

Table des matières

Déclaration	i
Remerciements	ii
Résumé	iii
Table des matières	iv
Liste des figures	v
1. Introduction	1
2. Historique du projet.....	2
3. L'ergonomie d'une plateforme	4
3.1 Les critères de l'ergonomie	4
3.1.1 La sobriété	4
3.1.2 La lisibilité	4
3.1.3 L'utilisabilité	4
3.1.4 La rapidité	5
3.1.5 L'interactivité	5
3.1.6 L'adaptabilité.....	5
3.1.7 L'accessibilité.....	5
3.2 Les bonnes pratiques d'ergonomie pour un moteur de recherche	6
3.2.1 Le design	6
3.2.2 Les fonctionnalités	7
3.2.3 L'affichage des résultats	8
4. Analyse et amélioration de la plateforme	10
4.1 Architecture des pages de recherche	10
4.2 Affichage des résultats de recherche	15
4.3 Processus de login/récupération de mot de passe	18
4.4 Corrections de problèmes d'affichages et de fonctionnement	20
4.4.1 Inversion d'informations sur la page « AnnéeScolaire » en vue mobile ..	20
4.4.2 Problème sur la page « vacances » en vue mobile	21
4.4.3 Modifications des conditions dans la requête d'affichage des activités..	22
4.4.4 Correction de la requête de recherche des activités vacances	22
4.5 Suppression des éléments de l'Etat de Genève	23
4.6 Suppression de l'image « Résultat de la recherche »	24
4.7 Changement des curseurs	24
4.8 Affichage des détails des favoris en vue mobile	26
4.9 Correction de l'impression des favoris	26
4.10 Mise en place d'un tableau de bord pour l'administrateur	27
5. Future amélioration de la plateforme.....	29
5.1 L'état actuel de la plateforme	29
5.2 Des futures fonctionnalités ?.....	29
6. Conclusion	30
Bibliographie.....	31

Liste des figures

Figure 1 - Illustration de l'avancée du projet.....	3
Figure 2 - Importance de la taille des champs de saisies	6
Figure 3 - Exemple de design avec le formulaire de recherche de YouTube	7
Figure 4 – Exemple d'autosuggestion de Google	7
Figure 5 - Exemple de recherche avancée sur la recherche d'image de Google.....	8
Figure 6 - Affichage du nombre total de résultats obtenus par une recherche sur Bing	9
Figure 7 - Contenu de la page avec les deux vues desktop et mobile	10
Figure 8 - Code effectuant l'affichage des informations	11
Figure 9 - Contenu de la page pour les filtres de recherche	11
Figure 10 - Affichage des informations pour une seule des vues	13
Figure 11 - Identification des div à adapter en fonction de la vue	14
Figure 12 - Code qui adapte l'affichage du filtre de recherche en fonction de la largeur de l'écran	14
Figure 13 - Exemple de résultats d'une recherche d'activités par un utilisateur	15
Figure 14 - Vue HTML des éléments qui permettent de naviguer entre les pages	16
Figure 15 - Code qui détermine le nombre total de pages	17
Figure 16 - Code permettant l'affichage paginé des informations	17
Figure 17 - Exemple de message d'erreur incrusté dans la page	18
Figure 18 - Code qui permet le fonctionnement de la fenêtre modale	20
Figure 19 - Inversion des champs au niveau de la vue mobile	20
Figure 20 - Nombre de places inconnues en vue mobile pour les activités de vacances	21
Figure 21 - Erreur sur l'attribut pour la gestion des favoris.....	21
Figure 22 - Modification des conditions de la requête récupérant les activités	22
Figure 23 - Récupération de l'identifiant de l'association	23
Figure 24 - Problème d'ergonomie lors de l'affichage des résultats.....	24
Figure 25 - Code HTML de la liste déroulante des âges	25
Figure 26 - Gestion de l'affichage des détails des activités en favori sur mobile	26
Figure 27 - Evènement de gestion de l'impression	27
Figure 28 - Appel du script d'envoi de mail aux organismes	28

1. Introduction

Le travail de Bachelor est le projet qui conclut ma formation à la Haute Ecole de Gestion. Après toutes ces années, j'ai décidé d'orienter ce travail dans le domaine du Web, qui est celui qui me plaît le plus et où je prends le plus de plaisir à travailler.

Le sujet de ce travail m'a été proposé par mon professeur de mémoire. Je voulais me retrouver dans des conditions de travail que l'on peut rencontrer dans le monde professionnel lorsque l'on se voit attribuer un projet. Le fait de partir d'un existant ou de rien ne m'importait que peu. Je recherchais un projet qui soit utile et qui ne serait pas mis de côté une fois ce mémoire terminé.

Il m'a donc été proposé de reprendre le projet « loisirsjeunes.ch » qui est une plateforme web qui répertorie une grande quantité d'activités pour les jeunes sur Genève. L'état compte à ce jour plus de 700 organismes proposant des activités. Un grand nombre de personnes ont travaillé sur le projet sans réellement avoir abouti à un résultat voulu. Le but est donc de rendre la plateforme la plus opérationnelle et ergonomique possible. De plus, il y avait pour objectif de mettre en production cette plateforme d'ici la rentrée scolaire 2017.

La conception de la plateforme a été développée en HTML, PHP, jQuery, CSS et la base de données est en MySQL. Toutes ces technologies m'ont été enseignées au cours de ma formation à la HEG, ce qui va me permettre de prendre en main le projet plus facilement du côté des technologies utilisées. En ce qui concerne les documents fournis, la prise en main a été facilitée, car tous les fichiers ont bien été séparés. Il était donc facile pour moi de me repérer et de trouver les informations nécessaires lors du développement.

Dans ce mémoire, j'ai donc décidé d'aborder la thématique de l'ergonomie au niveau d'une plateforme web. Dans un premier temps, je vais parler de l'historique du projet afin de comprendre ce qui s'est passé pour en être arrivé là. Ensuite, il sera question de définir ce qu'est l'ergonomie et les critères qui permettent de déterminer si une plateforme web est ergonomique. Puis, j'aborderai les bonnes pratiques à mettre en place lors d'un développement d'un projet web afin que la plateforme web soit la plus ergonomique possible une fois le projet terminé. Par la suite, je vous présenterai tout le travail d'analyse et d'amélioration que j'ai réalisé autour de la plateforme. Pour terminer, je vous donnerai mon avis sur l'état de la plateforme une fois opérationnelle, mon travail de recherche ainsi que sur l'analyse et l'amélioration apportée à cette plateforme et les conclusions que j'ai pu en tirer.

2. Historique du projet

Il est important de voir comment est née l'idée de ce projet et comment il a été géré tout au long de son développement afin de comprendre l'état actuel de la plateforme.

Auparavant, les activités se trouvaient toutes sur le site de l'état. Elles étaient stockées dans différentes bases de données. Les activités annuelles étaient stockées dans la base de données « hobby », quant aux activités de vacances, elles étaient enregistrées dans la base de données « info-vacances ».

Ces deux bases de données étaient gérées par un service de l'Etat de Genève. À chaque fois que des informations étaient transmises à l'état concernant des activités, ce service était chargé de les saisir dans la base de données correspondante. De plus, il était rémunéré pour mettre à jour ces informations.

Le site « www.ge.ch » mettait ainsi à disposition deux moteurs de recherche distincts afin de permettre à un utilisateur de pouvoir filtrer parmi toutes les activités disponibles dans chacune des bases.

Il a été décidé de lancer le projet de la création de la plateforme « loisirsjeunes.ch » afin de regrouper, de gérer et de proposer les données d'activités contenues dans les anciennes bases au sein de la nouvelle plateforme. Dorénavant, ce ne sera plus le service de l'état mais les organismes qui tiendront à jour leurs informations.

Un grand nombre d'intervenants ont pris part au développement de cette plateforme :

Le service des loisirs éducatif (SLE), qui n'existe plus aujourd'hui, était chargé de la saisie des informations des activités dans les anciennes bases.

La Direction Générale de l'Office de la Jeunesse (DGOJ) a été chargée de migrer ce projet du SLE vers le GLAJ-GE.

Le Groupe de liaison genevois des Associations de jeunesse (GLAJ-GE) s'occupera de la gestion de la plateforme une fois qu'elle sera mise en production.

La société BUXUM qui est une entreprise externe dans le domaine du graphisme, a été chargée de réaliser les maquettes de la plateforme ainsi que les pages de celle-ci au format HTML et CSS.

La Haute Ecole de Gestion de Genève (HEG) a programmé le moteur de recherche de la plateforme et a migré certaines données qui se trouvaient dans les anciennes bases ainsi que quelques éléments de maintenance.

Le projet a rencontré plusieurs problèmes au cours de son développement. Voici les points les plus importants qui ont fait que le projet en soit là à ce jour :

Tout d'abord, un problème majeur s'est révélé au moment de la migration du SLE vers le GLAJ. La DGOJ n'a pas mis un spécialiste métier dans le groupe de pilotage du projet afin de pouvoir aider et donner un avis dans la prise de décision.

Suite à cela, les maquettes de la plateforme, réalisées par BUXUM, ont été validées en fonction de leurs apparences et non à l'aide d'une analyse fonctionnelle. Cela est dû à l'absence d'élément métier dans le comité de pilotage.

