

# Cahier des spécifications

## v0.1

**Projet PFA - 2015/2016**

**Application mobile pour rediriger les usagers lors  
des pannes de transport sur le réseau de  
Bordeaux-Metropole**



**qucit**

**Responsable Pédagogique**  
**Contact Client**

Toufik Ahmed  
Nicolas Bonnotte

ENSEIRB-MATMECA  
Qucit

**Réalisateurs**

Mehdi Bounakhla  
Romain Rambaud  
Nathan Reavaille  
Simo Rihani  
Pierre Roux  
Victor St-Guilhem

# Introduction

## Contexte

### Description initiale du projet par le client

Le projet consiste à développer une application mobile multi-plateforme qui permet de rediriger les usagers en cas de panne de transport sur le réseau de Bordeaux Métropole.

Une première étape consiste en la collection de données à partir de sources différents (infotbc, twitter, qucit, ...) afin d'avoir une vision globale de l'état du réseau de la ville. Il faudra ensuite détecter les pannes dans le réseau de transport de bordeaux métropole.

La deuxième étape consiste à coder une application mobile simple multi-plateforme (basée sur la plateforme polymer de Google) afin de permettre aux utilisateurs d'enregistrer leurs trajets habituels et de recevoir des notifications si ce trajet est affecté par une pannes du réseau. Les notifications permettent non seulement d'informer l'utilisateur de l'existence d'un problème, mais aussi de lui proposer un itinéraire alternatif pour son déplacement.

### État d'avancement

Cette première version du cahier des spécifications intervient après une première réunion avec l'équipe Qucit, où différents besoin ont été précisés et développés. De nouvelles réunions étant toutes les deux semaines, ce document va être amené à évoluer rapidement.

