



# Stage de développement à la marie de Tours

# **RAPPORT D'ACTIVITÉS HEBDOMADAIRES**



Stage effectué du 25/05/2023 au 28/06/2023

Maître de stage: M. ROHART Thomas

Suivie de stage : Mme. CARAMIGEAS Béatrice

Formation concerné : BTS Services informatiques aux organisations

# **SOMMAIRE**

•	Remerciementpa	ige 3
•	Introduction pa	ige 4
•	Zoom sur l'établissementpa	ige 5
•	<ul> <li>Présentation de l'entreprise</li></ul>	ge 6 ige 7 ige 8 ige 8
	<ul> <li>Les tâches réalisées</li></ul>	ge 14 ge 17

# • Conclusion de mon stage

- o Ce qu'il m'a apporté au niveau compétence
- o Ce qu'il m'a apporté au niveau relationnel
- Conclusion

# I - Remerciement

Je remercie d'abord M. Rohart Thomas pour m'avoir donné la chance de rejoindre l'équipe de la DSI de la ville de Tours ainsi que de m'avoir partagé ses connaissances et de m'avoir offert la possibilité de travailler sur ce projet.

Je remercie toute l'équipe de la DSI pour leur accueil et leur soutien tout au long de mon stage. L'équipe étant amicale et soudée, cela a créé un bon environnement de travail où j'ai pu améliorer mes compétences.

Ce stage a été une expérience enrichissante pour moi et elle aura un impact positif sur mon avenir. Je suis reconnaissant pour tout ce que j'ai pu apprendre et les opportunités qui m'ont été donné pendant cette période de stage.

#### II - Introduction

Lors de ma formation en BTS Services Informatiques aux Organisations (SIO), j'ai effectué un stage de développement informatique qui m'a permis de mettre en pratique mes connaissances acquises au cours de ma formation et de découvrir l'environnement professionnel. Ce rapport de stage a pour but de présenter les missions qui m'ont été confiées, les compétences que j'ai développées, et les enseignements que j'ai pu tirer de cette expérience.

Au cours de ce stage, j'ai eu la possibilité de travailler sur un projet concret et de participer au développement d'une application mobile. J'ai pu me familiariser avec différents langages de programmation tels que le WLanguage, l'HTML, le JavaScript, le CSS et utiliser l'API Mapbox. Cette expérience m'a également permis de développer mes compétences relationnelles.

Dans cette introduction, je présenterai tout d'abord la collectivité dans laquelle j'ai effectué mon stage. Ensuite, je détaillerai les objectifs que j'ai fixés pour ce stage, la mission qui m'a été confiée et les problématiques que j'ai connues. Je parlerai des compétences que j'ai pu acquérir et développer au cours de cette expérience, et les apprentissages que j'ai tirés de ce stage en développement informatique.

Ce rapport de stage me donne l'occasion de réfléchir à mon expérience et de voir comment j'ai progressé. C'est aussi une possibilité pour moi de partager mes réflexions et mes idées sur le développement informatique et comment il affecte les entreprises.

#### III - Zoom sur l'établissement

#### A - Présentation de l'entreprise

Date de création: 1896 - 1904

Fondateur: Architecte Victor Laloux

Statut de l'entreprise : collectivité territoriale

Effectif: 2.700 agents environ

La Mairie de Tours est composée de 7 branches différentes dans lesquelles des sous branches se divisent. Les bureaux qui ci-situent sont les suivants : (Voir annexe 1)

- Cabinet du Maire
- Des méthodes
- Des ressources et moyens
- o des développements cultures et de l'administration générale
- o de l'éducation développement sportif et social
- des services techniques
- Direction des ressources humaines

Ces différents bureaux sont gérés par un directeur général des services qui est lui-même dirigé par le Maire. La collaboration de tous ces bureaux mène à bien les activités de la ville de Tours.

Le rôle d'une mairie est le même dans chaque ville de France :

elle a pour principale mission de satisfaire les besoins quotidiens de la population. Ses attributions sont multiples : état-civil, urbanisme et logement, écoles et équipements, activités culturelles, santé et aide sociale, police ...

Le maire représente l'autorité municipale et est le détenteur du pouvoir exécutif. Il est investi d'un rôle important et bénéficie d'une double fonction .

Placé sous l'autorité du préfet, il est agent de l'État et remplit des fonctions administratives, dont la publication des lois et règlements, l'organisation des élections et la légalisation des signatures. C'est un officier d'état civil et un officier de police judiciaire placé sous l'autorité du procureur de la République

#### B - Présentation du service de la DSI de la Mairie de Tours

La direction des systèmes d'information est la direction responsable du système d'information d'une entreprise. Elle est en charge de définir l'architecture du SI, concevoir, installer, déployer et exploiter le SI.

En l'occurrence la DSI de la Mairie de Tours gère la ville de Tours mais aussi ce que l'on appelle Tours métropole.

Elle est composée de différents services qui se divisent sur 4 étages. Les différents services sont les suivants :

- 1<sup>er</sup> étage : il s'agit de l'étage où se situent la création et les impressions des différentes affiches de la ville de Tours, mais aussi celles en interne . Une salle où une vingtaine d'imprimantes s'y trouve pour imprimer les affiches
- 2º étage: Tout le matériel de la DSI y est situé ici, écran, clavier, souris etc. Lorsqu'une demande de matériel est faite c'est ce service qui prend le relais. C'est aussi ce service qui gère les différents droits d'accès sur le réseau.
- 3<sup>e</sup> étage : lci se situent les serveurs de la ville de Tours. Ils hébergent différents sites et ils ont des hébergeurs sous-traitants.
- 4é étage: Pôle fonctionnel de la DSI de Tours: Ici se trouve notamment la directrice de la DSI Mme. CHATAIN Véronique, les développeurs qui créent les différentes applications etc...

Le 4<sup>é</sup> est donc mon lieu de stage, je suis dans le « lab ' », c'est ici que les applications sont développées . Il y a 4 développeurs qui s'occupent de coder des applications selon la demande des différents services de la mairie de Tours.

Ils doivent donc respecter un cahier des charges donné. Dans ce bureau travaillent des développeurs ainsi que différents acteurs de la DSI, comme le chef de projets numériques , le gestionnaire de la commande publique ou encore le directeur du pôle fonctionnel

#### C - Présentation de l'environnement matériel et logiciel

Durant le confinement covid, il y a eu du BYOD car beaucoup de personnel se sont retrouvés à utiliser leur pc personnel pour se connecter à l'environnement de travail. Aujourd'hui, le BYOD est plutôt proscrit.