Ce n'est qu'au lancement de la plateforme qu'ont été effectués les premiers tests et c'est à ce moment-là qu'ont surgis les premières incohérences. Des modifications ont été apportées un peu de tous les côtés, mais sans arriver au résultat final souhaité. L'illustration qui suit résume bien l'évolution de ce projet.

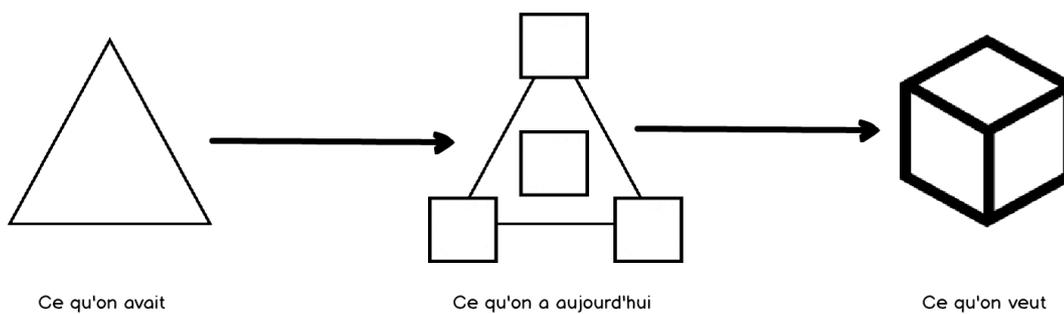


Figure 1 - Illustration de l'avancée du projet

Lors de la prise en main du projet, il n'est qu'à moitié fonctionnel. Un grand nombre de choses ont été faites en vitesse. Plusieurs fonctionnalités ne fonctionnent pas, certains problèmes font leur apparition et l'ergonomie de la plateforme peut être nettement améliorée afin de proposer un outil plus agréable à utiliser.

3. L'ergonomie d'une plateforme

L'ergonomie web consiste à mettre en place une plateforme qui propose sécurité, confort et efficacité dans l'utilisation de celle-ci.

3.1 Les critères de l'ergonomie

Après avoir vu la définition de l'ergonomie, il est important de comprendre les critères qui la décrivent et sur lesquels elle se repose afin de faciliter l'implémentation de celle-ci dans de futurs projets web.

3.1.1 La sobriété

Un site internet se doit d'être simple. Cela signifie qu'il doit contenir uniquement ce qui est essentiel afin de faire une bonne impression et d'être cohérent aux yeux de l'utilisateur.

3.1.2 La lisibilité

Le temps de lecture d'un texte sur un écran est 25% supérieur à une lecture sur papier. Pour cette raison, la clarté du texte est primordiale.

La clarté peut être améliorée en utilisant une bonne structure. Les paragraphes et les titres de différents niveaux aident l'utilisateur dans la lecture.

Il ne faut pas oublier d'organiser ses informations selon leurs degrés d'importance. Les éléments les plus importants doivent être représentés toujours en haut de la page.

3.1.3 L'utilisabilité

La navigation doit être facile pour l'utilisateur. Cela signifie que n'importe quelle information doit être accessible en faisant le moins de clics possibles. La « Règle des 3 clics » est un concept qui pousse à ce que n'importe quelle information soit accessible en moins de 3 clics pour un utilisateur.

L'utilisateur doit pouvoir se repérer sur le site. Il est conseillé de mettre le logo à un endroit précis et qu'il ne change pas d'emplacement, tout comme la navigation. La structure des pages est censée être la même afin que l'utilisateur comprenne qu'il se trouve toujours sur le même site.

La navigation ne doit pas être restreinte pour l'utilisateur. Il doit pouvoir aller sur la page qu'il lui semble en un seul clic.

L'URL de la page doit être explicite et visible à l'utilisateur afin qu'il puisse se repérer sur le site.

3.1.4 La rapidité

En général, un utilisateur n'attend pas plus de 15 secondes lors du chargement d'une page web. Il est donc nécessaire que cela soit le plus rapide possible. Attention tout de même, le temps peut dépendre de plusieurs facteurs tels que la connexion du visiteur, le contenu de la page web ou encore du serveur web.

Il est important de bien choisir ses images, car elles permettront de faire gagner du temps au chargement de la page. Il est préférable que les images se trouvent dans les bonnes dimensions afin de ne pas avoir à être redimensionnées.

3.1.5 L'interactivité

L'interactivité sur un site représente les interactions possibles entre un utilisateur et un site web. Elles sont mises en place à l'aide de liens hypertextes afin de permettre à l'utilisateur un grand nombre de chemins différents afin d'arriver au résultat souhaité. Il est important de savoir utiliser correctement cela, car un trop grand nombre de liens peuvent perturber la lecture de la plateforme.

Afin de ne pas perdre l'utilisateur au premier abord, il est préférable de ne pas dévoiler toutes les informations à première vue, mais plutôt de les découper. Ce découpage consiste à rediriger l'utilisateur à l'aide d'un lien afin d'accéder à la totalité de l'information.

Il est important de faciliter les échanges avec les visiteurs du site à l'aide d'un moyen de contact, même si ce n'est qu'une simple adresse de courrier électronique. Cela permettra de récolter les ressentis des utilisateurs pour le bon fonctionnement de la plateforme.

3.1.6 L'adaptabilité

Un site web qui s'adapte, c'est offrir une consultation confortable peu importe le support utilisé. L'utilisateur n'a pas à effectuer certaines manipulations afin de consulter le site web sinon cela pourrait détériorer son expérience.

3.1.7 L'accessibilité

Il est essentiel pour une plateforme de respecter les standards fixés par le W3C. Cela garantit que le site est consultable de façon correcte par n'importe quel logiciel client.

3.2 Les bonnes pratiques d'ergonomie pour un moteur de recherche

Un moteur de recherche est ce qui nous permet de chercher rapidement une information au sein d'un site web. On peut en voir de tout genres sur un grand nombre de sites, mais nous allons voir ensemble un certain nombre de conseils afin de réaliser au mieux un moteur de recherche.

3.2.1 Le design

Un moteur de recherche se doit de suivre certaines normes de webdesign pour permettre une meilleure utilisabilité de ses utilisateurs.

Dans la majorité des sites, l'emplacement d'un formulaire de recherche se trouve en général centré horizontalement sur une page. Dans les projets de e-commerce, il est même conseillé d'ajouter un filtre de recherche sur les pages ne contenant pas de formulaire afin d'améliorer l'accessibilité à l'information. Concernant les filtres de recherche, ils sont généralement en haut à droite d'une page web.

Le champ de saisie de l'information principale doit généralement être assez grand afin de pouvoir observer la saisie de l'utilisateur qui donnera suite à une requête.

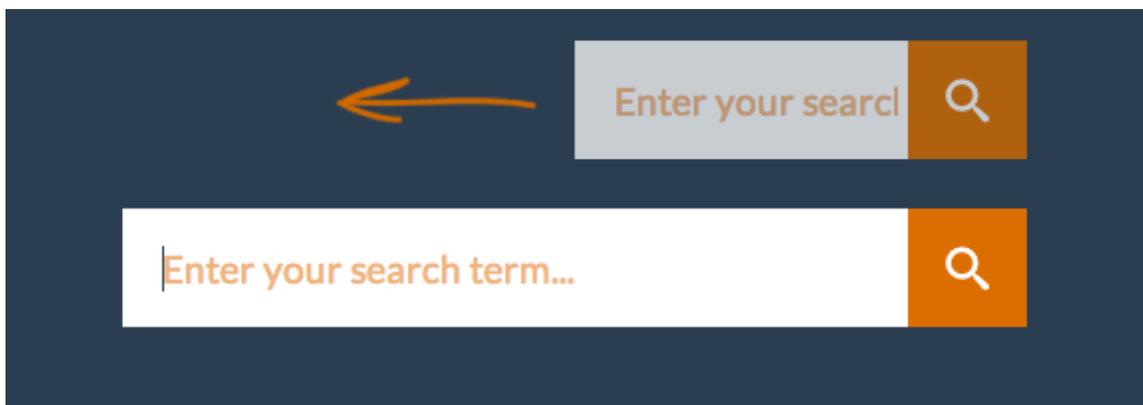


Figure 2 - Importance de la taille des champs de saisies¹

Il est préférable de ne pas limiter le nombre de caractères d'un champ de saisie. Dans le cas d'une restriction, il ne faut pas que cela limite l'utilisateur dans la requête qu'il souhaite faire.

¹ <http://blog.adimeo.com/9-conseils-pour-ameliorer-ergonomie-de-votre-moteur-de-recherche>

Les champs de saisies doivent être vides la première fois que l'utilisateur se rend sur le site. Cela lui évite de les vider lui-même avant de réaliser une requête.

La mise en place de « placeholder » dans les champs de saisie permet d'aider l'utilisateur dans la préparation de sa requête. Le « placeholder » est du texte présent dans un champ de saisie qui vient à disparaître dès que l'utilisateur effectue une saisie à l'intérieur de celui-ci.

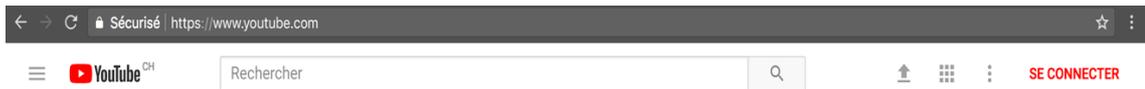


Figure 3 - Exemple de design avec le formulaire de recherche de YouTube ²

Il n'est pas nécessaire d'afficher une description à côté du champ de saisie. Ceci est dans le but de gagner en lisibilité. Ce point est souvent compensé par l'utilisation de « placeholder » au sein des champs de saisies.