## Présentation du document

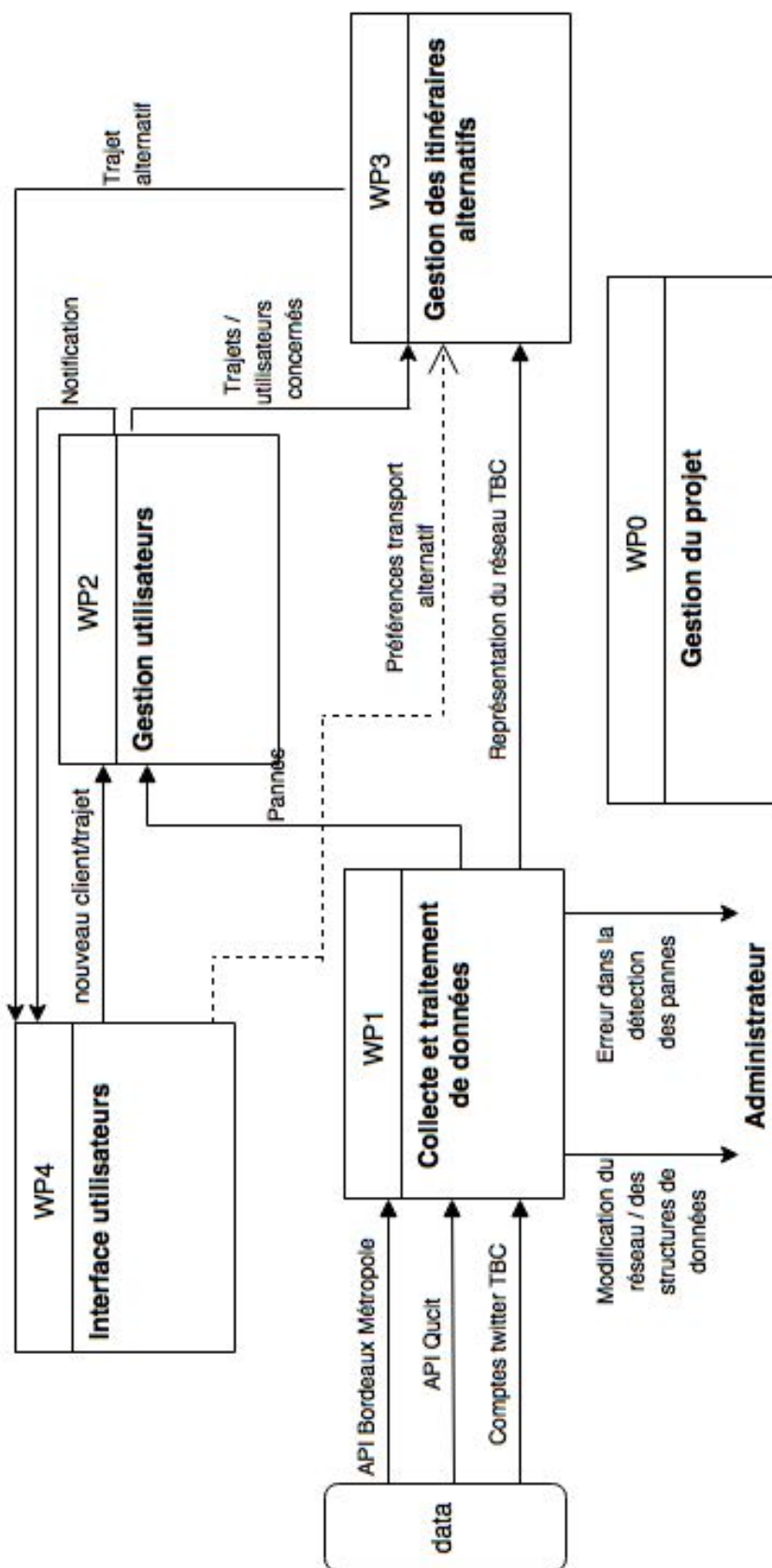
### Objet du document

L'objectif de ce dossier est de décrire l'aspect fonctionnel de l'application à produire, et ainsi de permettre aux réalisateurs de mieux cerner le besoin. Il fait donc la part belle à l'aspect visible de l'application, mais aussi après découpage logique, des fonctionnalités internes et des flux de données reliant les modules principaux identifiés.

### Guide de lecture

Il a été déterminer avec le responsable pédagogique que le travail à réaliser serait divisé en **Working Package** ou chaque Package correspondrait à un ensemble de fonctionnalités proches.

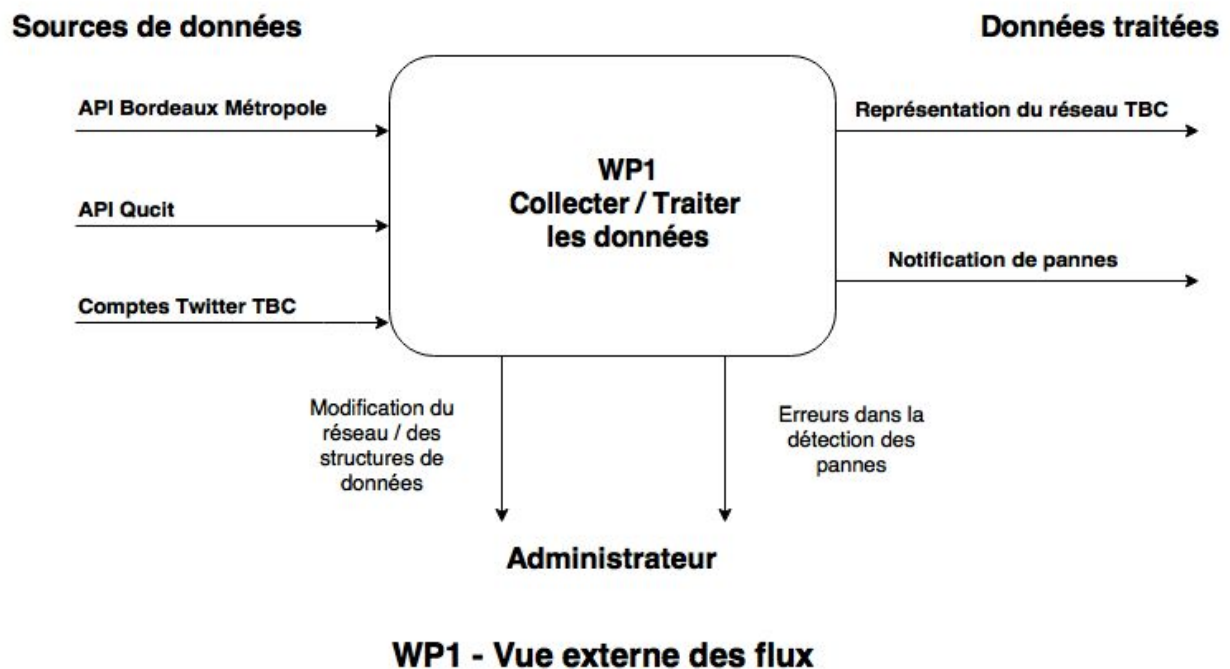
Chacun des quatre prochains chapitres présente donc un de ces Packages, dont les relations sont détaillées dans le digramme intitulé "Découpage du projet en Working Package", page suivante.