Au contraire, ce qui a beaucoup changé depuis le Covid, c'est le fait que la DSI propose désormais soit un PC fixe, soit un kit mobilité, c'est à dire pc portable + écran + clavier + souris. Les agents ont le choix, c'est une forme de CYOD.

Par défaut, les directeurs et directrices, les élues, les agents avec une responsabilité disposent souvent d'un smartphone. Le choix est restreint à 2 ou 3 modèles. Par exemple le smartphone incassable pour les agents sur le terrain, et un Samsung d'un modèle précis sous Android pour les autres.

Il y a des demandes spécifiques parfois comme un IPhone ou un Android en particulier. Dans ce cas, c'est souvent la direction métier qui finance si le besoin est vraiment validé. Il y a tout de même beaucoup de BYOD car de nombreux agents paramètres leur messagerie sur leur smartphone personnel, pour avoir leur agenda.

Actuellement une migration vers Windows 11 est en train de s'effectuer dans toute la DSI. Quelques ordinateurs portables sont encore sous Windows 10. Il y a un accès VPN de disponible. En effet, les utilisateurs peuvent accéder de chez eux à leurs lecteurs réseaux pour télé-travailler par exemple. Ils disposent tous d'un identifiant et des mots de passe.

Sur l'ordinateur de chacun des utilisateurs, se trouve différents logiciels. Tout d'abord :

- WebEx : c'est une application créée par Cisco, elle remplace Microsoft Teams. Elle permet donc de créer des canaux de discussion, faire des réunions, des appels vidéos, etc. Actuellement je suis dans 3 canaux de discussions communs au personnel de la DSI. (Voir annexe 2)
  - Un groupe nommé «Espace café »
  - Un groupe nommé « DSI », c'est ici que toutes les informations en rapport avec le bureau de la DSI circulent.
  - Un groupe « Centre de service ». Quand les utilisateurs ont des problèmes avec leurs matériel informatique, ils posent leurs questions ici. Une réponse est très vite donné en général.
- DSI Store est une application interne à la DSI de la mairie de Tours, qui permet tout comme Microsoft Store d'obtenir diverses applications comme par exemple WebEx cité au-dessus.
- Outlook est un client de courrier électronique. Chaque personne possède une adresse mail du type « x.nom@tours-metropole.fr » . Comme je viens de l'extérieur, j'ai une adresse mail un peu différente, elle ressemble à ceci : x.nom-ext@tours-metropole.fr (ext voulant dire extérieur).

#### D - Présentation du contexte de ma situation professionnelle

Créée en 2018 l'application nommée « Aristée Gestion » (Voir annexe)

permet aux jardiniers de tours-métropole de programmer des interventions, rechercher des arbres pour avoir plus d'informations (genre, essence, taille, etc.), ajouter des arbres, inventorier des arbres et pleins d'autres options que je présenterai.

Une version pour Android existe aussi (Voir annexe). En effet cela évite aux jardiniers de se déplacer avec un ordinateur qui peut-être encombrant. Les deux versions de cette application permettent d'afficher les arbres sur une map grâce à Google Maps, avec des marqueurs dont les couleurs diffèrent selon l'urgence d'intervention.

Depuis peu, Google a décidé de faire payer certains services de son API Maps. On ne peut donc plus accéder à certaines fonctionnalités. Il a donc été décidé de passer à une autre API. Celle-ci s'appelant MapBox .Elle possède 2 versions, « MapBox js », tout comme Google Maps permet d'afficher des maps sous différents formats. Le fonctionnement de cette API est calqué sur une API très connue nommée « Leaflet » . « MapBox js » n'évolue plus car une nouvelle version est apparue s'appelant « MapBox GL JS», celle-ci ne dépendant plus de Leaflet. Elle propose de très jolies cartes et beaucoup de fonctionnalités.

Une implantation avait été essayée en 2019, sans succès puisque cette API, bien qu'elle soit nouvelle, manquait de fonctionnalités dont l'application avait besoin. Le choix a donc été fait de rester sur l'ancienne version de MapBox, « MapBox js ». « MapBox js » a donc été implantée sur la version Windows de « Aristée Gestion », mais elle n'a pas encore été ajoutée sur la version Android.

#### E - Présentation de la mission confiée

Ma mission principale est donc d'implémenter l'API MapBox dans la version Android de « Aristée Gestion » tout en gardant toutes les fonctionnalités de base.

J'ai à ma disposition pour réaliser cela, une tablette Android de la marque Samsung et un émulateur directement dans le logiciel WinDev que je présenterai. Il émule une tablette de la marque « Nexus ». Il faut implanter la carte dans tous les onglets dans lesquels elle est utilisée.

Chaque onglet a sa propre fonctionnalité et tous les onglets sont donc différents. Il faut que j'implante cette carte pour chaque onglet tout en respectant les fonctionnalités de base de chaque onglet. L'utilisateur doit pouvoir selon des critères entrés, voir l'arbre apparaître sur la map et doit pouvoir ,lorsqu'il clique ou appuie dessus, consulter les informations en détail de l'arbre via une info-bulle (Voir annexe).

Pour réaliser cette mission je dois combiner 5 langages. Le W langage, l'HTML, le CSS, le JavaScript et le SQL . Je dois aussi manipuler le <u>GeoJson</u>. Je reviendrai plus en détail sur son fonctionnement et son utilité. Je dois aussi m'aider des documentations de l'API « MapBox.Js » et du site de WinDev qui propose une documentation ainsi qu'un forum pour m'aider à l'utilisation de ce nouveau langage.

### IV - Mon activité en tant que stagiaire chaque semaine

#### A- Les tâches réalisées

#### Semaine 1:

 Mon objectif était d'ajouter pas à pas l'API MapBox. Comme je l'ai dit plus haut l'application mobile possède plusieurs onglets. Un onglet nommé « Parcourir la carte » permet de consulter la map de la ville de Tours.

