegardés et votre itinéraire est mis à jour.
169     </p>
170     <p>
171         Vous pouvez également rajouter des photos a votre itiné
        raire en les ajoutant dans le champs "Photos".
172     </p>

```



```

173     <h2 id="profile">Page de profil</h2>
174     <p>
175         Pour accéder à votre profil, il faut cliquer sur le lien "
176         Profil" de la barre de navigation qui vous redirigera sur <a href
177         = "../profil.php" target="_blank">votre page profil</a>.
178     </p>
179     <p>
180         Toutes les informations de votre compte sont affichés dans
181         les champs prévus pour. Vous pouvez modifier les détails de votre
182         compte en changeant les champs et en cliquant sur le bouton "Save
183         infos".
184     </br>
185     Une fois le bouton "Save infos" cliqué, les informations
186     que vous avez renseigné ainsi que l'avatar de votre compte (si vous
187     en avez ajouté un) sont sauvegardés et mis à jour dans la base de
188     données.
189 </p>
190 </div>
191 </section>
192 <footer class="footer text-center">
193     <div class="container">
194         <div class="row">
195             <!-- Footer Location-->
196             <div class="col-lg-4 mb-5 mb-lg-0">
197                 <h4 class="mb-4">LOCATION</h4>
198                 <p class="pre-wrap lead mb-0">CFPT Informatique
199 Chemin Gérard-De-Ternier 10
200 1213 Lancy</p>
201             </div>
202             <!-- Footer Middle Section-->
203             <div class="col-lg-4 mb-5 mb-lg-0">
204                 <div>
205                     <!-- Footer About Text-->
206                     <div class="col-lg-4">
207                         <h4 class="mb-4">ABOUT TRAVLER</h4>
208                         <p class="pre-wrap lead mb-0">Travler is a project
209 made by Thomas Fujise for his CFC work in the CFPT</p>
210                     </div>
211                 </div>
212             </div>
213         </div>
214     </footer>
215 <!-- Copyright Section-->
216 <section class="copyright py-4 text-center text-white">
217     <div class="container"><small class="pre-wrap">Copyright
218     Travler 2020</small></div>
219 </section>
220 <!-- Scroll to Top Button (Only visible on small and extra-small
221 screen sizes)-->
222 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-
223 scroll-trigger d-block text-center text-white rounded" href="#page-
224 top"><i class="fa fa-chevron-up"></i></a></div>
225 <!-- Bootstrap core JS-->
226 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
227 jquery.min.js"></script>
228 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
229 bootstrap.bundle.min.js"></script>

```

```

215 <!-- Third party plugin JS-->
216 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
    /1.4.1/jquery.easing.min.js"></script>
217 <!-- Core theme JS-->
218 <script src="./assets/js/script.js"></script>
219 <!-- Include -->
220 <!-- Display maps on itineraries card with mapquest-->
221 <script src="./assets/js/home.js"></script>
222 </body>
223
224 </html>

```

Listing 7: help.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjectTPI.
5 *      Page             :  Login.
6 *      Brief            :  Login page.
7 *      Date             :  02.06.2020.
8 */
9
10 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FUserManager.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FSessionManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FCodeManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FPhotoManager.php';
14 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FCommentManager.php';
15 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/model/
    FUser.php';
16 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
17 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/view/
    FItineraryView.php';
18 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FRateManager.php';
19
20
21
22 if (session_status() == PHP_SESSION_NONE) {
23     session_start();
24 }
25
26 $itinerary = FItineraryManager::GetInstance()->GetById($_GET['id']);
27 ?>
28 <!DOCTYPE html>
29 <html lang="FR" dir="ltr">
30 <head>
31     <meta charset="utf-8">
32     <meta name="viewport" content="width=device-width, initial-scale
        =1, shrink-to-fit=no">
33     <meta name="description" content="">
34     <meta name="author" content="">

```

```

35 <title>Itinerary details</title>
36 <!-- Font Awesome icons (free version)-->
37 <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
38 /5.13.0/js/all.min.js" crossorigin="anonymous"></script>
39 <!-- Core theme CSS (includes Bootstrap)-->
40 <link href="./assets/css/styles.css" rel="stylesheet">
41 <!-- Fonts CSS-->
42 <link rel="stylesheet" href="./assets/css/heading.css">
43 <link rel="stylesheet" href="./assets/css/body.css">
44 <!-- Mapquest CSS and JS-->
45 <link type="text/css" rel="stylesheet" href="https://api.mqcdn.com
46 /sdk/mapquest-js/v1.3.2/mapquest.css"/>
47 <script src="https://api.mqcdn.com/sdk/mapquest-js/v1.3.2/mapquest
48 .js"></script>
49 </head>
50 <body>
51 <?php
52 //Show the good navbar depends if logged and if user is admin
53 if (FUserManager::getInstance()->isLoggedIn()) {
54     if (FUserManager::getInstance()->isAllowed($_SESSION['
55 userLogged'])) {
56         include "./inc/navbar/navbarAdmin.php";
57     }
58     else {
59         include "./inc/navbar/navbarLogged.php";
60     }
61 }
62 else {
63     include "./inc/navbar/navbarNotLogged.php";
64 }
65 ?>
66 <section class="page-section mb-0" id="itineraryDetails">
67 <div class="container">
68 <div class="row justify-content-center mt-4">
69 <div class="col-md-12">
70 <div class="card">
71 <div class="card-header">Itinerary details</div>
72 <input id="itineraryId" type="hidden" value="<?=
73 $itinerary->Id; ?>"/>
74 <div class="card-body">
75 <form name="itinerary">
76 <div class="form-group">
77 <div class="row">
78 <div class="col">
79 <label for="title">Title :</label>
80 <?php if($itinerary->User ==
81 FSessionManager::GetUserLoggedIn()->Id) :?>
82 <input type="text" class="form
83 -control" id="title" placeholder="Title" name="title" value="<?=
84 $itinerary->Title?>" required>
85 <?php else :?>
86 <input type="text" class="form
87 -control" id="title" placeholder="Title" name="title" value="<?=
88 $itinerary->Title?>" required readonly>
89 <?php endif;?>

```

```

82                                     <p id="errorTitle" class="
errmsg">Title already exist</p>
83                                     </div>
84                                 </div>
85                             </div>
86                             <div class="form-group">
87                                 <div class="row">
88                                     <div class="col">
89                                         <label for="description">
Description :</label>
90                                         <?php if($itinerary->User ==
FSessionManager::GetUserLogged()->Id) :?>
91                                             <textarea class="form-control"
id="description" rows="3"><?= $itinerary->Description?></textarea>
92                                         <?php else :?>
93                                             <textarea class="form-control"
id="description" rows="3" readonly><?= $itinerary->Description?></
textarea>
94                                         <?php endif; ?>
95                                         </div>
96                                     </div>
97                                 </div>
98                                 <div class="form-group">
99                                     <div class="row">
100                                         <div class="col">
101                                             <label for="itineraryCountry">
Country :</label>
102                                             <?php if($itinerary->User ==
FSessionManager::GetUserLogged()->Id) :?>
103                                                 <select name="itineraryCountry
" id="itineraryCountry" size="1" class="custom-select">
104                                                 <?php else :?>
105                                                 <select name="itineraryCountry
" id="itineraryCountry" size="1" class="custom-select" disabled>
106                                                 <?php endif;
107
108                                             $countryManager = new
FCodeManager();
109                                             foreach ($countryManager::
GetInstance()->getAllCountry() as $country) {
110                                                 if($itinerary->Country ==
$country->Code){
111                                                     echo "<option id=" .
$country->Code . " selected> " . $country->Name . "</option>";
112                                                 }
113                                                 else{
114                                                     echo "<option id=" .
$country->Code . "> " . $country->Name . "</option>";
115                                                 }
116                                             }
117                                         <?>
118                                         </select>
119                                     </div>
120                                     <div class="col">
121                                         <label for="distance">Distance (KM
) :</label>
122                                         <?php if($itinerary->User ==

```

```

123 FSessionManager::GetUserLogged()->Id) :?>
    <input class="form-control" id="
distance" type="number" value="<?= $itinerary->Distance?>" data-
decimals="1" min="1" max="999" step="0.1"/>
124 <?php else :?>
125 <input class="form-control" id="
distance" type="number" value="<?= $itinerary->Distance?>" data-
decimals="1" min="1" max="999" step="0.1" readonly/>
126 <?php endif; ?>
127 <p id="errorDistance" class="
errorMsg">Distance not valid</p>
128 </div>
129 </div>
130 </div>
131 <div class="form-group">
132 <div class="row">
133 <div class="col">
134 <div class="md-form md-outline">
135 <label for="duration">Duration
(Hours:Minutes):</label>
136 <?php if($itinerary->User ==
FSessionManager::GetUserLogged()->Id) :?>
137 <input type="time" id="
duration" class="form-control" placeholder="Select time" value="<?=
$itinerary->Duration ?>">
138 <?php else :?>
139 <input type="time" id="
duration" class="form-control" placeholder="Select time" value="<?=
$itinerary->Duration ?>" readonly>
140 <?php endif;?>
141 </div>
142 </div>
143 <?php if($itinerary->User ==
FSessionManager::GetUserLogged()->Id) : ?>
144 <div class="col">
145 <label for="itineraryImages">
Photos :</label>
146 <input type="file" class="form-
control-file" name="itineraryImages" id="itineraryImages">
147 </div>
148 <?php endif; ?>
149 </div>
150 </div>
151 <div class="form-group">
152 <div id="carouselControls" class="carousel
slide" data-ride="carousel">
153 <div class="carousel-inner">
154 <?= FItineraryView::
DisplayPhotosItineraries(FPhotoManager::GetInstance()->GetAllById(
$itinerary->Id))?>
155 </div>
156 <a class="carousel-control-prev" href=
"#carouselControls" role="button" data-slide="prev">
157 <span class="carousel-control-prev
-icon" aria-hidden="true"></span>
158 <span class="sr-only">Previous</
span>

```

```

159         </a>
160         <a class="carousel-control-next" href=
"#carouselControls" role="button" data-slide="next">
161             <span class="carousel-control-next
-icon" aria-hidden="true"></span>
162             <span class="sr-only">Next</span>
163         </a>
164     </div>
165 </div>
166 <div class="form-group">
167     <div class="row">
168         <div class="col">
169             <label for="mapItinerary">The
itinerary :</label>
170             <div id="mapItinerary" class="img-
fluid rounded mb-5" style="width: 100%; height: 630px;"></div>
171             </div>
172         </div>
173     </div>
174     <div class="form-group">
175         <div class="row">
176             <div class="col">
177                 <label for="rateAvg">Average Rate :</p>
178                 <p id="rateAvg"><?= FRateManager::
GetInstance()->GetAvgByItinerary($itinerary->Id); ?>(<?=
FRateManager::GetInstance()->GetNbByItinerary($itinerary->Id);?>)</
p>
179                 </div>
180                 <div class="col">
181                     <?php if(FSessionManager::
GetUserLoggedIn()->Role == 2) :?>
182                     <a href="#" class="btn btn-danger"
data-type="disable" data-target="#modalConfirm" data-toggle="modal
">Disable</a>
183                     <?php endif; ?>
184                 </div>
185             </div>
186         </div>
187         <?php if($itinerary->User == FSessionManager::
GetUserLoggedIn()->Id) : ?>
188             <div class="form-group">
189                 <div class="row">
190                     <p id="errorUpdate" class="errmsg">
Update fail, Please try again</p>
191                     <div class="col">
192                         <button type="button" id="cancel"
class="form-control btn btn-outline-danger">Cancel</button>
193                     </div>
194                     <div class="col">
195                         <button type="submit" id="save"
class="form-control btn btn-outline-primary" name="save">Save
itinerary</button>
196                     </div>
197                     <div class="col">
198                         <a href="#" class="btn btn-danger"
data-type="delete" data-target="#modalConfirm" data-toggle="modal"
>Delete</a>

```

```

199         </div>
200     </div>
201 </div>
202     <?php elseif(!FRateManager::GetInstance()->
HasAlreadyRate($itinerary->Id, FSessionManager::GetUserLogged()->Id
) && FSessionManager::GetUserLogged()->Id != 0) : ?>
203         <div class="form-group">
204             <form name="rating">
205                 <div class="row">
206                     <div class="col">
207                         <label for="rating">Rating (0-10)
208
: </label>
209                         <input class="form-control" id="
rating" type="number" value="" data-decimals="0" min="1" max="10"
step="1"/>
210                         <p id="errorRate" class="errorMsg"
>Error trying to add your rate, Please try again later</p>
211                         <p id="errorRateValid" class="
errorMsg">The rate you enter isn't valid, Please enter a valid rate
(0-10)</p>
212                         <p id="RateAdded" class="succesmsg
">Your rate has been added</p>
213                         </div>
214                     </div>
215                     <div class="row">
216                         <div class="col">
217                             <button type="submit" id="rate"
class="form-control btn btn-outline-primary" name="rate">Rate</
button>
218                             </div>
219                         </div>
220                     </form>
221                 </div>
222                 <?php elseif(FSessionManager::GetUserLogged()
->Id == 0) : ?>
223                     <div class="lead font-weight-bold">Create an
account to rate this itinerary</div>
224                     <?php else :?>
225                         <div class="lead font-weight-bold">You
already rate this itinerary</div>
226                         <?php endif;?>
227                     </div>
228                 </form>
229             </div>
230         </div>
231     </div>
232 </div>
233 </div>
234 </div>
235     <div class="modal fade" id="modalConfirm" tabindex="-1" role="
dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
236         <div class="modal-dialog" role="document">
237             <div class="modal-content">
238                 <div class="modal-header">
239                     <h5 class="modal-title" id="exampleModalLabel">-</h5>
240                     <button type="button" class="close" data-dismiss="

```

```

modal" aria-label="Close">
241         <span aria-hidden="true">&times;</span>
242     </button>
243 </div>
244 <div class="modal-body">
245     -
246 </div>
247 <div class="modal-footer">
248     <form action="" method="post">
249         <button type="button" class="btn btn-danger" data-
dismiss="modal">Cancel</button>
250         <input id="btnModal" class="btn btn-success" type
="submit" name="" value="Confirm">
251         <input id="hiddenModal" type="hidden" name=""
value="">
252     </form>
253 </div>
254 </div>
255 </div>
256 </div>
257 <?php if($itinerary->User != FSessionManager::GetUserLogged()->Id)
: ?>
258 <div class="container">
259 <div class="row justify-content-center mt-4">
260     <div class="col-md-12">
261         <div class="card">
262             <div class="card-header">Comment this itinerary</div>
263             <div class="card-body">
264                 <form name="comment">
265                     <div class="form-group">
266                         <div class="row">
267                             <div class="col">
268                                 <label for="comment">Post your
comment :</label>
269                                 <?php if(FSessionManager::
GetUserLogged()->Id == 0) :?>
270                                     <textarea class="form-control"
id="comment" placeholder="Create an account to comment this
itinerary" rows="2" disabled></textarea>
271                                 <?php else :?>
272                                     <textarea class="form-control" id
="comment" placeholder="Write a comment here ..." rows="2"></
textarea>
273                                 <?php endif;?>
274                                     <p id="errorComment" class="
errmsg">Your comment wasn't posted, Please try again</p>
275                                     <p id="CommentAdded" class="
succesmsg">Your comment has been added</p>
276                                 </div>
277                             </div>
278                             <div class="row">
279                                 <div class="col">
280                                     <?php if(FSessionManager::
GetUserLogged()->Id != 0) :?>
281                                         <button type="submit" id="sendComment
" class="form-control btn btn-outline-primary" name="btnComment">
Comment</button>

```



```

282         <?php endif;?>
283     </div>
284 </div>
285 </div>
286 </form>
287 </div>
288 </div>
289 </div>
290 </div>
291 <?php endif;?>
292 </section>
293 <div class="row d-flex justify-content-center mt-100 mb-100">
294 <div class="col-lg-6">
295     <div class="card">
296         <div class="card-body text-center">
297             <h4 class="card-title">Latest Comments</h4>
298         </div>
299         <div class="comment-widgets">
300             <!-- Comments Row -->
301             <?= FItineraryView::DisplayCommentsItineraries(
FCommentManager::GetInstance()->GetAllById($itinerary->Id)); ?>
302         </div>
303     </div>
304 </div>
305 </div>
306 </div>
307 <footer class="footer text-center">
308 <div class="container">
309     <div class="row">
310         <!-- Footer Location-->
311         <div class="col-lg-4 mb-5 mb-lg-0">
312             <h4 class="mb-4">LOCATION</h4>
313             <p class="pre-wrap lead mb-0">CFPT Informatique
314 Chemin Gérard-De-Ternier 10
315 1213 Lancy</p>
316         </div>
317         <!-- Footer Middle Section-->
318         <div class="col-lg-4 mb-5 mb-lg-0">
319
320         </div>
321         <!-- Footer About Text-->
322         <div class="col-lg-4">
323             <h4 class="mb-4">ABOUT TRAVLER</h4>
324             <p class="pre-wrap lead mb-0">Travler is a project
made by Thomas Fujise for his CFC work in the CFPT</p>
325         </div>
326     </div>
327 </div>
328 </footer>
329 <!-- Copyright Section-->
330 <section class="copyright py-4 text-center text-white">
331     <div class="container"><small class="pre-wrap">Copyright
Travler 2020</small></div>
332 </section>
333 <!-- Scroll to Top Button (Only visible on small and extra-small
screen sizes)-->
334 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-

```

```

        scroll-trigger d-block text-center text-white rounded" href="#page-
        top"><i class="fa fa-chevron-up"></i></a></div>
335 <!-- Bootstrap core JS-->
336 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
        jquery.min.js"></script>
337 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
        bootstrap.bundle.min.js"></script>
338 <!-- Third party plugin JS-->
339 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
        /1.4.1/jquery.easing.min.js"></script>
340 <!-- Core theme JS-->
341 <script src="./assets/js/script.js"></script>
342 <!-- Include JS -->
343 <script src="./constants/constants.js"></script>
344 <!-- Ajax call to send forms field -->
345 <script src="./assets/js/detailsItinerary.js"></script>
346 </body>
347 </html>

```

Listing 8: itineraryDetails.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  logout.
6 *      Brief            :  logout page.
7 *      Date             :  05.06.2020.
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
        manager/FSessionManager.php';
10
11 if (session_status() == PHP_SESSION_NONE){
12     session_start();
13 }
14 FSessionManager::Reset();
15
16 header("Location: index.php");
17 ?>

```

Listing 9: logout.php

```

1 <nav class="navbar navbar-expand-lg bg-secondary fixed-top" id="
        mainNav">
2 <div class="container"><a class="navbar-brand js-scroll-trigger"
        href=".\\index.php">TRAVLER</a>
3 <button class="navbar-toggler navbar-toggler-right font-weight-
        bold bg-primary text-white rounded" type="button" data-toggle="
        collapse" data-target="#navbarResponsive" aria-controls="
        navbarResponsive" aria-expanded="false" aria-label="Toggle
        navigation">Menu <i class="fas fa-bars"></i></button>
4 <div class="collapse navbar-collapse" id="navbarResponsive">
5 <ul class="navbar-nav ml-auto">
6 <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
        -0 px-lg-3 rounded js-scroll-trigger" href=".\\administration.php">
        MANAGE USERS</a>
7 </li>

```

```

8         <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
          -0 px-lg-3 rounded js-scroll-trigger" href=".\\logout.php">LOGOUT</a
          >
9         </li>
10        </ul>
11    </div>
12 </div>
13 </nav>

```

Listing 10: navbarAdmin.php

```

1 <nav class="navbar navbar-expand-lg bg-secondary fixed-top" id="
  mainNav">
2 <div class="container"><a class="navbar-brand js-scroll-trigger"
  href=".\\index.php">TRAVLER</a>
3   <button class="navbar-toggler navbar-toggler-right font-weight-
    bold bg-primary text-white rounded" type="button" data-toggle="
    collapse" data-target="#navbarResponsive" aria-controls="
    navbarResponsive" aria-expanded="false" aria-label="Toggle
    navigation">Menu <i class="fas fa-bars"></i></button>
4   <div class="collapse navbar-collapse" id="navbarResponsive">
5     <ul class="navbar-nav ml-auto">
6       <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
          -0 px-lg-3 rounded js-scroll-trigger" href=".\\createItinerary.php">
          CREATE ITINERARY</a>
7       </li>
8       <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
          -0 px-lg-3 rounded js-scroll-trigger" href=".\\myItineraries.php">MY
          ITINERARY</a>
9       </li>
10      <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
          -0 px-lg-3 rounded js-scroll-trigger" href=".\\profil.php">PROFILE</
          a>
11      </li>
12      <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
          -0 px-lg-3 rounded js-scroll-trigger" href=".\\help.php">HELP</a>
13      </li>
14      <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
          -0 px-lg-3 rounded js-scroll-trigger" href=".\\logout.php">LOGOUT</a
          >
15      </li>
16    </ul>
17  </div>
18 </div>
19 </nav>

```

Listing 11: navbarLogged.php

```

1 <nav class="navbar navbar-expand-lg bg-secondary fixed-top" id="
  mainNav">
2 <div class="container"><a class="navbar-brand js-scroll-trigger"
  href=".\\index.php">TRAVLER</a>
3   <button class="navbar-toggler navbar-toggler-right font-weight-
    bold bg-primary text-white rounded" type="button" data-toggle="
    collapse" data-target="#navbarResponsive" aria-controls="
    navbarResponsive" aria-expanded="false" aria-label="Toggle
    navigation">Menu <i class="fas fa-bars"></i></button>
4   <div class="collapse navbar-collapse" id="navbarResponsive">

```