## Découpage du projet en Working Packages

# WP1 - Collecte et Traitement des données

Ce Working Package a pour but de fournir l'ensemble des modules permettant de collecter, traiter et stocker les données du réseau de transport TBC et des pannes.



## Cas d'utilisation du WP : flux entrants

### Collecte des données : Réseau de transport (API Data Metropole)

- ➡ En tant que serveur, je dois fournir une représentation à jour de l'ensemble du réseau de transport TBC.

## Critères

- Afin de modéliser le réseau TBC, il est nécessaire d'avoir accès aux données suivantes :
  - les arrêts : les types (TRAM/BUS/BATEAU), l'emplacement géographique, les lignes qui y passent
  - la constitution des lignes : les types (TRAM/BUS/BATEAU), leurs arrêts, les directions et sens possibles, les boucles, l'existence d'arrêts partiels, et différentes variations,
  - les horaires de chaque ligne, en différenciant les périodes spécifiques : journée, soir, nuit, weekend/jour férié, période scolaire
- Les requêtes permettant d'obtenir ces données devront être faites automatiquement et périodiquement afin de toujours avoir une représentation la plus à jour du réseau de transport.

## Collecte des données : Pannes (Twitter)

- ➡ En tant que serveur, je dois détecter, traiter et stocker les pannes du réseau TBC.

## Critères

- Afin de modéliser les pannes du réseau, il est nécessaire d'avoir accès aux données suivantes :
  - les types de pannes possibles : modification programmée d'une ligne
  - la raison de chaque panne : pour travaux, événements
- Le traitement d'un message de panne se fait en reconnaissant la partie de réseau affectée (arrêt unique ou tronçons) et la durée de la panne si elle est donnée. Dans le meilleur des cas, cela permet donc d'obtenir des données spatio-temporelles qui vont affecter le graphe de modélisation du réseau..

## Collecte des données : Vélos (API Qucit)

- ➡ En tant que serveur, je dois fournir une représentation des stations de Vcub et de la disponibilité des vélos aux stations.

## Critères

- Afin de modéliser les stations de Vcub et la disponibilité des vélos aux stations, il est nécessaire d'avoir accès aux données suivantes :
  - les stations : l'emplacement géographique, le nombre de vélos disponibles et le nombre de places

## Détection de la non-conformité des données

- ➡ En tant que serveur, je dois détecter qu'un changement dans les structures de données des sources de données rend la modélisation du réseau et des pannes non conformes.
- ➡ En tant que serveur, je dois détecter que le traitement des messages de pannes est défectueux.

### Critères

- Lorsque le nombre d'erreurs de l'algorithme traitant les données du réseau TBC dépasse une valeur prédéfinie pendant un intervalle de temps, le serveur doit en avvertir l'administrateur pour que celui-ci effectue une correction des algorithmes de traitement
- Lorsqu'un nombre de message de pannes non détectés dépasse une valeur prédéfinie pendant un intervalle de temps, le serveur doit en avvertir l'administrateur pour que celui-ci effectue une correction des algorithmes de traitement des messages de pannes.

