FICHE technique épreuve E6

To do list

Dans le cadre de mon stage de première année de BTS SIO, j'ai pu participer au développement d'une application Mobile, l'application développé par Oxycar. Cette dernière a pour but de proposer un système de mis en place de covoiturage entre les employés d'une même entreprise. J'ai donc intégré une équipe de développement. Tous les membre de cette équipe travail sur des parties différentes du même projets. Cependant ils sont tous dépendant les uns des autres, car c'est un travail collaboratif. Pour réussir à avancer tous ensemble ils font des réunions chaque semaine pour savoir où est chacun, car si l'un n'avance pas les autres vont se retrouver bloqué. Lors de ces assemblées ils se mettent d'accord sur les objectifs à accomplir pour la semaine suivante. Afin de faciliter ce travail de groupe il est important de connaitre le sujet de sa propre mission et de pouvoir voir quand une personne à finit le travail dont vous avez besoin. Alors pour faciliter cela j'ai eu pour mission lors de mon stage de créer une application web qui doit afficher les missions de chacun, et pouvoir voir si elles sont validées, en cours ou pas commencées.

Cette application n'a pas pour but d'être utilisé pour surveiller les employés. Son objectif est d'aider le travail collaboratif, ainsi si une personne ne veut pas se servir de l'application pour quelconque raison il n'est pas obligé.

Les points clés :

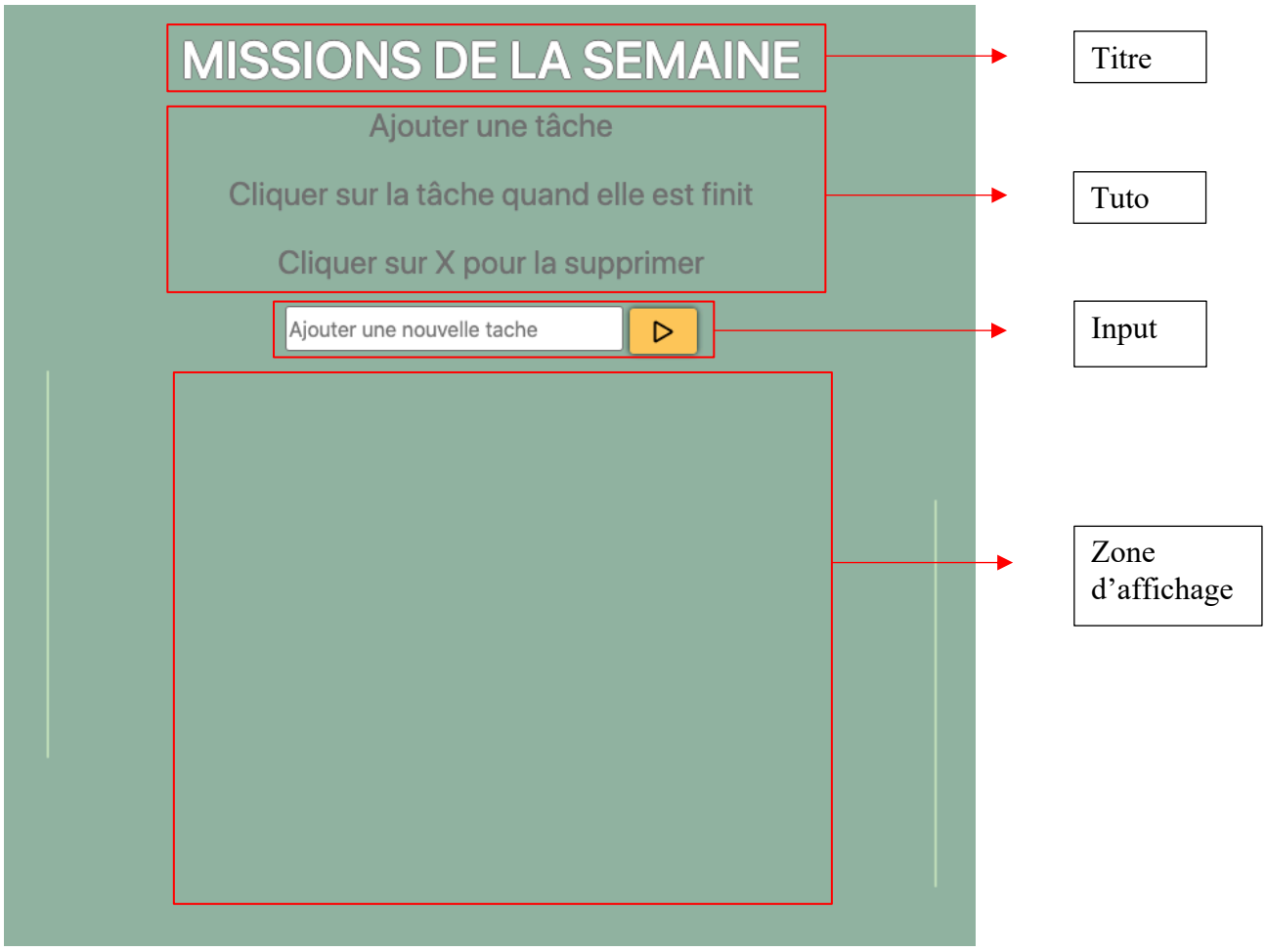
L'utilisateur doit pourvoir afficher ses missions, les valider et les supprimer. Aucune contrainte de temps lui est imposé dans la semaine. Aussi cette application est une page web qui n'utilisera pas de base de données car c'est un concept expérimental. Si le test « grandeur nature » est bénéfique et validée alors un développement d'une application plus conséquente pourra commencer.