3.2.2 Les fonctionnalités

Une fois le design du moteur de recherche prêt, on va pouvoir aborder certaines fonctionnalités importantes à implémenter sur un moteur de recherche afin de rendre son utilisation la plus agréable possible.

L'autosuggestion est fondamentale dans la réalisation d'un bon moteur de recherche. Cela fait gagner du temps à l'utilisateur. Il aura plus de chance de trouver des résultats pertinents à l'aide de requêtes déjà connues et ainsi, le risque de fautes réalisées par l'utilisateur est réduit. Le fait de garder en mémoire d'anciennes recherches effectuées peut être intéressant à mettre en place.



Figure 4 – Exemple d'autosuggestion de Google ³

² <https://www.youtube.com/>

Il faut faire attention à certaines choses lors de la mise en place de l'autosuggestion. Le fait de proposer des résultats est une bonne chose pour celui qui l'utilise, mais lui proposer énormément de résultats peut impacter négativement son expérience avec cet outil. D'ailleurs, une grande liste de résultats ne doit en aucun cas dépasser la ligne de flottaison qui correspond au bas de l'écran, ce qui obligerait l'utilisateur à scroller et voir le formulaire de recherche disparaître.

Par la suite, le formulaire doit être prêt à recevoir certaines saisies spéciales des utilisateurs. Il faut prévoir des requêtes telles que la chaîne vide ou qui contiendrait des caractères spéciaux afin de ne pas perturber l'utilisateur dans l'utilisation de l'outil.

La plupart des sites proposent une recherche avancée pour permettre à l'utilisateur d'affiner sa recherche. On peut mettre cela en place en ajoutant des filtres au formulaire qui vont trier les résultats obtenus sur différents critères. Cette recherche avancée peut être cachée de l'utilisateur et seulement affichée dans le cas où l'utilisateur en signale le besoin. Le fait de l'afficher par défaut ne pose pas de problème si c'est un réel besoin pour la plateforme.

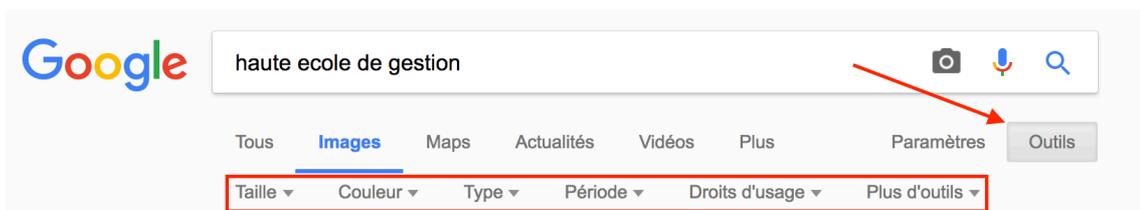


Figure 5 - Exemple de recherche avancée sur la recherche d'image de Google⁴

Pour terminer, il est déconseillé de mettre en place un bouton de réinitialisation de formulaire. Dans le cas où il est déclenché sans le vouloir, l'utilisateur se verrait mettre à zéro son formulaire.

3.2.3 L'affichage des résultats

La partie ergonomique est autant importante dans l'affichage des résultats que dans celle du formulaire de recherche. Une bonne lisibilité de celle-ci améliorera l'utilisabilité du moteur de recherche.

³ <https://www.google.ch/>

⁴ <https://www.google.ch/>

L'utilisateur doit facilement pouvoir soumettre à nouveau une requête afin d'affiner les résultats, ou d'en trouver dans le cas où aucun résultat n'est retourné. Le formulaire de recherche doit quant à lui contenir les informations de la requête effectuée auparavant pour que l'utilisateur n'ait pas à entrer toutes ses informations à nouveau.

De plus, il est primordial d'afficher le nombre de résultats retournés. Dans le cas où il y aurait un trop grand nombre de résultats, cela lui permettrait d'affiner sa recherche.

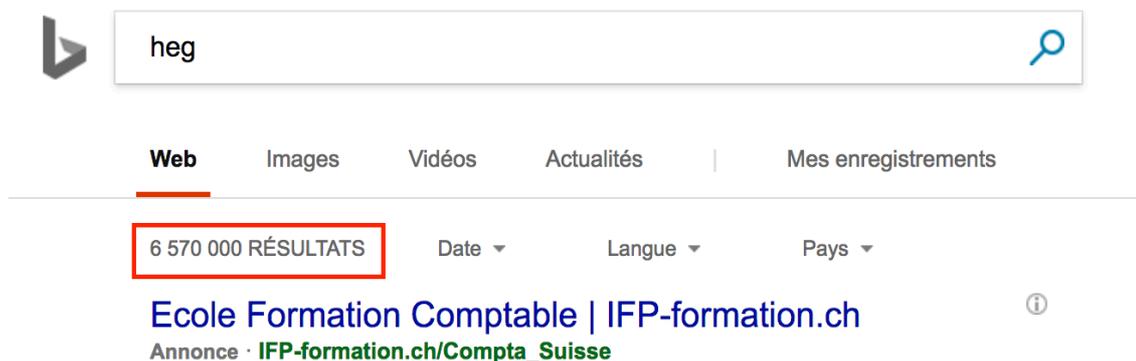


Figure 6 - Affichage du nombre total de résultats obtenus par une recherche sur Bing⁵

Par la même occasion, si une requête dépasse un certain nombre de résultats, il est judicieux de mettre à disposition de l'utilisateur une pagination afin de mieux se retrouver dans la consultation des résultats. On peut aussi proposer des tris sur les résultats afin de pouvoir les classer et faciliter la recherche de l'utilisateur.

Il est important pour le bon sens de l'utilisateur que le design de la page de résultats soit le même qui est utilisé pour la page du formulaire de recherche. Si l'on vient à ne pas répondre à ce critère, l'utilisateur peut se sentir perdu et finir par abandonner sa recherche. Il n'empêche pas que les deux se retrouvent sur la même page.

Pour finir, préciser le type de résultat qui va être retourné peut être utile en fonction du but de la plateforme. On peut aussi utiliser le stockage local afin d'améliorer la rapidité et réduire les charges sur le serveur.

⁵ <http://www.bing.com/>

4. Analyse et amélioration de la plateforme

Dans cette section, nous allons aborder toutes les modifications apportées à la plateforme, tant au niveau de l'ergonomie que des fonctionnalités qu'elle propose. Chacun des points suivants fera le lien avec le travail de recherche en précisant quel point d'ergonomie n'a pas été respecté ou dans le cas d'une fonctionnalité, la source du problème et les modifications faites afin qu'elle soit opérationnelle.

4.1 Architecture des pages de recherche

Le premier problème rencontré est un problème de conception d'architecture qui peut impacter sur la rapidité de la plateforme. Voici les différents critères ergonomiques non respectés lors du développement :

- Sobriété
- Rapidité
- Adaptabilité
- Accessibilité

La plateforme « loisirsjeunes.ch » a mis en place deux types de vues différentes : une vue desktop et une vue mobile. Le problème ne se voit pas à première vue mais est remarquable lorsqu'on se penche de plus près sur le code réalisé par le développeur.

Il a mis en place dans chacune des pages les deux types de vues. C'est à dire que lorsque l'on récupère les données de la base, elles seront affichées sur la page dans les deux formats qui sont desktop et mobile. Pour pallier à cela, le développeur a juste masqué la vue qui ne l'intéressait pas en fonction de l'appareil utilisé, à l'aide du CSS.

```
▼<div id="content mobile_tab">
  ▼<div class="centre_tab">
    <!-- TABLEAU DESKTOP - TAB -->
    ▼<div class="result_tab">
      ▼<table>
        ▶<thead>...</thead>
        ▼<tbody id="liste_activites">
          ▶<tr>...</tr>
          ▶<tr>...</tr>
          ▶<tr>...</tr>
          ▶<tr>...</tr>
          ▶<tr>...</tr>
          </tbody>
        </table>
      </div>
    <!-- TABLEAU MOBILE -->
    ▼<div class="result_tab_mobile">
      ▶<div class="line_tab_mobile">...</div>
      ▶<div class="line_tab_mobile">...</div>
      ▶<div class="line_tab_mobile">...</div>
      ▶<div class="line_tab_mobile">...</div>
      ▶<div class="line_tab_mobile">...</div>
    </div>
  </div>
```

Figure 7 - Contenu de la page avec les deux vues desktop et mobile

Pour ce qui est du code, on retrouve sur toutes les pages le même bout de code ci-dessous. Il va exécuter une boucle sur le tableau qui contient les informations que l'on a récupérées de la base de données.

Pour chacune des informations, on va appeler deux fonctions. La première se charge d'ajouter l'affichage de l'information dans la vue desktop et la seconde, d'ajouter l'affichage dans la vue mobile.