## Cas d'utilisation du WP : flux sortants

### Fournir les données complètes au module d'itinéraires alternatifs (WP2)

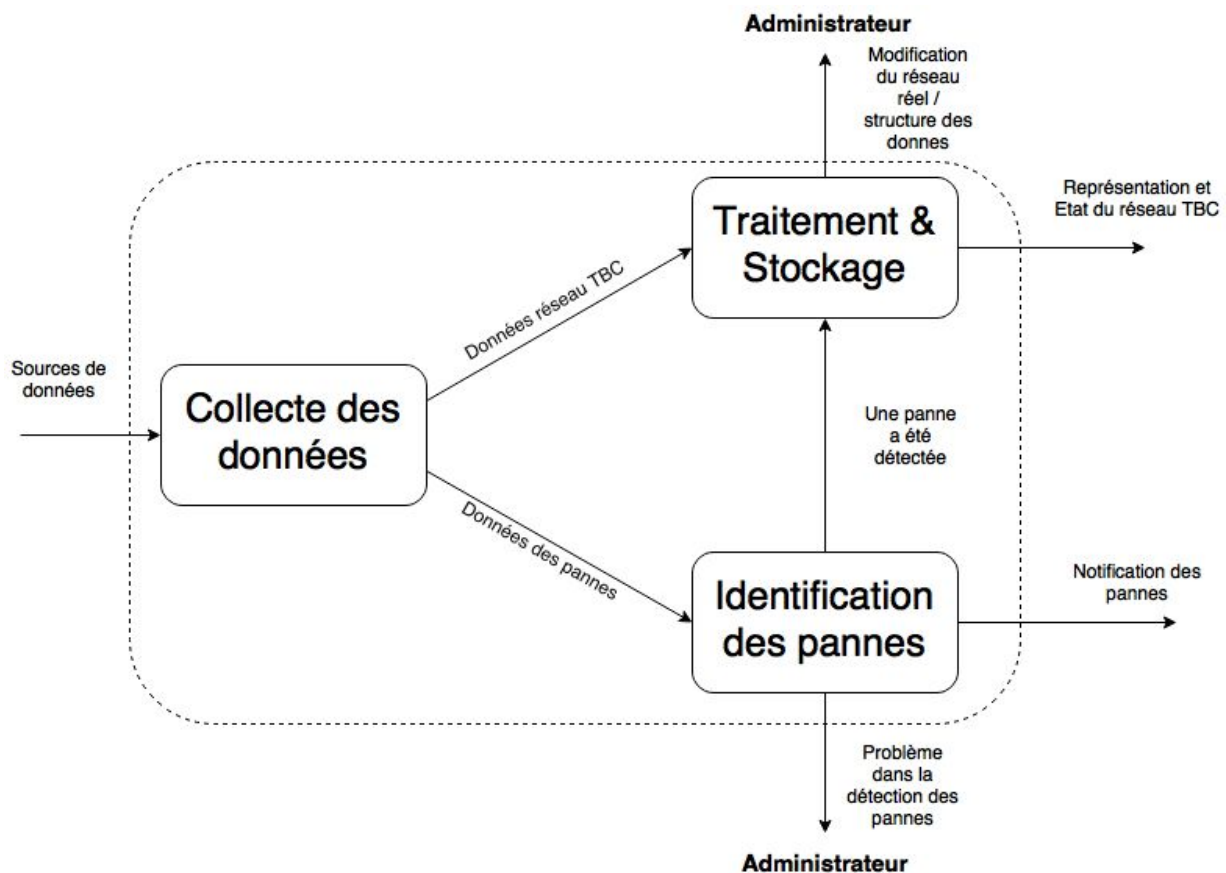
- ➡ En tant que serveur, je doit fournir, au module de recherche d'itinéraires alternatifs, toutes les données nécessaires à son algorithme.

### Critères

- Afin de réaliser l'application complète, les autres nécessaires ont besoin d'une d'accéder à un modèle de représentation du réseau TBC. Ce Working Package fait donc office d'API interne pour les autres modules de l'application.

### Notification des pannes à la Gestion des utilisateurs (WP3)

- ➡ En tant que serveur, je doit notifier le module de Gestion utilisateur de chaque panne détectée sur le réseau TBC.



### WP1 - Vue interne des flux

Le Working Package 1 est donc composé de 3 modules principaux :

- **Collecte des données** à partir de plusieurs sources : API Data Metropole, API QUCIT, et API Twitter. Cette collecte devra être périodique
- **Traitement et stockages**
- **Identification des pannes**

Actuellement nous avons volontairement fusionné la description des fonctionnalités et des critères des modules de Collecte des données avec celui de Traitement & Stockage. Lorsque le moment sera venu de détailler plus techniquement les interactions entre ces deux modules, nous le ferons.