Le sujet étant assez cours il n'y aura que 2 grandes parties, une concernant l'interface de l'application et l'autre au sujet de la logique du fonctionnement. Ce projet a été réalisé avec 3 langages de programmations, HTML 5, CSS 3 et JavaScript.

Une fois terminée et validée par mon maitre de stage, l'application pourra être mis en test.

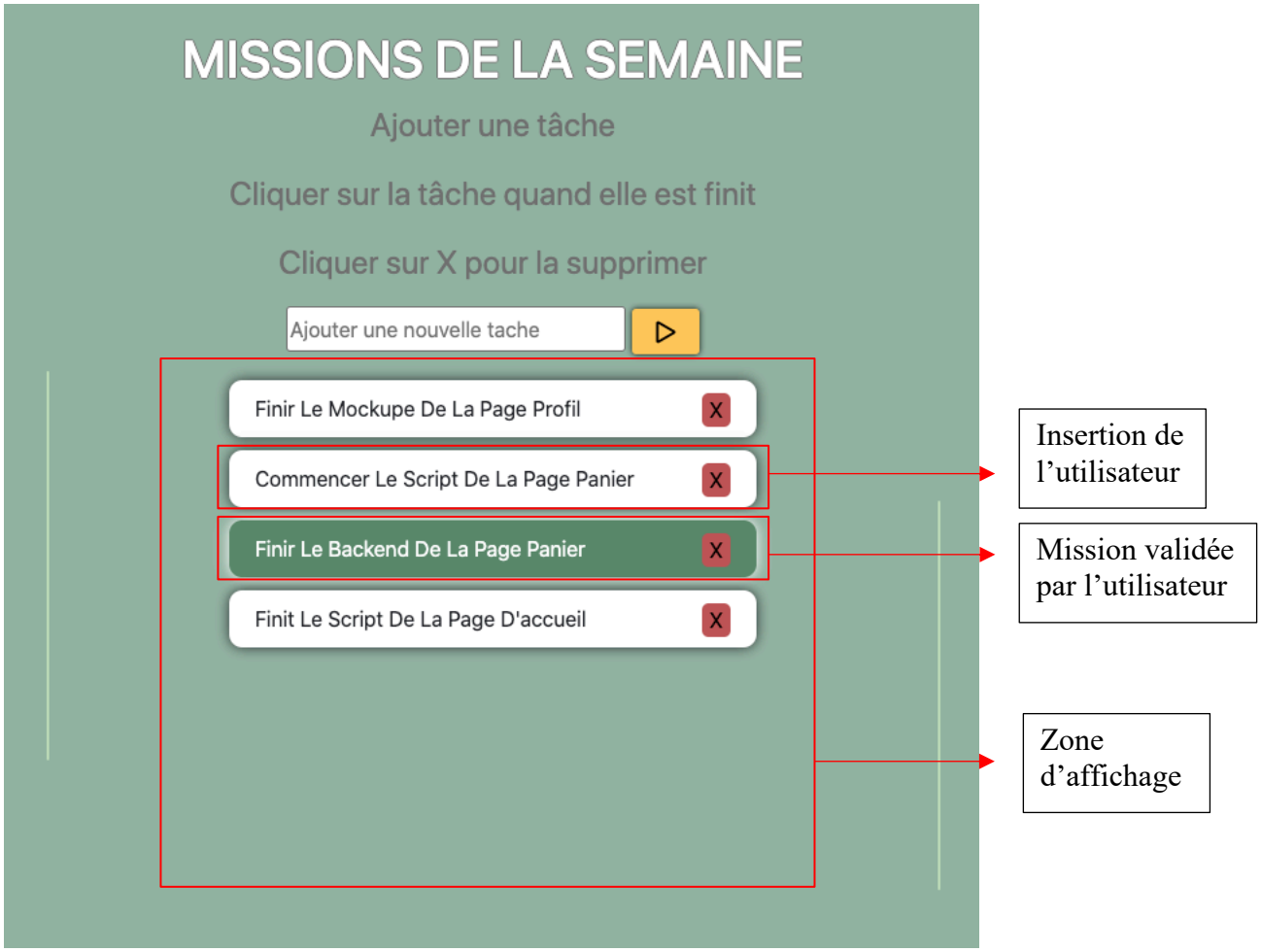
1. L'interface

L'interface doit être simple, rapide à utiliser et lire. Aussi étant donné que c'est une phase expérimentale il peut utile d'afficher un cours tuto sur l'interface. Le visuel de base est conforme à celui-ci-dessous.



Comme indiqué ci-dessus, un espace est prévu pour l'affiche. Ainsi cette zone affichera ce que les utilisateurs rentrent dans « l'input ».