J'ai donc décidé de commencer la nouvelle carte ici. En effet c'était le plus simple puisqu' aucun critère de recherche n'est demandé, il n'y pas non plus besoin d'afficher d'informations, juste afficher la carte. Je me suis aidé de l'application « Aristée Gestion » Windows puisque la carte via l'API MapBox était implantée dans celle-ci. Je précise que l'onglet « Parcourir la carte » n'est pas directement disponible dans la version Windows de l'application, j'ai donc dû adapter le code selon mon besoin.

J'ai obtenu des résultats satisfaisants dès le 3e jour (Voir annexe). Suite à cela, j'ai décidé d'implanter la carte dans les autres onglets. Je suis actuellement encore en train de travailler dessus.

#### Semaine 2:

J'ai travaillé sur l'implémentation de la carte sur les autres onglets. Le premier concerné a été l'onglet « recherche multicritères, consultation, et mise à jour de l'inventaire des arbres ». Le fonctionnement de cet onglet est assez simple. L'utilisateur saisi des critères de recherches d'un arbre, il a plusieurs choix puis, selon ces critères la carte est affiché avec les arbres et leur description (voir annexe vidéo). Grâce à des marqueurs, on peut localiser un arbre facilement dans la ville et lorsque qu'un appui est effectué sur l'un des arbres une info-bulle apparaît avec les différentes informations de l'arbre. Les caractéristiques des marqueurs sont créées grâce à l'utilisation de GeoJson.

Le GeoJson permet de décrire des données de type point, ligne, chaîne de caractères, polygone, ainsi que des ensembles et sous-ensembles de ces types de données et d'y ajouter des attributs d'information qui ne sont pas spatiaux. Voici un exemple de GeoJson utilisé pour le projet actuelle.

Il est généré via WinDev, une fois que l'utilisateur a choisi ses critères de recherches. Puis via le JavaScript et l'API MapBox, il est appliqué sur la carte générée auparavant. Une fois que cela est fait le fichier est enregistré puis chargé dans le Champ-Html. Il est possible de zoomer et de dézoomer via le pincement des doigts. Plus on zoom, plus des groupes d'arbres se forment. Sur ces groupes d'arbres, il y a un nombre qui signifie le nombre d'arbre situé dans cette zone. Enfin, lorsque la souris se place sur le marqueur, on peut voir la zone dans laquelle les arbres se situent. Sur la gauche, on peut voir la liste des arbres ainsi

que leurs différentes informations. Et en bas à droite, on peut voir le nombre d'arbres trouvé pour les critères choisis.

#### Semaine 3:

Cette semaine, j'ai avancé sur le projet notamment sur la partie lecture/écriture des coordonnées des marqueurs. Lors de la précédente semaine, j'étais bloqué sur le fait de pouvoir écrire dans un fichier externe les coordonnées en x et en y, de chaque arbre. Grâce à mes différentes recherches, j'ai pu les écrire dans un fichier externe un peu particulier, mais qui fonctionne très bien. Le « localStrorage » des navigateurs. Tous les navigateurs ont un « localStorage » qui peut stocker des fichiers. Il existe 2 types de « localStorage »

Le localStorage basique et le localStorage dit de session. On parle de localStorage de session pour dire que chaque fichier stocker dans celui-ci n'est créé et disponible que lorsque la page est ouverte.

#### I : le localStorage « basique » :

Toutes les données sont partagées entre tous les onglets et fenêtres de la même origine.

Les principales caractéristiques du localStorage sont :

• Les données n'expirent pas. Elles restent après le redémarrage du navigateur, et même le redémarrage du système d'exploitation.

Par exemple, si l'on exécute ce code :

localStorage.setItem('test', 1); → Ici nous définissons un fichier avec comme nom « test » et la valeur « 1 » cela signifie qu'à l'intérieur de celui-ci sera écrit « 1 ».

Et l'on ferme puis rouvre le navigateur ou ouvre simplement la même page dans une fenêtre différente, alors on pourra obtenir la valeur du fichier comme ceci :

alert( localStorage.getItem('test') ); // 1 l'alert nous retournera 1

On peut voir qu'ici 2 fonctions on était utilisé sur le localStorage :

- o « setItem() » → Qui prend un paramètre une clé et sa valeur (autrement dit le nom du fichier et ce qu'il y aura écrit à l'intérieur )
- o « getItem() » → Qui ne prend-elle qu'un seul paramètre la clé du fichier, elle nous retournera sa valeur.

Nous devons seulement être sur la même origine (domaine/port/protocole), le chemin de l'url peut être différent.

Le localStorage est partagé entre toutes les fenêtres ayant la même origine, donc si nous définissons les données dans une fenêtre, la modification devient visible dans un autre « localStorage »

#### II : le localStorage de session :

L'objet .sessionStorage est utilisé beaucoup moins souvent que localStorage Les propriétés et les méthodes sont les mêmes, mais c'est beaucoup plus limité :

- Il n' existe uniquement dans l'onglet actuel du navigateur ;
- Un autre onglet avec la même page aura un stockage différent.
- Il est tout de même partagé entre les « iframes » dans le même onglet (en supposant qu'ils proviennent de la même origine).
- Les données « survivent » à l'actualisation de la page, mais pas à la fermeture/ouverture de l'onglet.

#### Par exemple, si l'on exécute ce code :

```
sessionStorage.setItem('test', 1);
```

Et que nous actualisons ensuite la page, on pourra toujours accéder à ces données

```
alert( sessionStorage.getItem('test') ); // après le rafraîchissement : 1
```

Mais si l'on ouvre la même page dans un autre onglet et que l'on réessaye d'accéder aux données, le code ci-dessus renvoie « null » , ce qui signifie qu'il n'a rien trouvé.

Voici les différentes fonction qui peuvent être utilisé avec le localStorage mais aussi bien avec le sessionStorage :

- « setItem(key, value) » Paire clé / valeur de stockage.
- « getltem(key) » Obtenir la valeur par clé.
- « removeltem(key) » Retirez la clé avec sa valeur.
- « clear() » tout supprimer.
- « key(index) » Obtenir le numéro de clé à l'index passé en paramètre
- « length » le nombre d'objets stockés.
- « Object.keys » Utilisez pour obtenir toutes les clés.

Les navigateurs modernes prennent également en charge la « BroadCast Channel API, intéressante pour la communication inter-fenêtre de même origine, elle est plus complète,

mais moins prise en charge.