```

5         <ul class="navbar-nav ml-auto">
6             <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
-0 px-lg-3 rounded js-scroll-trigger" href=".\\login.php">LOGIN</a>
7                 </li>
8             <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
-0 px-lg-3 rounded js-scroll-trigger" href=".\\register.php">
REGISTER</a>
9                 </li>
10            <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px
-0 px-lg-3 rounded js-scroll-trigger" href=".\\help.php">HELP</a>
11                </li>
12        </ul>
13    </div>
14 </div>
15 </nav>

```

Listing 12: navbarNotLogged.php

```

1 <?php
2 /*
3 *     Auteur           :  Fujise Thomas .
4 *     Projet          :  ProjectTPI .
5 *     Page            :  Profil .
6 *     Description     :  Profil page .
7 *     Date début projet :  04.06.2020 .
8 */
9 //requirements
10 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
manager/FUserManager.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
manager/FSessionManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
manager/FCodeManager.php';
13
14
15 if (session_status() == PHP_SESSION_NONE) {
16     session_start();
17 }
18 //If user isn't logged in
19 if(FSessionManager::getUserLogged() == null){
20     header("Location: index.php");
21     exit();
22 }
23 $userLogged = FSessionManager::getUserLogged();
24 ?>
25 <!DOCTYPE html>
26 <html lang="FR" dir="ltr">
27 <head>
28     <meta charset="utf-8">
29     <meta name="viewport" content="width=device-width, initial-scale
=1, shrink-to-fit=no">
30     <meta name="description" content="">
31     <meta name="author" content="">
32     <title>Profil</title>
33     <!-- Font Awesome icons (free version)-->
34     <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
/5.13.0/js/all.min.js" crossorigin="anonymous"></script>
35     <!-- Core theme CSS (includes Bootstrap)-->

```

```

36 <link href="./assets/css/styles.css" rel="stylesheet">
37 <!-- Fonts CSS-->
38 <link rel="stylesheet" href="./assets/css/heading.css">
39 <link rel="stylesheet" href="./assets/css/body.css">
40 <!-- Maquest CSS and JS Load-->
41 <link type="text/css" rel="stylesheet" href="https://api.mqcdn.com
/sdk/mapquest-js/v1.3.2/mapquest.css"/>
42 <script src="https://api.mqcdn.com/sdk/mapquest-js/v1.3.2/mapquest
.js"></script>
43
44 </head>
45 <body id="page-top">
46 <?php
47
48 //Show the good navbar depends if logged and if user is admin
49 if (FUserManager::getInstance()->isLoggedIn()) {
50     if (FUserManager::getInstance()->isAllowed(FSessionManager::
getUserLogged())) {
51         include "../inc/navbar/navbarAdmin.php";
52     }
53     else {
54         include "../inc/navbar/navbarLogged.php";
55     }
56 }
57 else {
58     include "../inc/navbar/navbarNotLogged.php";
59 }
60
61 ?>
62 <section class="page-section mb-0" id="formLogin">
63     <div class="container">
64         <div class="row justify-content-center mt-4">
65             <div class="col-md-12">
66                 <div class="card">
67                     <div class="card-header">Profil</div>
68                     <div class="card-body">
69                         <form name="profil">
70                             <div class="form-group">
71                                 <div class="row">
72                                     <div class="col">
73                                         <label for="nickname">Nickname :</
74                                         label>
75                                             <input type="text" class="form
-control" id="nickname" placeholder="Nickname" name="nickname"
76                                             value="<?= $userLogged->Nickname?" required>
77                                             <p id="errorNickname" class="
78                                             errorMsg">Nickname already exist</p>
79                                             </div>
80                                         </div>
81                                     <div class="form-group">
82                                         <div class="row">
83                                             <div class="col">
84                                                 <label for="name">Name :</
85                                                 label>
86                                                 <input type="text" class="form
-control" id="name" placeholder="Name" name="name" value="<?=

```

```

$userLogged->Name?>">
84                                     </div>
85                                     <div class="col">
86                                         <label for="surname">Surname
: </label>
87                                     <input type="text" class="form
-control" id="surname" placeholder="Surname" name="surname" value="
<?= $userLogged->Surname?>">
88                                     </div>
89                                     </div>
90                                 </div>
91                                 <div class="form-group">
92                                     <div class="row">
93                                         <div class="col">
94                                             <label for="userBio">Bio :</label>
95                                             <textarea class="form-control" id=
"userBio" rows="3"><?= $userLogged->Bio?></textarea>
96                                         </div>
97                                     </div>
98                                 </div>
99                                 <div class="form-group">
100                                     <div class="row">
101                                         <div class="col">
102                                             <label for="userCountry">Country
: </label>
103                                     <select name="userCountry" id="
userCountry" size="1" class="custom-select">
104                                         <?php
105                                             $countryManager = new
FCodeManager();
106                                             foreach ($countryManager::
GetInstance()->getAllCountry() as $country) {
107                                                 if($userLogged->Country ==
$country->Code){
108                                                     echo "<option id=" .
$country->Code . " selected> " . $country->Name . "</option>";
109                                                 }
110                                                 else{
111                                                     echo "<option id=" .
$country->Code . "> " . $country->Name . "</option>";
112                                                 }
113                                             }
114                                         <?>
115                                         </select>
116                                     </div>
117                                 </div>
118                             </div>
119                             <div class="form-group">
120                                 <div class="row">
121                                     <div class="col">
122                                         <label for="avatar">Avatar :</
label>
123                                         
124                                         <input type="file" class="form-
control-file" name="avatar" id="avatar">
125

```

```

126         </div>
127     </div>
128 </div>
129     <div class="form-group">
130         <div class="row">
131             <p id="errorUpdate" class="errmsg">
Update fail, Please try again</p>
132             <div class="col">
133                 <button type="button" id="cancel"
class="form-control btn btn-outline-danger">Cancel</button>
134             </div>
135             <div class="col">
136                 <button type="submit" id="save"
class="form-control btn btn-outline-primary" name="formProfil">Save
infos</button>
137             </div>
138         </div>
139     </div>
140 </div>
141 </div>
142 </form>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </section>
150 <footer class="footer text-center">
151 <div class="container">
152     <div class="row">
153         <!-- Footer Location-->
154         <div class="col-lg-4 mb-5 mb-lg-0">
155             <h4 class="mb-4">LOCATION</h4>
156             <p class="pre-wrap lead mb-0">CFPT Informatique
157 Chemin Gérard-De-Ternier 10
158 1213 Lancy</p>
159         </div>
160         <!-- Footer Middle Section-->
161         <div class="col-lg-4 mb-5 mb-lg-0">
162
163         </div>
164         <!-- Footer About Text-->
165         <div class="col-lg-4">
166             <h4 class="mb-4">ABOUT TRAVLER</h4>
167             <p class="pre-wrap lead mb-0">Travler is a project
made by Thomas Fujise for his CFC work in the CFPT</p>
168         </div>
169     </div>
170 </div>
171 </footer>
172 <!-- Copyright Section-->
173 <section class="copyright py-4 text-center text-white">
174     <div class="container"><small class="pre-wrap">Copyright
Travler 2020</small></div>
175 </section>
176 <!-- Scroll to Top Button (Only visible on small and extra-small

```

```

        screen sizes)-->
177 <div class="scroll-to-top d-lg-none position-fixed"><a class="js-
        scroll-trigger d-block text-center text-white rounded" href="#page-
        top"><i class="fa fa-chevron-up"></i></a></div>
178 <!-- Bootstrap core JS-->
179 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/
        jquery.min.js"></script>
180 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
        bootstrap.bundle.min.js"></script>
181 <!-- Third party plugin JS-->
182 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing
        /1.4.1/jquery.easing.min.js"></script>
183 <!-- Core theme JS-->
184 <script src="./assets/js/script.js"></script>
185 <!-- Include constants-->
186 <script src="./constants/constants.js"></script>
187 <!-- Display maps on itineraries card with mapquest-->
188 <script src="./assets/js/profil.js"></script>
189 </body>
190 </html>

```

Listing 13: profil.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjectTPI.
5 *      Page             :  verified.
6 *      Brief            :  page to verify user for account
        activation.
7 *      Date             :  02.06.2020.
8 */
9
10 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
        manager/FuserManager.php';
11
12 $token = $_GET['token'];
13 $email = $_GET['email'];
14 if(FuserManager::getInstance()->ActivateAccount($token, $email)){
15     echo "Your account has been activated";
16     //Go on index.php after 3s
17     header( "Refresh:3; url=./index.php", true, 303);
18 }
19 else{
20     echo "Your link doesn't work";
21 }
22 ?>

```

Listing 14: verified.php

```

1 <?php
2 //Requirements
3 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
        manager/FuserManager.php';
4
5 //=====
6 // Tests
7 //=====

```



```

8
9
10 //Check Register function
11 $userWillRegister = FUserManager::GetInstance()->Register("John@Doe.
    com", "JohnDoe59", "Super2012");
12 $userWontRegister = FUserManager::GetInstance()->Register("John@Doe.
    com", "JohnDoe95", "Super2012");
13 $userWontRegister2 = FUserManager::GetInstance()->Register("Doe@john.
    com", "JohnDoe59", "Super2012");
14 if($userWillRegister == TRUE && $userWontRegister == FALSE &&
    $userWontRegister2 == FALSE){
15     echo "Register : Work !";
16 }
17 else{
18     echo "Register : Error !";
19 }
20
21 //Check Login function
22 $userWillLogin = FUserManager::GetInstance()->Login(['userEmail' => "
    John@Doe.com", 'userPwd' => "Super2012"]);
23 $userWontLogin = FUserManager::GetInstance()->Login(['userEmail' => "
    John@Doe.com", 'userPwd' => "Super"]);
24 if($userWillLogin != FALSE && $userWontLogin == FALSE){
25     echo "Login : Work !";
26     var_dump($userWillLogin);
27 }
28 else{
29     echo "Login : Error !";
30 }
31
32 //Check GetHashPassword function
33 if(FUserManager::GetInstance()->GetHashPassword("John@Doe.com") !=
    FALSE){
34     echo "GetHashPassword : Work !";
35 }
36 else{
37     echo "GetHashPassword : Error !";
38 }
39
40 //Check IsAllowed function
41 $userIsAllowed = new FUser(1, "John@Doe.com", "JohnDoe59","John","Doe"
    ,"dd", "sdsf", "CH", 2, 2);
42 $userIsntAllowed = new FUser(1, "John@Doe.com", "JohnDoe59","John","
    Doe","dd", "sdsf", "CH", 2, 1);
43 if(FUserManager::GetInstance()->IsAllowed($userIsAllowed) == TRUE &&
    FUserManager::GetInstance()->IsAllowed($userIsntAllowed) == FALSE){
44     echo "IsAllowed : Work !";
45 }
46 else{
47     echo "IsAllowed : Error !";
48 }
49
50 //Check ActivateAccount function
51 if(FUserManager::GetInstance()->ActivateAccount('
    f42e0b9cddb7e52702dc3e02b242f212312b171e5b733e98c46ac70fbb64bd86',
    "John@Doe.com")){
52     echo "ActivateAccount : Work !";

```

```

53 }
54 else{
55     echo "ActivateAccount : Error !";
56 }
57
58 //Check VerifyActivation function
59 if(FUserManager::GetInstance()->VerifyActivation("JohnDoe59") == TRUE
    && FUserManager::GetInstance()->VerifyActivation("David") == FALSE)
    {
60     echo "VerifyActivation : Work !";
61 }
62 else{
63     echo "VerifyActivation : Error !";
64 }
65
66 ?>

```

Listing 15: FUserManagerTest.php

```

1  /* latin */
2  @font-face {
3      font-family: 'SB Body';
4      font-style: italic;
5      font-weight: 400;
6      src: local('Lato Italic'), local('Lato-Italic'),
7           url(https://fonts.gstatic.com/s/lato/v16/S6u8w4BMUTPHjxsAXC-
8           qNiXg7Q.woff2) format('woff2');
9      unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02
10     C6, U+02DA, U+02DC,
11     U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U
12     +2215, U+FEFF, U+FFFD;
13 }
14 /* latin */
15 @font-face {
16     font-family: 'SB Body';
17     font-style: italic;
18     font-weight: 700;
19     src: local('Lato Bold Italic'), local('Lato-BoldItalic'),
20         url(https://fonts.gstatic.com/s/lato/v16/
21         S6u_w4BMUTPHjxsI5wq_Gwftx9897g.woff2)
22         format('woff2');
23     unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02
24     C6, U+02DA, U+02DC,
25     U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U
26     +2215, U+FEFF, U+FFFD;
27 }
28 /* latin */
29 @font-face {
30     font-family: 'SB Body';
31     font-style: normal;
32     font-weight: 400;
33     src: local('Lato Regular'), local('Lato-Regular'),
34         url(https://fonts.gstatic.com/s/lato/v16/
35         S6uyw4BMUTPHjx4wXiWtFCc.woff2) format('woff2');
36     unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02
37     C6, U+02DA, U+02DC,
38     U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U
39     +2215, U+FEFF, U+FFFD;

```

```

31 }
32 /* latin */
33 @font-face {
34     font-family: 'SB Body';
35     font-style: normal;
36     font-weight: 700;
37     src: local('Lato Bold'), local('Lato-Bold'),
38         url(https://fonts.gstatic.com/s/lato/v16/
39         S6u9w4BMUTPHh6UVSwiPGQ3q5d0.woff2) format('woff2');
40     unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02
41     C6, U+02DA, U+02DC,
42     U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U
43     +2215, U+FEFF, U+FFFD;
44 }
45 /* Change color for error messages */
46 .errormsg{
47     color: red;
48 }
49 /* Comment panel CSS */
50 @import url(http://fonts.googleapis.com/css?family=Calibri
51 :400,300,700);
52
53 .mt-100 {
54     margin-top: 100px
55 }
56
57 .mb-100 {
58     margin-bottom: 100px
59 }
60
61 .comment-widgets {
62     position: relative;
63     margin-bottom: 10px
64 }
65
66 .comment-widgets .comment-row {
67     border-bottom: 1px solid transparent;
68     padding: 14px;
69     display: -webkit-box;
70     display: -ms-flexbox;
71     display: flex;
72     margin: 10px 0
73 }
74
75 .p-2 {
76     padding: 0.5rem !important
77 }
78
79 .comment-text {
80     padding-left: 15px
81 }
82
83 .w-100 {
84     width: 100% !important
85 }
86
87 .m-b-15 {

```

```

84     margin-bottom: 15px
85 }
86
87 .btn-sm {
88     padding: 0.25rem 0.5rem;
89     font-size: 0.76563rem;
90     line-height: 1.5;
91     border-radius: 1px
92 }
93
94 .btn-cyan {
95     color: #fff;
96     background-color: #27a9e3;
97     border-color: #27a9e3
98 }
99
100 .btn-cyan:hover {
101     color: #fff;
102     background-color: #1a93ca;
103     border-color: #198bbe
104 }
105
106 .comment-widgets .comment-row:hover {
107     background: rgba(0, 0, 0, 0.05)
108 }

```

Listing 16: body.css

```

1  /* latin */
2  @font-face {
3      font-family: 'SB Heading';
4      font-style: normal;
5      font-weight: 400;
6      src: local('Montserrat Regular'), local('Montserrat-Regular'),
7           url(https://fonts.gstatic.com/s/montserrat/v14/JTUSjIg1_i6t8kCHKm459WRhyyTh89ZNpQ.woff2)
8           format('woff2');
9      unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02
10     C6, U+02DA, U+02DC,
11     U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U
12     +2215, U+FEFF, U+FFFD;
13 }
14 /* latin */
15 @font-face {
16     font-family: 'SB Heading';
17     font-style: normal;
18     font-weight: 700;
19     src: local('Montserrat Bold'), local('Montserrat-Bold'),
20         url(https://fonts.gstatic.com/s/montserrat/v14/JTURjIg1_i6t8kCHKm45_dJE3gnD_vx3rCs.woff2)
21         format('woff2');
22     unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02
23     C6, U+02DA, U+02DC,
24     U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U
25     +2215, U+FEFF, U+FFFD;
26 }

```

Listing 17: heading.css

```

1 $(document).ready(() => {
2     $('#btnModal').click(DisableAccount);
3     $('#modalDisable').on('show.bs.modal', function (event) {
4         var button = $(event.relatedTarget)
5         var type = button.data('type')
6         $('#modalDisable').modal('show')
7         if (type == "disable") {
8             var nickname = button.data('nickname')
9             var modal = $(this)
10             modal.find('.modal-title').text('Are you sure you want to
disable this account')
11             modal.find('.modal-body').text('Disable the account : ' +
nickname)
12             $('#btnModal').attr('name', 'disableUser');
13             $('#hiddenModal').attr('name', 'hiddenNick');
14             $('#hiddenModal').attr('value', nickname);
15         }
16     });
17 });
18
19 function DisableAccount(event){
20     if (event) {
21         event.preventDefault();
22     }
23
24     let nickname = $('#hiddenModal').val();
25
26     if(nickname.length == 0){
27         $('#errorNickname').show().delay(3000).fadeOut(1000);
28         return;
29     }
30     var formData = new FormData();
31     formData.append("nickname", nickname);
32
33     $.ajax({
34         method: 'POST',
35         url: './app/api/disableAccount.php',
36         data: formData,
37         dataType: 'json',
38         contentType: false,
39         processData: false,
40
41         success: function(data){
42             switch(data.ReturnCode){
43                 case ACCOUNT_DISABLED:
44                     window.location = "./administration.php";
45                     break;
46                 case DISABLE_ACCOUNT_FAIL:
47                     $('#errorUpdate').show().delay(3000).fadeOut(1000)
;
48                     break;
49                 default:
50                     alert("-");
51                     break;
52             }
53         },
54

```

```

55     error: function (jqXHR){
56         error = "Error :";
57         switch(jqXHR.status){
58             case INVALID_JSON:
59                 error = error + jqXHR.status + "invalid json";
60                 break;
61             case FILE_NOT_FOUND:
62                 error = error + jqXHR.status + "Can't find
createItinerary.php";
63                 break;
64         }
65         alert(error);
66     }
67 });
68 }

```

Listing 18: administration.js

```

1 //Return code
2
3 //Register
4 const REGISTER_DONE = 0;
5 const REGISTER_FAIL = 1;
6 const REGISTER_DIFFERENT_PASSWORD = 2;
7 const REGISTER_PASSWORD_DONT_MATCH_REQUIREMENTS = 3;
8 //Login
9 const LOGIN_DONE = 10;
10 const LOGIN_ACCOUNT_NOT_ACTIVATE = 11;
11 const LOGIN_FAIL = 12;
12 const LOGIN_ACCOUNT_BLOCKED = 13;
13 //Update Info
14 const INFOS_UPDATED = 30;
15 const INFOS_UPDATED_FAIL = 31;
16 //Create itinerary
17 const ITINERARY_CREATE = 40;
18 const ITINERARY_CREATE_FAIL = 41;
19 const ITINERARY_DISTANCE_NOT_VALID = 42;
20 //Update Itinerary
21 const ITINERARY_UPDATE = 70;
22 const ITINERARY_UPDATE_FAIL = 71;
23
24 //Add comment
25 const COMMENT_ADDED = 50;
26 const COMMENT_ADD_FAIL = 51;
27 //Add rate
28 const RATE_ADDED = 60;
29 const RATE_ADD_FAIL = 61;
30 const RATE_NOT_VALID = 62;
31 //Disable account
32 const ACCOUNT_DISABLED = 80;
33 const DISABLE_ACCOUNT_FAIL = 81;
34 //Disable itinerary
35 const ITINERARY_DISABLED = 90;
36 const DISABLE_ITINERARY_FAIL = 91;
37 //Delete itinerary
38 const ITINERARY_DELETED = 100;
39 const DELETE_ITINERARY_FAIL = 101;
40

```

```

41
42
43 //Error
44 const INVALID_JSON = 200;
45 const FILE_NOT_FOUND = 404;

```

Listing 19: constants.js

```

1 $(document).ready(() => {
2     InitMap();
3     $('#errmsg').hide();
4     $('#createItinerary').click(CreateItinerary);
5     $('#cancel').click(Cancel);
6 });
7 /* @var array Array with all waypoints's info */
8 var waypoints = [];
9
10 var formData;
11 /**
12  * @brief Create an itinerary on form submit
13  *
14  * @param {*} event
15  *
16  * @return void
17  */
18 function CreateItinerary(event) {
19     if (event) {
20         event.preventDefault();
21     }
22     // init
23     let title = $("#title").val();
24     let country = $("#startCountry option:selected").attr("id");
25     let description = $("#description").val();
26     let duration = $("#duration").val();
27     let distance = $("#distance").val();
28     let itineraryImages = $("#itineraryImages").prop('files').length >
29     0 ? $('#itineraryImages').prop('files')[0] : null;
30
31     let fWaypointsArray = [];
32     //Array with fWaypoint format
33     for(let i = 0; i<=waypoints.length-1;i++){
34         fWaypointsArray.push({
35             "Index" : i+1,
36             "Address" : waypoints[i].street + ", " + waypoints[i].
37             adminArea5 + ", " + waypoints[i].adminArea3 + " " + waypoints[i].
38             postalCode + " " + waypoints[i].adminArea1,
39             "Longitude" : waypoints[i].latLng.lng,
40             "Latitude" : waypoints[i].latLng.lat
41         });
42     }
43     //Create data to send
44     formData = new FormData();
45     formData.append("title", title);
46     formData.append("country", country);
47     formData.append("description", description);
48     formData.append("duration", duration);
49     formData.append("distance", distance);
50     formData.append("media[]", itineraryImages);