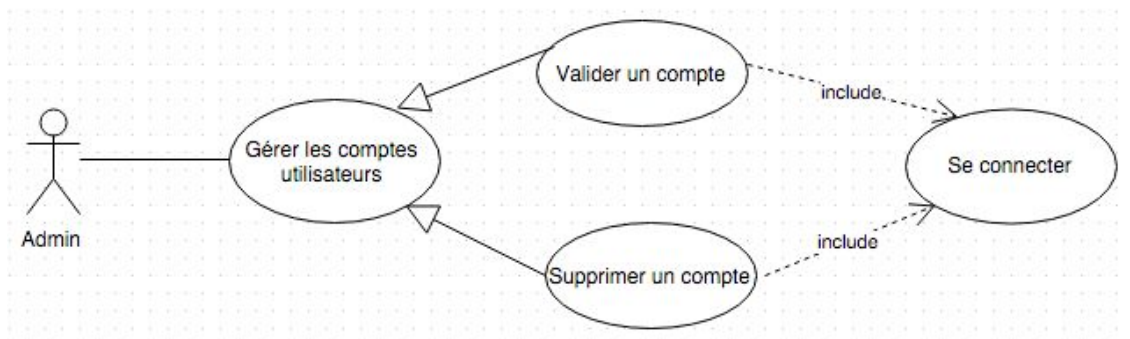
## WP2 - Gestion des utilisateurs

### Connexion et inscription

- ➡ En tant qu'administrateur, je souhaite que lorsqu'un utilisateur s'enregistre sur l'application, ses informations soient stockées sur le serveur.
- ➡ En tant qu'administrateur, je souhaite qu'un utilisateur puisse se connecter à l'application si il est enregistré.
- ➡ En tant qu'administrateur, je souhaite qu'un utilisateur enregistré puisse supprimer son compte de l'application.

### Critères

- Un utilisateur qui s'enregistre sur l'application doit avoir ses données stockées sur la base de données "User" de l'application.
- La base de données doit contenir une table "User" qui doit avoir les champs nécessaires pour stocker les informations demandées lors de l'inscription sur l'UI.
- Lors d'une inscription, si l'ensemble des informations saisies sont correctes, un mail de confirmation est envoyé à l'adresse renseignée lors de l'inscription.
- Dans le cas d'une tentative de connexion, le serveur devra vérifier que le login renseigné est présent dans la base de données, et que le mot de passe associé est correct. Si c'est le cas, la connexion est validée. Dans le cas contraire, le serveur renvoie un code d'erreur à l'UI.
- Dans le cas d'une demande de suppression de compte depuis l'UI, les informations relatives à l'utilisateur seront supprimées dans la base de données.
- Si, lors d'une nouvelle inscription, le mail renseigné par l'utilisateur est déjà présent dans la base de données, l'inscription n'est pas enregistrée en BDD et un message d'erreur est renvoyé à l'UI.
- Si une requête de nouveau mot de passe est envoyée, le système vérifier que l'adresse mail concernée est bien présente dans la table "User". Si c'est le cas, un mail sera envoyé contenant un lien de réinitialisation du mot de passe.
- Une fois que l'utilisateur s'est connecté sur l'application, les identifiants seront automatiquement enregistré sur le téléphone pour éviter à l'utilisateur de les taper à chaque lancement.





## Gestion des trajets

- ➔ En tant qu'administrateur, je souhaite que l'ensemble des trajets de chaque utilisateur soient enregistrés en BDD.
- ➔ En tant qu'administrateur, je souhaite communiquer les trajets concernés par la panne à l'algorithme de création d'itinéraires alternatifs.
- ➔ En tant qu'administrateur, je souhaite qu'une notification soit envoyée à l'utilisateur afin de le prévenir d'une panne.

## Critères

- La base de données doit contenir une table "Trajet" avec les champs nécessaires à l'enregistrement d'un nouveau trajet : heure de départ, heure d'arrivée, transport empruntés, arrêt de départ, arrêt d'arrivée. Il pourra indiquer des correspondances, et pour chacune d'entre elles, il sera demandé l'arrêt de départ d'arrivée et le transport emprunté.
- Lorsqu'un utilisateur renseigne un nouveau trajet, l'ensemble des informations qu'il aura rentré devront être enregistrées en base de données.
- Lorsque le module de collecte et de traitement de données communique l'information comme quoi une panne est signalée sur le réseau de transport, le module de gestion utilisateur doit vérifier l'ensemble de ses données "Trajet".
- Si un trajet est concerné par une panne reçue, une notification doit être envoyée sur l'interface utilisateur.
- Si un trajet est concerné par une panne reçue, le numéro du trajet avec les informations d'horaire, de point de départ et d'arrivée sont communiqués au module de gestion des itinéraires.
- Un trajet proposé devra être compatible avec le réseau de transport TBC (arrêt existant).
- En cas d'incompatibilité avec le réseau de transport TBC, le serveur renverra une erreur à l'interface utilisateur.

## Gestion des trajets alternatifs

- ➡ En tant que serveur, je souhaite envoyer uniquement aux utilisateurs concernés les trajets itinéraires calculés par le module "itinéraires alternatifs".
- ➡ En tant que serveur, je souhaite envoyer le détails du trajet alternatif choisi par l'utilisateur.
- ➡ En tant que serveur, je souhaite pouvoir gérer les changements de direction de l'utilisateur en direct en cas de représentation "Maps".

