

Adaptation des Interfaces

Membres de l'équipe

Anthony Loroscio (anthonyloroscio@gmail.com)

Alicia Marin (amarin@polytech.unice.fr)

Thomas Monzein-Stocky (thomas.monzein@gmail.com)

Thibaut Terris (thibaut.terris@gmail.com)

Technologies

- **Responsive Web Design** : Bootstrap (+ Angular) [Thibaut]
- **Web-Component** : VueJS [Alicia]
- **Cross-Platform** : Ionic [Thomas]
- **Native** : Swift [Anthony]

Description de l'application

L'application qui va être conçue est à destination des utilisateurs de bus. Elle a pour but de mettre en avant différents types d'informations relatifs au réseau de bus urbain ainsi qu'aux trajets quotidiens de l'utilisateur.

Fonctionnalités

L'application sera composée de plusieurs pages accessibles via des menus. Chaque page renverra vers une fonctionnalité précise de l'application :

- Afficher une carte
- Afficher la tables des horaires

En plus de cela, l'application permettra d'accéder à certaines fonctionnalités propres aux périphériques sur lequel elle est lancée :

- Ajouter un événement régulier à son calendrier personnel
- Être notifié avant que le bus arrive
- Bénéficier d'une sélection automatique du lieu de départ en fonction de la localisation

Tâches

Sur Web :

- Afficher la carte complète
- Afficher la table des horaires complète
- Suivre un bus en “live”
- Créer un événement régulier dans le calendrier

Sur Mobile :

- Afficher partiellement la carte en fonction des arrêts choisis
- Afficher partiellement la table des horaires en fonction de l’heure actuelle
- Géolocaliser l’utilisateur pour définir automatiquement la zone de départ
- Recevoir une notification 3 minutes avant que le bus arrive à l’arrêt de départ

Liens entre l’application et l’adaptation

Utilisateur : Voyageur commutant en bus consultant l’application chez lui, ou à l’extérieur.

À l’environnement : Le type de notification sera différent. Par exemple sur téléphone, une notification sera envoyé 3 minutes avant l’arrivée du bus. De plus, l’arrêt de départ sera automatiquement sélectionné sur téléphone grâce à la géolocalisation.

Au dispositif : L’affichage de l’application sera différente entre l’application web et mobile. En particulier, cela va affecter l’affichage de la carte et des horaires.

De plus, nous pourrons comparer l’utilisation des périphériques suivant le dispositif sur lequel tourne l’application.

Scénarios

Edward : Utilisateur régulier qui a sauvegardé ses préférences (lignes et heure)

Alphonse : Utilisateur ponctuel

Adaptation à l’utilisateur :

Consulter la carte

- Edward veut consulter la carte, les données seront affichées selon ses préférences (trajets favoris)
- Alphonse veut consulter la carte, les données sont affichées de manière globale

Adaptation à l’environnement :

Se renseigner sur l'itinéraire

- Edward est chez lui et veut connaître tous les trajets possibles (interface web).
- Lorsqu'Edward sera en déplacement, il consultera sur son appli mobile l'itinéraire (carte) qu'il avait choisi chez lui

Se renseigner sur les horaires

- Edward est chez lui ou à son travail, il veut obtenir des informations sur des lignes. Lorsqu'il passe sur mobile, il obtient les informations (horaires/arrêts) qu'il a choisi avant
- Edward est à l'arrêt de bus, lorsqu'il consultera l'application, l'arrêt de départ sera pré-selectionné

Être conscient du trafic

- Edward est sur chez lui ou à son travail, il sera notifié en cas de **retard ou avance** du bus dont il a l'habitude de prendre.
- Edward est à son arrêt de bus, il sera notifié en cas de **retard** du bus

Adaptation au dispositif :

Choisir son arrêt de départ

- Chez lui sur son ordinateur il se rend sur l'interface web et devra choisir son arrêt de départ et d'arrivé avec l'heure prévue.
- Au café d'en face sur son téléphone il se rend sur l'application, rentre les informations concernant le trajet qui l'intéresse. Il n'aura pas besoin d'entrer l'arrêt de départ, ni l'heure prévue, qui seront récupérés automatiquement de son téléphone.

Défi pour chacune des technologies :

Bootstrap (+Angular) : Évaluer le rendu sur grand écran ainsi que sur très petit écran

VueJS : Réutiliser les mêmes composants dans des contextes différents

Ionic : Utiliser l'accès aux capteurs

Swift : Comparer l'accès aux capteurs avec Ionic par exemple