```

```

48     formData.append("waypoints", JSON.stringify(fWaypointsArray));
49
50
51     if (title.length == 0) {
52         $("#title").css("border-color", "red");
53         $("#title").focus();
54         return;
55     } else {
56         $("#title").css("border-color", "");
57     }
58
59     if (description.length == 0) {
60         $("#description").css("border-color", "red");
61         $("#description").focus();
62         return;
63     } else {
64         $("#description").css("border-color", "");
65     }
66
67     if (duration.length == 0) {
68         $("#duration").css("border-color", "red");
69         $("#duration").focus();
70         return;
71     } else {
72         $("#duration").css("border-color", "");
73     }
74
75     if(distance.length == 0){
76         $("#distance").css("border-color", "red");
77         $("#distance").focus();
78         return;
79     } else{
80         $("#distance").css("border-color", "");
81     }
82
83     if(waypoints.length < 2){
84         $("#mapItinerary").css("border-color", "red");
85         alert("Veuillez placer votre itinéraire");
86         return;
87     } else{
88         $("#mapItinerary").css("border-color", "");
89     }
90
91     $.ajax({
92         method: 'POST',
93         url: './app/api/createItinerary.php',
94         data: formData,
95         dataType: 'json',
96         contentType: false,
97         processData: false,
98
99         success: function(data){
100             switch(data.ReturnCode){
101                 case ITINERARY_CREATE :
102                     window.location = "./myItineraries.php";
103                     break;
104                 case ITINERARY_CREATE_FAIL:

```



```

105         $('#errorCreate').show().delay(3000).fadeOut(1000);
106         break;
107         case ITINERARY_DISTANCE_NOT_VALID:
108             $('#errorDistance').show().delay(3000).fadeOut(1000);
109             break;
110         default:
111             alert("-");
112             break;
113     }
114 },
115
116     error: function (jqXHR){
117         error = "Error :";
118         switch(jqXHR.status){
119             case INVALID_JSON :
120                 error = error + jqXHR.status + "invalid json";
121                 break;
122             case FILE_NOT_FOUND :
123                 error = error + jqXHR.status + "Can't find login.php";
124                 break;
125         }
126         alert(error);
127     }
128 });
129 }
130
131 /**
132  * @brief Init the map where the user can place his itinerary
133  *
134  * @return void
135  */
136 function InitMap(){
137     L.mapquest.key = 'xTHtqDgrfGDrRKxzBKyFtDdkqRS4Uu8V';
138
139     var map = L.mapquest.map('mapItinerary', {
140         center: [46.165320, 6.072530],
141         layers: L.mapquest.tileLayer('map'),
142         zoom: 13,
143         zoomControl: false
144     });
145
146     L.control.zoom({
147         position: 'topright'
148     }).addTo(map);
149
150     L.mapquest.directionsControl({
151         routeSummary: {
152             enabled: false
153         },
154         narrativeControl: {
155             enabled: false,
156             compactResults: false
157         }
158     }).addTo(map);
159
160     //When marker moove
161     map.on('moveend', function(event) {

```

```

162     let marker = event.target;
163     for (let prop in marker._layers) {
164         if (!marker._layers.hasOwnProperty(prop)) continue;
165
166         //console.log(marker._layers[prop].locations);
167         if(marker._layers[prop].locations != undefined){
168             for(let i = 0;i<marker._layers[prop].locations.length;i
169 ++){
170                 waypoints[i] = marker._layers[prop].locations[i];
171             }
172             // locationIndex- I am assuming 0 for start marker and 1 for
173 end marker.
174             if (marker._layers[prop].locationIndex === 1) {
175                 let latLong = marker._layers[prop]._latlng;
176                 //console.log(latLong)
177             }
178         }
179     });
180 }
181 /**
182  * @brief Cancel the form
183  *
184  * @param {*} event
185  *
186  * @return void
187  */
188 function Cancel(event){
189     if (event) {
190         event.preventDefault();
191     }
192     window.location = "../index.php";
193 }

```

Listing 20: createItinerary.js

```

1 $(document).ready(() => {
2     InitMap();
3     $('#save').click(SaveItinerary);
4     $('#sendComment').click(AddComment);
5     $('#rate').click(AddRate);
6     $('#cancel').click(Cancel);
7     $('.errmsg').hide();
8     $('.succesmsg').hide();
9     $('#btnModalDisable').click(DisableItinerary);
10    // Set modal for confirmation
11    $('#modalConfirm').on('show.bs.modal', function (event) {
12        var button = $(event.relatedTarget)
13        var type = button.data('type')
14        $('#modalConfirm').modal('show')
15        if (type == "disable") {
16            var modal = $(this)
17            modal.find('.modal-title').text('Are you sure you want to
18disable this itinerary')
19            modal.find('.modal-body').text('Disable the itinerary : '
+ $('#title').val())
20            $('#btnModal').attr('name', 'disableItinerary');

```

```

20         $('input[name = "disableItinerary"]').click(
DisableItinerary);
21     }else if(type == "delete"){
22         var modal = $(this)
23         modal.find('.modal-title').text('Are you sure you want to
delete this itinerary')
24         modal.find('.modal-body').text('Delete the itinerary : ' +
$("#title").val())
25         $('#btnModal').attr('name', 'deleteItinerary');
26         $('input[name = "deleteItinerary"]').click(DeleteItinerary)
;
27     }
28 });
29 });
30 /* @var array Array with all waypoints's info */
31 var waypoints = [];
32
33 var formData;
34 /**
35  * @brief Save itinerary's details change
36  *
37  * @param {*} event
38  */
39 function SaveItinerary(event) {
40     if (event) {
41         event.preventDefault();
42     }
43     // init
44     let itineraryId = $("#itineraryId").val();
45     let title = $("#title").val();
46     let description = $("#description").val();
47     let country = $("#itineraryCountry option:selected").attr("id");
48     let duration = $("#duration").val();
49     let distance = $("#distance").val();
50     let itineraryImages = $("#itineraryImages").prop('files').length >
0 ? $('#itineraryImages').prop('files')[0] : null;
51
52     let fWaypointsArray = [];
53     //Array with fWaypoint format
54     for(let i = 0; i<=waypoints.length-1;i++){
55         fWaypointsArray.push({
56             "Index" : i+1,
57             "Address" : waypoints[i].street + ", " + waypoints[i].
adminArea5 + ", " + waypoints[i].adminArea3 + " " + waypoints[i].
postalCode + " " + waypoints[i].adminArea1,
58             "Longitude" : waypoints[i].latLng.lng,
59             "Latitude" : waypoints[i].latLng.lat
60         });
61     }
62     //Create data to send
63     formData = new FormData();
64     formData.append("itineraryId", itineraryId);
65     formData.append("title", title);
66     formData.append("country", country);
67     formData.append("description", description);
68     formData.append("duration", duration);
69     formData.append("distance", distance);

```

```

70     formData.append("media[]", itineraryImages);
71     formData.append("waypoints", JSON.stringify(fWaypointsArray));
72
73
74     if (title.length == 0) {
75         $("#title").css("border-color", "red");
76         $("#title").focus();
77         return;
78     } else {
79         $("#title").css("border-color", "");
80     }
81
82     if (description.length == 0) {
83         $("#description").css("border-color", "red");
84         $("#description").focus();
85         return;
86     } else {
87         $("#description").css("border-color", "");
88     }
89
90     if (duration.length == 0) {
91         $("#duration").css("border-color", "red");
92         $("#duration").focus();
93         return;
94     } else {
95         $("#duration").css("border-color", "");
96     }
97
98     if(distance.length == 0){
99         $("#distance").css("border-color", "red");
100         $("#distance").focus();
101         return;
102     } else{
103         $("#distance").css("border-color", "");
104     }
105
106     if(waypoints.length < 2){
107         $("#mapItinerary").css("border-color", "red");
108         alert("Veuillez placer votre itinéraire");
109         return;
110     } else{
111         $("#mapItinerary").css("border-color", "");
112     }
113
114     $.ajax({
115         method: 'POST',
116         url: './app/api/updateItinerary.php',
117         data: formData,
118         dataType: 'json',
119         contentType: false,
120         processData: false,
121
122         success: function(data){
123             switch(data.ReturnCode){
124                 case ITINERARY_UPDATE:
125                     window.location = "./myItineraries.php";
126                     break;

```

```

127         case ITINERARY_UPDATE_FAIL:
128             $('#errorUpdate').show().delay(3000).fadeOut(1000);
129         break;
130         case ITINERARY_DISTANCE_NOT_VALID:
131             $('#errorDistance').show().delay(3000).fadeOut(1000)
132     };
133     break;
134     default:
135         alert("-");
136         break;
137     }
138 },
139     error: function (jqXHR){
140         error = "Error :";
141         switch(jqXHR.status){
142             case INVALID_JSON:
143                 error = error + jqXHR.status + "invalid json";
144             break;
145             case FILE_NOT_FOUND:
146                 error = error + jqXHR.status + "Can't find
createItinerary.php";
147             break;
148         }
149         alert(error);
150     }
151 });
152 }
153 /**
154  * @brief Add a comment to an itinerary
155  *
156  * @param {*} event
157  *
158  * @return void
159  */
160 function AddComment(event) {
161     if (event) {
162         event.preventDefault();
163     }
164     //Send title to use php function "GetByTitle" after because title's
unique
165     let title = $("#title").val();
166
167     let comment = $("#comment").val();
168
169     formData = new FormData();
170     formData.append("title", title);
171     formData.append("comment", comment);
172
173     if (comment.length == 0) {
174         $("#comment").css("border-color", "red");
175         $("#comment").focus();
176         return;
177     } else {
178         $("#comment").css("border-color", "");
179     }
180 }

```

```

181     $.ajax({
182         method: 'POST',
183         url: './app/api/addComment.php',
184         data: formData,
185         dataType: 'json',
186         contentType: false,
187         processData: false,
188
189         success: function(data){
190             switch(data.ReturnCode){
191                 case COMMENT_ADDED:
192                     $('#CommentAdded').show().delay(3000).fadeOut
193                     (1000);
194                     break;
195                 case COMMENT_ADD_FAIL:
196                     $('#errorComment').show().delay(3000).fadeOut
197                     (1000);
198                     break;
199                 default:
200                     alert("-");
201                     break;
202             }
203         },
204
205         error: function (jqXHR){
206             error = "Error :";
207             switch(jqXHR.status){
208                 case INVALID_JSON:
209                     error = error + jqXHR.status + "invalid json";
210                     break;
211                 case FILE_NOT_FOUND:
212                     error = error + jqXHR.status + "Can't find
213                     createItinerary.php";
214                     break;
215             }
216             alert(error);
217         }
218     });
219 }
220 /**
221  * @brief Add a rate to an itinerary
222  *
223  * @param {*} event
224  *
225  * @return void
226  */
227 function AddRate(event) {
228     if (event) {
229         event.preventDefault();
230     }
231     //Send title to use php function "GetByTitle" after because title
232     's unique
233     let title = $("#title").val();
234
235     let rate = $("#rating").val();
236
237     formData = new FormData();

```

```

234     formData.append("title", title);
235     formData.append("rate", rate);
236
237     if (rate.length == 0) {
238         $("#comment").css("border-color", "red");
239         $("#comment").focus();
240         return;
241     } else {
242         $("#comment").css("border-color", "");
243     }
244
245     $.ajax({
246         method: 'POST',
247         url: './app/api/addRate.php',
248         data: formData,
249         dataType: 'json',
250         contentType: false,
251         processData: false,
252
253         success: function(data){
254             switch(data.ReturnCode){
255                 case RATE_ADDED:
256                     $('#RateAdded').show().delay(3000).fadeOut(1000);
257                     break;
258                 case RATE_ADD_FAIL:
259                     $('#errorRate').show().delay(3000).fadeOut(1000);
260                     break;
261                 case RATE_NOT_VALID:
262                     $('#errorRateValid').show().delay(3000).fadeOut
(1000);
263                     break;
264                 default:
265                     alert("-");
266                     break;
267             }
268         },
269
270         error: function (jqXHR){
271             error = "Error :";
272             switch(jqXHR.status){
273                 case INVALID_JSON:
274                     error = error + jqXHR.status + "invalid json";
275                     break;
276                 case FILE_NOT_FOUND:
277                     error = error + jqXHR.status + "Can't find
createItinerary.php";
278                     break;
279             }
280             alert(error);
281         }
282     });
283 }
284 /**
285  * @brief Init the map where the user can modify his itinerary
286  *
287  * @return void
288  */

```

```

289 function InitMap(){
290     L.mapquest.key = 'xTHtqDgrfGDrRKxzBKyFtDdkqRS4Uu8V';
291
292     $.getJSON('./app/api/getItineraryById.php?id=' + $('#itineraryId')
293     .val(), function(data) {
294         var mapInit = [];
295         var waypointsArray = [];
296         var map = L.mapquest.map('mapItinerary', {
297             center: [data.Waypoints[0].Latitude, data.Waypoints[0].
298             Longitude],
299             layers: L.mapquest.tileLayer('map'),
300             zoom: 13,
301             zoomControl: false
302         });
303         mapInit.push({
304             start: data.Waypoints[0].Address,
305             end: data.Waypoints[data.Waypoints.length - 1].Address
306         });
307         /*Add waypoints in array */
308         for(let i = 1; i< data.Waypoints.length-1 ; i++){
309             waypointsArray.push(data.Waypoints[i].Latitude + ', ' +
310             data.Waypoints[i].Longitude);
311         }
312         if(waypointsArray.length > 0){
313             mapInit[0].waypoints = waypointsArray;
314         }
315         L.control.zoom({
316             position: 'topright'
317         }).addTo(map);
318
319         L.mapquest.directionsControl({
320             routeSummary: {
321                 enabled: false
322             },
323             narrativeControl: {
324                 enabled: false,
325                 compactResults: false
326             },
327             startInput:{
328                 compactResults: true,
329                 disabled: false,
330                 location: data.Waypoints[0].Address,
331                 placeholderText: 'Starting point or click on the map
332                 ...',
333                 geolocation: {
334                     enabled: true
335                 }
336             },
337             endInput:{
338                 compactResults: true,
339                 disabled: false,
340                 location: data.Waypoints[data.Waypoints.length - 1].
341                 Address,
342                 placeholderText: 'Destination',
343                 geolocation: {
344                     enabled: true
345                 }
346             }
347         });
348     });
349 }

```



```

341     }
342     }).addTo(map);
343     //console.log(L.mapquest.directions().route(mapInit[0]));
344     L.mapquest.directions().route(mapInit[0]);
345
346     //When marker moove
347     map.on('moveend', function(event) {
348         let marker = event.target;
349         for (let prop in marker._layers) {
350             if (!marker._layers.hasOwnProperty(prop)) continue;
351
352             //console.log(marker._layers[prop].locations);
353             if(marker._layers[prop].locations != undefined){
354                 for(let i = 0;i<marker._layers[prop].locations.
length;i++){
355                     waypoints[i] = marker._layers[prop].locations[i];
356                 }
357             }
358             // locationIndex- I am assuming 0 for start marker and 1
359             for end marker.
360             if (marker._layers[prop].locationIndex === 1) {
361                 let latLong = marker._layers[prop]._latlng;
362                 //console.log(latLong)
363             }
364         }
365     });
366 });
367
368
369 }
370 /**
371  * @brief Disable the itinerary
372  *
373  * @param {*} event
374  *
375  * @return void
376  */
377 function DisableItinerary(event){
378     if (event){
379         event.preventDefault();
380     }
381
382     let idItinerary = $("#itineraryId").val();
383     let title = $("#title").val();
384
385     //Create data to send
386     formData = new FormData();
387     formData.append("itineraryId", idItinerary);
388     formData.append("title", title);
389
390
391     $.ajax({
392         method: 'POST',
393         url: './app/api/disableItinerary.php',
394         data: formData,
395         dataType: 'json',

```

```

396         contentType: false,
397         processData: false,
398
399         success: function(data){
400             switch(data.ReturnCode){
401                 case ITINERARY_DISABLED:
402                     window.location = "./index.php";
403                     break;
404                 case DISABLE_ITINERARY_FAIL:
405                     $('#errorDisable').show().delay(3000).fadeOut
(1000);
406                     break;
407                 default:
408                     alert("-");
409                     break;
410             }
411         },
412
413         error: function (jqXHR){
414             error = "Error :";
415             switch(jqXHR.status){
416                 case INVALID_JSON:
417                     error = error + jqXHR.status + "invalid json";
418                     break;
419                 case FILE_NOT_FOUND:
420                     error = error + jqXHR.status + "Can't find
createItinerary.php";
421                     break;
422             }
423             alert(error);
424         }
425     });
426 }
427 /**
428  * @brief Delete the itinerary
429  *
430  * @param {*} event
431  *
432  * @return void
433  */
434 function DeleteItinerary(event){
435     if (event){
436         event.preventDefault();
437     }
438
439     let idItinerary = $("#itineraryId").val();
440
441     //Create data to send
442     formData = new FormData();
443     formData.append("itineraryId", idItinerary);
444
445     $.ajax({
446         method: 'POST',
447         url: './app/api/deleteItinerary.php',
448         data: formData,
449         dataType: 'json',

```

```

451     contentType: false,
452     processData: false,
453
454     success: function(data){
455         switch(data.ReturnCode){
456             case ITINERARY_DELETED:
457                 window.location = "./index.php";
458                 alert("Your itinerary has been deleted");
459                 break;
460             case DELETE_ITINERARY_FAIL:
461                 $('#errorDelete').show().delay(3000).fadeOut(1000)
462         };
463         break;
464         default:
465             alert("-");
466             break;
467     }
468 },
469
470     error: function (jqXHR){
471         error = "Error :";
472         switch(jqXHR.status){
473             case INVALID_JSON:
474                 error = error + jqXHR.status + "invalid json";
475                 break;
476             case FILE_NOT_FOUND:
477                 error = error + jqXHR.status + "Can't find
createItinerary.php";
478                 break;
479         }
480         alert(error);
481     }
482 });
483 }
484 /**
485  * @brief Cancel the form
486  *
487  * @param {*} event
488  *
489  * @return void
490  */
491 function Cancel(event){
492     if (event) {
493         event.preventDefault();
494     }
495     window.location = "./index.php";
496 }

```

Listing 21: detailsItinerary.js

```

1 /**
2  *      Author           :   Fujise Thomas.
3  *      Project          :   ProjetTPI.
4  *      Page             :   Home.js
5  *      Brief            :   Display maps on itinerary card on home
6  *      Date             :   02.06.2020.

```

```

7  */
8  var maps = [];
9  $(document).ready(function(){
10
11      $("#itineraryFilter").click(itineraryFilter);
12      $("#itineraryFilterCancel").click(cancelFilter);
13      L.mapquest.key = 'xTHtqDgrfGDrRKxzBKyFtDdkqRS4Uu8V';
14      //Show maps after refresh when user use the filter
15      $('body').bind('beforeunload',function(){
16          DisplayMaps();
17      });
18      DisplayMaps();
19  });
20  function DisplayMaps(){
21      /* call the php that has the php array which is json_encoded */
22      $.getJSON('./app/api/getAllItineraries.php', function(data) {
23          var waypointsArray = [];
24          var mapInit = [];
25          /* data will hold the php array as a javascript object */
26          $.each(data, function(key, val) {
27              if($('#map'+val.Id).length){
28                  let map = L.mapquest.map('map' + val.Id, {
29                      center: [val.Waypoints[0].Latitude, val.Waypoints
30                      [0].Longitude],
31                      layers: L.mapquest.tileLayer('map'),
32                      zoom: 8
33                  });
34                  L.marker([val.Waypoints[0].Latitude, val.Waypoints[0].
35                      Longitude], {
36                      icon: L.mapquest.icons.marker({
37                          primaryColor: '#03fc39',
38                          secondaryColor: '#03fc39',
39                          shadow: true,
40                          size: 'md',
41                          symbol: 'S'
42                      })
43                  }).addTo(map);
44                  L.marker([val.Waypoints[val.Waypoints.length - 1].
45                      Latitude, val.Waypoints[val.Waypoints.length - 1].Longitude], {
46                      icon: L.mapquest.icons.marker({
47                          primaryColor: '#fc5203',
48                          secondaryColor: '#fc5203',
49                          shadow: true,
50                          size: 'md',
51                          symbol: 'E'
52                      })
53                  }).addTo(map);
54                  for(let i = 1; i< val.Waypoints.length-1 ; i++){
55                      L.marker([val.Waypoints[i].Latitude, val.Waypoints
56                      [i].Longitude], {
57                          icon: L.mapquest.icons.marker({
58                              primaryColor: '#2803fc',
59                              secondaryColor: '#2803fc',
60                              shadow: true,
61                              size: 'md',
62                              symbol: 'W'
63                          })
64                      }

```

```

60         }).addTo(map);
61     }
62     map.fitBounds([
63         [val.Waypoints[0].Latitude, val.Waypoints[0].
Longitude],
64         [val.Waypoints[val.Waypoints.length - 1].Latitude,
val.Waypoints[val.Waypoints.length - 1].Longitude]
65     ]);
66     if(waypointsArray.length > 0){
67         mapInit[0].waypoints = waypointsArray;
68     }
69     maps.push(map);
70
71     /*mapInit.push({
72         start: val.Waypoints[0].Address,
73         end: val.Waypoints[val.Waypoints.length - 1].
Address
74     });*/
75 }
76
77 /*Add waypoints in array */
78 /*for(let i = 1; i< val.Waypoints.length-1 ; i++){
79     waypointsArray.push(val.Waypoints[i].Latitude + ', ' +
val.Waypoints[i].Longitude);
80 }
81 if(waypointsArray.length > 0){
82     mapInit[0].waypoints = waypointsArray;
83 }*/
84 //console.log(mapInit[key]);
85 //console.log(L.mapquest.directions().route(mapInit[0]));
86 //maps[key].directions().route(mapInit[key]).addTo(map);
87 //waypointsArray = [];
88 //mapInit = [];
89 });
90 });
91
92 }
93 /**
94  * @brief Filter to display itinerary in a specific order
95  *
96  * @param {} event
97  *
98  * @return void
99  */
100 function itineraryFilter(event){
101     if(event){
102         event.preventDefault();
103     }
104
105     let country = $("#countryFilter option:selected").attr("id");
106     let rateMin = $("#rateFilter option:selected").val();
107     let distanceMin = $("#distanceMinFilter").val();
108     let distanceMax = $("#distanceMaxFilter").val();
109     let durationMin = $("#durationMinFilter").val();
110     let durationMax = $("#durationMaxFilter").val();
111
112     var formData = new FormData();