Dans mon cas, je suis encore allé plus en détail pour utiliser cette fonction. Lorsque j'ai récupéré les cordonnées, de chaque arbre j'ai dû les liés à leur ID. Pour réaliser ceci j'ai créé un fichier Json de cette forme :

Dans un premier temps, j'accédais, via la console du navigateur, au Json créé pour voir si tout fonctionnait bien pour accéder aux Json, il suffit de : json.arbres.coordinates // ce qui retourne ceci : {x: 1670, y: 275}

coordonnées On peut accéder aux en Χ et en У comme ceci : 1670 ison.arbres.coordinates.x,// ce qui retournera json.arbres.coordinates.y // : 275 retournera ce qui

Une fois lié, j'ai réussi via le « localStorage » à écrire les informations de l'arbre puis à y accéder via la console du navigateur. Cette fois-ci, pour y accéder, j'ai du Parser le fichier en Json, car en effet une fois écrit, le fichier n'est plus un Json en tant que telle. J'ai utilisé la fonction Json.Parse(//nomDuFichierAParser ) Une fois cela réaliser, je pouvais accéder au JSon via :

var fichierJSON = localStorage.getItem("fichierTest"); var fichierJSONObjet = JSON.parse(fichierJSON);

fichierJSONObjet.arbres[0].coordinates.x // Ici j'accède au fichier précédemment parsé
Je lui dis que je veux accéder à l'arbre à l'index 0 du tableau puis que je veux accéder à
ses coordonnées en x

Pour finaliser entièrement mon idée « d'auto clic » il faut donc que je passe en paramètre, à la fonction « elementFromPoint() » les coordonnées précédemment écrit dans le fichier Json puis que je lui applique la fonction « .click() » ce qui donne : document.elementFromPoint(fichierJSONObjet.arbres[0].coordinates.x, fichierJSO-NObjet.arbres[0].coordinates.y).click() :

Et voilà maintenant l'auto clic est fonctionnelle. Il faut maintenant que j'arrive l'implanter dans le champHtml de WinDev.

#### <u>Semaine 4 :</u>

Cette semaine, j'ai réussi à réaliser différentes choses dans Aristee gestion.

- La première chose que j'ai réalisée a été de lier le champ-HTML avec la zone répétée dans Aristée Gestion. Grâce à différentes fonctions Javascript, à mes connaissances en Wlangage j'ai pu réaliser cela assez rapidement. Maintenant, lorsque l'utilisateur appuie, sur une ligne de la zone répétée (par exemple, il veut savoir où se situe l'arbre 8510.), la carte se redirige sur l'arbre correspondant. On a décidé d'enlever l'info-bulle qui apparaît lorsqu'un arbre est sélectionné, car ses informations apparaissent déjà dans la zone répétée.
- 2. Une fois que j'ai réussi à lier le champ html à la zone répétée, j'ai ajouté une fonctionnalité qui n'était pas forcément demandée dans l'application, mais qui était plus pratique pour l'utilisateur. M. Rohart m'avait parlé de cette fonctionnalité, qui consistait à faire grossir et/ou changer un marquer de couleurs lorsque celui-ci était sélectionné. J'ai donc crée une fonction JavaScript qui permet de faire cela. (voir annexe)

#### Semaine 5:

Lors de cette semaine, j'ai pu terminer l'implantions de la carte dans tous les autres onglets de l'application. Quelques bugs sont encore à régler lors de la semaine 6.

- 1.
  J'ai donc pu finir l'onglet de « Recherche multicritères, consultation et mise à jour de l'inventaire des arbres » cet onglet à particulièrement était long à finir dû au fait de devoir trouver une façon de lier le champ html ainsi que la zone répète et surtout mettre en œuvre cette idée.
- 2. Le deuxième onglet que j'ai accompli a été l'onglet « Ajouter un nouvel arbre dans l'inventaire ». Via cet onglet les utilisateurs peuvent ajouter un arbre selon des critères bien précis (que l'on appelle une fiche) (voir annexe) et peuvent choisir , via un marqueur sur la carte, où l'arbre se situe. Selon où est placé le marqueur les coordonnées en longitude et latitude changent. Aussi, j'ai ajouté une fonctionnalité qui permet que lorsque l'utilisateur appuie n'importe où sur la carte, le marqueur vienne se positionner sur la position de l'appui. (voir annexe)
- 3. Ensuite, j'ai réalisé un onglet très similaire qui lui a pour but d'ajouter un groupe d'arbres au lieu d'un seul et même arbre, l'utilisateur peut ensuite créer une fiche de ce groupe d'arbres (voir annexe). Encore une fois ,via un marqueur , l'utilisateur vient placer sur la carte où se situe le groupe d'arbres et voie les coordonnées bouger selon où le marqueur est placé. L'onglet une particularité, car il est lié à un autre onglet qui, lui, permet de faire des recherches de groupe d'arbres (voir annexe). Lorsque l'on clique sur l'un des groupes d'arbres, il y a alors la fiche du groupe d'arbres qui est ouverte dans l'onglet précédemment cité. On peut alors modifier la fiche du groupe d'arbres.

Pour réaliser ces deux onglets, j'ai utilisé la fonction WL.Execute() qui prend en paramètre un JSON. Ce JSON contient la longitude et la latitude du marqueur. Il est mis à jour à chaque fois que le marqueur à finie de bouger via l'événement « move end »(voir annexe). Pour rappel la fonction WL.Execute() permet d'exécuter une procédure Wlangage dans du code JavaScript. J'ai donc pu via cette fonction crée une méthode qui réaffecte à la variable les longitudes et latitudes actuelles du marqueur.

Une fonctionnalité que j'ai aussi pu ajouter est le fait que lorsque l'utilisateur appui sur un arbre sur la carte, il soit directement sélectionné dans la zone répétée. Cela rend beaucoup plus facile le travail à l'utilisateur (en effet, il n'a donc pas besoin de rechercher dans toute la liste l'arbre souhaité.). De plus, toutes les informations de l'arbre sont donc mises en surbrillance et consultable facilement.

#### B - Les problèmes rencontrés

#### Semaine 1 :

Dès mon arrivée, j'ai rencontré des problèmes majeurs que je vais expliquer.