Le problème que l'on rencontre ici est qu'au chargement de la page les deux vues sont chargées alors que seulement une seule est utilisée. Cela a pour effet de rendre les pages plus volumineuses et cela implique un chargement plus long car la même information est affichée à deux reprises.

```
function afficherActivites() {
  $('#liste_activites').html("");
  $('#result_tab_mobile').html("");
  for (var acti of activites) {
    $('#liste_activites').append(ligneActivite(acti));
    $('#result_tab_mobile').append(ligneActiviteMobile(acti));
  }
  if (activites.length==0) {
    $('#liste_activites').html("<tr><td colspan='6'>Aucun résultat ne correspond à votre recherche</td></tr>");
  }
  openCloseTab();
  $('#favorisAE').on('click', function(evt){
    clickfavoris(evt);
  });
}
```

Figure 8 - Code effectuant l'affichage des informations

On rencontre le même problème avec les filtres de recherche de ces mêmes pages. On s'aperçoit que les deux filtres sont existants dans la page mais que l'un est masqué par rapport à l'autre en utilisant le CSS.

```
<form action="#" method="post" id="rechercheActiviteForm"
onsubmit="return false">
  <div id="modify_search_spacing">
    <div class="input_field">...</div>
    <div class="input_field">...</div>
    <div class="input_field">...</div>
    <div class="input_field">...</div>
    <div class="input_field range">...</div>
    <div class="input_field">...</div>
    <div class="input_field">...</div>
  </div>
  <div class="modify_search_mobile" style="display: block;">
    <select id="domaineActiviteMob" name="domaineActiviteMob"
placeholder="Domaine d'activité" class="modernSelect
modify_select" style="display: none;" aria-disabled=
"false" tabindex="0">...</select>
    <span>...</span>
    <select id="typeActiviteMob" name="typeActiviteMob"
placeholder="Type" class="modernSelect modify_select"
style="display: none;" aria-disabled="false" tabindex="0">
...</select>
    <span>...</span>
    <select id="lieuMob" name="lieuMob" placeholder="Lieu"
class="modernSelect" style="display: none;" aria-disabled=
```

Figure 9 - Contenu de la page pour les filtres de recherche

Il faut donc simplifier cela en mettant en place un seul filtre de recherche qui s'adaptera en fonction de la taille de l'écran. En ce qui concerne les informations qui découlent de la recherche, il faut les afficher uniquement dans le format adéquat qui est utilisé par l'utilisateur.

Modifications apportées

On va tout d'abord se concentrer sur l'affichage des informations. On a pu voir plus haut que lors de l'affichage des informations, une boucle se charge de parcourir un tableau et d'ajouter l'information dans les deux vues possibles qui sont desktop et mobile. Il faut faire en sorte d'ajouter au DOM l'information uniquement dans la vue qui nous intéresse.

Pour remédier à cela, la solution qui a été mise en place est totalement gérée par jQuery. Au chargement de la page, on récupère la largeur de l'écran qui nous permettra de déterminer sous quelle forme il faudra afficher les informations. Il est également important de mettre en place l'évènement « `resize()` » au niveau de la fenêtre dans le but de modifier l'affichage des informations si le client vient à redimensionner la fenêtre de son navigateur ou lors d'un changement d'orientation sur mobile. Si cela se produit, on appelle à nouveau la méthode qui se charge d'afficher les informations mais en lui indiquant sous quel format il doit les afficher.

Sur l'image ci-dessous, on peut voir le résultat final obtenu pour la fonction qui s'occupe de l'affichage des informations. Avant de lancer l'une des deux boucles d'affichage, on vérifie dans quelle vue se trouve l'utilisateur à l'aide d'un booléen.⁶ Les valeurs sont attribuées en fonction de la largeur de l'écran.

A travers ce test, on sait quelle boucle va être exécutée, laquelle va se charger d'ajouter les informations au bon endroit dans le DOM. Dans une vue de type desktop, on ajoutera l'information au DOM au format spécifique pour desktop⁷ et dans le cas d'une vue mobile, on ajoutera au format spécifique pour mobile.⁸ Avec cette solution, la page ne contient plus qu'une fois l'information.

⁶ Référence à la flèche bleue sur l'image

⁷ Référence à la flèche verte sur l'image

⁸ Référence à la flèche orange sur l'image

```

function afficherActivites() {
    $('#liste_activites').html("");
    $('#result_tab_mobile').html("");
    if(activites.length <= activiteFin){
        activiteFin = activites.length-1;
    }
    //si on est en vue desktop
    if/vueMobileInfo == false){
        for (var i = activiteDebut; i <= activiteFin; i++) {
            $('#liste_activites').append(ligneActivite(activites[i]));
        };
    }else{//si on est en vue mobile
        for (var i = activiteDebut; i <= activiteFin; i++) {
            $('#result_tab_mobile').append(ligneActiviteMobile(activites[i]));
        };
    }
}

```

Figure 10 - Affichage des informations pour une seule des vues

Maintenant que l'on s'est occupé de l'affichage des informations, on va se pencher sur les deux filtres de recherches qui se trouvent sur ces pages. Comme vu précédemment, on s'était rendu compte que la balise de formulaire contenait deux filtres de recherches, un pour chacune des vues de la plateforme.

En analysant de plus près les deux filtres, on peut s'apercevoir qu'ils sont pareils à deux exceptions près. Chaque élément du filtre desktop est englobé dans une balise « div » et le « div » qui englobe la totalité du filtre de recherche a un attribut différent l'un de l'autre.

On va donc jouer avec les attributs des balises « div » à l'aide de jQuery afin de passer facilement d'un filtre en vue desktop à mobile et réciproquement. Avant tout, le filtre mobile ne servant plus à rien est supprimé des pages. On va adapter celui du desktop pour répondre à la demande de l'utilisateur de la plateforme. Pour cela, on va ajouter au niveau de l'HTML un identifiant à chacune des balises « div » dont les attributs vont changer en fonction de la vue souhaitée.

On peut voir, sur l'image ci-dessous, qu'on va ajouter un attribut « class »⁹ pour identifier le div qui englobe le filtre car l'attribut « id » disparaîtra dans la vue mobile. Pour ce qui est des autres balises « div », on va les identifier avec un attribut « id »¹⁰ afin de gérer facilement l'ajout ou suppression de l'attribut « class ».

⁹ Référence au rectangle orange sur l'image

¹⁰ Référence au rectangle violet sur l'image

```

<form action="#" method="post" id="rechercheActiveForm" onsubmit="return false">
<!-- Formulaire -->
<div id="modify_search_spacing" class="div_modify_search_spacing">
<div id="formulaireAdapte" class="input_field">
  <select id="domaineActive" name="domaineActive" placeholder="Domaine d'activité"
    <option value="-1" selected="selected">Domaine d'activité</option>
    <?php echo($domaine->domainesOptionHTML());?>
  </select>
</div>
<div id="formulaireAdapte" class="input_field">
  <select id="typeActive" name="typeActive" placeholder="Type" class="modernSelect"
    <option value="-1" selected="selected">Type</option>

```

Figure 11 - Identification des div à adapter en fonction de la vue

Maintenant que l'on a fini avec l'identification des balises qui nous intéressent, on va se pencher sur le code. On va réutiliser les événements qu'on avait déclarés et utilisés pour l'affichage des informations. Les événements « ready() » et « resize() » vont permettre d'adapter en tout temps le filtre de recherche en fonction de la taille de l'écran.

Tout d'abord, l'évènement « ready() » va permettre d'adapter le filtre de recherche au format mobile si nécessaire lors du chargement de la page car c'est le format desktop qui est affiché par défaut.

Ensuite, l'évènement « resize() » va permettre, comme pour l'affichage des informations, d'adapter l'affichage du filtre dans le cas d'un redimensionnement de la page sur un navigateur ou lors d'un changement d'orientation sur mobile. À chaque redimensionnement, on vérifie la largeur de l'écran et si avant le redimensionnement le filtre était affiché différemment du format voulu actuellement, alors c'est seulement à ce moment-là que l'on va apporter les modifications au filtre.

Enfin, sur l'image ci-dessous, le code encadré en rouge sera exécuté lorsque l'on passe en affichage mobile. Il va enlever le style appliqué sur la « div » de chaque élément du filtre et enlever l'attribut « id » sur le « div » qui englobe tous les éléments pour les mêmes raisons. Quant au code encadré en vert, il va faire le contraire du code précédent afin d'afficher le filtre au format desktop.

```

$(window).resize(function() {
  largeurFenetre = $(window).width();
  //si on passe en vue mobile pour la recherche
  if (largeurFenetre <= 1024 && vueMobileRecherche == false){
    vueMobileRecherche = true;
    $(".div #formulaireAdapte").removeClass("input_field");
    $(".div_modify_search_spacing").addClass("modify_search_mobile");
    $(".div_modify_search_spacing").removeAttr("id");
  }
  //si on passe en vue desktop pour la recherche
  if (largeurFenetre > 1024 && vueMobileRecherche == true){
    vueMobileRecherche = false;
    $(".div #formulaireAdapte").addClass("input_field");
    $(".div_modify_search_spacing").removeClass("modify_search_mobile");
    $(".div_modify_search_spacing").attr("id", "modify_search_spacing");
    $(".div_modify_search_spacing").css('display', '');
  }
}

```

Figure 12 - Code qui adapte l'affichage du filtre de recherche en fonction de la largeur de l'écran

4.2 Affichage des résultats de recherche

Le problème ici est beaucoup plus flagrant pour l'utilisateur de la plateforme. Voici les différents critères ergonomiques non respectés lors du développement :

- Lisibilité
- Rapidité
- Interactivité

Lorsqu'une personne effectue une recherche, elle peut spécifier certains paramètres qui seront pris en compte pour filtrer toutes les données présentes dans la base. Mais en fonction de la recherche effectuée par l'utilisateur, on peut se retrouver avec plus d'une centaine d'informations à afficher et qui s'affichent les unes à la suite des autres à n'en plus finir.