```

```

113     if (country != "None") {
114         formData.append("countryFilter", country);
115     }
116     if(rateMin != "0"){
117         formData.append("rateFilter", rateMin);
118     }
119     if(distanceMin != 0){
120         formData.append("distanceMin", distanceMin);
121     }
122     if(distanceMax != 0){
123         formData.append("distanceMax", distanceMax);
124     }
125     if(durationMin != 0){
126         formData.append("durationMin", durationMin);
127     }
128     if(durationMax != 0){
129         formData.append("durationMax", durationMax);
130     }
131
132     if($.isEmptyObject(formData)){
133         return;
134     }
135
136     $.ajax({
137         method: 'POST',
138         url: './app/api/getAllItineraries.php',
139         data: formData,
140         dataType: 'json',
141         contentType: false,
142         processData: false,
143
144         success: function(data){
145             location.reload();
146         },
147
148         error: function (jqXHR){
149             error = "Error :";
150             switch(jqXHR.status){
151                 case INVALID_JSON :
152                     error = error + jqXHR.status + "invalid json";
153                     break;
154                 case FILE_NOT_FOUND :
155                     error = error + jqXHR.status + "Can't find login.
156 php";
157                     break;
158             }
159             alert(error);
160         });
161     }
162
163     function cancelFilter(event){
164         if(event){
165             event.preventDefault();
166         }
167
168         $.ajax({

```

```

169     method: 'POST',
170     url: './app/api/cancelFilter.php',
171
172     success: function(data){
173         location.reload();
174     },
175
176     error: function (jqXHR){
177         error = "Error :";
178         switch(jqXHR.status){
179             case INVALID_JSON :
180                 error = error + jqXHR.status + "invalid json";
181                 break;
182             case FILE_NOT_FOUND :
183                 error = error + jqXHR.status + "Can't find login.
184 php";
185                 break;
186             }
187             alert(error);
188         });
189     }

```

Listing 22: home.js

```

1 $(document).ready(() => {
2     $('#login').click(login);
3     $('.errmsg').hide();
4 });
5
6 /**
7  * @brief Login with ajax
8  *
9  * @param {*} event
10  */
11 function login(event) {
12     if (event) {
13         event.preventDefault();
14     }
15
16     // init
17     let username = $("#username").val();
18     let password = $("#password").val();
19
20     if (username.length == 0) {
21         $("#username").css("border-color", "red");
22         $("#username").focus();
23         return;
24     } else {
25         $("#username").css("border-color", "");
26     }
27
28     if (password.length == 0) {
29         $("#password").css("border-color", "red");
30         $("#password").focus();
31         return;
32     } else {
33         $("#password").css("border-color", "");

```

```

34     }
35
36     $.ajax({
37         method: 'POST',
38         url: './app/api/login.php',
39         data: {'username': username, 'password': password},
40         dataType: 'json',
41
42         success: function(data){
43             switch(data.ReturnCode){
44                 case LOGIN_DONE :
45                     window.location = "./index.php";
46                     break;
47                 case LOGIN_ACCOUNT_NOT_ACTIVATE :
48                     $('#errorActivation').show().delay(3000).fadeOut
(1000);
49                     break;
50                 case LOGIN_ACCOUNT_BLOCKED :
51                     $('#errorBlocked').show().delay(3000).fadeOut
(1000);
52                     break;
53                 case LOGIN_FAIL :
54                     $('#errorLogin').show().delay(3000).fadeOut(1000);
55                     break;
56                 default:
57                     alert("-");
58                     break;
59             }
60         },
61
62         error: function (jqXHR){
63             error = "Error :";
64             switch(jqXHR.status){
65                 case INVALID_JSON:
66                     error = error + jqXHR.status + "invalid json";
67                     break;
68                 case FILE_NOT_FOUND:
69                     error = error + jqXHR.status + "Can't find login.
php";
70                     break;
71             }
72             alert(error);
73         }
74     });
75 }

```

Listing 23: login.js

```

1 $(document).ready(() => {
2     $('#save').click(SaveInfos);
3     $('#cancel').click(Cancel);
4     $('#errmsg').hide();
5 });
6
7 /**
8  * @brief Save user's infos
9  *
10  * @param {*} event

```



```

11  */
12 function SaveInfos(event) {
13     if (event) {
14         event.preventDefault();
15     }
16
17     // init
18     let nickname = $("#nickname").val();
19     let name = $("#name").val();
20     let surname = $("#surname").val();
21     let userBio = $("#userBio").val();
22     let country = $("#userCountry option:selected").attr("id");
23     let avatar = ($("#avatar").prop('files').length > 0 ? $('#avatar').
prop('files')[0] : null;
24
25     //Create data to send
26     let formData = new FormData();
27     formData.append("nickname", nickname);
28     formData.append("name", name);
29     formData.append("surname", surname);
30     formData.append("userBio", userBio);
31     formData.append("country", country);
32     formData.append("media[]", avatar);
33
34     if (nickname.length == 0) {
35         $("#nickname").css("border-color", "red");
36         $("#nickname").focus();
37         return;
38     } else {
39         $("#nickname").css("border-color", "");
40     }
41
42     $.ajax({
43         method: 'POST',
44         url: './app/api/updateProfil.php',
45         data: formData,
46         dataType: 'json',
47         contentType: false,
48         processData: false,
49
50         success: function(data){
51             switch(data.ReturnCode){
52                 case INFOS_UPDATED :
53                     window.location = "./index.php";
54                     alert("Your infos has been updated");
55                     break;
56                 case INFOS_UPDATED_FAIL :
57                     $('#errorUpdate').show().delay(3000).fadeOut(1000)
;
58                     break;
59                 default:
60                     alert("-");
61                     break;
62             }
63         },
64
65         error: function (jqXHR){

```

```

66         error = "Error :";
67         switch(jqXHR.status){
68             case INVALID_JSON:
69                 error = error + jqXHR.status + "invalid json";
70                 break;
71             case FILE_NOT_FOUND:
72                 error = error + jqXHR.status + "Can't find
updateProfil.php";
73                 break;
74             }
75             alert(error);
76         }
77     });
78 }
79 /**
80  * @brief Cancel the form
81  *
82  * @param {*} event
83  *
84  * @return void
85  */
86 function Cancel(event){
87     if (event) {
88         event.preventDefault();
89     }
90     window.location = "../index.php";
91 }

```

Listing 24: profil.js

```

1 $(document).ready(() => {
2     $('#errmsg').hide();
3     $('#registerUser').click(register);
4 });
5
6 /**
7  * Register with ajax
8  * @param {*} event
9  */
10 function register(event) {
11     if (event) {
12         event.preventDefault();
13     }
14     // Init
15     let nickname = $('#nicknameUser').val();
16     let email = $('#emailUser').val();
17     let password = $('#password').val();
18     let verifyPassword = $('#verifyPassword').val();
19
20     //Create data to send
21     let formData = new FormData();
22     formData.append("nickname", nickname);
23     formData.append("email", email);
24     formData.append("password", password);
25     formData.append("verifyPassword", verifyPassword);
26
27
28

```

```

29     if (nickname.length == 0) {
30         $("#nicknameUser").css("border-color", "red");
31         $("#nicknameUser").focus();
32         return;
33     } else {
34         $("#nicknameUser").css("border-color", "");
35     }
36
37     if (email.length == 0) {
38         $("#emailUser").css("border-color", "red");
39         $("#emailUser").focus();
40         return;
41     } else {
42         $("#emailUser").css("border-color", "");
43     }
44
45     if (password.length == 0) {
46         $("#password").css("border-color", "red");
47         $("#password").focus();
48         return;
49     } else {
50         $("#password").css("border-color", "");
51     }
52
53     if (verifyPassword.length == 0) {
54         $("#verifyPassword").css("border-color", "red");
55         $("#verifyPassword").focus();
56         return;
57     } else {
58         $("#verifyPassword").css("border-color", "");
59     }
60
61     $.ajax({
62         method: 'POST',
63         url: './app/api/register.php',
64         data: formData,
65         dataType: 'json',
66         contentType: false,
67         processData: false,
68
69         success: function(data){
70             switch(data.ReturnCode){
71                 case REGISTER_DONE :
72                     window.location = "./login.php";
73                     break;
74                 case REGISTER_FAIL :
75                     $('#errorParam').show().delay(3000).fadeOut(1000);
76                     break;
77                 case REGISTER_DIFFERENT_PASSWORD:
78                     $('#errorDifferentPassword').show().delay(3000).
79 fadeOut(1000);
80                     break;
81                 case REGISTER_PASSWORD_DONT_MATCH_REQUIREMENTS:
82                     $('#errorPasswordMatch').show().delay(3000).
83 fadeOut(1000);
84                     break;
85                 default:

```

```

84         alert("-");
85         break;
86     }
87 },
88
89     error: function (jqXHR){
90         error = "Error :";
91         switch(jqXHR.status){
92             case INVALID_JSON:
93                 error = error + jqXHR.status + "invalid json";
94                 break;
95             case FILE_NOT_FOUND:
96                 error = error + jqXHR.status + "Can't find
register.php";
97                 break;
98             }
99         alert(error);
100     }
101 });
102 }

```

Listing 25: register.js

```

1 <?php
2 /*
3 *     Author           :  Fujise Thomas.
4 *     Project          :  ProjetTPI.
5 *     Page             :  addComment.
6 *     Brief            :  api to add a comment to an itinerary used
for ajax call.
7 *     Date             :  04.06.2020.
8 */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
manager/FItineraryManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
manager/FSessionManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
manager/FCommentManager.php';
14
15 if(session_status() == PHP_SESSION_NONE){
16     session_start();
17 }
18
19 //Init
20 $title = filter_input(INPUT_POST, "title", FILTER_SANITIZE_STRING);
21 $comment = filter_input(INPUT_POST, "comment", FILTER_SANITIZE_STRING)
;
22 $idUser = FSessionManager::GetUserLogged()->Id;
23 $idItinerary = FItineraryManager::GetInstance()->GetByTitle($title)->
Id;
24 // if all field aren't empty
25 if (strlen($title) > 2 && strlen($comment) > 0) {
26     if(FCommentManager::GetInstance()->AddToItinerary($idItinerary,
$idUser, $comment)) {
27         echo '{ "ReturnCode": 50, "Message": "Comment added"}';
28         exit();

```

```

29     }
30     echo '{ "ReturnCode": 51, "Message": "Add comment fail"}';
31     exit();
32 }
33
34 ?>

```

Listing 26: addComment.php

```

1 <?php
2 /*
3 *     Author           :  Fujise Thomas.
4 *     Project          :  ProjetTPI.
5 *     Page             :  addRate.
6 *     Brief            :  api to add a rate to an itinerary used
7 *     Date             :  04.06.2020.
8 */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
12     manager/FItineraryManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
14     manager/FSessionManager.php';
15 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
16     manager/FRateManager.php';
17
18 if(session_status() == PHP_SESSION_NONE){
19     session_start();
20 }
21
22 //Init
23 $title = filter_input(INPUT_POST, "title", FILTER_SANITIZE_STRING);
24 $rate = filter_input(INPUT_POST, "rate", FILTER_SANITIZE_STRING);
25 $idUser = FSessionManager::GetUserLogged()->Id;
26 $idItinerary = FItineraryManager::GetInstance()->GetByTitle($title)->
27     Id;
28 // if all field aren't empty
29 if (strlen($title) > 2 && strlen($rate) > 0) {
30     if($rate < 10 && $rate > 0){
31         if(FRateManager::GetInstance()->AddToItinerary($idItinerary,
32             $idUser, $rate)) {
33             echo '{ "ReturnCode": 60, "Message": "Rate added"}';
34             exit();
35         }
36         echo '{ "ReturnCode": 61, "Message": "Rate add fail"}';
37         exit();
38     }
39     echo '{ "ReturnCode": 62, "Message": "Rate not valid"}';
40     exit();
41 }
42
43 ?>

```

Listing 27: addRate.php

```

1 <?php
2 /*

```

```

3  *      Author      :  Fujise Thomas.
4  *      Project     :  ProjetTPI.
5  *      Page        :  cancelFilter.
6  *      Brief       :  api to cancel the itinerary filter used
    for ajax call.
7  *      Date        :  04.06.2020.
8  */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'../ProjectTPI/src/www/app/
    manager/FSessionManager.php';
12
13 if(session_status() == PHP_SESSION_NONE){
14     session_start();
15 }
16
17 FSessionManager::UnsetItineraryFilter();
18
19 ?>

```

Listing 28: cancelFilter.php

```

1  <?php
2  /*
3  *      Author      :  Fujise Thomas.
4  *      Project     :  ProjetTPI.
5  *      Page        :  createItinerary.
6  *      Brief       :  api to create itinerary used for ajax
    call.
7  *      Date        :  04.06.2020.
8  */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'../ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'../ProjectTPI/src/www/app/
    manager/FSessionManager.php';
13
14 if(session_status() == PHP_SESSION_NONE){
15     session_start();
16 }
17
18 //Init
19 $title = filter_input(INPUT_POST, "title", FILTER_SANITIZE_STRING);
20 $country = filter_input(INPUT_POST, "country", FILTER_SANITIZE_STRING)
    ;
21 $description = filter_input(INPUT_POST, "description",
    FILTER_SANITIZE_STRING);
22 $duration = filter_input(INPUT_POST, "duration",
    FILTER_SANITIZE_STRING);
23 $distance = filter_input(INPUT_POST, "distance", FILTER_VALIDATE_FLOAT
    );
24 $waypoints = json_decode($_POST['waypoints']);
25 $idUser = FSessionManager::GetUserLogged()->Id;
26
27 if(isset($_FILES["media"])){
28     $photos = $_FILES["media"];
29 }

```

```

30 //Array for all images uploaded encoded in base 64
31 $srcPhotos = [];
32 // if all field aren't empty
33 if (strlen($title) > 2 && strlen($description) > 0 && strlen($duration
    ) > 0) {
34     if($distance < 999 || $distance > 1) {
35         // if user add photos to his itinerary
36         if(!isset($photos) || !is_uploaded_file($photos['tmp_name'][0])){
37             if(FItineraryManager::GetInstance()->Create($title,
                $description, $duration, $distance, $country, $waypoints,$idUser))
            {
38                 echo '{ "ReturnCode": 40, "Message": "Create itinerary
                done"}';
39                 exit();
40             }
41             echo '{ "ReturnCode": 41, "Message": "Create itinerary fail"}'
            ;
42             exit();
43         }
44         else{
45             for($i = 0;$i<count($photos['tmp_name']);$i++){
46                 $data = file_get_contents($photos['tmp_name'][$i]);
47                 //Get MIME type
48                 //Need to uncomment the line : extension=fileinfo in php.
                ini
49                 $finfo = new finfo(FILEINFO_MIME_TYPE);
50                 $mime = $finfo->file($photos['tmp_name'][$i]);
51                 array_push($srcPhotos, 'data:'. $mime. ';base64,'.
                base64_encode($data));
52             }
53             if(FItineraryManager::GetInstance()->Create($title,
                $description, $duration, $distance, $country, $waypoints,$idUser,
                $srcPhotos)) {
54                 echo '{ "ReturnCode": 40, "Message": "Create itinerary
                done"}';
55                 exit();
56             }
57             echo '{ "ReturnCode": 41, "Message": "Create itinerary fail"}'
            ;
58             exit();
59         }
60     }
61     echo '{ "ReturnCode": 42, "Message": "Distance not valid"}';
62     exit();
63 }
64
65 ?>

```

Listing 29: createItinerary.php

```

1 <?php
2 /*
3 *     Author           :   Fujise Thomas.
4 *     Project          :   ProjetTPI.
5 *     Page             :   deleteItinerary.
6 *     Brief            :   api to disable the account used for ajax
    call.
7 *     Date             :   04.06.2020.

```

```

8  */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'../ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
12
13 if(session_status() == PHP_SESSION_NONE){
14     session_start();
15 }
16
17 $idItinerary = filter_input(INPUT_POST, "itineraryId",
    FILTER_SANITIZE_STRING);
18
19 if(FItineraryManager::GetInstance()->Delete($idItinerary)){
20     echo '{ "ReturnCode": 100, "Message": "Itinerary deleted"}';
21     exit();
22 }else{
23     echo '{ "ReturnCode": 101, "Message": "Itinerary delete fail"}';
24     exit();
25 }
26
27 ?>

```

Listing 30: deleteItinerary.php

```

1 <?php
2 /*
3 *      Author          :  Fujise Thomas.
4 *      Project         :  ProjetTPI.
5 *      Page            :  disableAccount.
6 *      Brief           :  api to disable the account used for ajax
    call.
7 *      Date            :  04.06.2020.
8 */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'../ProjectTPI/src/www/app/
    manager/FUserManager.php';
12
13 if(session_status() == PHP_SESSION_NONE){
14     session_start();
15 }
16
17 $nickname = filter_input(INPUT_POST, "nickname",
    FILTER_SANITIZE_STRING);
18
19 if(FUserManager::GetInstance()->DisableAccount($nickname)){
20     echo '{ "ReturnCode": 80, "Message": "Account disabled"}';
21     exit();
22 }else{
23     echo '{ "ReturnCode": 81, "Message": "Disable account fail"}';
24     exit();
25 }
26
27 ?>

```

Listing 31: disableAccount.php



```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  disableItinerary.
6 *      Brief            :  api to disable the itinerary used for
7 *                        ajax call.
8 *      Date              :  04.06.2020.
9 */
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
12     manager/FItineraryManager.php';
13 if(session_status() == PHP_SESSION_NONE){
14     session_start();
15 }
16
17 $idItinerary = filter_input(INPUT_POST, "itineraryId",
18     FILTER_SANITIZE_STRING);
19 $title = filter_input(INPUT_POST, "title", FILTER_SANITIZE_STRING);
20 if(FItineraryManager::GetInstance()->Disable($idItinerary, $title)){
21     echo '{ "ReturnCode": 90, "Message": "Itinerary disabled"}';
22     exit();
23 }else{
24     echo '{ "ReturnCode": 91, "Message": "Disable itinerary fail"}';
25     exit();
26 }
27
28 ?>

```

Listing 32: disableItinerary.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  getAllItineraries.
6 *      Brief            :  Api to get all itineraries store in
7 *                        database used for ajax call.
8 *      Date              :  02.06.2020.
9 */
10 /* Requirements */
11 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
12     manager/FItineraryManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
14     manager/FSessionManager.php';
15
16 if (session_status() == PHP_SESSION_NONE) {
17     session_start();
18 }
19 /* Set document type to text/javascript */
20 header("Content-type: text/javascript");
21
22 $country = filter_input(INPUT_POST, "countryFilter",
23     FILTER_SANITIZE_STRING);

```

```

21 $rateMin = filter_input(INPUT_POST, "rateFilter",
    FILTER_SANITIZE_STRING);
22 $durationMin = filter_input(INPUT_POST, "durationMin",
    FILTER_SANITIZE_STRING);
23 $durationMax = filter_input(INPUT_POST, "durationMax",
    FILTER_SANITIZE_STRING);
24 $distanceMin = filter_input(INPUT_POST, "distanceMin",
    FILTER_VALIDATE_FLOAT);
25 $distanceMax = filter_input(INPUT_POST, "distanceMax",
    FILTER_VALIDATE_FLOAT);
26
27 if($country == null && $rateMin == null && $durationMin == null &&
    $durationMax == null && $distanceMin == null && $distanceMax ==
    null){
28     /* Encode the array with all itineraries as json. */
29     echo json_encode(FItineraryManager::GetInstance()->GetAll());
30 }
31 else{
32     FSessionManager::SetItineraryFilter([
33         'country' => $country,
34         'rateMin' => $rateMin,
35         'durationMin' => $durationMin,
36         'durationMax' => $durationMax,
37         'distanceMin' => $distanceMin,
38         'distanceMax' => $distanceMax
39     ]);
40     echo json_encode(FItineraryManager::GetInstance()->
    GetAllWithFilter(FSessionManager::GetItineraryFilter()));
41 }
42
43 ?>

```

Listing 33: getAllItineraries.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  getItineraryById.
6 *      Brief            :  Api to get the FItinerary object with
    itinerary id used for ajax call
7 *      Date             :  03.06.2020.
8 */
9 /* Requirements */
10 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FSessionManager.php';
12
13 /* Set document type to text/javascript */
14 header("Content-type: text/javascript");
15 $id = $_GET['id'];
16
17 /* Encode the object FItinerary as json. */
18 echo json_encode(FItineraryManager::GetInstance()->GetById($id));
19 ?>