- M. Rohart avait installé tous les logiciels nécessaires au bon déroulement de mon stage. Malheureusement, des problèmes de licence WinDev sont apparus. WinDev fonctionnait très bien mais sa version mobile, quant à elle, affichait une erreur de dongle (voir annexe). Le terme dongle provient de l'anglais "dongle" qui a pour signification pendouiller. Il est utilisé dans l'informatique pour désigner les composants électroniques que l'on branche sur un ordinateur afin d'apporter une fonctionnalité supplémentaire à ce dernier, tel que les prises HDMI reliant à un écran, les clés USB, etc. Ici la fonctionnalité supplémentaire est la licence WinDev qui fonctionne par clé USB.
- Le second problème est arrivé lorsque l'on a essayé d'importer le projet « Aristée Gestion » via un GDS (Gestionnaire de sources) intégré directement dans WinDeV. On peut comparer cela à l'utilisation de GIT. (Voir annexe)
- 3.

  Le troisième problème a été d'implanter la carte dans la version Android d'« Aristée gestion ». Sur la version Windows de l'application un champ « ActiveX » est utilisé pour afficher la carte. Le champ ActiveX permet de gérer simplement les contrôles ActiveX. Les contrôles ActiveX sont des composants réutilisables utilisant la technologie ActiveX et pouvant être intégré dans une fenêtre. En quelques sortes, on peut implanter pleins de contrôle externe à WinDev. Dans mon exemple c'est une carte qui est implantée via l'API MapBox. Malheureusement ce champ ActiveX n'est pas disponible dans la version Android, j'ai donc dû m'adapter à cela.
- 4.
  Le dernier problème connu a été lors du test de l'application Aristée Gestion sur la tablette qui m'était fournie. Lors de l'émulation sur la tablette une erreur se produisait et faisait planter l'application (voir annexe). En effet la tablette

essayait de se connecter au serveur de test. Mais il n'est pas disponible sur la tablette.

#### Semaine 2:

1. Le premier problème connu cette semaine a été la génération du fichier GeoJson qui, tout comme le Json, fonctionne avec des paires clé → valeur.

Ces clés et valeurs sont normalement marquées entre guillemets, mais je me suis vite aperçu, lors de mes tests, que les fichiers GeoJson que je générais étaient erronés. Les paires clés  $\rightarrow$  valeurs sont normalement marqué qu'avec une quotte et non deux, de plus Visual Studio m'indiquait une erreur. Le fichier GeoJson étant mal généré, il était impossible d'afficher les marqueurs sur la carte.

Le deuxième problème est un problème que je n'ai pas encore résolu. Une fois la carte générée et la liste d'arbres affichés. L'utilisateur doit pouvoir, lorsqu'il appuie sur un arbre dans la liste, voir apparaître sur la carte l'info-bulle de l'arbre sélectionné. Il faudrait donc que j'arrive à faire communiquer le champ html avec la liste (que l'on appelle une zone répétée). Il faut savoir que le champ html peut être considéré comme un navigateur intégré. Il faut donc « simplement » que j'accède au code actuellement exécuté sur la page pour récupérer l'ID des marqueurs. Une fois cela réalisé, je pourrais récupérer leur positon en x et y, puis exécuter un clic virtuel pour afficher l'info-bulle.

#### Semaine 3:

- 1. Le premier problème rencontré cette semaine a été plutôt une partie compliquée de mon avancement. En effet, la semaine dernière, j'ai eu l'idée d'écrire les coordonnées de chaque marqueur dans un fichier Json à part. Un problème s'est vite posé. Il est presque impossible d'écrire via JavaScript des fichiers en local sur une machine. Pour des raisons de sécurité, JavaScript n'est pas capable de faire cela.
- 2. Le second problème connu à était lors de mes différents tests sur la tablette : Lorsque l'on fait un débogage sur la tablette. Une version « test » de l'application est générée qui est appelé le « Go » suivi du nom de l'application. Par exemple sur l'application sur laquelle je travaille cela donne « Go Aristée Gestion ». Tous fonctionnaient bien jusqu'à la semaine dernière où le « Go » ne fonctionnait plus. En fait la génération du Go sur l'ordinateur fonctionné, mais au bout de quelques secondes, il s'arrêtait sans aucune erreur, et sur la tablette rien ne s'affichait.

#### Semaine 4

1. J'ai dû revoir ma méthode d'approche pour utiliser mes données Json dans Windev. En effet, mon idée avait été d'écrire via le localStorage, les coordonnées en x et en y de chaque arbre selon la fenêtre actuellement consulté. Le problème que j'ai connu est que les champ-html de Windev n'ont pas de localStorage, j'ai donc du réfléchir à une autre approche. J'avais

- notamment un erreur qui me montrait bien que le champ-html n'avais pas de localStorage (Voir Annexe).
- 2. Le second problème connu a été pour lire les coordonnées encodées en Json en Wlangage. En effet, énormément de fonctionnalité Windev ne sont pas disponible en WinDev mobile. Notamment une fonction : « ChaineVersJson » qui comme son nom l'indique permet de convertir une chaîne de caractère un fichier Json. N'étant pas disponible en Windev Mobile, j'ai dû m'adapter autrement.

#### Semaine 5 :

- 1. Le premier problème rencontré a été lors de l'implantation du changement de couleur des marqueurs. En effet lors de la semaine 4, j'avais implanté cette fonctionnalité, mais je n'avais pas entièrement testé avec différents cas. Notamment le cas où l'utilisateur appuyait sur un marquer d'une couleur et ensuite un marqueur d'une autre couleur. Je me suis vite aperçu que le deuxième marqueur appuyé premier la couleur du premier et vice-versa.
- 2. Le second problème rencontré a été lorsque j'ai voulu utiliser la fonction WL.Execute () dans l'onglet d'un ajout d'arbre. Je voulais passer 2 paramètres à la fonction. Le premier paramètre étant la longitude et le second la latitude. Malgré plusieurs essais, la fonction me retournait une erreur très bizarre (voir annexe). J'ai donc dû m'adapter à cette situation.
- 3. Le troisième problème connu sera résolu la semaine 6. Dans la zone répétée, on peut via un « Tirer- relâcher » mettre à jour la liste des arbres. Actuellement, lors de ce « Tirer-relâcher, » l'application cesse de fonctionner et affiche une erreur (voir annexe). Ce problème se situe dans tous les onglets mise à jour.
- 4. Le quatrième problème sera aussi résolu lors de la semaine 6. Il vient de l'onglet « Ajouter un nouveau groupe d'arbres dans l'inventaire ». Lorsque l'on veut accéder à la fiche du groupe d'arbres, tout fonctionne très bien. Mais lorsque l'on fait retour et que l'on sélectionne un autre groupe d'arbres la fiche du groupe d'arbres précédent apparaît. Il faut refaire une deuxième fois retour puis re-séléctionné le groupe d'arbres voulu pour que la fiche de celui-ci apparaisse.

