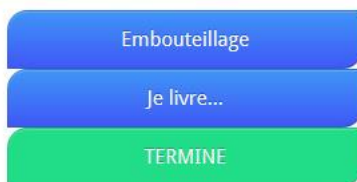


Langage d'entrée/sortie – Syntaxe

Dans le cadre d'un logiciel qui peut facilement **s'adapter au support** de son utilisateur nomade, nous avons choisi de faire une **application web** pour le **livreur**. Elle n'est pas seulement web, elle est également « **responsive design** ».

En effet, les livreurs affiliés au programme « **Optimod'Lyon** » ne sont pas tous équipés de la même technologie. Certains utilisent probablement leurs **smartphones**, d'autres leurs **tablettes**, ou encore des **ordinateurs** de bords. Il suffit d'une connexion internet (3G, 4G, Edge ou encore satellite) et d'un navigateur web.



Notre interface adapte ainsi sa forme sans léser le contenu. Elle a fait le sujet d'une double réflexion :

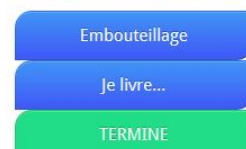
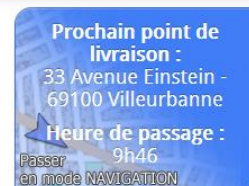
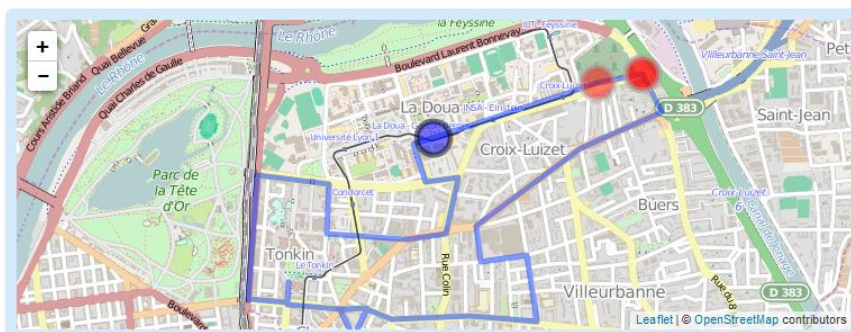
- À quoi ressemble l'interface sur **un grand écran** ?
- À quoi ressemble l'interface sur **un petit écran** ?

Pour mettre en œuvre un **design** dit « **responsive** » ou « **adaptable** », le framework « **Bootstrap** » de **Twitter** nous donne tous les outils pour concevoir et réaliser une **interface adaptable**. Elle fonctionne sur le principe de classes **CSS** et de code **JavaScript** à inclure et à utiliser selon certaines règles.

La syntaxe de notre langage est en fin de compte celui de l'**HTML** (sa DTD), nous avons **réalisé** en partie cette application web.

La spécification des **fenêtres** et les **objets** utilisés seront décrits dans le lexique du langage.

La réalisation est **accessible** à cette adresse (code source accessible) : www.spownik.net/devIHM



Langage d'entrée/sortie – Lexique

Notre application doit donc être en « responsive design ». Nous sommes partis d'une page conventionnelle en HTML pour en définir le design de base