```

Listing 34: getItineraryById.php

```

1 <?php
2 /*
3 *      Author           :   Fujise Thomas.
4 *      Project          :   ProjetTPI.
5 *      Page             :   login.
6 *      Brief            :   api to log the user used for ajax call.
7 *      Date             :   19.05.2020.
8 */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
    manager/FUserManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
    manager/FSessionManager.php';
13
14 if(session_status() == PHP_SESSION_NONE){
15     session_start();
16 }
17
18 //Init
19 $username = filter_input(INPUT_POST, "username",
    FILTER_SANITIZE_STRING);
20 $password = filter_input(INPUT_POST, "password",
    FILTER_SANITIZE_STRING);
21
22 if(strlen($username) > 0 && strlen($password) > 0) {
23
24     if(strpos($username, '@')) {
25         $userLogged = FUserManager::GetInstance()->Login(['userEmail'
    => $username, 'userPwd' => $password]);
26     } else {
27         $userLogged = FUserManager::GetInstance()->Login(['
    userNickname' => $username, 'userPwd' => $password]);
28     }
29
30     if($userLogged !== false) {
31         if(FUserManager::GetInstance()->VerifyActivation($userLogged->
    Nickname)==1){
32             echo '{ "ReturnCode": 11, "Message": "Account not activate
    ." }';
33             exit();
34         }
35         elseif(FUserManager::GetInstance()->VerifyActivation(
    $userLogged->Nickname)==3){
36             echo '{ "ReturnCode": 13, "Message": "Account blocked." }';
37             exit();
38         }
39         FSessionManager::setUserLogged($userLogged);
40         $_SESSION['isLoggedIn'] = true;
41
42         echo '{ "ReturnCode": 10, "Message": "Login done" }';
43         exit();
44     }
45
46     echo '{ "ReturnCode": 12, "Message": "Wrong username or password" }
    ' ;
47 }

```

## Listing 35: login.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  register.
6 *      Brief            :  api to register the user used for ajax
                          call.
7 *      Date             :  19.05.2020.
8 */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FUserManager.php';
12
13 if(session_status() == PHP_SESSION_NONE){
14     session_start();
15 }
16
17 //Init
18 $nickname = filter_input(INPUT_POST, "nickname",
    FILTER_SANITIZE_STRING);
19 $email = filter_input(INPUT_POST, "email", FILTER_SANITIZE_EMAIL);
20 $password = filter_input(INPUT_POST, "password",
    FILTER_SANITIZE_STRING);
21 $verifyPassword = filter_input(INPUT_POST, "verifyPassword",
    FILTER_SANITIZE_STRING);
22
23 // Regex with min 8 char, 1 maj and 1 number
24 $regex = "/^(?=\w{8,})(?=[^a-z]*[a-z])(?=[^A-Z]*[A-Z])(\w*\d)\w*/";
25
26 // if all field aren't empty
27 if (strlen($nickname) > 2 && strlen($email) > 0 && strlen($password) >
    0 && strlen($verifyPassword) > 0) {
28     // if regex match with password
29     if ((preg_match($regex, $password))) {
30         // Both password match
31         if ($password == $verifyPassword) {
32             // if account is register
33             if(FUserManager::GetInstance()->Register($email, $nickname
    , $password)) {
34                 echo '{ "ReturnCode": 0, "Message": "Register done"}';
35                 exit();
36             }
37             echo '{ "ReturnCode": 1, "Message": "Register fail"}';
38             exit();
39         }
40         echo '{ "ReturnCode": 2, "Message": "Different passwords"}';
41         exit();
42     }
43     echo '{ "ReturnCode": 3, "Message": "Password dont match
    requirements"}';
44     exit();
45 }
46

```

47 ?>

Listing 36: register.php

```
1 <?php
2 /*
3 *      Author      :  Fujise Thomas.
4 *      Project     :  ProjetTPI.
5 *      Page        :  updateItinerary.
6 *      Brief       :  api to update an itinerary used for ajax
7 *                  call.
8 *      Date        :  05.06.2020.
9 */
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
12     manager/FItineraryManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
14     manager/FSessionManager.php';
15 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
16     manager/FWaypointsManager.php';
17
18 if(session_status() == PHP_SESSION_NONE){
19     session_start();
20 }
21
22 //Init
23 $id = filter_input(INPUT_POST, "itineraryId",
24     FILTER_SANITIZE_NUMBER_INT);
25 $title = filter_input(INPUT_POST, "title", FILTER_SANITIZE_STRING);
26 $country = filter_input(INPUT_POST, "country", FILTER_SANITIZE_STRING)
27     ;
28 $description = filter_input(INPUT_POST, "description",
29     FILTER_SANITIZE_STRING);
30 $duration = filter_input(INPUT_POST, "duration",
31     FILTER_SANITIZE_STRING);
32 $distance = filter_input(INPUT_POST, "distance", FILTER_VALIDATE_FLOAT
33     );
34 $waypoints = json_decode($_POST['waypoints']);
35
36 $idUser = FSessionManager::GetUserLogged()->Id;
37 $idItinerary = FItineraryManager::GetInstance()->GetById($id)->Id;
38
39 if(isset($_FILES["media"])){
40     $photos = $_FILES["media"];
41 }
42 //Array for all images uploaded encoded in base 64
43 $srcPhotos = [];
44 // if all field aren't empty
45 if (strlen($title) > 2 && strlen($description) > 0 && strlen($duration
46     ) > 0) {
47     if($distance < 999 || $distance > 1) {
48         // if user add photos to his itinerary
49         if(!isset($photos) || !is_uploaded_file($photos['tmp_name'][0])){
50             if(FItineraryManager::GetInstance()->Update($idUser,
51                 $idItinerary, $title, $description, $duration, $distance, $country,
52                 $waypoints)) {
```

```

43         echo '{ "ReturnCode": 70, "Message": "Itinerary updated"}'
44     ;
45     exit();
46 }
47 echo '{ "ReturnCode": 71, "Message": "Itinerary update fail"}'
48 ;
49 exit();
50 }
51 else{
52     for($i = 0;$i<count($photos['tmp_name']);$i++){
53         $data = file_get_contents($photos['tmp_name'][$i]);
54         //Get MIME type
55         //Need to uncomment the line : extension=fileinfo in php.
56         ini
57             $finfo = new finfo(FILEINFO_MIME_TYPE);
58             $mime = $finfo->file($photos['tmp_name'][$i]);
59             array_push($srcPhotos, 'data:'.$mime.';base64,'.
60             base64_encode($data));
61     }
62     if(FItineraryManager::GetInstance()->Update($idUser,
63     $idItinerary, $title, $description, $duration, $distance, $country,
64     $waypoints, $srcPhotos)) {
65         echo '{ "ReturnCode": 70, "Message": "Itinerary updated"}'
66     ;
67         exit();
68     }
69     echo '{ "ReturnCode": 71, "Message": "Itinerary update fail"}'
70 ;
71     exit();
72 }
73 }
74 echo '{ "ReturnCode": 42, "Message": "Distance not valid"}';
75 exit();
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

Listing 37: updateItinerary.php

```

1 <?php
2 /*
3 *     Author           : Fujise Thomas.
4 *     Project          : ProjetTPI.
5 *     Page             : updateProfil.
6 *     Brief            : api to update the profil used for ajax
7 *     Date             : 04.06.2020.
8 */
9
10 //requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
12     manager/FUserManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
14     manager/FSessionManager.php';
15
16 if(session_status() == PHP_SESSION_NONE){
17     session_start();
18 }

```

```

17
18 //Init
19 $nickname = filter_input(INPUT_POST, "nickname",
    FILTER_SANITIZE_STRING);
20 $name = filter_input(INPUT_POST, "name", FILTER_SANITIZE_STRING);
21 $surname = filter_input(INPUT_POST, "surname", FILTER_SANITIZE_STRING)
    ;
22 $userBio = filter_input(INPUT_POST, "userBio", FILTER_SANITIZE_STRING)
    ;
23 $country = filter_input(INPUT_POST, "country", FILTER_SANITIZE_STRING)
    ;
24
25 $idUser = FSessionManager::GetUserLogged()->Id;
26
27 if(isset($_FILES["media"])){
28     $avatar = $_FILES["media"];
29 }
30 //Get current user's avatar
31 $srcAvatar = FSessionManager::GetUserLogged()->Avatar;
32 // if the required field isn't't empty
33 if (strlen($nickname) > 2) {
34     // if user don't add avatar
35     if(!isset($avatar) || !is_uploaded_file($avatar['tmp_name'][0])){
36         if(FUserManager::GetInstance()->UpdateInfos($idUser, $nickname
    , $name, $surname, $userBio, $country, $srcAvatar)) {
37             FSessionManager::SetUserLogged(FUserManager::GetInstance()
    ->GetById($idUser));
38             echo '{ "ReturnCode": 30, "Message": "Infos updated"}';
39             exit();
40         }
41         echo '{ "ReturnCode": 31, "Message": "Update fail"}';
42         exit();
43     }
44     else{
45         $data = file_get_contents($avatar['tmp_name'][0]);
46         //Get MIME type
47         //Need to uncomment the line : extension=fileinfo in php.ini
48         $finfo = new finfo(FILEINFO_MIME_TYPE);
49         $mime = $finfo->file($avatar['tmp_name'][0]);
50         $srcAvatar = 'data:'.$mime.';base64,'.base64_encode($data);
51         if(FUserManager::GetInstance()->UpdateInfos($idUser, $nickname
    , $name, $surname, $userBio, $country, $srcAvatar)) {
52             FSessionManager::SetUserLogged(FUserManager::GetInstance()
    ->GetById($idUser));
53             echo '{ "ReturnCode": 30, "Message": "Infos updated"}';
54             exit();
55         }
56         echo '{ "ReturnCode": 31, "Message": "Update fail"}';
57         exit();
58     }
59 }
60 }
61
62 ?>

```

Listing 38: updateProfil.php

```
1 <?php
```

```

2  /*
3  *      Author          :  Fujise Thomas.
4  *      Project         :  ProjetTPI.
5  *      Page            :  getMyItineraries.
6  *      Brief           :  Api to get only the itineraries created
    by the user logged used for ajax call.
7  *      Date            :  02.06.2020.
8  */
9  /* Requirements */
10 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
    manager/FSessionManager.php';
12
13 /* Set document type to text/javascript */
14 header("Content-type: text/javascript");
15
16 /* Encode the array with all itineraries as json. */
17 echo json_encode(FItineraryManager::GetInstance()->GetAllByUserId(
    FSessionManager::getUserLogged()->Id));
18 ?>

```

Listing 39: getMyItineraries.php

```

1  <?php
2  /*
3  *      Author          :  Fujise Thomas.
4  *      Project         :  ProjetTPI.
5  *      Page            :  FAdminView.
6  *      Brief           :  home view.
7  *      Date            :  02.06.2020.
8  */
9  //Requirements
10 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
    manager/FUserManager.php';
11
12 /**
13  * Represents admin panel page view.
14  */
15 class FAdminView{
16     /**
17      * @brief Class constructor
18      */
19     function __construct(){}
20     /**
21      * @brief Display all users registered
22      *
23      * @param array $users all users to display
24      *
25      * @return string code html to display
26      */
27     public static function DisplayUsers($users) : string{
28         $result = '';
29         $status = '';
30         $action = '';
31         if($users == FALSE)
32         {
33             $result .= <<<EX

```



```

34         <div class="lead font-weight-bold">You don't create
        itinerary yet</div>
35         EX;
36     }
37     else{
38         for($i = 0;$i<count($users);$i++){
39             switch($users[$i]->Status){
40                 case 1:
41                     $status = "Not activate";
42                     $action = '<a href="#" class="btn btn-danger"
data-type="disable" data-nickname="' . $users[$i]->Nickname . '"
data-target="#modalDisable" data-toggle="modal">Disable</a>';
43                     break;
44                 case 2:
45                     $status = "Activated";
46                     $action = '<a href="#" class="btn btn-danger"
data-type="disable" data-nickname="' . $users[$i]->Nickname . '"
data-target="#modalDisable" data-toggle="modal">Disable</a>';
47
48                     break;
49                 case 3:
50                     $status = "Blocked";
51                     $action = '<a href="#" class="btn btn-success"
data-type="disable" data-nickname="' . $users[$i]->Nickname . '"
data-target="#modalDisable" data-toggle="modal">Enable</a>';
52                     break;
53             }
54             $result .= <<<EX
55                 <tr>
56                     <td>{$users[$i]->Nickname}</td>
57                     <td>{$users[$i]->Email}</td>
58                     <td>{$users[$i]->Name}</td>
59                     <td>{$users[$i]->Surname}</td>
60                     <td>{$users[$i]->Country}</td>
61                     <td>{$status}</td>
62                     <td>{$action}</td>
63                 </tr>
64             EX;
65         }
66     }
67     return $result;
68 }
69
70 }
71
72 ?>

```

Listing 40: FAdminView.php

```

1 <?php
2 /*
3 *      Author           :   Fujise Thomas.
4 *      Project          :   ProjetTPI.
5 *      Page             :   FHomeView.
6 *      Brief            :   home view.
7 *      Date             :   02.06.2020.
8 */
9 //Requirements

```

```

10 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FUserManager.php';
11
12 /**
13  * Represents home page view.
14  */
15 class FItineraryView{
16     /**
17      * @brief Class constructor
18      */
19     function __construct(){}
20 /**
21  * @brief Display all itineraries
22  *
23  * @param array $itineraries all itineraries to display
24  *
25  * @return string code html to display
26  */
27 public static function DisplayItineraries($itineraries) : string{
28     $result = '';
29     if($itineraries == FALSE)
30     {
31         $result .= <<<EX
32         <div class="lead font-weight-bold">You don't create itinerary yet
33         </div>
34         EX;
35     }
36     else{
37         for($i = 0;$i<count($itineraries);$i++){
38             $result .= self::DisplayItinerary($itineraries[$i]);
39         }
40     }
41     return $result;
42 }
43
44 /**
45  * @brief Display one itinerariy
46  *
47  * @param FItinary $itineraries all itineraries to display
48  *
49  * @return string code html to display
50  */
51 private static function DisplayItinerary($itinerary) : string{
52     $result = '';
53     if($itinerary == FALSE)
54     {
55         $result .= <<<EX
56         <div class="lead font-weight-bold">No itinerary available here,
57         maybe create on ?</div>
58         EX;
59     }
60     else{
61         $result .= <<<EX
62         <div class="col-md-6 col-lg-4 mb-5">
63         <div class="portfolio-item mx-auto card" data-toggle="modal" data-
64         target="#portfolioModal{$itinerary->Id}">
65         <div id="map{$itinerary->Id}" class="card-img-top" style="width:

```

```

100%; height: 200px;"></div>
63 <div class="card-body">
64 <a href="./itineraryDetails.php?id={$itinerary->Id}" ><h5 class="
card-title">{$itinerary->Title}</h5></a>
65 <p class="card-text">{$itinerary->Description}</p>
66 <a href="#" id="info{$itinerary->Id}" class="btn btn-primary">More
infos</a>
67 </div>
68 </div>
69 </div>
70 EX;
71 }
72 return $result;
73 }
74 /**
75  * @brief Display all photos uploaded for a itinerary
76  *
77  * @param array $photos All the photos to display
78  *
79  * @return string code html to display
80  */
81 public static function DisplayPhotosItineraries($photos){
82     $result = '';
83     if($photos != FALSE){
84         for($i = 0;$i<count($photos);$i++){
85             if($i == 0){
86                 $result .= <<<EX
87                 <div class="carousel-item active">
88                     <img class="d-block w-100" height="450px" src
="{$photos[$i]->Image}" alt="Photo from itinerary">
89                 </div>
90                 EX;
91             }else{
92                 $result .= <<<EX
93                 <div class="carousel-item">
94                     
95                 </div>
96                 EX;
97             }
98         }
99     }
100     return $result;
101 }
102 /**
103  * @brief Display all comments added for a itinerary
104  *
105  * @param array $comments All the comments to display
106  *
107  * @return string code html to display
108  */
109 public static function DisplayCommentsItineraries($comments){
110     $result = '';
111     if($comments != FALSE){
112         for($i = 0;$i<count($comments);$i++){
113             $result .= <<<EX

```

```

114         <div class="d-flex flex-row comment-row m-t-0">
115             <div class="p-2"></div>
116             <div class="comment-text w-100">
117                 <h6 class="font-medium">{ $comments[$i]->
User->Nickname}</h6> <span class="m-b-15 d-block">{ $comments[$i]->
Comment}</span>
118                 <div class="comment-footer"> <span class="
text-muted float-right">{ $comments[$i]->PostedAt}</span>
119                 </div>
120             </div>
121             EX;
122         }
123     }
124     return $result;
125 }
126 }
127
128 ?>

```

Listing 41: FltineraryView.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  FCodeManager.
6 *      Brief            :  code manager.
7 *      Date             :  03.06.2020.
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/config/
config.php';
10 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/model/
FCountry.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/app/
manager/FDatabaseManager.php';
12 /**
13 * Class FCodeManager, manager for all code table in the db
14 */
15 class FCodeManager extends FDatabaseManager{
16     /**
17     * @var static $instance the instance for the manager
18     */
19     private static $instance;
20     /**
21     * @brief Class constructor, init all field from table 'COUNTRIES'
22     */
23     function __construct(){
24         $this->tableCountries = "COUNTRIES";
25         $this->fieldCode = "ISO2";
26         $this->fieldName = "NAME";
27     }
28     /**
29     * @brief Create the initial instance for the manager or get it
30     * @return $instance
31     */
32     public static function GetInstance(){
33         if(!self::$instance){

```

```

34         self::$instance = new FCodeManager();
35     }
36     return self::$instance;
37 }
38 /**
39  * @brief Get all country in the database
40  *
41  */
42 public function getAllCountry() {
43     $result = array();
44     $query = <<<EX
45         SELECT '{$this->fieldName}', '{$this->fieldCode}'
46         FROM '{$this->tableCountries}'
47         ORDER BY '{$this->fieldName}'
48     EX;
49
50     try {
51         $req = $this::getDb()->prepare($query);
52         $req->execute();
53
54         while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
55         FETCH_ORI_NEXT)){
56             $country = new FCountry($row[$this->fieldName], $row[
57             $this->fieldCode]);
58             array_push($result, $country);
59         }
60
61         return count($result) > 0 ? $result : FALSE;
62     } catch(PDOException $e) {
63         echo "Can't read the database" . $e->getMessage();
64
65         return FALSE;
66     }
67 }
68 }
69 ?>

```

Listing 42: FCodeManager.php

```

1 <?php
2 /*
3  *      Author           :  Fujise Thomas.
4  *      Project          :  ProjetTPI.
5  *      Page             :  FCommentManager.
6  *      Brief            :  comment manager.
7  *      Date             :  04.09.2019.
8  */
9
10 //Requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
12     manager/FDatabaseManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
14     manager/FUserManager.php';
15 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/model/
16     FComment.php';
17 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
18     manager/FMailerManager.php';

```

```

15 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FItineraryManager.php';
16
17 /**
18  * Comment's manager, manager for comment table
19  */
20 class FCommentManager extends FDatabaseManager{
21
22     /**
23      * @var static $instance the instance for the manager
24      * */
25     private static $instance;
26     /**
27      * @brief Class constructor, init all field from table 'COMMENTS'
28      */
29     function __construct() {
30         $this->tableName = "COMMENTS";
31         $this->fieldComment = "COMMENT";
32         $this->fieldDate = "COMMENT_DATE";
33         $this->fieldUser = "USERS_ID";
34         $this->fieldItinerary = "ITINERARY_ID";
35     }
36     /**
37      * @brief Create the initial instance for the manager or get it
38      * @return $instance
39      */
40     public static function GetInstance(){
41         if(!self::$instance){
42             self::$instance = new FCommentManager();
43         }
44         return self::$instance;
45     }
46     /**
47      * @brief Get all comment from an itinerary with his unique's id
48      *
49      * @param int $itineraryId Itinerary's unique id
50      *
51      * @return array Array with all comments
52      */
53     public function GetAllById(int $itineraryId) {
54         $result = array();
55
56         $query = <<<EX
57             SELECT '{$this->fieldComment}','{$this->fieldDate}','{
58 $this->fieldUser}'
59             FROM '{$this->tableName}'
60             WHERE '{$this->fieldItinerary}' = :itineraryId
61             ORDER BY '{$this->fieldDate}'
62         EX;
63
64         try{
65             $req = $this->getDb()->prepare($query);
66             $req->bindParam(':itineraryId', $itineraryId, PDO::
67 PARAM_INT);
68             $req->execute();
69
70             while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::

```

```

100         FETCH_ORI_NEXT)) {
101             $comment = new FComment($row[$this->fieldComment],
102                                     $row[$this->fieldDate], FUserManager::GetInstance()->GetById($row[
103                                     $this->fieldUser]));
104             array_push($result, $comment);
105         }
106     } catch(PDOException $e){
107         return FALSE;
108     }
109     return count($result) > 0 ? $result : FALSE;
110 }
111 /**
112  * @brief Add a comment to an itinerary and send mail to owner
113  *
114  * @param int $idItinerary itinerary's id
115  * @param int $idUser user's id
116  * @param string $comment the comment the user post
117  *
118  * @return bool true if insert success, else false
119  */
120 public function AddToItinerary(int $idItinerary, int $idUser,
121 string $comment){
122     $query = <<<EX
123         INSERT INTO '{$this->tableName}' ('{$this->fieldComment
124     }, '{$this->fieldDate}', '{$this->fieldUser}', '{$this->
125     fieldItinerary}')
126         VALUES (:comment, :date, :idUser, :idItinerary)
127     EX;
128     try{
129         $date = date("Y-m-d H:i:s");
130         $req = $this->getDb()->prepare($query);
131         $req->bindParam(':comment', $comment, PDO::PARAM_STR);
132         $req->bindParam(':date', $date, PDO::PARAM_STR);
133         $req->bindParam(':idUser', $idUser, PDO::PARAM_INT);
134         $req->bindParam(':idItinerary', $idItinerary, PDO::
135     PARAM_INT);
136         $req->execute();
137
138         FMailerManager::sendMail("Comment added on your itinerary"
139     , array(FUserManager::GetInstance()->GetById(FItineraryManager::
140     GetInstance()->GetById($idItinerary)->User)->Email), FMailerManager
141     ::getCommentAddMail($comment, FUserManager::GetInstance()->GetById(
142     $idUser)->Nickname, FItineraryManager::GetInstance()->GetById(
143     $idItinerary)->Title));
144     } catch(PDOException $e){
145         return FALSE;
146     }
147     //Done
148     return TRUE;
149 }
150 }
151 ?>

```

Listing 43: FCommentManager.php