#### C – Les actions menées et les recherches effectuées pour leur résolution

#### Semaine 1:

1.

Le premier problème a été résolu facilement : La clé USB contenant le dongle n'est pas directement branché sur le pc portable mais sur les serveurs de la mairie via un hub USB qui est relié à un Windows Server qui gère le réseau. Il a donc fallu demander à un technicien réseau de débrancher et rebrancher la clé dongle. Il arrive parfois que les licences bugent.

On s'est rendu compte du bug lorsque M. Rohart a essayé à son tour d'utiliser WinDev mobile et que la même erreur est apparue.

2.

La résolution du second problème a été très simple.

Lors de la première connexion via mon compte, celle-ci n'était pas autorisée à accéder au gestionnaire. M. Rohart a dû créer un utilisateur sur cette GDS pour que je puisse me connecter. Une fois cela réalisé j'ai pu directement me connecter au GDS et récupérer le projet. Lors de la connexion il faut entrer Nom du serveur, utilisateur mot de passe et un port d'accès.

Ce a été un vrai défi pour moi.

En effet le champ active X n'étant pas disponible j'ai dû m'adapter en utilisant ce qu'on appelle un « champ HTML ». Le champ d'affichage HTML permet d'afficher une page Web dans une application. Ainsi, sans utiliser de navigateur externe, WINDEV Mobile peut afficher une page d'un site. Ce champ est basé sur Chromium CEF.

3. Pour le dernier problème, je ne l'ai pas résolu de moi-même. En effet, M. Rohart ayant créé le projet, lui seul savait où allait pour éviter que la tablette essaie de se connecter au serveur de test lors du démarrage de l'application. Il a fallu modifier une ligne de code pour que cela fonctionne correctement.

#### Semaine 2:

Pour résoudre le problème du GeoJson je suis tout simplement allé sur leur site officiel où se situent des exemples de codes. J'y ai vite remarqué que la construction

du GeoJson que je générais n'était pas bonne. Je l'ai su aussi grâce à VisualStudio Code qui me générait des erreurs. N'y connaissant rien au GeoJson, j'ai pu découvrir tout plein de fonctionnalités très utile et j'ai compris l'utilité de celui-ci. (Voir annexe)

Pour le second problème, j'ai cherché sur énormément de sites comme stackoverflow ou encore le site de PcSoft. J'ai eu beaucoup de pistes que je vais citer :

- La première piste que j'ai eue a été sur stackoverflow, une personne avait besoin de localiser une div, sans ID, dans sa page HTML pour y modifier la couleur. On lui a proposé d'utiliser la fonction « elementFromPoint(x,y) » (x et y étant les coordonnées de l'élément recherché).
- Dans un second temps, j'avais cherché si l'on pouvait via l'élément récupérer, lui appliquer la fonction « .click()» . La fonction « . click() » permet de simuler un clic page html. J'ai donc appliqué cette fonction à la fonction « elementFromPoint(x,y) » donnait finalement се qui me « elementFromPoint(x,y) .click() ». J'arrivais donc à "appuyer" sur un élément de ma page virtuellement après l'avoir récupéré. Le problème ici est que j'ai rentré des coordonnées en dur. Les marqueurs ont eu des coordonnées qui bougent selon le le dézoom de Comme je l'ai cité au-dessus, il ne me manque plus qu'à réussir à récupérer les coordonnées du marqueur voulu.

#### Semaine 3:

- 1. Pour résoudre le problème, je me suis redirigé vers « StackOverflow » et « MDN Web Docs » Ici, j'ai pu y trouver les informations nécessaires en rapport au localStorage et au sessionStorage. Des exemples y étaient donnés, je m'en suis donc inspiré. Ensuite, j'ai pu trouver via des sites complémentaires des informations pour parser les fichiers en Json et les exploiter pour mon besoin.
- 2. Pour résoudre le second problème, j'ai dû directement chercher sur la tablette. Après, près de 30 minutes de recherches, j'ai su où se situait le problème. Malgré différentes recherches sur Internet et sur le forum de WinDev, aucune réponse n'était donnée. Au début, j'ai directement cru que le problème était lié à WinDev, mais finalement pas du tout ; du côté WinDev, tout fonctionnait bien.

Lorsque que le « Go » avait été lançais la première fois sur la tablette, une icône est apparue (voir annexe). On peut voir la différence entre les 2 applications (celle en prod et celle en test). J'ai vite remarqué que l'icône de « Go Aristée Gestion » n'était plus là quand je voulais réaliser les tests de débogage.

Il s'avère que plusieurs comptes utilisateur étaient dans la tablette, mais que pour le mien l'application n'était plus installé. J'ai donc dû la désinstaller proprement pour tous les utilisateurs puis une fois cela fait, j'ai relancé un débogage, l'icône est réapparue et l'application, c'est lancez.

#### Semaine 4:

1. Pour résoudre mon premier problème, j'ai dû réfléchir à une nouvelle méthode d'écriture de fichier. Au lieu d'écrire dans le localStorage du navigateur avec JavaScript, j'ai dû me pencher sur l'écriture de fichier via le Wlangage. Lors de la génération de mon Json, une fonction très puissante, que l'on appelle « WL.Execute » permet d'exécuter une procédure wlangage dans du code java script (voir annexe). Cette méthode prend en paramètre le Json qui est généré à chaque mouvement et zoom/dézoom de la carte. Si il n'était pas déjà créé, il crée le fichier, sinon il écrase les anciennes données et les rem-places par les nouvelles. Voici comment est enregistré le Json généré : (voir annexe)

2.