Si on regarde de plus près l'état actuel de la plateforme sur l'image qui suit, le rectangle en bleu clair représente l'affichage d'une des informations souhaitées par l'utilisateur. Mais ce dont il faut faire attention ici, c'est le curseur en haut à droite de l'image qui montre l'énorme quantité d'informations qui est contenue dans la page.



AJOUTER À MA SÉLECTION	ACTIVITÉ	LIEU	JOUR	ÂGE	DOMAINE D'ACTIVITÉ ET TYPE
☆	1ÈRE ANNÉE FORMATION MUSICALE	COLLONGE-BELLERIVE	LUNDI 16h20 à 17h10	DE 7 À 10 ANS	MUSIQUE: Solfège
☆	2ÈME ANNÉE FORMATION MUSICALE	COLLONGE-BELLERIVE	LUNDI 17h50 à 18h40	DE 7 À 12 ANS	MUSIQUE: Solfège
☆	3ÈME ANNÉE FORMATION MUSICALE	COLLONGE-BELLERIVE	JEUDI 17h15 à 18h05	DE 9 À 16 ANS	MUSIQUE: Solfège
☆	4ÈME ANNÉE FORMATION MUSICALE	COLLONGE-BELLERIVE	JEUDI 16h20 à 17h10	DE 9 À 16 ANS	MUSIQUE: Solfège
☆	ACTIVITÉ SCOUTE	VEYRIER	SAMEDI 14h00 à 17h00	DE 6 À 12 ANS	NATURE ET ENVIRONNEMENT: Scoutisme
☆	AÏKIDO ENFANTS GROTTES	GROTTES	MARDI 17h30 à 18h30	DE 7 À 12 ANS	SPORT ET JEU: Arts martiaux

Figure 13 - Exemple de résultats d'une recherche d'activités par un utilisateur

La solution préconisée est de mettre en place une pagination des informations afin de ne pas surcharger la page avec des centaines d'activités d'un seul coup.

Modifications apportées

Pour commencer, on va ajouter au niveau de l'HTML les éléments qui nous permettront de naviguer entre les différentes pages. On va donc ajouter deux boutons qui vont permettre à l'utilisateur soit de passer à la page suivante, soit de revenir à la page précédente. Entre les deux boutons, on affichera sur quelle page se trouve l'utilisateur dans la consultation des informations. Ces éléments ont été ajoutés à chaque page de recherche. Voici le résultat obtenu actuellement.



Figure 14 - Vue HTML des éléments qui permettent de naviguer entre les pages

Une fois la vue opérationnelle, on va s'occuper de gérer les actions possibles à l'aide de jQuery. Sur chacun des boutons de la vue, on va ajouter l'évènement « click() » qui va nous permettre d'afficher les informations qui suivent ou qui précèdent celles qui sont affichées actuellement.

Par la même occasion, on va masquer les boutons si on se trouve à l'une des extrémités de l'affichage ou s'il n'y a pas de navigation possible. Maintenant que les évènements sont en place, on va se pencher sur l'affichage paginé de ces informations.

Lorsqu'un utilisateur déclenche une recherche, on va récupérer un tableau contenant les informations à afficher. On va regarder si le nombre total d'informations à afficher est divisible par le nombre d'informations affichées sur une page.¹¹ Si c'est le cas, le résultat de cette division est le nombre total de pages sur lesquelles on peut naviguer.¹² Dans le cas contraire, on prend la partie entière du résultat de cette division et on lui ajoute 1.¹³ Une fois le nombre de pages connu, on va demander à afficher les informations.

¹¹ Référence à la flèche jaune sur l'image

¹² Référence à la flèche bleue sur l'image

¹³ Référence à la flèche bleue sur l'image

```

// Requête
$.post("anneescolaire AJAX.php", data, function(res, status){
  // Supprimer activités actuelles Desktop et mobile
  // Ajouter le résultat de la recherche
  activites = JSON.parse(res);
  $('#pageSuivante').css('visibility','visible');
  if(activites.length % pagination == 0){
    pageTotal = activites.length / pagination;
  }else{
    pageTotal = Math.trunc(activites.length / pagination)+1;
  }
  if(pageTotal == 1){
    $('#pageSuivante').css('visibility','hidden');
  }
  afficherActivites();
}

```

Figure 15 - Code qui détermine le nombre total de pages

Au niveau de la fonction d'affichage des informations, il y a 3 points importants pour le bon fonctionnement de tout cela. Tout d'abord, on doit vérifier que la variable de fin de boucle ne dépasse pas la taille du tableau que l'on doit afficher.¹⁴ Cette valeur varie en fonction des clics de la part de l'utilisateur en se promenant entre les différentes pages. Ce test n'est qu'utile lorsque l'utilisateur arrive sur la dernière page.

Par la suite, on va se charger d'exécuter la boucle d'affichage qui correspond à la vue de l'utilisateur.¹⁵ Finalement, une fois l'affichage terminé, on affiche l'état de la navigation qui est la page où se trouve actuellement l'utilisateur.¹⁶

```

function afficherActivites() {
  $('#liste_activites').html("");
  $('#result_tab_mobile').html("");
  if(activites.length <= activiteFin){
    activiteFin = activites.length-1;
  }
  //si on est en vue desktop
  if/vueMobileInfo == false){
    for (var i = activiteDebut; i <= activiteFin; i++) {
      $('#liste_activites').append(ligneActivite(activites[i]));
    };
  }else{//si on est en vue mobile
    for (var i = activiteDebut; i <= activiteFin; i++) {
      $('#result_tab_mobile').append(ligneActiviteMobile(activites[i]));
    };
  }

  if (activites.length==0) {
    $('#liste_activites').html("<tr><td colspan='6'>Aucun résultat ne correspond à votre recherche</td></tr>");
    $('#div_pagination').css('display','none');
  }else{
    $('#div_pagination').css('display','block');
    $('#maPage').html("");
    $('#maPage').html("Page "+pageActuelle+" / "+pageTotal);
  }
}

```

Figure 16 - Code permettant l'affichage paginé des informations

¹⁴ Référence au rectangle bleu sur l'image

¹⁵ Référence aux rectangles oranges sur l'image

¹⁶ Référence au rectangle vert sur l'image

4.3 Processus de login/récupération de mot de passe

On aborde ici un problème de lisibilité par rapport aux messages d'erreurs qui sont intégrés au sein de la page de login. Voici les différents critères ergonomiques non respectés lors du développement :

- Lisibilité
- Utilisabilité
- Interactivité

Lorsqu'un utilisateur se rend sur une des pages de login, deux possibilités s'offrent à lui. La première est de saisir ses informations de connexion afin d'accéder à ses privilèges d'administrateur ou ceux d'un organisme et la deuxième est de signaler à la plateforme qu'il a oublié son mot de passe.

Dans les deux cas, des messages d'informations s'affichent sur la page juste au-dessus du formulaire. Le premier notifie l'utilisateur qu'il a saisi des données de connexion qui sont erronées et le second l'avertit que son nouveau mot de passe lui a été envoyé si le mail saisi juste avant existe dans la base de données.

Gestion des données - Espace réservé aux administrateurs



The image shows a login form titled "Gestion des données - Espace réservé aux administrateurs". At the top, there is a red-bordered error message box with a warning icon and the text "Authentification impossible" and "Vous devez saisir votre login (email) et votre mot de passe". Below the error message, there are two input fields: "Login*" and "Mot de passe*". Below the input fields, there is a note "* Champs obligatoires". At the bottom of the form, there are two buttons: "OK" and "MOT DE PASSE PERDU". A red arrow points from the error message box to the "Mot de passe*" input field.

Figure 17 - Exemple de message d'erreur incrusté dans la page

Si on se penche sur le code réalisé, tout est géré en PHP. Dans le cas d'un échec de login ou lors du signalement d'un oubli de mot de passe, le code va instancier des variables qui contiennent les textes qui seront affichés à l'utilisateur dans les messages d'avertissement. Ces variables vont permettre de gérer l'affichage du côté HTML. En effet, au chargement de la page, on vérifie si ces variables ont été créées et dans ce cas, on va afficher du code HTML en plus dans la page qui correspond au message d'avertissement qui informe l'utilisateur qu'une des deux actions a eu lieu.

La solution qui sera mise en place est d'afficher ces mêmes messages d'avertissement mais sous forme de fenêtre modale, afin de bloquer le formulaire tant que l'on n'a pas cliqué sur le message affiché.

Modifications apportées

Toutes les modifications qui ont été faites dans cette partie se retrouvent à deux endroits qui sont : la page de login pour les organismes et celle pour les administrateurs du site. L'affichage des messages d'informations en fenêtre modale est géré en jQuery. La partie PHP n'a pas été touchée.