```

1 <?php
2 /*

```

```

3  *      Author      :  Fujise Thomas.
4  *      Project     :  ProjetTPI.
5  *      Page        :  FDatabaseManager.
6  *      Brief       :  Database's manager.
7  *      Date        :  28.05.2020.
8  *      Version     :  1.0.
9  */
10 //Requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'./ProjectTPI/src/www/config/
    config.php';
12 /**
13  * Class FDatabase
14  */
15 class FDatabaseManager {
16     /**
17      * @var static $db the instance for the manager
18      * */
19     private static $db ;
20     /**
21      * @brief Class constructor, create a new database connection if
22      one doesn't exist.
23      */
24     private function __construct(){}
25     /**
26      * @brief Like the constructor, __clone is private so nobody can
27      clone the instance.
28      */
29     private function __clone(){}
30     /**
31      * @brief Create the initial connection to the database or get the
32      instance
33      *
34      * @return PDO $db PDO instance
35      */
36     public static function getDb() {
37         if(!self::$db){
38             try{
39                 $connectString = DB_DBTYPE.':host='.DB_HOST.';dbname='
40 .DB_NAME;
41                 self::$db = new PDO($connectString, DB_USER, DB_PASS,
42 array('charset'=>'utf8'));
43                 self::$db->setAttribute(PDO::ATTR_ERRMODE, PDO::
44 ERRMODE_EXCEPTION);
45                 }catch(PDOException $e){
46                     echo "EDatabase Error: ".$e;
47                 }
48             }
49             return self::$db;
50         }
51     }
52 }
53 ?>

```

Listing 44: FDatabaseManager.php

```

1 <?php
2 /*
3  *      Author      :  Fujise Thomas.
4  *      Project     :  ProjetTPI.

```



```

5 *      Page           : FItineraryManager.
6 *      Brief          : itinerary manager.
7 *      Date           : 04.09.2019.
8 */
9
10 //Requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FDatabaseManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FMailerManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/model/
    FItinerary.php';
14 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FWaypointsManager.php';
15 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FCommentManager.php';
16 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/app/
    manager/FPhotoManager.php';
17 require_once $_SERVER['DOCUMENT_ROOT'].'ProjectTPI/src/www/constants/
    constants.php';
18 /**
19  * Itinerary's manager, manager for the table itinerary in the db
20  */
21 class FItineraryManager extends FDatabaseManager{
22
23     /**
24      * @var static $instance the instance for the manager
25      */
26     private static $instance;
27     /**
28      * @brief Class constructor, init all field from table 'ITINERARY'
29      */
30     function __construct() {
31         $this->tableName = "ITINERARY";
32         $this->fieldId = "ID";
33         $this->fieldTitle = "TITLE";
34         $this->fieldRating = "RATING";
35         $this->fieldDescription = "DESCRIPTION";
36         $this->fieldDuration = "DURATION";
37         $this->fieldDistance = "DISTANCE";
38         $this->fieldCountry = "COUNTRIES_ISO2";
39         $this->fieldStatus = "STATUS_ID";
40         $this->fieldUser = "USERS_ID";
41     }
42     /**
43      * @brief Create the initial instance for the manager or get it
44      * @return $instance
45      */
46     public static function GetInstance(){
47         if(!self::$instance){
48             self::$instance = new FItineraryManager();
49         }
50         return self::$instance;
51     }
52     /**
53      * @brief Get all itinerary
54      */

```

```

55     * @return array Array of Fltinereary with all itinerary
56     */
57     public function GetAll(){
58         $result = array();
59
60         $query = <<<EX
61         SELECT '{$this->fieldId}', '{$this->fieldTitle}', '{$this-
->fieldRating}', '{$this->fieldDescription}', '{$this->fieldDuration
}' , '{$this->fieldDistance}', '{$this->fieldCountry}', '{$this->
fieldStatus}', '{$this->fieldUser}'
62         FROM '{$this->tableName}'
63         WHERE '{$this->fieldStatus}' = 2
64         ORDER BY '{$this->fieldRating}' DESC
65     EX;
66
67     try{
68         $req = $this::getDb()->prepare($query);
69         $req->execute();
70
71         while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
FETCH_ORI_NEXT)){
72             $itinerary = new Fltinereary($row[$this->fieldId], $row
[$this->fieldTitle], $row[$this->fieldRating], $row[$this->
fieldDescription], $row[$this->fieldDuration], $row[$this->
fieldDistance],
73             $row[$this->fieldCountry], $row[$this->fieldStatus],
FWaypointManager::GetInstance()->GetAllById($row[$this->fieldId]),
FCommentManager::GetInstance()->GetAllById($row[$this->fieldId]),
74             FPhotoManager::GetInstance()->GetAllById($row[$this->
fieldId]), $row[$this->fieldUser]);
75             array_push($result, $itinerary);
76         }
77     }catch(PDOException $e){
78         return FALSE;
79     }
80     return count($result) > 0 ? $result : FALSE;
81 }
82 /**
83  * @brief Get all itinerary with a filter
84  *
85  * @param array array with all filters to apply
86  *
87  * @return array Array of Fltinereary with all itinerary filtered
88  */
89     public function GetAllWithFilter(array $filters){
90         $filters += [
91             'country' => null,
92             'rateMin' => null,
93             'durationMin' => null,
94             'durationMax' => null,
95             'distanceMin' => null,
96             'distanceMax' => null
97         ];
98         //Extract the key of the array has variables
99         extract($filters);
100
101         $result = array();

```

```

102
103     $filtersValue = [];
104     $filtersParam = [];
105
106     $firstCondition = false;
107     $query = <<<EX
108     SELECT '{$this->fieldId}', '{$this->fieldTitle}', '{$this
->fieldRating}', '{$this->fieldDescription}', '{$this->fieldDuration
}' , '{$this->fieldDistance}', '{$this->fieldCountry}', '{$this->
fieldStatus}', '{$this->fieldUser}'
109         FROM '{$this->tableName}'
110         WHERE '{$this->fieldStatus}' = 2
111     EX;
112     if($country !== null){
113         $query .= <<<EX
114         AND '{$this->fieldCountry}' = :country
115         EX;
116         array_push($filtersParam, ":country");
117         array_push($filtersValue, $country);
118
119     }
120     if($rateMin !== null){
121         $query .= <<<EX
122         AND '{$this->fieldRating}' >= :rating
123         EX;
124
125         array_push($filtersParam, ":rating");
126         array_push($filtersValue, $rateMin);
127     }
128     if($durationMin !== null){
129         $query .= <<<EX
130         AND '{$this->fieldDuration}' >= :durationMin
131         EX;
132         array_push($filtersParam, ":durationMin");
133         array_push($filtersValue, $durationMin);
134     }
135     if($durationMax !== null){
136         $query .= <<<EX
137         AND '{$this->fieldDuration}' <= :durationMax
138         EX;
139         array_push($filtersParam, ":durationMax");
140         array_push($filtersValue, $durationMax);
141     }
142     if($distanceMin !== null){
143         $query .= <<<EX
144         AND '{$this->fieldDistance}' >= :distanceMin
145         EX;
146         array_push($filtersParam, ":distanceMin");
147         array_push($filtersValue, $distanceMin);
148     }
149     if($distanceMax !== null){
150         $query .= <<<EX
151         AND '{$this->fieldDistance}' <= :distanceMax
152         EX;
153         array_push($filtersParam, ":distanceMax");
154         array_push($filtersValue, $distanceMax);
155     }

```

```

156     $query .= <<<EX
157         ORDER BY '{$this->fieldRating}' DESC
158     EX;
159     try{
160         $req = $this::getDb()->prepare($query);
161         for($i = 0;$i<count($filtersValue);$i++){
162             $req->bindParam($filtersParam[$i], $filtersValue[$i],
PDO::PARAM_STR);
163         }
164         $req->execute();
165
166         while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
FETCH_ORI_NEXT)){
167             $itinerary = new Fltinerary($row[$this->fieldId], $row
[$this->fieldTitle], $row[$this->fieldRating], $row[$this->
fieldDescription], $row[$this->fieldDuration], $row[$this->
fieldDistance],
168                 $row[$this->fieldCountry], $row[$this->fieldStatus],
FWaypointManager::GetInstance()->GetAllById($row[$this->fieldId]),
FCommentManager::GetInstance()->GetAllById($row[$this->fieldId]),
169                 FPhotoManager::GetInstance()->GetAllById($row[$this->
fieldId]), $row[$this->fieldUser]);
170             array_push($result, $itinerary);
171         }
172     }catch(PDOException $e){
173         return FALSE;
174     }
175     return count($result) > 0 ? $result : FALSE;
176 }
177 /**
178  * @brief Get all itinerary from an user
179  *
180  * @return array Array of Fltinerary with all itinerary
181  */
182 public function GetAllByUserId(int $idUser){
183     $result = array();
184
185     $query = <<<EX
186         SELECT '{$this->fieldId}', '{$this->fieldTitle}', '{$this
->fieldRating}', '{$this->fieldDescription}', '{$this->fieldDuration
}', '{$this->fieldDistance}', '{$this->fieldCountry}', '{$this->
fieldStatus}', '{$this->fieldUser}'
187         FROM '{$this->tableName}'
188         WHERE '{$this->fieldUser}' = :idUser
189         AND '{$this->fieldStatus}' = 2
190     EX;
191
192     try{
193         $req = $this::getDb()->prepare($query);
194         $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
195         $req->execute();
196
197         while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
FETCH_ORI_NEXT)){
198             $itinerary = new Fltinerary($row[$this->fieldId], $row
[$this->fieldTitle], $row[$this->fieldRating], $row[$this->
fieldDescription], $row[$this->fieldDuration], $row[$this->

```

```

fieldDistance],
199         $row[$this->fieldCountry], $row[$this->fieldStatus],
FWaypointManager::GetInstance()->GetAllById($row[$this->fieldId]),
FCommentManager::GetInstance()->GetAllById($row[$this->fieldId]),
200         FPhotoManager::GetInstance()->GetAllById($row[$this->
fieldId]), $row[$this->fieldUser]);
201         array_push($result, $itinerary);
202     }
203     }catch(PDOException $e){
204         return FALSE;
205     }
206     return count($result) > 0 ? $result : FALSE;
207 }
208 /**
209  * @brief Get a FItinerary object for the itinerary with the id
210  *
211  * @param int Itinerary's id
212  *
213  * @return FItinerary Object FItinerary
214  */
215 public function GetById(int $idItinerary){
216     $query = <<<EX
217     SELECT '{$this->fieldId}', '{$this->fieldTitle}', '{$this-
->fieldRating}', '{$this->fieldDescription}', '{$this->fieldDuration
}' , '{$this->fieldDistance}', '{$this->fieldCountry}', '{$this->
fieldStatus}', '{$this->fieldUser}'
218     FROM '{$this->tableName}'
219     WHERE '{$this->fieldId}' = :idItinerary
220     EX;
221     $itinerary = "";
222     try{
223         $req = $this::getDb()->prepare($query);
224         $req->bindParam(":idItinerary", $idItinerary, PDO::
PARAM_INT);
225         $req->execute();
226
227         while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
FETCH_ORI_NEXT)){
228             $itinerary = new FItinerary($row[$this->fieldId], $row
[$this->fieldTitle], $row[$this->fieldRating], $row[$this->
fieldDescription], $row[$this->fieldDuration], $row[$this->
fieldDistance],
229             $row[$this->fieldCountry], $row[$this->fieldStatus],
FWaypointManager::GetInstance()->GetAllById($row[$this->fieldId]),
FCommentManager::GetInstance()->GetAllById($row[$this->fieldId]),
230             FPhotoManager::GetInstance()->GetAllById($row[$this->
fieldId]), $row[$this->fieldUser]);
231         }
232     }catch(PDOException $e){
233         return FALSE;
234     }
235     return $itinerary != "" ? $itinerary : FALSE;
236 }
237 /**
238  * @brief Create new itinerary
239  *
240  * @param string $title Itinerary's title

```

```

241     * @param string $description Itinerary's description
242     * @param string $duration Itinerary's duration
243     * @param double $distance Itinerary's distance
244     * @param string $preview Itinerary's preview
245     * @param string $country Code iso2 for itinerary's started land
246     * @param array $waypoints Array<FWaypoint> array with all
itinerary's waypoints
247     * @param array $photos Array<FPhoto> array with all itinerary's
photos
248     * @param int $idUser owner's id
249     */
250     public function Create($title, $description, $duration, $distance,
    $country, $waypoints, $idUser, $photos = []){
251         $query = <<<EX
252         INSERT INTO '{$this->tableName}('{{{$this->fieldTitle}', '{$
    $this->fieldDescription}', '{$this->fieldDuration}', '{$this->
    fieldDistance}', '{$this->fieldCountry}', '{$this->fieldStatus}', '{$
    $this->fieldUser}'')
253         VALUES(:title, :description, :duration, :distance, :
    country, :status, :idUser)
254         EX;
255
256         try{
257             //Set itinerary's default status
258             $status = ITINERARY_STATUS_DEFAULT;
259
260             $this::getDb()->beginTransaction();
261             $req = $this::getDb()->prepare($query);
262             $req->bindParam(':title', $title, PDO::PARAM_STR);
263             $req->bindParam(':description', $description, PDO::
PARAM_STR);
264             $req->bindParam(':duration', $duration, PDO::PARAM_STR);
265             $req->bindParam(':distance', $distance, PDO::PARAM_STR);
266             $req->bindParam(':country', $country, PDO::PARAM_STR);
267             $req->bindParam(':status', $status, PDO::PARAM_STR);
268             $req->bindParam(':idUser', $idUser, PDO::PARAM_INT);
269             $req->execute();
270             $itineraryId = $this::getDb()->lastInsertId();
271             if(!empty($photos)){
272                 FPhotoManager::GetInstance()->Create($photos,
    $itineraryId);
273             }
274             FWaypointManager::GetInstance()->Create($waypoints,
    $itineraryId);
275
276             $this::getDb()->commit();
277         }catch(PDOException $e){
278             $this::getDb()->rollBack();
279             return FALSE;
280         }
281         return TRUE;
282     }
283     /**
284     * @brief Update itinerary's details
285     *
286     * @param int $idUser owner's id
287     * @param int $idItinerary itinerary's id

```

```

288     * @param string $title itinerary's title
289     * @param string $country itinerary's country
290     * @param string $description itinerary's description
291     * @param string $duration itinerary's duration
292     * @param string $distance itinerary's distance
293     * @param array $waypoints Array<FWaypoint> array with all
itinerary's waypoints
294     * @param array $photos Array<FPhoto> array with all itinerary's
photos
295     *
296     * @return bool true if update success, else false
297     */
298     public function Update($idUser,$idItinerary, $title, $description,
$duration, $distance, $country, $waypoints = [],$photos = []){
299         $query = <<<EX
300             UPDATE '{$this->tableName}'
301             SET '{$this->fieldTitle}' = :title, '{$this->
fieldDescription}' = :description, '{$this->fieldDuration}' = :
duration, '{$this->fieldDistance}' = :distance, '{$this->
fieldCountry}' = :country
302             WHERE '{$this->fieldId}' = :idItinerary
303             AND '{$this->fieldUser}' = :idUser
304         EX;
305
306         try{
307             $this::getDb()->beginTransaction();
308             $req = $this::getDb()->prepare($query);
309             $req->bindParam(':title', $title, PDO::PARAM_STR);
310             $req->bindParam(':description', $description, PDO::
PARAM_STR);
311             $req->bindParam(':distance', $distance, PDO::PARAM_STR);
312             $req->bindParam(':duration', $duration, PDO::PARAM_STR);
313             $req->bindParam(':country', $country, PDO::PARAM_STR);
314             $req->bindParam(':idItinerary', $idItinerary, PDO::
PARAM_INT);
315             $req->bindParam(':idUser', $idUser, PDO::PARAM_INT);
316             $req->execute();
317             if(!empty($photos)){
318                 FPhotoManager::GetInstance()->Create($photos,
$idItinerary);
319             }
320             if(!empty($waypoints)){
321                 FWaypointManager::GetInstance()->Delete($idItinerary);
322                 FWaypointManager::GetInstance()->Create($waypoints,
$idItinerary);
323             }
324
325
326             $this::getDb()->commit();
327         } catch(PDOException $e){
328             $this::getDb()->rollBack();
329             return FALSE;
330         }
331         return TRUE;
332     }
333     /**
334     * @brief Update the rating of an itinerary

```

```

335  *
336  * @param float new avg rating
337  * @param int itinerary's id
338  *
339  * @return bool true if update succes, else false
340  */
341 public function UpdateRating($newRating, $idItinerary){
342     $query = <<<EX
343         UPDATE '{$this->tableName}'
344         SET '{$this->fieldRating}' = :rating
345         WHERE '{$this->fieldId}' = :idItinerary
346     EX;
347
348     try{
349         $req = $this::getDb()->prepare($query);
350         $req->bindParam(':rating', $newRating, PDO::PARAM_STR);
351         $req->bindParam(':idItinerary', $idItinerary, PDO::
PARAM_INT);
352         $req->execute();
353     }catch(PDOException $e){
354         return FALSE;
355     }
356     return TRUE;
357
358 }
359 /**
360  * @brief Get the itinerary by his title (title is unique)
361  *
362  * @param string $title title of the itinerary
363  *
364  * @return FItninerary The itinerary
365  */
366 public function GetByTitle($title){
367     $query = <<<EX
368         SELECT '{$this->fieldId}', '{$this->fieldTitle}', '{$this
->fieldRating}', '{$this->fieldDescription}', '{$this->fieldDuration
}' , '{$this->fieldDistance}', '{$this->fieldCountry}', '{$this->
fieldStatus}', '{$this->fieldUser}'
369         FROM '{$this->tableName}'
370         WHERE '{$this->fieldTitle}' = :title
371     EX;
372     $itinerary = "";
373     try{
374         $req = $this::getDb()->prepare($query);
375         $req->bindParam(":title", $title, PDO::PARAM_STR);
376         $req->execute();
377
378         while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
FETCH_ORI_NEXT)){
379             $itinerary = new FItninerary($row[$this->fieldId], $row
[$this->fieldTitle], $row[$this->fieldRating], $row[$this->
fieldDescription], $row[$this->fieldDuration], $row[$this->
fieldDistance],
380             $row[$this->fieldCountry], $row[$this->fieldStatus],
FWaypointManager::GetInstance()->GetAllById($row[$this->fieldId]),
FCommentManager::GetInstance()->GetAllById($row[$this->fieldId]),
381             FPhotoManager::GetInstance()->GetAllById($row[$this->

```



```

fieldId]), $row[$this->fieldUser]);
382     }
383     }catch(PDOException $e){
384         return FALSE;
385     }
386     return $itinerary != "" ? $itinerary : FALSE;
387 }
388 /**
389  * @brief Disable itinerary
390  *
391  * @param int $idItinerary itinerary's id
392  * @param string $title itinerary's title
393  *
394  * @return bool true if disable success, else false
395  */
396 public function Disable($idItinerary, $title){
397     $query = <<<EX
398         UPDATE '{$this->tableName}'
399         SET '{$this->fieldStatus}' = 3
400         WHERE '{$this->fieldId}' = :idItinerary
401     EX;
402     try{
403         $req = $this::getDb()->prepare($query);
404         $req->bindParam(':idItinerary', $idItinerary, PDO::
PARAM_INT);
405         $req->execute();
406
407         FMailerManager::sendMail("Your itinerary has been disabled
", array(FUserManager::GetInstance()->GetById($this::GetById(
$idItinerary)->User)->Email), FMailerManager::
getDisableItineraryMail($title));
408     } catch(PDOException $e){
409         return FALSE;
410     }
411     return TRUE;
412 }
413 /**
414  * @brief Delete itinerary
415  *
416  * @param int $idItinerary itinerary's id
417  *
418  * @return bool true if delete success, else false
419  */
420 public function Delete($idItinerary){
421     $query = <<<EX
422         DELETE FROM '{$this->tableName}'
423         WHERE '{$this->fieldId}' = :idItinerary
424     EX;
425     try{
426         $req = $this::getDb()->prepare($query);
427         $req->bindParam(':idItinerary', $idItinerary, PDO::
PARAM_INT);
428         $req->execute();
429
430     } catch(PDOException $e){
431         return FALSE;
432     }

```

```

433         return TRUE;
434     }
435 }
436 ?>

```

Listing 45: FIineraryManager.php