Lors de mes premiers essais pour réussir à lire les informations que j'avais enregistrées, j'ai essayé de transformer le Json en ce qu'on appelle un « Variant » en Wlangage. Je pensais pouvoir accéder aux propriétés de mon Json via le NomDuVariant. MaPropriété. Cela fonctionnait très bien lorsque j'étais sur l'émulateur nexus, mais quand je me retrouvais sur la tablette, plus rien ne fonctionnait. J'ai donc dû changer ma méthode. Cette fois-ci, j'ai demandé de l'aide sur un discord WinDev collaboratif qui m'ont gentiment expliqué comment faire. J'ai du tout simplement déclarer une variable en tant que Json, puis j'accédais à mon Json via leNomDeLa-Variable. MaPropiété. J'ai dû ensuite créer une fonction qui selon la ligne sélectionnée, va chercher les coordonnées correspondantes, puis j'ai appliqué la fonction « docu-ment.elementFromPoint(x,y).click() » qui permet d'appuyer (voir annexe)

#### Semaine 5:

1.

La résolution du premier problème a été plutôt une tâche compliquée, mais j'ai finalement réussi. J'ai recherché le problème et je me suis aperçu d'un problème dans ma fonction. J'ai donc dû changer ma fonction JavaScript qui stock maintenant la couleur de l'ancien marqueur dans une variable et lui rapplique une fois que l'utilisateur appui sur un autre marqueur. (voir annexe)

2.

Pour le second problème, j'ai dû faire de multiples tests, car quelque chose d'illogique se produisait. En effet, sur la documentation de Wlangage, il est clairement écrit que la fonction « WL.Execute » peut prendre de 1 à 10 paramètres inclus. J'avais donc décidait de passer deux paramètres à ma fonction (la latitude et la longitude) .

L'erreur n'étant pas très explicite j'ai dû me débrouiller moi-même. Je me suis finalement aperçu que WL.Execute() ne prend que 1 paramètre pas plus ni moins. M. Rohart, mon maître de stage, ma confirmée cette information. En effet, si la fonction ne prend aucun paramètre elle retourne une erreur, si elle prend plus d'1 para-mètre aussi. J'ai donc décider de

retourner un JSON qui ensuite a été accessible très facilement pour réaffecter les bonnes valeurs au champ latitude et longitude Les deux autres problèmes seront résolus lors de la semaine 6.

# V - Conclusion de mon stage

#### A - Ce qu'il m'a apporté au niveau compétence technique

Ce stage m'a permis d'acquérir de nombreuses compétences en développement informatique.

Tout d'abord d'un point de vue programmation. En effet grâce à ce stage, j'ai pu développer de nouvelles compétences en HTML, CSS, JavaScript. J'ai même découvert un nouveau langage que l'on appelle « Wlangage ». Celui-ci est très puissant pour créer des GUI ou même des applications mobile. J'ai exploré les multiples fonctionnalités de ces langages et appris à les combiner de manière optimale. Par ailleurs, l'utilisation de Mapbox m'a ouvert les portes de la visualisation de données géographiques, enrichissant ainsi mes compétences en cartographie interactive.

Dans un second temps d'un point de vue organisation.

En effet, j'ai pu découvrir qu'est-ce qu'un gestionnaire de source et comment m'en servir. J'ai pu découvrir des nouveaux outils de gestions de projets très puissants et des méthodes de débogages intéressantes.

#### B – Ce qu'il m'a apporté au niveau relationnel

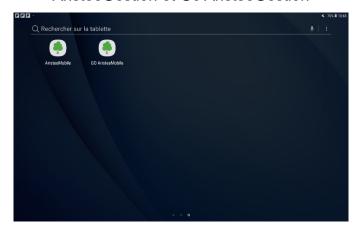
En ce qui concerne le volet relationnel, ce stage m'a offert l'occasion de découvrir le milieu professionnel et découvrir les différents métiers de l'informatique que je ne connaissais pas. J'ai pu partager mes idées, résoudre des problèmes et bénéficier de l'expertise de mon maître de stage. Cette expérience a renforcé ma capacité à communiquer, et à m'adapter à différents styles de travail.

#### C - Conclusion

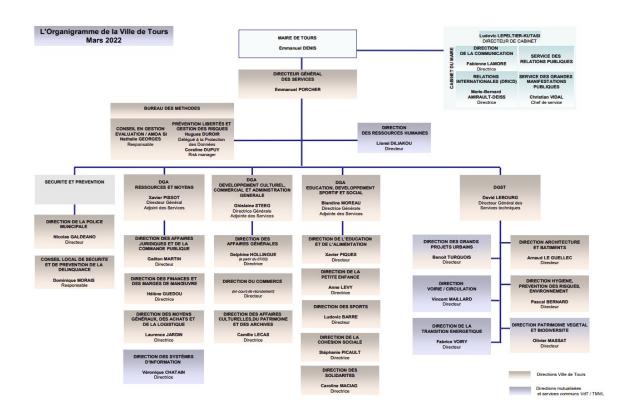
En conclusion, ce stage en développement informatique avec ces différents langages de programmation a été une opportunité pour améliorer mes compétences techniques, renforcer mes aptitudes relationnelles et approfondir ma vision du domaine de la programmation. Je suis reconnaissant d'avoir eu cette opportunité et je suis convaincu que les connaissances et l'expérience acquises pendant ce stage me seront bénéfiques tout au long de ma carrière dans le domaine de l'informatique.

# <u>Annexe</u>

« AristeeGestion et Go AristeeGestion »

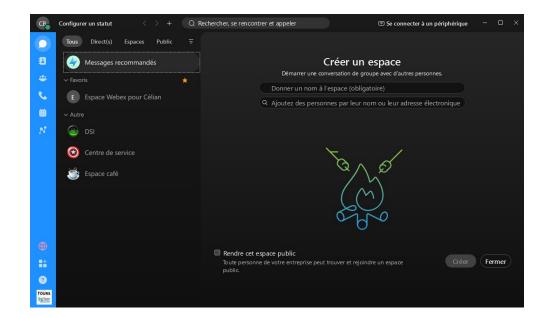


# « Organigramme de la ville de Tours »



# Annexe 2

#### « WebEx »

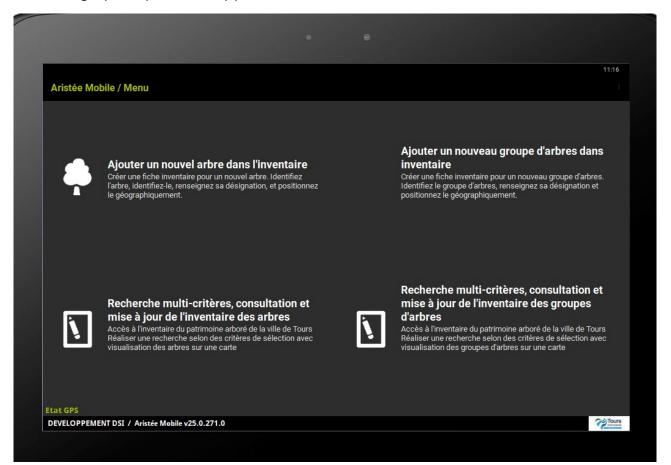


« Page principale de l'application « Aristée Gestion » version Windows »