## Critères

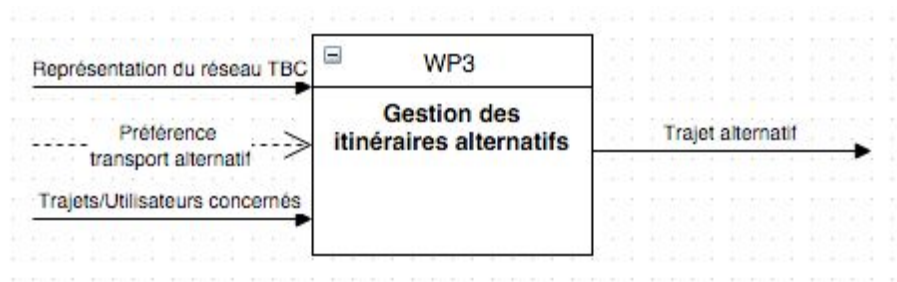
- Un utilisateur ne devra recevoir un itinéraire alternatif à la seule condition que son trajet enregistré soit perturbé.
- Si il y a plusieurs trajets renseignés par l'utilisateur, le serveur devra vérifier que la panne intervient au jour et à l'heure auquel l'utilisateur a indiqué qu'il emprunterait ce trajet.
- Lors de l'envoi d'itinéraire alternatif, l'utilisateur devra choisir celui qui lui convient le mieux. Une fois ce choix fait, le serveur devra communiquer le détail de l'itinéraire à l'utilisateur.
- Les détails de chaque itinéraire seront de la forme "Départ de cet arrêt à XhXX. Arrivé à cet arrêt à XhXX". L'ensemble des transports utilisés devront aussi être présenté.
- L'utilisateur aura à sa disposition deux représentations : une liste d'instruction en format texte ou une carte interactive indiquant en temps réel les déplacements à effectuer.

## WP3 - Gestion des itinéraires alternatifs

- ➔ En tant que serveur, je souhaite créer un itinéraire alternatif pour un trajet utilisateur lorsqu'une panne est détectée.
- ➔ En tant que serveur, je souhaite communiquer les nouveaux itinéraires à l'utilisateur.

### Critères

- Les pannes seront remontées depuis le module de récolte et traitement des données.
- Lorsqu'une panne est détectée, le module de gestion utilisateur communique les trajets impactés par cette panne.
- Pour chaque trajet impacté, le module devra calculer un nouvel itinéraire.
- Cet itinéraire devra utiliser la partie du transport TBC opérationnel. Pour cela, le module pourra se baser sur la représentation du réseau réalisé par le module récolte et traitement des données.
- Pour le calcul d'itinéraire alternatif, le module se basera sur les algorithmes proposés par l'API Direction de Google Maps. Dans le cas où ils ne seraient pas exploitables, il conviendra d'en développer un à part entière.
- L'algorithme en question devra prendre en entrée la représentation du réseau à l'instant de la panne, l'arrêt de départ et d'arrivée du trajet utilisateur, et l'horaire de départ.
- Pour trouver un nouvel itinéraire, il utilisera l'ensemble des transports à sa disposition, c'est à dire tramway, bus, et vélo.
- L'algorithme devra retourner au moins un itinéraire alternatif.
- Dans un second temps, l'algorithme pourra retourner plusieurs itinéraires afin de laisser le choix à l'utilisateur, notamment sur les types de transport.
- Une fois le nouvel itinéraire calculé, celui ci est renvoyé à la gestion utilisateur.