```

1  <?php
2  /*
3  *      Author           :  Fujise Thomas.
4  *      Project          :  ProjectTPI.
5  *      Page             :  FMailerManager.
6  *      Brief            :  mailer's manager.
7  *      Date             :  20.05.2020.
8  */
9  require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/
    swiftmailer5/lib/swift_required.php';
10 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/config/
    config.php';
11 //Class mailer, contains all function for the mailer
12 class FMailerManager{
13
14     /**
15      * @var static $instance the instance for the manager
16      * */
17     private static $instance;
18
19     private function __construct(){}
20     private function __clone(){}
21     /**
22      * @brief Create the initial instance for the manager or get it
23      * @return $instance
24      */
25     public static function getInstance(){
26         if(!self::$instance){
27             self::$instance = Swift_SmtpTransport::newInstance(
28 EMAIL_SERVER, EMAIL_PORT, EMAIL_TRANS)
29                 ->setUsername(EMAIL_USERNAME)
30                 ->setPassword(EMAIL_PASSWORD);
31         }
32         return self::$instance;
33     }
34     /**
35      * @brief Send mail function
36      *
37      * @param string $subject mail subject
38      * @param string $userEmail user email
39      * @param string $body html body for the mail
40      *
41      * @return bool true if the mail's send, else false
42      */
43     public static function sendMail(string $subject, array $userEmail,
44 string $body) : bool{
45
46         try {
47             $instance = self::getInstance();
48             $mailer = Swift_Mailer::newInstance($instance);
49             $message = Swift_Message::newInstance();

```

```

48         $message->setSubject($subject);
49         $message->setFrom(array('travler.infos@gmail.com' => '
Travler Infos'));
50         $message->setTo($userEmail);
51
52         $message->setBody($body,'text/html');
53         $result = $mailer->send($message);
54         return true;
55     } catch (Swift_TransportException $e) {
56         echo "Problème d'envoi de message: ".$e->getMessage();
57         return false;
58     }
59
60 }
61 /**
62  * @brief Get the activation mail body
63  *
64  * @param string $token the token activation
65  *
66  * @return string activation mail body
67  */
68 public static function getActivationMail(string $token, string
$userEmail) : string{
69     $body = <<<EOT
70     <html>
71         <head></head>
72         <body>
73             <p>Hello,</p>
74             <br>
75             <p>Thank's for the interest you had for our website,
please click on the following link to finalize your account
activation : <a href="localhost/ProjectTPI/src/www/verified.php?
token={$token}&email={$userEmail}">Activate</a></p>
76             <br>
77             <p>Sincerely,</p>
78             <p>Team Travler</p>
79         </body>
80     </html>
81     EOT;
82     return $body;
83 }
84 /**
85  * @brief Get the mail body for owner when a user comment his
itinerary
86  *
87  * @param string $comment the comment the user posted
88  * @param string $nicknameUser user's nickname of the user who
posted the comment
89  * @param string $itineraryTitle itinerary's title
90  *
91  * @return string mail body
92  */
93 public static function getCommentAddMail(string $comment, string
$nicknameUser, string $itineraryTitle) : string{
94     $body = <<<EOT
95     <html>
96         <head></head>

```

```

97         <body>
98             <p>Hello,</p>
99             </br>
100             <p>The user <b>{$nicknameUser}</b> comment your
itinerary <b>{$itineraryTitle}</b> with the following comment : <i
>{$comment}</i>.</p>
101             <p>Feel free to check it by yourself on your itinerary
page</p>
102             </br>
103             <p>Sincerely,</p>
104             <p>Team Travler</p>
105         </body>
106     </html>
107     EOT;
108     return $body;
109 }
110
111 /**
112  * @brief Get the mail body for user when his account is blocked
by an admin
113  *
114  *
115  * @return string mail body
116  */
117 public static function getDisableAccountMail() : string{
118     $body = <<<EOT
119     <html>
120         <head></head>
121         <body>
122             <p>Hello,</p>
123             </br>
124             <p>Your account has been suspend by an administrator
for an indefinite period.</p>
125             <p>From now on you can't login with your account on
our website</p>
126             </br>
127             <p>Sincerely,</p>
128             <p>Team Travler</p>
129         </body>
130     </html>
131     EOT;
132     return $body;
133 }
134
135 /**
136  * @brief Get the mail body for user when his itinerary is blocked
by an admin
137  *
138  * @param string $title the title of the itinerary
139  *
140  * @return string mail body
141  */
142 public static function getDisableItineraryMail($title) : string{
143     $body = <<<EOT
144     <html>
145         <head></head>
146         <body>

```

```

147         <p>Hello,</p>
148         </br>
149         <p>Your itinerary : {$title} has been suspend by an
administrator for an indefinite period.</p>
150         <p>From now on your itinerary cant be seen on our
website</p>
151         </br>
152         <p>Sincerely,</p>
153         <p>Team Travler</p>
154     </body>
155 </html>
156 EOT;
157 return $body;
158 }
159
160
161 }
162
163 ?>

```

Listing 46: FMailerManager.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  FPhotoManager.
6 *      Brief            :  photo manager.
7 *      Date             :  04.09.2019.
8 */
9
10 //Requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
manager/FDatabaseManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/model/
FPhoto.php';
13
14 /**
15 * Photo's manager, manager for photos table in db
16 */
17 class FPhotoManager extends FDatabaseManager{
18
19     /**
20      * @var static $instance the instance for the manager
21      * */
22     private static $instance;
23     /**
24      * @brief Class constructor, init all field from table 'PHOTOS'
25      */
26     function __construct() {
27         $this->tableName = "PHOTOS";
28         $this->fieldId = "ID";
29         $this->fieldImage = "IMAGE";
30         $this->fieldItinerary = "ITINERARY_ID";
31     }
32     /**
33      * @brief Create the initial instance for the manager or get it
34      * @return $instance

```

```

35     */
36     public static function GetInstance(){
37         if(!self::$instance){
38             self::$instance = new FPhotoManager();
39         }
40         return self::$instance;
41     }
42     /**
43      * @brief Get all photo from an itinerary with his unique's id
44      *
45      * @param int $itineraryId Itinerary's unique id
46      *
47      * @return array Array with all photos
48      */
49     public function GetAllById(int $itineraryId) {
50         $result = array();
51
52         $query = <<<EX
53             SELECT '{$this->fieldImage}'
54             FROM '{$this->tableName}'
55             WHERE '{$this->fieldItinerary}' = :itineraryId
56         EX;
57
58         try{
59             $req = $this::getDb()->prepare($query);
60             $req->bindParam(':itineraryId', $itineraryId, PDO::
PARAM_INT);
61             $req->execute();
62
63             while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
FETCH_ORI_NEXT)){
64                 $photo = new FPhoto($row[$this->fieldImage]);
65                 array_push($result, $photo);
66             }
67         }catch(PDOException $e){
68             return FALSE;
69         }
70         return count($result) > 0 ? $result : FALSE;
71     }
72     /**
73      * @brief Create photos for itinerary
74      *
75      * @param array $photos array with the images encoded in base 64
76      * @param int itinerary's id
77      *
78      * @return bool true if the creation success, else false
79      */
80     public function Create(array $photos, int $itineraryId) {
81         $query = <<<EX
82             INSERT INTO '{$this->tableName}' ('{$this->fieldImage}',
 '{$this->fieldItinerary}')
83             VALUES(:image, :idItinerary)
84         EX;
85
86         try{
87             $req = $this::getDb()->prepare($query);
88             for($i = 0;$i<count($photos);$i++)

```

```

89         {
90             $req->bindParam(':image', $photos[$i], PDO::PARAM_STR)
91             ;
92             $req->bindParam(':idItinerary', $itineraryId, PDO::
PARAM_INT);
93             $req->execute();
94         }
95     }catch(PDOException $e){
96         return FALSE;
97     }
98     //Done
99     return TRUE;
100 }
101
102 }
103 ?>

```

Listing 47: FPhotoManager.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  FRateManager.
6 *      Brief            :  rate manager.
7 *      Date             :  05.06.2020.
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/config/
config.php';
10 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
manager/FDatabaseManager.php';
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
manager/FItineraryManager.php';
12 /**
13 * Class FRateManager, manager for table rating in db
14 */
15 class FRateManager extends FDatabaseManager{
16     /**
17     * @var static $instance the instance for the manager
18     */
19     private static $instance;
20     /**
21     * @brief Class constructor, init all field from table 'RATING'
22     */
23     function __construct(){
24         $this->tableName = "RATING";
25         $this->fieldRate = "RATE";
26         $this->fieldItinerary = "ITINERARY_ID";
27         $this->fieldUser = "USERS_ID";
28     }
29     /**
30     * @brief Create the initial instance for the manager or get it
31     * @return $instance
32     */
33     public static function GetInstance(){
34         if(!self::$instance){
35             self::$instance = new FRateManager();

```

```

36     }
37     return self::$instance;
38 }
39 /**
40  * @brief Check if user has already rate an itinerary
41  *
42  * @param int $idItinerary itinerary's id
43  * @param int $idUser user's id
44  *
45  * @return bool true if he already rate, else false
46  */
47 public function HasAlreadyRate($idItinerary, $idUser){
48     $query = <<<EX
49         SELECT '{$this->fieldRate}'
50         FROM '{$this->tableName}'
51         WHERE '{$this->fieldItinerary}' = :idItinerary
52         AND '{$this->fieldUser}' = :idUser
53     EX;
54
55     try {
56         $req = $this::getDb()->prepare($query);
57         $req->bindParam(":idItinerary", $idItinerary, PDO::
PARAM_INT);
58         $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
59         $req->execute();
60
61         if($req->rowCount() > 0){
62             return TRUE;
63         }
64         else{
65             return FALSE;
66         }
67     } catch(PDOException $e) {
68         return FALSE;
69     }
70 }
71 /**
72  * @brief Add a rate to an itinerary
73  *
74  * @param int $idItinerary itinerary's id
75  * @param int $idUser user's id
76  * @param string $rate the rate the user enter
77  *
78  * @return bool true if insert success, else false
79  */
80 public function AddToItinerary(int $idItinerary, int $idUser,
string $rate){
81     $query = <<<EX
82         INSERT INTO '{$this->tableName}' ('{$this->fieldRate}', '{$
$this->fieldItinerary}', '{$this->fieldUser}')
83         VALUES (:rate, :idItinerary, :idUser)
84     EX;
85     try{
86         $req = $this::getDb()->prepare($query);
87         $req->bindParam(':rate', $rate, PDO::PARAM_INT);
88         $req->bindParam(':idItinerary', $idItinerary, PDO::
PARAM_INT);

```



```

89         $req->bindParam(':idUser', $idUser, PDO::PARAM_INT);
90         $req->execute();
91
92         FItineraryManager::GetInstance()->UpdateRating(str_replace
93 ('',',', ','), $this::GetAvgByItinerary($idItinerary),$idItinerary);
94     }catch(PDOException $e){
95         return FALSE;
96     }
97     //Done
98     return TRUE;
99 }
100 /**
101  * @brief Get the average rate of an itinerary
102  *
103  * @param int $idItinerary itinerary's id
104  *
105  * @return double the average
106  */
107 public function GetAvgByItinerary(int $idItinerary){
108     $query = <<<EX
109     SELECT AVG('{ $this->fieldName }')
110     FROM '{ $this->tableName }'
111     WHERE '{ $this->fieldItinerary }' = :idItinerary
112     EX;
113
114     try{
115         $req = $this::getDb()->prepare($query);
116         $req->bindParam(':idItinerary', $idItinerary, PDO::
117 PARAM_INT);
118         $req->execute();
119
120         $result = number_format($req->fetch(PDO::FETCH_ASSOC)['AVG
121 (''. $this->fieldName. ''')'],2,',','');
122     }catch(PDOException $e){
123         return FALSE;
124     }
125     //Done
126     return $result;
127 }
128 /**
129  * @brief Get the number of rate on an itinerary
130  *
131  * @param int $idItinerary itinerary's id
132  *
133  * @return int the number
134  */
135 public function GetNbByItinerary(int $idItinerary){
136     $query = <<<EX
137     SELECT '{ $this->fieldName }'
138     FROM '{ $this->tableName }'
139     WHERE '{ $this->fieldItinerary }' = :idItinerary
140     EX;
141
142     try{
143         $req = $this::getDb()->prepare($query);
144         $req->bindParam(':idItinerary', $idItinerary, PDO::
145 PARAM_INT);

```

```

142         $req->execute();
143
144         $result = $req->rowCount();
145     }catch(PDOException $e){
146         return FALSE;
147     }
148     //Done
149     return $result;
150 }
151 }
152 ?>

```

Listing 48: FRateManager.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  FSessionManager.
6 *      Brief            :  session manager.
7 *      Date             :  21.05.2020.
8 */
9 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/model/
    FUser.php';
10 /**
11 * Class FSessionManager, to manage session
12 */
13 class FSessionManager {
14     /**
15      * @brief Class constructor
16      */
17     function __construct(){}
18     /**
19      * @brief Set the user logged in session
20      *
21      * @param TUser $user the user logged
22      *
23      * @return bool true if the user is set, else false;
24      */
25     public static function SetUserLogged(FUser $user) : bool{
26         if($user !== null)
27         {
28             $_SESSION['userLogged'] = $user;
29             return true;
30         }
31         else{
32             return false;
33         }
34     }
35     /**
36      * @brief Get the user logged in session or if user isn't logged,
37      * get a FUser object for unregistered user
38      *
39      * @return TUser $_SESSION['userLogged'] the user
40      */
41     public static function GetUserLogged() : ?FUser{
42         if(isset($_SESSION['userLogged'])){
43             return $_SESSION['userLogged'];

```

```

43         }else{
44             //Id 0 for visitor account
45             return new FUser(0, "John@Doe.com", "Visitor", "John", "
Doe", "", "", "", "2", "1");
46         }
47
48     }
49     /**
50     * @brief Clear session
51     *
52     * @return bool true if session clear success, else false
53     */
54     public static function Reset() : bool{
55         if(ini_get("session.use_cookies")){
56             setcookie(session_name(), '', 0);
57         }
58         session_destroy();
59
60         return session_destroy() ? true : false;
61     }
62     /**
63     * @brief Set a filter for itinerary display
64     *
65     * @param array All the filter to apply
66     *
67     * @return bool true if filter's set, else false
68     */
69     public static function SetItineraryFilter(array $filters){
70         if($filters != null){
71             $_SESSION['filters'] = $filters;
72             return TRUE;
73         }
74         else{
75             return FALSE;
76         }
77     }
78     /**
79     * @brief Get the filter for itinerary display
80     *
81     * @return array All the filter to apply
82     */
83     public static function GetItineraryFilter(){
84         if(isset($_SESSION['filters'])){
85             return $_SESSION['filters'];
86         }else{
87             return FALSE;
88         }
89     }
90     /**
91     * @brief Unset the itinerary filters
92     *
93     * @return bool true if unset done, else false
94     */
95     public static function UnsetItineraryFilter(){
96         if(isset($_SESSION['filters'])){
97             unset($_SESSION['filters']);
98             return TRUE;

```

```

99         }
100         else{
101             return FALSE;
102         }
103     }
104 }
105 ?>

```

Listing 49: FSessionManager.php

```

1 <?php
2 /*
3 *      Author           :  Fujise Thomas.
4 *      Project          :  ProjetTPI.
5 *      Page             :  FUserManager.
6 *      Brief            :  user manager.
7 *      Date             :  04.09.2019.
8 */
9
10 //Requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FDatabaseManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FMailerManager.php';
13 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/model/
    FUser.php';
14 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/constants/
    constants.php';
15
16 /**
17 * User's manager, manager for table users in db
18 */
19 class FUserManager extends FDatabaseManager{
20
21     /**
22      * @var static $instance the instance for the manager
23      */
24     private static $instance;
25     /**
26      * @brief Class constructor, init all field from table 'USERS'
27      */
28     function __construct() {
29         $this->tableName = "USERS";
30         $this->fieldId = "ID";
31         $this->fieldEmail = "EMAIL";
32         $this->fieldNickname = "NICKNAME";
33         $this->fieldPassword = "PASSWORD";
34         $this->fieldToken = "TOKEN";
35         $this->fieldName = "NAME";
36         $this->fieldSurname = "SURNAME";
37         $this->fieldBio = "BIO";
38         $this->fieldAvatar = "AVATAR";
39         $this->fieldStatus = "STATUS_ID";
40         $this->fieldRole = "ROLES_ID";
41         $this->fieldCountry = "COUNTRIES_ISO2";
42     }
43     /**
44      * @brief Get User if login informations are correct

```

```

45  *
46  * @param array $args Array with login informations.
47  *
48  * @return FUser||FALSE
49  */
50 public function Login(array $args) {
51     $args += [
52         'userEmail' => null,
53         'userNickname' => null,
54         'userPwd' => null
55     ];
56
57     // Extract the keys of the array has variables
58     extract($args);
59
60     $loginField = "";
61     $loginValue = "";
62     $salt = "";
63     if($userEmail !== null){
64         $loginValue = $userEmail;
65         $loginField = $this->fieldEmail;
66
67     } else if($userNickname !== null){
68         $loginValue = $userNickname;
69         $loginField = $this->fieldNickname;
70
71     } else {
72         return false;
73     }
74
75     if($userPwd === null){
76         return false;
77     }
78
79     //$pwd = hash("sha256", $userPwd . $salt);
80
81     if(password_verify($userPwd, $this::GetHashPassword(
82 $loginValue))){
83
84         $query = <<<EX
85             SELECT '{$this->fieldId}', '{$this->fieldEmail}', '{$this-
86 ->fieldNickname}', '{$this->fieldName}', '{$this->fieldSurname}', '{$
87 $this->fieldBio}', '{$this->fieldAvatar}', '{$this->fieldCountry}',
88             '{$this->fieldStatus}', '{$this->fieldRole}'
89             FROM '{$this->tableName}'
90             WHERE '{$loginField}' = :connectValue
91         EX;
92
93         try {
94             $req = $this::getDb()->prepare($query);
95             $req->bindParam(':connectValue', $loginValue, PDO::
96 PARAM_STR);
97             $req->execute();
98
99             $result = $req->fetch(PDO::FETCH_ASSOC);
100
101             return $result !== false & > 0 ? new FUser($result[$this->

```

```

fieldId], $result[$this->fieldEmail], $result[$this->fieldNickname
], $result[$this->fieldName], $result[$this->fieldSurname], $result
[$this->fieldBio], $result[$this->fieldAvatar], $result[$this->
fieldCountry], $result[$this->fieldStatus], $result[$this->
fieldRole]) : FALSE;
97     } catch(PDOException $e) {
98         return FALSE;
99     }
100 }
101 else{
102     return FALSE;
103 }
104 }
105 /**
106  * @brief Register new user
107  *
108  * @param string $userEmail Email
109  * @param string $userNickname Nickname
110  * @param string $userPwd Password
111  * @param string $userCountry Country
112  * @param string $userBirthday Birthday
113  *
114  * @return bool register state
115  */
116 public function Register(string $userEmail, string $userNickname,
string $userPwd) : bool {
117     $query = <<<EX
118         INSERT INTO '{$this->tableName}'('{$this->fieldNickname}',
'{$this->fieldEmail}', '{$this->fieldPassword}', '{$this->
fieldAvatar}', '{$this->fieldStatus}', '{$this->fieldRole}', '{
$this->fieldToken}')
119         VALUES(:userNickname, :userEmail, :userPassword, :
userAvatar, :userStatus, :userRole, :token)
120     EX;
121
122     $userPwd = password_hash($userPwd, PASSWORD_DEFAULT);
123
124     $token = hash('sha256', $userEmail . microtime());
125     //account not activate
126     $userStatus = 1;
127     //role 1 = user
128     $userRole = 1;
129
130     $defaultAvatar = DEFAULT_USER_LOGO;
131
132     try {
133         $req = $this::getDb()->prepare($query);
134         $req->bindParam(':userNickname', $userNickname, PDO::
PARAM_STR);
135         $req->bindParam(':userEmail', $userEmail, PDO::PARAM_STR);
136         $req->bindParam(':userPassword', $userPwd, PDO::PARAM_STR)
;
137         $req->bindParam(':userAvatar', $defaultAvatar, PDO::
PARAM_STR);
138         $req->bindParam(':userStatus', $userStatus, PDO::PARAM_INT
);
139         $req->bindParam(':userRole', $userRole, PDO::PARAM_INT);

```

```

140         $req->bindParam(':token', $token, \PDO::PARAM_STR);
141         $req->execute();
142
143         FMailerManager::sendMail("Account Activation", array(
144             $userEmail), FMailerManager::getActivationMail($token,$userEmail));
145
146         } catch(PDOException $e){
147             return FALSE;
148         }
149         //Done
150         return TRUE;
151     }
152     /**
153     * @brief Get hash password
154     *
155     * @param int $idUser user's id
156     *
157     * @return string hash password
158     */
159     public function GetHashPassword(string $loginValue) : string {
160         $query = <<<EX
161             SELECT '{$this->fieldPassword}'
162             FROM '{$this->tableName}'
163             WHERE EMAIL = :userEmail
164             OR NICKNAME = :userNickname;
165         EX;
166         try{
167             $req = $this::getDb()->prepare($query);
168             $req->bindParam(':userEmail', $loginValue, PDO::PARAM_STR);
169             $req->bindParam(':userNickname', $loginValue, PDO::PARAM_STR);
170             $req->execute();
171             $result = $req->fetch(PDO::FETCH_ASSOC);
172             }catch(PDOException $e){
173                 return FALSE;
174             }
175             //Done
176             return $result ==! FALSE ? $result['PASSWORD'] : FALSE;
177         }
178     /**
179     * @brief Check if user is admin or not
180     *
181     * @param TUser $user User to check
182     *
183     * @return bool true if admin, false if user
184     */
185     public function IsAllowed(FUser $user) : bool {
186         return $user->Role == 2 ? TRUE : FALSE;
187     }
188     /**
189     * @brief Check if user is logged or not
190     *
191     * @return bool true if user's logged, else false
192     */
193     public function IsLogged() : bool {
194         return isset($_SESSION['userLogged']) ? TRUE : FALSE;
195     }
196     /**