En ce qui concerne la page HTML, on va enlever les tests qui servaient à vérifier l'existence des variables contenant les messages qui permettaient d'afficher les messages à l'utilisateur. À la place, on va ajouter une balise « div » où se trouvait ces tests et différencier les deux types de messages en leur attribuant des « id » différents à chacun d'entre eux. Cela nous permettra de gérer leur affichage avec jQuery. La dernière modification apportée à l'HTML est l'ajout d'une autre balise « div » juste après la balise de « body » qui n'a aucun contenu mais qui va nous permettre de griser le fond de l'écran et mettre en avant le message d'erreur à l'utilisateur sous forme de fenêtre modale.

Du côté jQuery, la gestion de la fenêtre modale est faite à l'aide de deux événements. Le premier événement « ready() » va permettre, une fois que la page sera chargée, de vérifier si les variables contenant les textes d'erreurs ont été instanciées et donc de savoir si on doit afficher l'une des deux fenêtres modales. Seulement une des deux fenêtres peut avoir ces variables qui sont instanciées. On va donc vérifier dans le DOM, où la valeur des variables est normalement insérée, s'il contient des informations.¹⁷

Dans le cas où l'une des deux fenêtres possède un message d'erreur à l'intérieur, alors on va déclencher l'affichage de celui-ci.¹⁸ Le second événement « click() » va se charger de fermer la fenêtre modale une fois que l'utilisateur clique dessus afin d'afficher à nouveau le formulaire de login.¹⁹

¹⁷ Référence à la flèche verte sur l'image

¹⁸ Référence à la flèche rouge sur l'image

¹⁹ Référence au rectangle bleu sur l'image

```

$(document).ready(function(){
//si problème de connexion on affiche message d'info en fenetre odal
if($("#messageErreur .content .title").text() != ""){
    $("#opac").css("display","block");
    $("#messageErreur").css("display","block");
}
//si demande un nouveau mot de passe on affiche message d'info en fenetre modal
if($("#messageDetail .content .title").text() != ""){
    $("#opac").css("display","block");
    $("#messageDetail").css("display","block");
}
});

//fait disparaitre la fenetre modal lorsqu'on clique dessus
$(".modal").on("click", function(evt) {
    $("#opac").css("display","none");
    $(".modal").css("display","none");
});

```

Figure 18 - Code qui permet le fonctionnement de la fenêtre modale

4.4 Corrections de problèmes d’affichages et de fonctionnement

Le mandant m’a fait part de certains problèmes au niveau de l’affichage des informations qui est incorrect et de fonctionnalités non-opérationnelles.

4.4.1 Inversion d’informations sur la page « AnnéeScolaire » en vue mobile

Ce problème va impacter le critère ergonomique de la lisibilité. Il s’agit d’une inversion des informations au niveau de la vue mobile sur la page des activités pour l’année scolaire. Les informations concernant les champs « domaine et types » ainsi que « lieu » des activités ont été inversées.²⁰ Ceci est un simple problème d’inattention de la part du développeur précédent, il suffit simplement d’inverser au niveau du code les variables des deux champs en question.

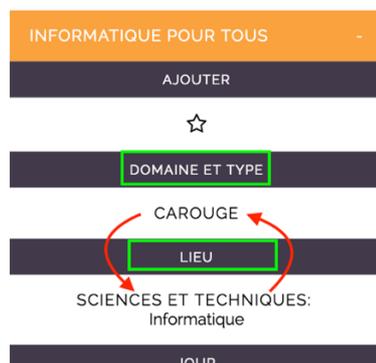


Figure 19 - Inversion des champs au niveau de la vue mobile

²⁰ Référence à la flèche en rouge sur l’image

4.4.2 Problème sur la page « vacances » en vue mobile

4.4.2.1 Variables non définies

Les deux problèmes suivants se situent tous les deux au niveau de la vue mobile des activités pour les vacances. Le premier est un problème d’affichage d’information concernant les places totales et disponibles pour une des activités. Ce problème touche de nouveau le critère ergonomique de la lisibilité.

Les variables indiquées dans le code étant censées contenir ces informations sont inconnues et donc les valeurs affichées sont « undefined ». ²¹ Le développeur précédent s’est trompé au niveau de l’appel des variables et le navigateur, ne connaissant pas les variables, affiche alors ce message. Pour résoudre le problème, il a fallu simplement changer ces variables inexistantes par celles qui contiennent réellement les informations.



Figure 20 - Nombre de places inconnues en vue mobile pour les activités de vacances

4.4.2.2 Ajout d’activités en favori impossible

Le second problème est un souci de fonctionnement. L’ajout d’une activité de vacances aux favoris ne fonctionne pas sur la vue mobile. En effet, le problème est que l’attribut ajouté à la balise contenant l’étoile de favori, qui sert à ajouter l’évènement d’ajout ou de suppression d’un favori, n’est pas le bon. Il est donc normal que, lorsque l’on clique sur l’étoile en vue mobile, rien ne se passe. L’attribut actuel est celui qui a été assigné à la balise de la page des activités de l’année scolaire. En remplaçant cet attribut par l’attribut correct, le problème est donc résolu. ²² Cette erreur d’inattention est sûrement la cause d’un « copier/coller » du code qui avait été produit pour la page de recherche des activités de l’année scolaire.

```
ligne += '<p class="plus_moins">+</p>';  
ligne += '</div>';  
ligne += '<div class="content_onglet">';  
ligne += '<p class="title_tab">Ajouter</p>';  
ligne += '<p class="content_tab favorisAE" ><i data-id="'+activite.ses_id+'''  
ligne += '<p class="title_tab">Faitière</p>';  
ligne += '<p class="content_tab"><span class="text_orange">';  
if (activite.org_clai==1) {
```

Figure 21 - Erreur sur l’attribut pour la gestion des favoris

²¹ Référence au rectangle en rouge sur l’image

²² Référence au rectangle en vert sur l’image

4.4.3 Modifications des conditions dans la requête d'affichage des activités

Un des autres points relevés concerne les activités affichées dans les pages de recherche des activités annuelles et des vacances. Actuellement, la plateforme affiche des activités qui sont déjà terminées. Le mandant souhaite que les activités affichées soient encore en cours ou que la date de début soit comprise dans les 12 mois qui suivent. Ce point relève du niveau fonctionnel de la plateforme.

Au niveau de la requête actuelle, on récupère les activités des 12 mois qui précèdent et qui suivent la date à laquelle la recherche est faite. Donc, on modifie les conditions de la requête de la façon suivante : on stocke la date actuelle ainsi que celle qu'il sera dans un an à partir du moment où est effectuée la recherche.²³ Puis, on récupère les activités qui ont une date de début inférieure à celle qu'il sera dans 12 mois et qui se termine plus tard que la date actuelle.²⁴ Ceci va permettre d'afficher les activités dans la durée d'une année au maximum et ainsi de cacher celles qui se sont terminées.

```
$clauses[] = 'org_statut_fk= :STATUT_OK,'  
// Affichage publique des activités ne dépassant pas 12 mois  
$dans12MoisStr = dateDans(1);  
$now = dateDans(0);  
$clauses[] = "act_debut<='$dans12MoisStr'";  
$clauses[] = "act_fin >='$now'";
```

Figure 22 - Modification des conditions de la requête récupérant les activités

4.4.4 Correction de la requête de recherche des activités vacances

Un autre problème dont s'est aperçu le mandant, est que la recherche d'activités pour les vacances ne prend pas en compte l'association qu'on lui indique dans le filtre de recherche. Le problème est que la requête nécessite de recevoir l'identifiant de l'association, mais ce qui lui est fourni est le nom de celle-ci. C'est un autre problème fonctionnel de la plateforme.

Actuellement, l'utilisateur dispose d'un champ de saisie qui lui permet de filtrer les associations affichées à l'aide d'une liste pour choisir parmi l'une d'entre elles. Lorsqu'il choisit une des associations présentes dans la liste, cela va automatiquement copier le nom de celle-ci dans le champ de saisie. On va donc récupérer l'id de l'association choisie à l'aide de jQuery et de l'évènement « change() » qui est exécuté une fois que l'utilisateur sort du champ de saisie.

²³ Référence au carré vert sur l'image

²⁴ Référence au carré rouge sur l'image

Lorsque l'évènement est exécuté par l'utilisateur, on va parcourir toutes les balises « option » qui contiennent les organismes afin de trouver si une d'elles possède le même texte que celui contenu dans le champ de saisie. Si cela est le cas, on va accéder à la valeur de la balise « option », qui correspond à l'id de l'association.²⁵ Si ce n'est pas le cas, cela va nous retourner comme valeur « undefined » qui signifie que la valeur contenue dans le champ de saisie ne correspond à aucune des associations de la liste. On va donc effacer ce qui est contenu dans le champ de saisie et ne pas prendre en compte d'association lors de la recherche d'activités.²⁶

```
$('#organisme').on("change", function (evt) {  
    //id assoc contient l'id de l'association a rechercher  
    idAssoc = $(".organismes option").filter(function() {  
        return $(this).text() === $('#organisme').val();  
    }).first().attr("value");  
    //pas d'association trouver  
    if(idAssoc == null){  
        idAssoc = -1;  
        $('#organisme').val('');  
    }  
});
```

Figure 23 - Récupération de l'identifiant de l'association

4.5 Suppression des éléments de l'Etat de Genève

Le problème abordé ici est un souci dû à la gestion du projet. Le directeur adjoint de l'OEJ²⁷ a demandé à ce que l'on supprime de la plateforme les éléments d'affichage faisant référence à l'Etat de Genève. Plus particulièrement, d'enlever le logo qui se trouve en haut de chaque page et ainsi que le copyright en bas de celle-ci. Le header et footer étant le même sur chaque page, il a été correctement dissocié des contenus des pages afin de simplifier les modifications. Il a simplement fallu supprimer les « div » qui contiennent les informations que l'on ne veut plus voir affichées sur la plateforme.