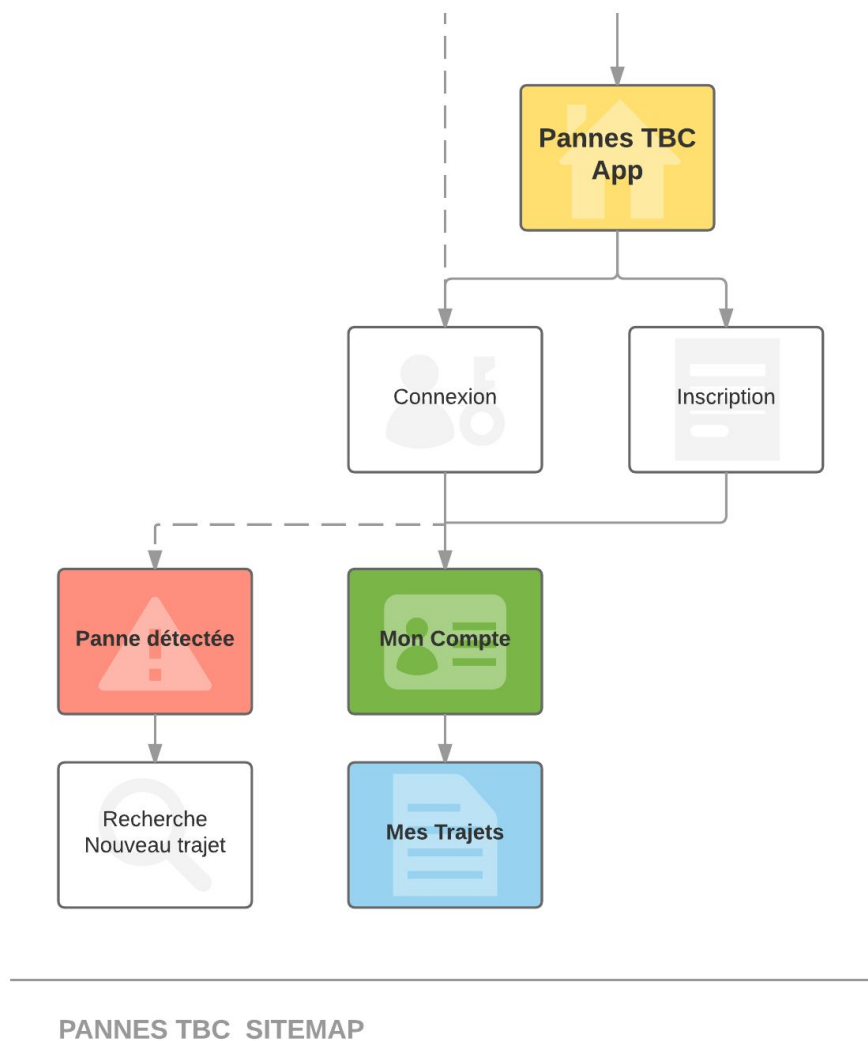


## WP4 - Interface utilisateur

L'ensemble des pages de l'application seront basées sur la charte graphique du client, que ce soit pour la police ou les couleurs.

On retrouvera en haut de l'application une barre de menu, permettant d'accéder aux différentes parties de l'application, ainsi qu'une redirection vers l'accueil depuis le logo.

Afin de suivre les dernières tendances en application mobile, et d'exploiter au mieux Polymer, l'application sera "Flat Design".



## Connexion/inscription

- ➡ En tant qu'utilisateur, je souhaite pouvoir me créer un compte sur l'application.
- ➡ En tant qu'utilisateur, je souhaite pouvoir me connecter à l'application si je possède déjà un compte.

### Critères

- L'utilisateur sera dirigé vers une page d'inscription lors du premier lancement de l'application
- Cette page d'inscription contiendra les champs suivants : nom, prénom, email, mot de passe. En ce qui concerne le champ email et mot de passe, un contrôle sur les caractères sera fait (présence d'un @, de point pour le email, pas de caractère accentué dans le mot de passe).
- Passé la première utilisation de l'application, lorsque l'utilisateur lancera l'application, il sera dirigé vers une page de connexion si il ne s'est jamais connecté sur le périphérique utilisé. Passé la première connexion, les identifiants seront stockés sur le périphérique, connectant l'utilisateur automatiquement à chaque lancement de l'application.
- La page de connexion contiendra les champs suivants : email, mot de passe. Un lien mot de passe oublié sera disponible en dessous ces deux champs.
- Dans le cas d'un mot de passe oublié, il sera possible d'en demander un nouveau en cliquant sur le lien "Mot de passe oublié ?". Ce lien redirigera vers une page demandant l'adresse mail de l'utilisateur afin de lui envoyer un lien de réinitialisation du mot de passe, si cette adresse email correspond bien à un utilisateur enregistré.

## Gestion du compte

- ➡ En tant qu'utilisateur, je veux pouvoir changer mon mot de passe.
- ➡ En tant qu'utilisateur, je veux pouvoir me déconnecter.

### Critères

- Depuis la page d'accueil, il sera possible d'accéder à une page "Mon compte" où les différentes options de gestion de compte sont accessibles.
- Sur cette page se trouvera un bouton permettant l'utilisateur de se déconnecter. Lors de la déconnexion, les identifiants sont automatiquement effacés du téléphone pour éviter la connexion automatique.
- Sur cette page on retrouvera aussi un lien afin d'arriver sur un formulaire de changement de mot de passe. Tout changement de passe devra être confirmé par mail.