```

```

196     * @brief Create the initial instance for the controller or get it
197     * @return $instance
198     */
199     public static function GetInstance(){
200         if(!self::$instance){
201             self::$instance = new FUserManager();
202         }
203         return self::$instance;
204     }
205     /**
206     * @brief Activate account with the token
207     *
208     * @param string $token activation token
209     *
210     * @return bool true if account's activation success, else false
211     */
212     public function ActivateAccount(string $token, string $userEmail)
213     : bool{
214         try{
215             $query = <<<EX
216                 UPDATE '{$this->tableName}'
217                 SET '{$this->fieldStatus}' = 2
218                 WHERE '{$this->fieldToken}' = :token
219                 AND '{$this->fieldEmail}' = :userEmail
220             EX;
221             $req = $this::getDb()->prepare($query);
222             $req->bindParam(':token', $token, \PDO::PARAM_STR);
223             $req->bindParam(':userEmail', $userEmail, \PDO::PARAM_STR)
224             ;
225             $req->execute();
226
227             } catch(PDOException $e){
228                 return FALSE;
229             }
230             //Done
231             return TRUE;
232         }
233     /**
234     * @brief Verify if account is activated
235     *
236     * @param string $nickname user's nickname
237     *
238     * @return bool true if account's activated, else false
239     */
240     public function VerifyActivation(string $nickname){
241         $query = <<<EX
242             SELECT '{$this->fieldStatus}'
243             FROM '{$this->tableName}'
244             WHERE '{$this->fieldNickname}' = :nickname
245         EX;
246         try{
247             $req = $this::getDb()->prepare($query);
248             $req->bindParam(':nickname', $nickname, PDO::PARAM_STR);
249             $req->execute();
250
251             $result = $req->fetch();
252             //return status code

```



```

251         return $result[0];
252     }catch(PDOException $e){
253         return FALSE;
254     }
255 }
256 /**
257  * @brief Get user by id
258  *
259  * @param int $idUser user's unique id
260  *
261  * @return FUser The user with the id
262  */
263 public function GetById(int $idUser){
264     $query = <<<EX
265     SELECT '{$this->fieldId}', '{$this->fieldEmail}', '{$this->fieldNickname}', '{$this->fieldName}', '{$this->fieldSurname}', '{$this->fieldBio}', '{$this->fieldAvatar}', '{$this->fieldCountry}', '{$this->fieldStatus}', '{$this->fieldRole}'
266     FROM '{$this->tableName}'
267     WHERE '{$this->fieldId}' = :idUser
268     EX;
269
270     try{
271         $req = $this::getDb()->prepare($query);
272         $req->bindParam(':idUser', $idUser, PDO::PARAM_INT);
273         $req->execute();
274
275         $result = $req->fetch(PDO::FETCH_ASSOC);
276         return $result != false & $result[0] ? new FUser($result[$this->fieldId], $result[$this->fieldEmail], $result[$this->fieldNickname], $result[$this->fieldName], $result[$this->fieldSurname], $result[$this->fieldBio], $result[$this->fieldAvatar], $result[$this->fieldCountry], $result[$this->fieldStatus], $result[$this->fieldRole]) : FALSE;
277     }catch(PDOException $e){
278         return FALSE;
279     }
280
281 }
282 /**
283  * @brief Get user by nickname
284  *
285  * @param string $nickname user's unique nickname
286  *
287  * @return FUser The user with the nickname
288  */
289 public function GetByNickname(string $nickname){
290     $query = <<<EX
291     SELECT '{$this->fieldId}', '{$this->fieldEmail}', '{$this->fieldNickname}', '{$this->fieldName}', '{$this->fieldSurname}', '{$this->fieldBio}', '{$this->fieldAvatar}', '{$this->fieldCountry}', '{$this->fieldStatus}', '{$this->fieldRole}'
292     FROM '{$this->tableName}'
293     WHERE '{$this->fieldNickname}' = :nickname
294     EX;
295
296     try{

```

```

297         $req = $this::getDb()->prepare($query);
298         $req->bindParam(':nickname', $nickname, PDO::PARAM_STR);
299         $req->execute();
300
301         $result = $req->fetch(PDO::FETCH_ASSOC);
302         return $result != false > 0 ? new FUser($result[$this->
fieldId], $result[$this->fieldEmail], $result[$this->fieldNickname
], $result[$this->fieldName], $result[$this->fieldSurname], $result
[$this->fieldBio], $result[$this->fieldAvatar], $result[$this->
fieldCountry], $result[$this->fieldStatus], $result[$this->
fieldRole]) : FALSE;
303     }catch(PDOException $e){
304         return FALSE;
305     }
306
307 }
308 /**
309  * @brief Get all users
310  *
311  * @return array Array of FUser with all users
312  */
313 public function GetAll(){
314     $result = array();
315
316     $query = <<<EX
317     SELECT '{$this->fieldId}', '{$this->fieldEmail}', '{$this-
->fieldNickname}', '{$this->fieldName}', '{$this->fieldSurname}', '{$
$this->fieldBio}', '{$this->fieldAvatar}', '{$this->fieldCountry}',
    '{$this->fieldStatus}', '{$this->fieldRole}'
318     FROM '{$this->tableName}'
319     EX;
320
321     try{
322         $req = $this::getDb()->prepare($query);
323         $req->execute();
324
325         while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
FETCH_ORI_NEXT)){
326             $users = new FUser($row[$this->fieldId], $row[$this->
fieldEmail], $row[$this->fieldNickname], $row[$this->fieldName],
            $row[$this->fieldSurname], $row[$this->fieldBio], $row[$this->
fieldAvatar], $row[$this->fieldCountry], $row[$this->fieldStatus],
            $row[$this->fieldRole]);
327             array_push($result, $users);
328         }
329     }catch(PDOException $e){
330         return FALSE;
331     }
332     return count($result) > 0 ? $result : FALSE;
333 }
334 /**
335  * @brief Update user's infos
336  *
337  * @param int $idUser user's id
338  * @param string $nickname user's nickname
339  * @param string $name user's name
340  * @param string $surname user's surname

```

```

341     * @param string $bio user's bio
342     * @param string $country user's residence country code
343     */
344     public function UpdateInfos($idUser, $nickname, $name, $surname,
345     $bio, $country, $avatar){
346         $query = <<<EX
347             UPDATE '{$this->tableName}'
348             SET '{$this->fieldName}' = :nickname, '{$this->
349     fieldName}' = :name, '{$this->fieldSurname}' = :surname, '{$this->
350     fieldBio}' = :bio, '{$this->fieldCountry}' = :country, '{$this->
351     fieldAvatar}' = :avatar
352             WHERE '{$this->fieldId}' = :idUser
353         EX;
354
355         try{
356             if(empty($avatar)){
357                 $avatar = DEFAULT_USER_LOGO;
358             }
359             $req = $this::getDb()->prepare($query);
360             $req->bindParam(':nickname', $nickname, PDO::PARAM_STR);
361             $req->bindParam(':name', $name, PDO::PARAM_STR);
362             $req->bindParam(':surname', $surname, PDO::PARAM_STR);
363             $req->bindParam(':bio', $bio, PDO::PARAM_STR);
364             $req->bindParam(':country', $country, PDO::PARAM_STR);
365             $req->bindParam(':avatar', $avatar, PDO::PARAM_STR);
366             $req->bindParam(':idUser', $idUser, PDO::PARAM_INT);
367             $req->execute();
368         } catch(PDOException $e){
369             return FALSE;
370         }
371         return TRUE;
372     }
373     /**
374     * @brief Disable account by nickname
375     *
376     * @param string $nickname user's nickname
377     *
378     * @return bool true if disable success, else false
379     */
380     public function DisableAccount($nickname){
381         $query = <<<EX
382             UPDATE '{$this->tableName}'
383             SET '{$this->fieldStatus}' = 3
384             WHERE '{$this->fieldNickname}' = :nickname
385         EX;
386         try{
387             $req = $this::getDb()->prepare($query);
388             $req->bindParam(':nickname', $nickname, PDO::PARAM_STR);
389             $req->execute();
390
391             FMailerManager::sendMail("Your account has been disabled",
392             array($this::GetByNickname($nickname)->Email), FMailerManager::
393             getDisableAccountMail());
394         } catch(PDOException $e){
395             return FALSE;
396         }
397         return TRUE;

```

```

392     }
393
394 }
395 ?>

```

Listing 50: FUserManager.php

```

1  <?php
2  /*
3   *      Author      :   Fujise Thomas.
4   *      Project     :   ProjetTPI.
5   *      Page        :   FWaypointManager.
6   *      Brief       :   Waypoint manager.
7   *      Date        :   04.09.2019.
8   */
9
10 //Requirements
11 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/
    manager/FDatabaseManager.php';
12 require_once $_SERVER['DOCUMENT_ROOT'].'/ProjectTPI/src/www/app/model/
    FWaypoint.php';
13
14 /**
15  * Waypoints's manager, manager for table waypoints in db
16  */
17 class FWaypointManager extends FDatabaseManager{
18
19     /**
20      * @var static $instance the instance for the manager
21      */
22     private static $instance;
23     /**
24      * @brief Class constructor, init all field from table 'WAYPOINTS'
25      */
26     function __construct() {
27         $this->tableName = "WAYPOINTS";
28         $this->fieldIndex = "INDEX";
29         $this->fieldAddress = "ADDRESS";
30         $this->fieldLongitude = "LONGITUDE";
31         $this->fieldLatitude = "LATITUDE";
32         $this->fieldItinerary = "ITINERARY_ID";
33     }
34     /**
35      * @brief Create the initial instance for the manager or get it
36      * @return $instance
37      */
38     public static function GetInstance(){
39         if(!self::$instance){
40             self::$instance = new FWaypointManager();
41         }
42         return self::$instance;
43     }
44     /**
45      * @brief Get all waypoints from an itinerary with his unique's id
46      *
47      * @param int $itineraryId itinerary's unique id
48      *
49      * @return array Array with all waypoint from the itinerary

```

```

50  */
51  public function GetAllById(int $itineraryId) {
52      $result = array();
53
54      $query = <<<EX
55      SELECT '{$this->fieldIndex}', '{$this->fieldAddress}', '{$this->fieldLongitude}', '{$this->fieldLatitude}'
56      FROM '{$this->tableName}'
57      WHERE '{$this->fieldItinerary}' = :itineraryId
58      ORDER BY '{$this->fieldIndex}'
59      EX;
60
61      try{
62          $req = $this::getDb()->prepare($query);
63          $req->bindParam(':itineraryId', $itineraryId, PDO::
PARAM_INT);
64          $req->execute();
65
66          while($row=$req->fetch(PDO::FETCH_ASSOC, PDO::
FETCH_ORI_NEXT)){
67              $waypoint = new FWaypoint($row[$this->fieldIndex],
$row[$this->fieldAddress], $row[$this->fieldLongitude], $row[$this->
fieldLatitude]);
68              array_push($result, $waypoint);
69          }
70          }catch(PDOException $e){
71              return FALSE;
72          }
73          return count($result) > 0 ? $result : FALSE;
74      }
75  /**
76   * @brief Create new waypoints for itinerary
77   *
78   * @param array Array<FWaypoint> with all waypoints to add
79   * @param int $idItinerary itinerary's id
80   *
81   * @return bool true if insert's done else false.
82   */
83  public function Create(array $waypoints, $idItinerary){
84      $query = <<<EX
85      INSERT INTO '{$this->tableName}' ('{$this->fieldIndex}',
 '{$this->fieldAddress}', '{$this->fieldLongitude}', '{$this->
fieldLatitude}', '{$this->fieldItinerary}')
86      VALUES(:index, :address, :longitude, :latitude, :
idItinerary)
87      EX;
88
89      try{
90          $req = $this::getDb()->prepare($query);
91          for($i = 0; $i < count($waypoints); $i++)
92          {
93              $req->bindParam(':index', $waypoints[$i]->Index, PDO::
PARAM_INT);
94              $req->bindParam(':address', $waypoints[$i]->Address,
PDO::PARAM_STR);
95              $req->bindParam(':longitude', $waypoints[$i]->
Longitude, PDO::PARAM_STR);

```

```

96         $req->bindParam(':latitude', $waypoints[$i]->Latitude,
PDO::PARAM_STR);
97         $req->bindParam(':idItinerary', $idItinerary, PDO::
PARAM_INT);
98         $req->execute();
99     }
100
101     } catch(PDOException $e){
102         return FALSE;
103     }
104     //Done
105     return TRUE;
106 }
107 /**
108  * @brief Delete all waypoints from an itinerary
109  *
110  * @param int $idItinerary itinerary's id
111  *
112  * @return bool true if delete success, else false
113  */
114 public function Delete($idItinerary){
115     $query = <<<EX
116         DELETE FROM '{$this->tableName}'
117         WHERE '{$this->fieldItinerary}' = :idItinerary
118     EX;
119
120     try{
121         $req = $this::getDb()->prepare($query);
122         $req->bindParam(':idItinerary', $idItinerary, PDO::
PARAM_INT);
123         $req->execute();
124     } catch(PDOException $e){
125         return FALSE;
126     }
127     return TRUE;
128 }
129
130 }
131 ?>

```

Listing 51: FWaypointsManager.php

```

1 <?php
2 /*
3  *      Author           :   Fujise Thomas.
4  *      Project          :   ProjetTPI.
5  *      Page             :   FComment.
6  *      Brief            :   Comment's model.
7  *      Date             :   28.05.2020.
8  *      Version          :   1.0.
9  */
10
11 /**
12  * Represents an comment.
13  */
14 class FComment{
15
16     /**

```

```

17     * @brief Class constructor, create an comment with specified
18     details
19     *
20     * @param string $CommentParam comment's text
21     * @param string $PostedAtParam comment's date
22     * @param FUser $UserParam comment's user
23     */
24     public function __construct(string $CommentParam = "", string
25     $PostedAtParam = "", FUser $UserParam)
26     {
27         $this->Comment = $CommentParam;
28         $this->PostedAt = $PostedAtParam;
29         $this->User = $UserParam;
30     }
31     /**
32     * @var string comment's text
33     */
34     public $Comment;
35
36     /**
37     * @var string date and hour where the comment was posted
38     */
39     public $PostedAt;
40
41     /**
42     * @var FUser comment's owner
43     */
44     public $User;
45 }
?>

```

Listing 52: FComment.php

```

1 <?php
2 /*
3 *     Author           : Fujise Thomas.
4 *     Project          : ProjetTPI.
5 *     Page             : FCountry.
6 *     Brief            : Country's model.
7 *     Date             : 28.05.2020.
8 *     Version          : 1.0.
9 */
10
11 /**
12 * Represents an country.
13 */
14 class FCountry{
15     /**
16     * @brief Class constructor, create an country with iso2 code and
17     name
18     *
19     * @param string $NameParam country's name
20     * @param string $CodeParam country's iso2 code
21     */
22     public function __construct(string $NameParam = "", string
23     $CodeParam = "")
24     {

```

```

23     $this->Name = $NameParam;
24     $this->Code = $CodeParam;
25 }
26
27 /**
28  * @var string Country's name
29  */
30 public $Name;
31
32 /**
33  * @var string Country's iso2 code
34  */
35 public $Code;
36 }
37 ?>

```

Listing 53: FCountry.php

```

1 <?php
2 /*
3  *      Author           :  Fujise Thomas.
4  *      Project          :  ProjetTPI.
5  *      Page             :  FItinerary.
6  *      Brief            :  Itinerary's model.
7  *      Date             :  28.05.2020.
8  *      Version          :  1.0.
9  */
10
11 /**
12  * Represents an itinerary.
13  */
14 class FItinerary{
15
16     /**
17      * @brief Class constructor, create an itinerary with specified
18      details
19      *
20      * @param int $IdParam itinerary's id
21      * @param string $TitleParam itinerary's title
22      * @param string $RatingParam itinerary's average rating
23      * @param string $DescriptionParam itinerary's description
24      * @param string $DurationParam itinerary's duration
25      * @param string $DistanceParam itinerary's distance
26      * @param string $PreviewParam itinerary's preview image encoded
27      in base 64
28      * @param string $CountryParam itinerary's country
29      * @param string $StatusParam itinerary's status
30      * @param Array<FWaypoint> $WaypointsParam Array with all
31      itinerary's waypoints.
32      * @param Array<FComment> $CommentsParam Array with all itinerary's
33      comments.
34      * @param Array<FPhotos> $PhotosParam Array with all itinerary's
35      photos.
36      * @param int $UserParam Owner's id
37      */
38     public function __construct(int $IdParam , string $TitleParam = ""
39     , $RatingParam , string $DescriptionParam = "" , string
40     $DurationParam = "" , string $DistanceParam = "" , string

```



```

34 $CountryParam = "", int $StatusParam , array $WaypointsParam,
35 $CommentsParam, $PhotosParam, int $UserParam)
36 {
37     $this->Id = $IdParam;
38     $this->Title = $TitleParam;
39     $this->Rating = $RatingParam;
40     $this->Description = $DescriptionParam;
41     $this->Duration = $DurationParam;
42     $this->Distance = $DistanceParam;
43     $this->Country = $CountryParam;
44     $this->Status = $StatusParam;
45     $this->Waypoints = $WaypointsParam;
46     $this->Comments = $CommentsParam;
47     $this->Photos = $PhotosParam;
48     $this->User = $UserParam;
49 }
50
51 /**
52  * @var int Itinerary's unique id
53  */
54 public $Id;
55
56 /**
57  * @var string Itineray's title
58  */
59 public $Title;
60
61 /**
62  * @var string Itinerary's rating
63  */
64 public $Rating;
65
66 /**
67  * @var string Itinerary's description
68  */
69 public $Description;
70
71 /**
72  * @var string Itinerary's duration
73  */
74 public $Duration;
75
76 /**
77  * @var string Itinerary's distance
78  */
79 public $Distance;
80
81 /**
82  * @var string Itinerary's country
83  */
84 public $Country;
85
86 /**
87  * @var int Itinerary's status
88  */
89 public $Status;
90
91 /**
92  * @var array Array with all itinerary's waypoints

```

```

89     */
90     public $Waypoints;
91     /**
92     * @var array Array with all itinerary's comments
93     */
94     public $Comments;
95     /**
96     * @var array Array with all itinerary's photos
97     */
98     public $Photos;
99     /**
100    * @var int owner's id
101    */
102    public $User;
103 }
104 ?>

```

Listing 54: FItninerary.php

```

1 <?php
2 /*
3 *     Author           :  Fujise Thomas.
4 *     Project          :  ProjetTPI.
5 *     Page             :  FComment.
6 *     Brief            :  Comment's model.
7 *     Date             :  28.05.2020.
8 *     Version          :  1.0.
9 */
10
11 /**
12 * Represents an comment.
13 */
14 class FComment{
15
16     /**
17     * @brief Class constructor, create an comment with specified
18     details
19     *
20     * @param string $CommentParam comment's text
21     * @param string $PostedAtParam comment's date
22     * @param FUser $UserParam comment's user
23     */
24     public function __construct(string $CommentParam = "", string
25 $PostedAtParam = "", FUser $UserParam)
26     {
27         $this->Comment = $CommentParam;
28         $this->PostedAt = $PostedAtParam;
29         $this->User = $UserParam;
30     }
31     /**
32     * @var string comment's text
33     */
34     public $Comment;
35
36     /**
37     * @var string date and hour where the comment was posted
38     */
39     public $PostedAt;

```

```

38
39     /**
40      * @var FUser comment's owner
41      */
42     public $User;
43
44 }
45 ?>

```

Listing 55: FComment.php

```

1 <?php
2 /*
3      Author           :  Fujise Thomas.
4      Project          :  ProjetTPI.
5      Page             :  FComment.
6      Brief            :  Comment's model.
7      Date             :  28.05.2020.
8      Version          :  1.0.
9 */
10
11 /**
12  * Represents an comment.
13  */
14 class FComment{
15
16     /**
17      * @brief Class constructor, create an comment with specified
18      details
19      *
20      * @param string $CommentParam comment's text
21      * @param string $PostedAtParam comment's date
22      * @param FUser $UserParam comment's user
23      */
24     public function __construct(string $CommentParam = "", string
25 $PostedAtParam = "", FUser $UserParam)
26     {
27         $this->Comment = $CommentParam;
28         $this->PostedAt = $PostedAtParam;
29         $this->User = $UserParam;
30     }
31     /**
32      * @var string comment's text
33      */
34     public $Comment;
35
36     /**
37      * @var string date and hour where the comment was posted
38      */
39     public $PostedAt;
40
41     /**
42      * @var FUser comment's owner
43      */
44     public $User;
45 }

```

45 ?>

Listing 56: FComment.php

```
1 <?php
2 /*
3 *      Author          :   Fujise Thomas.
4 *      Project         :   ProjetTPI.
5 *      Page            :   FComment.
6 *      Brief           :   Comment's model.
7 *      Date            :   28.05.2020.
8 *      Version         :   1.0.
9 */
10
11 /**
12 * Represents an comment.
13 */
14 class FComment{
15
16     /**
17     * @brief Class constructor, create an comment with specified
18     details
19     *
20     * @param string $CommentParam comment's text
21     * @param string $PostedAtParam comment's date
22     * @param FUser $UserParam comment's user
23     */
24     public function __construct(string $CommentParam = "", string
25 $PostedAtParam = "", FUser $UserParam)
26     {
27         $this->Comment = $CommentParam;
28         $this->PostedAt = $PostedAtParam;
29         $this->User = $UserParam;
30     }
31     /**
32     * @var string comment's text
33     */
34     public $Comment;
35
36     /**
37     * @var string date and hour where the comment was posted
38     */
39     public $PostedAt;
40
41     /**
42     * @var FUser comment's owner
43     */
44     public $User;
45 }
```

Listing 57: FComment.php