²⁵ Référence au rectangle turquoise sur l'image

²⁶ Référence au rectangle jaune sur l'image

²⁷ Office de l'enfance et de la jeunesse

4.6 Suppression de l'image « Résultat de la recherche »

Le critère ergonomique non respecté ici est au niveau de la sobriété de la plateforme.

Il se trouve que lors de la conception du design, on a décidé de placer en dessous du formulaire de recherche une image assez grande afin de préciser que les résultats se trouvent juste après celle-ci. Le problème est que cela porte à confusion, car quand un utilisateur effectue une recherche, s'il ne scroll pas vers le bas, il n'y a pour lui aucun changement au niveau de la page. Alors qu'en réalité, les résultats se trouvent juste après cette image. Il a été décidé avec l'accord du directeur adjoint de l'OEJ, de supprimer cette image du contenu des pages afin d'en améliorer son ergonomie.

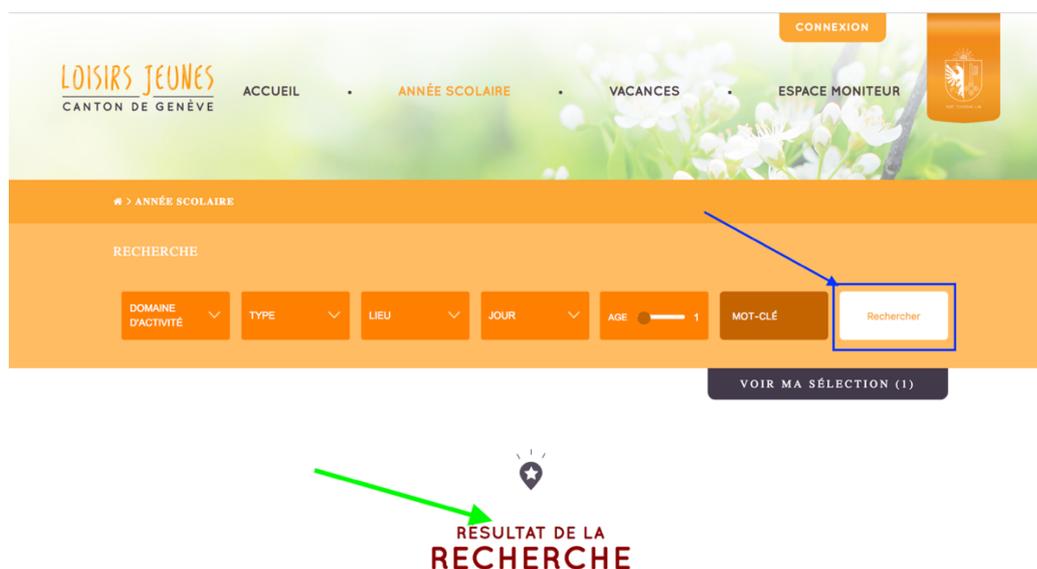


Figure 24 - Problème d'ergonomie lors de l'affichage des résultats

4.7 Changement des curseurs

Le mandant m'a fait part de certaines de ses impressions concernant les curseurs utilisés dans les formulaires de recherche des activités. Il a relevé des problèmes d'affichage de ses curseurs lors de la consultation de la plateforme avec le navigateur « Firefox ».

Il n'est pas réellement convaincu de l'utilisation d'un curseur dans ce cas précis. Il a, par la même occasion, indiqué vouloir étendre l'âge de la recherche non plus allant jusqu'à 16 ans mais jusqu'à 18 ans maximum et d'effectuer une recherche sans prendre l'âge en compte. Cette modification implique qu'il faudra également adapter cette nouvelle limite au niveau des formulaires d'ajouts d'activités.

Ces problèmes sont dû à la gestion du projet et une mauvaise définition du cahier des charges.

La solution qui a été définie avec le mandant est de changer les curseurs des formulaires en listes déroulantes, de fixer l'âge maximum à 18 ans et de pouvoir rechercher sans prendre l'âge en compte dans la recherche.

Modifications apportées

On va donc dans un premier temps apporter les modifications au niveau de l'HTML en remplaçant les curseurs par des listes déroulantes. Les choses à prendre en compte lors de cette modification sont de ne pas oublier de mettre les bons attributs sur la balise « div »²⁸ afin que cela s'adapte en vue mobile ou desktop et aussi sur la balise « select »²⁹ afin de récupérer la valeur choisie lors de la recherche. Pour ce qui est d'étendre l'âge à 18 ans du côté de la liste déroulante, il suffit d'ajouter les balises « option » nécessaires et sans oublier d'ajouter également une balise avec comme valeur -1 afin de ne pas prendre en compte l'âge si on ne veut pas qu'il soit spécifié.³⁰

Au niveau du jQuery, la seule modification à apporter est sur la page du formulaire d'ajout d'une activité. On va modifier la valeur de la constante qui contient l'âge maximum pour une activité afin de permettre d'aller jusqu'à l'âge maximum de participation à une activité, c'est-à-dire 18 ans.

```
</div>
<div id="formulaireAdapte" class="input_field">
  <select id="age" name="age" placeholder="Age" class="modernSelect">
    <option value="-1" selected="selected">Age</option>
    <option value="4">4 ans</option>
    <option value="5">5 ans</option>
    <option value="6">6 ans</option>
    <option value="7">7 ans</option>
    <option value="8">8 ans</option>
    <option value="9">9 ans</option>
    <option value="10">10 ans</option>
    <option value="11">11 ans</option>
    <option value="12">12 ans</option>
    <option value="13">13 ans</option>
    <option value="14">14 ans</option>
    <option value="15">15 ans</option>
    <option value="16">16 ans</option>
    <option value="17">17 ans</option>
    <option value="18">18 ans</option>
  </select>
</div>
<div id="formulaireAdapte" class="input_field">
```

Figure 25 - Code HTML de la liste déroulante des âges

²⁸ Référence au rectangle orange sur l'image

²⁹ Référence au rectangle turquoise sur l'image

³⁰ Référence aux rectangles verts sur l'image

4.8 Affichage des détails des favoris en vue mobile

Contrairement à la vue desktop, on peut voir uniquement le nom de l'activité que l'on a mis en favori lorsque l'utilisateur va sur « ma sélection » en vue mobile. Il est donc nécessaire d'avoir les mêmes informations et actions que sur la vue desktop.

Ce problème concerne le côté fonctionnel de la plateforme.

Tout d'abord, on va reprendre le même affichage que sur les pages de recherche et on va adapter cela en affichant seulement les informations qui sont affichées aussi sur la vue desktop. On va remplacer l'étoile qui était présente sur la page de recherche par une croix afin de permettre à l'utilisateur d'enlever une activité de ses favoris. Une fois la vue opérationnelle, on va s'occuper de la gestion de l'évènement.

Actuellement, les détails de l'activité sont cachés de l'utilisateur même lorsqu'il clique sur l'activité. A l'aide de jQuery et de l'évènement « click() », on va pouvoir afficher ou cacher les informations d'un des favoris. Lorsque l'utilisateur clique sur l'un des favoris, on va vérifier si les détails de celui-ci sont affichés ou non.³¹ S'ils sont cachés, on affiche les détails³² et s'ils sont affichés, on les cache³³.

```
//affiche ou cache la description de l'activité en favoris
$('#top_selection_onglet_tab').on('click', function(evt){
    if($(this).parent().children().last().css('display') == 'none'){
        $(this).parent().children().last().css('display', 'block');
    }else{
        $(this).parent().children().last().css('display', 'none');
    }
});
```

Figure 26 - Gestion de l'affichage des détails des activités en favori sur mobile

4.9 Correction de l'impression des favoris

Le critère ergonomique soulevé à ce point est celui de la lisibilité lors de l'impression des favoris d'un utilisateur.

Le problème a seulement lieu lorsque l'on se trouve sur une page qui affiche les détails d'une activité ou d'un poste de moniteur. En effet, lorsque l'utilisateur souhaite imprimer ses favoris, dans la prévisualisation de son impression apparait la liste de ses favoris en plus du fond de la page qui contient les détails de l'élément qu'a consulté celui-ci.

³¹ Référence au rectangle jaune sur l'image

³² Référence à la flèche orange sur l'image

³³ Référence à la flèche verte sur l'image

Modifications apportées

On va résoudre ce problème à l'aide de jQuery. Tout d'abord, on va déplacer l'évènement « `click()` » qui a été ajouté au niveau de l'HTML et le mettre dans le fichier JS afin de pouvoir utiliser les propriétés de jQuery. Une fois que l'utilisateur aura cliqué sur le bouton d'impression de ses favoris, on va tout d'abord cacher le contenu de la page afin qu'il n'apparaisse pas sur l'impression.³⁴ Ensuite, on peut lancer la prévisualisation de l'impression de la page qui ne contiendra que la liste des favoris.³⁵ Une fois le travail d'impression terminé, on va afficher à nouveau le contenu de la page pour l'utilisateur afin de reprendre la navigation sur la plateforme au même endroit qu'il l'avait laissé.³⁶

```
$("#print").on('click', function(evt){  
→ $("#content").css('display', 'none');  
→ window.print();  
→ $("#content").css('display', 'block');  
  return false;  
});
```

Figure 27 - Evènement de gestion de l'impression

4.10 Mise en place d'un tableau de bord pour l'administrateur

Le problème abordé ici est dû à la gestion du projet et une mauvaise évaluation du cahier des charges.