## Gestion des trajets

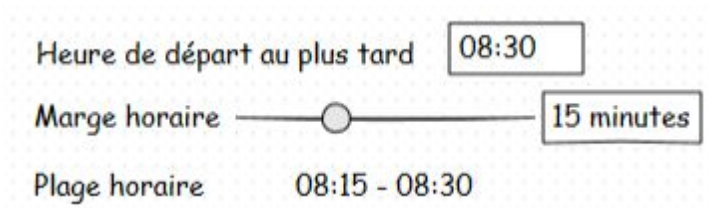
- ➡ En tant qu'utilisateur, je veux pouvoir créer mes différents trajets quotidiens.
- ➡ En tant qu'utilisateur, je veux pouvoir modifier mes différents trajets quotidiens.
- ➡ En tant qu'utilisateur, je veux pouvoir supprimer mes différents trajets quotidiens.
- ➡ En tant qu'utilisateur, je veux pouvoir visualiser l'ensemble de mes trajets sur la semaine à venir.

## Critères

- Depuis la page d'accueil, il sera possible d'accéder à une page "Mes trajets". On retrouvera sur cette page la semaine à venir présentée sous forme d'emploi du temps. Sur cet emploi du temps, on retrouvera les différents trajets renseignés par l'utilisateur aux horaires correspondant.
- Sur cette page se trouvera aussi un bouton "+" pour ajouter de nouveaux trajets. Un formulaire s'ouvrira et demandera les informations suivantes : point de départ (station/adresse ?), point d'arrivée (station/adresse ?), heure de départ, transport privilégié (tram, bus, vcub). (Piste d'amélioration : proposer point intermédiaire dans le trajet). Une fois le trajet validé, il sera affiché sur l'emploi du temps.
- En cliquant sur un trajet existant de l'emploi du temps, une nouvelle page s'ouvrira affichant les détails du trajet. Sur cette page on retrouvera un bouton modifié permettant de modifier n'importe quel champ du trajet. Enfin, un bouton supprimer sera également présent afin de permettre la suppression des trajets.
- En cliquant sur un trajet existant, l'utilisateur aura aussi la possibilité de visualiser son trajet sur une carte interactive.

## Améliorations

- Il pourrait être intéressant de proposer à l'utilisateur une marge horaire en plus de l'horaire de départ. En effet il est très peu probable qu'un utilisateur parte à chaque fois à l'heure exacte qu'il a renseigné.



## Notifications et gestion des pannes

- ➡ En tant qu'utilisateur, je veux recevoir une notification sur mon téléphone lorsqu'une panne est détectée sur un de mes trajets.
- ➡ En tant qu'utilisateur, je veux avoir accès à au moins un itinéraire alternatif en transport en commun lorsqu'une panne est détectée.
- ➡ En tant qu'utilisateur, je veux pouvoir sélectionner le trajet alternatif proposé qui me convient le plus.
- ➡ En tant qu'utilisateur, je veux pouvoir suivre mon trajet alternatif en temps réel depuis une carte interactive.

## Critères

- Lors de l'envoi d'une alerte par le serveur concernant une panne sur le réseau de transport TBC, une notification sera envoyée au téléphone afin qu'il affiche à l'utilisateur "Attention, votre transport de XXh sur la ligne XX est actuellement hors service."
- Si l'utilisateur lance l'application depuis la notification sur son téléphone, il sera redirigé directement vers la page proposant des itinéraires alternatifs. Dans le cas contraire, un indicateur graphique (type "1" rouge) sera présent dans le menu de l'application pour l'inciter à ouvrir cette page.
- Sur la page d'itinéraire alternatif, l'utilisateur aura accès à au moins un itinéraire alternatif. Les informations affichées à ce stade étant le point de départ, le point d'arrivée, le transport utilisé, les nouvelles lignes, et les horaires de départ sur chacune d'entre elles.
- En cliquant sur un des trajets alternatifs, il sera redirigé vers une carte représentant le nouveau trajet. Il aura alors accès à un bouton "Démarrer", lui permettant de choisir ce nouveau trajet, et de le suivre.
- En cliquant sur "Démarrer", il pourra alors soit suivre son trajet en temps réel sur une carte (à condition que la géolocalisation soit activée), soit avoir la liste des différentes étapes du trajet. Les différentes étapes seront de la forme "Départ - Arrivée, Heure départ, Transport, Ligne". Il conviendra d'intégrer les logos des différentes lignes (A, 10 etc...) pour améliorer l'ergonomie de l'application.