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>OptimodLyon - Livreur Mode</title>
</head>
<body >
  <div class="navbar navbar-fixed-top">
    <div class="navbar-inner">
      <div class="container-fluid">
        <a class="brand" href="index.php">OL</a>
        <div class="nav-collapse collapse">
          <ul class="nav">
          </ul>
        </div><!--/.nav-collapse -->
      </div>
    </div>
  </div>
  <div class="container-fluid" >
    <div class="row-fluid" >
      <div class="span10" >
        <div class="hero-unit">
        </div>
        <div class="span3" >
        </div><!--/span-->
      </div><!--/row-->
    </div><!--/.fluid-container-->
  </div>
</body>
</html>
```

Nous avons mis en place les conteneurs « container-fluid » qui est la classe de **Bootstrap** permettant de faire du design un **design responsive** et donc « **flexible** », il est flexible sur une ligne (row-fluid). L'intérieur se décompose en **deux zones** (div) :

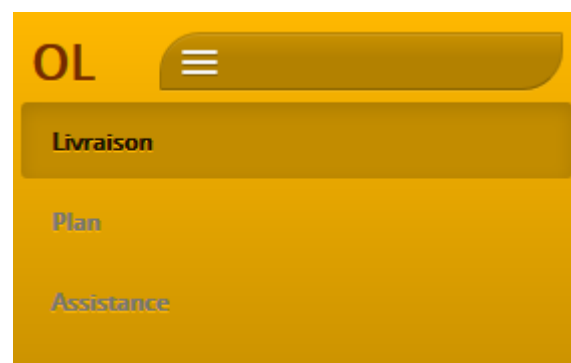
- **Hero Unit** (la zone d'affichage de taille **7 sur 10**, c'est la plus large réservée à la **Carte/GPS**)
- Un **span de taille 3 sur 10** présentant les **actions** possibles et les **informations** contextuelles

Ces deux zones sont arrangées de manière à **s'adapter** selon la largeur de l'écran. En utilisant correctement les classes de **Bootstrap**, les tailles des boutons et des objets HTML se redimensionnent en conséquence.



Ainsi nous avons une page qui **se lit sur la largeur** sur une tablette et un ordinateur, tant qu'elle **se lit en hauteur** sur un mobile.

De même que pour la barre de navigation (en-tête du site), nous avons utilisé les outils de **Bootstrap** et sa classe « **nav-bar** » et « **nav-collapse** » qui permettent une barre de navigation flexible, mais également compacte si besoin. Ainsi le menu du site web se compresse en un menu déroulant lorsque la largeur de l'écran est trop petite pour l'afficher entièrement.



La carte s'est intégrée parfaitement avec le framework « Leaflet » (OpenStreetMap), mais ce n'est pas le cas avec la vue GPS, faite grâce à l'application Google StreetView. En effet, elle est réalisée avec une balise <iframe>, qui ne se redimensionne pas par défaut et demande une méthode d'aménagement spécifique pour qu'elle puisse le faire. Voici les règles CSS utilisées pour contourner le problème de redimensionnement des iframes :

```
.iframe-responsive-wrapper
{
    position: relative;
}

.iframe-responsive-wrapper .iframe-ratio
{
    display: block;
    width: 100%;
    height: 350px;
}