Une fois cette action réalisée, le page sera affiché comme ci-dessous.



Il est donc possible de voir les missions à réaliser et celles effectuées. Un utilisateur peut choisir de laisser la mission qu'il a fini afficher dans l'attente d'une validation par exemple, il pourra la supprimer avec la petite croix par la suite.

2. Logique derrière l'application

Dans cette partie nous allons nous intéresser au script JavaScript par le biais de screenshot du code.

Commençons par déclarer les variables.

```
var enterButton = document.getElementById("enter");
var inputTxt = document.getElementById("userInputTxt");
var ul = document.querySelector("ul");
var item = document.getElementsByTagName("li");
```

- « enterButton », récupère le bouton avec l'id « enter » qui permet d'ajouter une nouvelle tâche.
- « inputTxt », récupère le texte entré dans le input qui a comme id « userInputTxt ».
- « ul », contient ce qui se que renferme l'élément « ul ».
- « item », contient ce qui se trouve dans l'élément « li » de la page html.

Les fonctions :

La première fonction permet de retourner la taille de la valeur de la chaine de caractère rentré par l'utilisateur, illustré si après.

```
function inputLength() {
    return inputTxt.value.length;
}
```

La fonction « createListElement » suivante est la fonction principale du programme, faisant plusieurs lignes, je vais découper les explications par bloc.

```
function createListElement() {

    //
    var li = document.createElement("li");
    //ajoute la valeur écrite
    li.appendChild(document.createTextNode(inputTxt.value));
    //ajoute à la liste
    ul.appendChild(li);
    //reset l'inputTxt
    inputTxt.value = "";
```

La première partie permet de créer une balise « li » qui contient le texte rentré par l'utilisateur, et de placer le tout dans la balise « ul » afin de créer une liste d'élément.
La dernière ligne permet de réinitialiser la champ texte sur l'interface.

```
//ajoute la classe quand...
function crossOut() {

    li.classList.toggle("done");
}

//... on clique sur une ligne
li.addEventListener("click", crossOut);
```

Dans la suite de la fonction, une fonction crossOut qui permet de colorer la ligne. Cela signifie que la mission est validée.
« classList » renvoie une chaine composée de la liste des classes.
« Toggle » fonctionne comme un interrupteur.
« done » renvoie à la classe css qui a pour rôle de colorer la couleur du fond de l'élément en vert.
La dernière ligne lance la fonction « crossOut » quand l'utilisateur clique sur un élément « li ».

La dernière partie de la fonction « createListElement » créé un bouton sur le nouvel élément. Ce dernier permet de supprimer la mission.

```
//Créé un bouton dans le li
var dBtn = document.createElement("button");
dBtn.appendChild(document.createTextNode("X"));
li.appendChild(dBtn);
//quand on clique on lance la fonction delete
dBtn.addEventListener("click", deleteListItem);

//quand on clique sur la croix on donne la classe css "delete"
function deleteListItem() {
    li.classList.add("delete");
}
```

Commence pa créer une variable « dBTN » qui contient un nouvel élément « bouton ». On lui ajoute un texte « X » pour la symbolique. Puis la variable est ajoutée à « li ».
« dBTN » lancera la fonction « deleteListItem » lorsque l'utilisateur clique sur le bouton.
La fonction « deleteListItem » donne la classe css « delete » à la ligne « li » sur laquelle l'utilisateur aura cliqué sur la croix.

Les deux dernières fonctions du script JavaScript permettent chacune de lancer la fonction principale du script.

La première le permet, lorsque l'utilisateur clique sur le bouton avec l'id « enter »,

```
function addListAfterClick() {
//si y'a qqch dans l'input on créé
    if (inputLength() > 0) {
        createListElement();
    }
}
enterButton.addEventListener("click", addListAfterClick);
```

- Le « if » permet de vérifier le champ que l'utilisateur doit remplir n'est pas vide.
- La dernière ligne met le bouton avec l'id « enter » en « écoute », il lancera la fonction ci-dessus quand on clique sur lui.

```
function addListAfterKeypress(event) {
    if (inputLength() > 0 && event.which === 13) {
        createListElement();
    }
}
inputTxt.addEventListener("keypress", addListAfterKeypress);
```

La deuxième fonction a la même utilité que la première, c'est-à-dire lancer la fonction principale. Mais la manière est différente, dans cette situation l'utilisateur appuie sur le bouton entrer pour déclencher la fonction principale.

- La balise « if » vérifie que l'utilisateur a écrit dans le champ et qu'il appuie sur la touche entrer. Si les deux conditions sont validées alors la fonction « creaListElement » est lancé.

Conclusion

L'application mis à disposition sera un nouvel outil d'organisation pour les employés. Il sera possible voir clairement quand une mission est finie sans devoir demander.
De grosses améliorations sont à prévoir pour rendre l'application plus complète. Nous pouvons penser à l'affiche des délais de chaque mission, à l'affichage des missions par employé… Cependant pour continuer le projet, l'application doit trouver une utilisation concrète dans l'entreprise. La suite du développement se fera selon les retours des utilisateurs.