Au vu du grand nombre d'organismes qui seront présents sur le site, l'administrateur de la plateforme m'a fait part du besoin de tenir au courant ces organismes afin qu'ils mettent à jour leurs activités et le nombre de places disponibles pour chacune d'elles. Il a été décidé de mettre en place un tableau de bord qui va permettre à l'administrateur d'envoyer des mails en un clic à un grand nombre d'organismes en fonction du besoin.

Dans un premier temps, il a fallu créer et mettre en place une nouvelle page qui sera le tableau de bord. Cette page est accessible uniquement par un compte administrateur. Si on essaie d'y accéder sans en avoir les droits, on est directement redirigé sur la page de login pour les administrateurs.

³⁴ Référence à la flèche rouge sur l'image

³⁵ Référence à la flèche verte sur l'image

³⁶ Référence à la flèche jaune sur l'image

Ensuite, au niveau de l'interface, il y a des boutons avec une description du travail qui est effectué par chacun d'entre eux afin que l'administrateur sache lequel utiliser en fonction de son besoin. Une fois la vue opérationnelle, la logique derrière cela sera gérée à l'aide d'événements jQuery et de script PHP.

Chacun des boutons va réagir à l'événement « click() ». Lors du déclenchement de celui-ci, on va exécuter une requête HTTP afin de lancer un script PHP qui va se charger d'envoyer un mail aux organismes afin de les notifier d'un besoin au niveau de la plateforme.³⁷

Chacun des boutons va lancer un script différent qui ne change que sur le contenu qui sera présent dans le mail et dans la requête effectuée afin de récupérer les mails des organismes à notifier. Une fois la requête exécutée, l'administrateur est informé par une alerte du déroulement de l'opération.³⁸

```
$("#mailsPlacesVacances").on("click", function(evt) {
    desactiverBouton();
    $.ajax({
        method: "GET",
        url: "./view/script/mail_alerte_places_vacances.php",
        dataType: "script",
        success : function(){
            alert("Les mails ont bien été envoyé !");
        },
        error : function(){
            alert("Erreur !");
        }
    });
    activerBouton();
});
```

Figure 28 - Appel du script d'envoi de mail aux organismes

³⁷ Référence au rectangle orange sur l'image

³⁸ Référence au rectangle vert sur l'image

5. Future amélioration de la plateforme

La plateforme est maintenant dans un état opérationnel. Nous allons voir ensemble l'état actuel de celle-ci concernant les critères ergonomiques et voir certaines fonctionnalités pouvant venir à être implémentées dans le futur.

5.1 L'état actuel de la plateforme

En ce qui concerne la sobriété de l'application, elle a été nettement améliorée. C'était sûrement l'un des critères les plus importants à traiter. Le code a été hautement simplifié en enlevant le développement qui a été réalisé à double pour chacune des vues.

Pour ce qui est de la lisibilité, elle était la source de plusieurs problèmes au sein de la plateforme. Il sera beaucoup plus agréable d'utiliser cet outil dorénavant. La pagination des informations est un des éléments qui a franchement amélioré ce point.

L'utilisabilité était déjà très bien mise en place à la prise en main du projet, ce n'était donc pas la principale source de problème. Un utilisateur n'est à aucun moment perdu au sein de la plateforme.

Le critère de la rapidité était crucial pour la plateforme. Plus de 700 organismes sont enregistrés au sein de celle-ci. Cela signifie qu'il y a énormément de données à traiter, raison pour laquelle il est donc très important de prendre en compte ce point lors de futurs développements.

Les critères restants ont été généralement bien mis en place, donc très peu de modifications ont été réalisées les concernant.

5.2 Des futures fonctionnalités ?

Suite aux entretiens avec le mandant, de futures fonctionnalités peuvent être implémentées pour rendre la plateforme plus agréable à utiliser pour les utilisateurs et simplifier le travail des administrateurs.

L'administrateur du site aurait souhaité pouvoir réaliser des exportations au format CSV des organismes et des activités de la base. Cela pourrait venir s'ajouter à son tableau de bord.

L'administrateur possède une page avec tous les organismes. Malheureusement, aucune méthode de tri n'a été mise en place, c'est pourquoi il serait utile de l'implémenter vu le grand nombre d'organismes.

6. Conclusion

Ce mémoire pourra aider des personnes dans de futurs projets à directement implémenter de manière correcte des critères ergonomiques à leur plateforme et ainsi mieux gérer leur projet.

La plateforme, dont la mise en production remonte au début septembre 2017, est actuellement disponible en ligne à l'adresse « www.loisirsjeunes.ch ».

Lors de la réalisation de ce travail, j'ai principalement amélioré l'ergonomie et les fonctionnalités de la plateforme, car à ma prise en main, elle n'était pas en état d'être mise en production. La plupart des fonctionnalités ne marchaient pas correctement. Une grande partie du développement a été doublée pour répondre à la vue mobile et desktop.

Grâce à ce mémoire, j'ai pu mettre en pratique les compétences que j'ai acquises au long de cette formation. Tout d'abord, par la réalisation d'une recherche d'un sujet en particulier. J'ai dû trouver des informations, les analyser ainsi que les comparer avec d'autres afin de pouvoir les exploiter par la suite. Ensuite, par la gestion de l'humain. En effet, j'ai dû rencontré le mandant qui sera en charge de la plateforme. J'ai discuté avec lui des améliorations à apporter à la plateforme et dû m'adapter aussi à de nouvelles demandes qui ont surgi durant le déroulement du projet. J'ai aussi rencontré les personnes en charge du déploiement du projet afin de leur faire part de mon avancement et de mes impressions. Pour terminer, toutes les compétences acquises dans le domaine du web. Le fait de reprendre un projet en cours peut demander un temps conséquent à la compréhension de celui-ci. Le fait que la plateforme n'était pas totalement opérationnelle, m'a fait passer énormément de temps dans de la lecture de code afin de remonter à la source des problèmes.

J'espère avoir pu proposer une plateforme opérationnelle et que les améliorations répondent aux besoins du mandant.

La réalisation de ce mémoire a été une expérience enrichissante et positive pour mon futur parcours professionnel.

Bibliographie

CommentCaMarche - *Ergonomie d'un site web*, [Consulté le 2017-09-05] Disponible à l'adresse : <http://www.commentcamarche.net/contents/1255-ergonomie-d-un-site-web>

InfoWebMaster - *Ergonomie web*, [Consulté le 2017-09-05] Disponible à l'adresse : <http://glossaire.infowebmaster.fr/ergonomie-web/>

Wikipédia - Site web adaptatif, [Consulté le 2017-09-05] Disponible à l'adresse : https://fr.wikipedia.org/wiki/Site_web_adaptatif

InfoWebMaster - Améliorer l'efficacité d'un formulaire de recherche, [Consulté le 2017-09-11] Disponible à l'adresse : http://www.infowebmaster.fr/131_news-efficacite-moteur-recherche-interne.html

Scala Dream Team – 7 Conseils pour réaliser son site en responsive design, [Consulté le 2017-09-11] Disponible à l'adresse : <http://www.groupe-scala.com/7-conseils-pour-realiser-son-site-en-responsive-design/>

Adimeo CEO – 9 Conseils pour améliorer l'ergonomie de votre moteur de recherche, [Consulté le 2017-09-11] Disponible à l'adresse : <http://blog.adimeo.com/9-conseils-pour-ameliorer-ergonomie-de-votre-moteur-de-recherche>

Tony Archambeau – (Etude de cas) Ergonomie des moteurs de recherche interne, [Consulté le 2017-09-11] Disponible à l'adresse : <http://tonyarchambeau.com/blog/276-analyse-moteur-recherche-interne/>

Charles Henri – 10 Conseils d'ergonomie basés sur des études de recherche, [Consulté le 2017-09-11] Disponible à l'adresse : <https://www.kryzalid.net/journal-expert-conception-web/10-conseils-d-ergonomie-bases-sur-des-etudes-de-recherche>

Olivier Sauvage Wexperience – Améliorez votre moteur de recherche, [Consulté le 2017-09-11] Disponible à l'adresse : <http://www.capitaine-commerce.com/2014/05/13/38066-ameliorer-votre-moteur-de-recherche/>

Sebastien – 10 Règles d'utilisabilité pour concevoir votre moteur de recherche, [Consulté le 2017-09-11] Disponible à l'adresse : <https://www.testapic.com/informations-pratiques/actualites/best-practices/10-regles-utilisabilite-concevoir-moteur-de-recherche/>

Usabilis – Vitesse de chargement des pages web : un critère ergonomique, [Consulté le 2017-09-11] Disponible à l'adresse : <http://www.usabilis.com/vitesse-chargeement-pages-web-critere-ergonomique/>

Alexandre Legge – Optimisez l'ergonomie de votre site, [Consulté le 2017-09-11] Disponible à l'adresse : <https://business.trustedshops.fr/blog/optimizez-ergonomie-site/>