.iframe-responsive-wrapper iframe
{
    position: relative;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
}
```



Pour ce qui est du tableau de bord lors du démarrage de l'application, les utilisateurs peuvent apercevoir la tournée complète avec les points de livraison ainsi que leur position GPS (au centre de l'écran).

En cliquant sur le bouton « **Prochain point de livraison ...** », l'utilisateur entre en mode GPS, où ce dernier guide le livreur jusqu'au prochain point.

Il peut également cliquer sur le bouton « **Embouteillage** » pour signaler au superviseur qui le suit qu'il est bloqué sur la route.

C'est une fois arrivé chez un client que le livreur clique sur « **Je livre...** », et entre en phase de réception des colis par le client.

En cliquant sur « **TERMINE** », le livreur prévient le superviseur que la tournée est terminée et lui confirme la bonne réception des informations quant à la tournée effectuée.

Objets présents : **Client, Itinéraire, Plages Horaires, Graphe, Points, Tronçons**

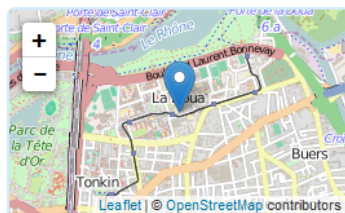
Le mode « GPS » guide l'utilisateur jusqu'au prochain point de livraison. Il indique l'heure de passage estimé par le GPS.

Il y a la possibilité de revenir au tableau de bord (accueil) de l'application en cliquant sur « **Retour livraison...** »

La zone de GPS est plus grande pour une meilleure lisibilité.

L'avantage de ce mode est qu'il lance la procédure de réception du colis dès que le point GPS coïncide avec le point de livraison.

Objets présents : **Client, Tronçons, Points, Plages Horaires**



Bloqué ici :
10 rue de la Physique - 69100
Villeurbanne

**Temps écoulé dans
l'embouteillage :**
0 min 3 sec

Embouteillage terminé

Assistance

Si le livreur se retrouve coincé dans un embouteillage, il doit alors avertir son superviseur en cliquant sur « **Embouteillage** » dans le tableau de bord. Depuis cette interface, il peut observer l'endroit où il est bloqué et le temps depuis lequel il est bloqué dans l'embouteillage.

Il peut également contacter le superviseur en cliquant sur « **Assistance** », ouvrant la page de contact pour pouvoir dialoguer avec ce dernier.

Objets présents : **Client, Points, Graphe.**



Votre superviseur :
Simon GAZON

Envoyer un mail

Appeler superviseur...

Retour Livraison

En cliquant sur un bouton « **Assistance** » sur une page ou dans le menu de l'application, le livreur pourra voir une page de contact où sont réunies les informations du superviseur associé à la tournée.

Il peut lui **envoyer un mail** ou encore **l'appeler** via une application Skype ou un système de téléphonie Voice IP. Il suffit de cliquer sur l'un des deux boutons « **Envoyer un mail** » ou « **Appeler superviseur...** » pour lancer une application externe.

Objets présents : **Client, Superviseur, Graphe, Point**



Vous êtes au
33 avenue Albert Einstein 69100
Villeurbanne

Réception
Aucun problème n'est survenu

Client absent

Colis refusé/Litige

Inaccessible

Autres...

Retour livraison

Une fois le livreur arrivé à un **point de livraison**. Il doit alors, sur le tableau de bord, cliquer sur « **Je livre...** », le faisant entrer dans la procédure de **réception du colis par le client**.

Il y retrouve **l'adresse de livraison**, la carte avec **son emplacement**, mais également (non affiché sur la maquette) les **coordonnées du Client** (nom, prénom, numéro de téléphone).

Depuis cette interface, il peut **signaler si un litige** est survenu lors de l'accès au client. S'il n'a pas pu accéder au point de livraison, ou bien si le client était absent, etc. Si une raison particulière doit être précisée, il peut la renseigner en cliquant sur « **Autres...** » où une boîte de dialogue lui demandera de **saisir cette raison**.

Toutes ces raisons, une fois cliquées, **demandent une confirmation** de la part du livreur, évitant ainsi de cliquer par mégarde sur un bouton non souhaité.

Le bouton « **Réception** » fait passer à l'étape suivante : le livreur est donc face au client, près à réceptionner le colis dans les meilleures conditions.

Dans une nouvelle interface, **l'adresse de facturation** se doit d'être confirmée par le client. Si un litige se présente au livreur, il doit le renseigner en cliquant sur **le bouton adéquat**, ou préciser une autre raison comme décrite précédemment.

Encore une fois, si tout se passe bien, le livreur peut passer à la signature électronique. Ici, on demande au client de dessiner, avec **son doigt ou un stylet**, une belle signature attestant de la bonne réception du colis et du paiement de la facture. Le livreur, si un problème survient ou si la signature ne convient pas, dispose d'un bouton « **Effacer** », réinitialisant la zone de signature à un bloc vide.

La nature du problème ?

OK Annuler

Réception
Aucun problème n'est survenu

Client absent

L'adresse de facturation
33 Avenue Einstein - 69100
Villeurbanne

Signature
Colis livré avec succès

Colis refusé/Litige

Autres...

Retour

Valider
Terminer la livraison

Effacer

Retour



Si tout s'est bien passé, le livreur valide la transaction en cliquant sur « **Valider** », l'application **retourne au tableau de bord** et la tournée peut continuer.

Au cours de cette procédure, le livreur peut revenir à l'étape précédente à tout moment en cliquant sur un bouton « **Retour** »