Annexe 4

### « Page principale de l'application « Aristée Gestion » version Android »



#### Annexe 5

#### « Info-bulle »



Annexe 6
« Nouvelle carte intégré via MapBox »

# Carte précédente :

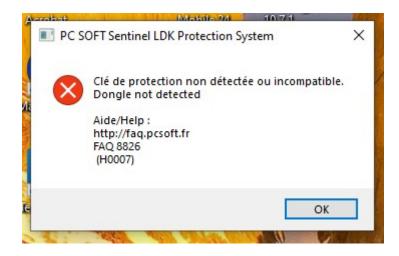


Carte actuelle intégrée via MapBox (possibilité de recherche de rue ainsi que changement de format de carte) :



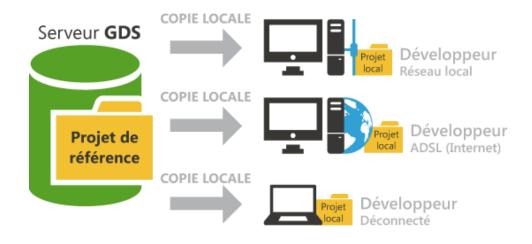
# Annexe 7

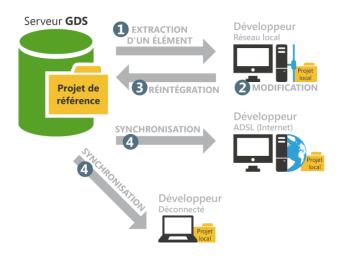
« Erreur « dongle » WinDev »



# Annexe 8

« Fonctionnement GDS (gestionnaire de source) »





« Erreur lors du test de l'application »





# Annexe 11 « Exemple de GeoJson »

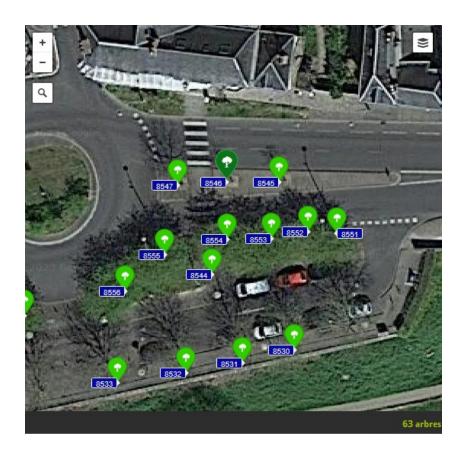
# Annexe 12

« Erreur GeoJson »

```
Property keys must be doublequoted json

View Problem (Alt+F8) No quick fixes available
```

« Exemple de clic sur un marqueur »



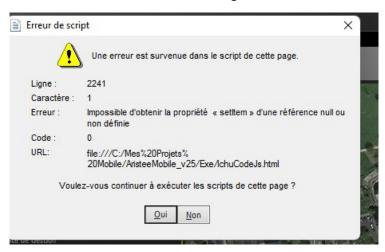
« Fonction permettant de changer le marqueur de couleur »

```
let selectedMarker = null;
function highlightMarker(marker, oldIcon, newIcon) {
    if (selectedMarker) {
        // Remettre à l'état de base le marqueur précédemment sélectionné
        selectedMarker.setIcon(oldIcon);
        selectedMarker.setZIndexOffset(900);
    }

    // Mettre en surbrillance le marqueur sélectionné
    marker.setIcon(newIcon);
    selectedMarker.setZIndexOffset(1000);
}
```

# Annexe 15

« Erreur dû au localStorage inexistant »



```
mp.on("moveend", function () {
       if (json.arbres.length > geoJson.features.length) {
           json.arbres.splice(0, json.arbres.length - 1);
        json.arbres.push(
               "id": f.properties.id,
               "coordinates": mp.latLngToContainerPoint(m.getLatLng())
        newJson = JSON.stringify(json)
      Write(newJson)
   mp.on("zoomend", function () {
       if (json.arbres.length > geoJson.features.length) {
           json.arbres.splice(0, json.arbres.length - 1);
       json.arbres.push({
           "id": f.properties.id,
           "coordinates": mp.latLngToContainerPoint(m.getLatLng())
       newlson = ISON.stringify(json)
     Write(newJson)
   console.log(json)
function Write(e) {
   WL.Execute("Ecrire", e);
</script>
```

Annexe 17

« Procédure wLangage « Ecrire » appelé côté javascript permettant d'écrire le Json en locale »

```
InIDFichier est un entier
lbufCodeJSUTF8 est une chaîne ANSI

lbufCodeJSUTF8 = ChaîneVersUTF8(i)
//fRepEnCours(SysRepCarteStockage() + "\AristeeMobile")

fCrée(fRepEnCours()+ "\ison.json")
lnIDFichier = fOuvre(fRepEnCours() + "\ison.json", foEcriture)
fEcrit(lnIDFichier,lbufCodeJSUTF8)
fFerme(lnIDFichier)
```

«code javascript permettant de rechercher les coordonnées de l'arbre sélectionné selon son ID et de simuler un clic sur les coordonnées passé en paramètre »\_

```
lchuFic est une entier
 lContent est une JSON = fChargeTexte(fRepEnCours()+"\json.json")
∄js est une chaîne = [
     function findObjectById(id,jsonData) {
     for (var i = 0; i < jsonData.arbres.length; i++) {
     if (jsonData.arbres[i].id == id) {
     return jsonData.arbres[i].coordinates
     return null; // Retourne null si l'ID n'est pas trouvé
    var result = findObjectById(
 js += ZR_Arbre[ZR_Arbre].ATT_ID
∃js += [
 js+=lContent
∃js+=[
 );
     document.elementFromPoint(result.x, result.y + 6).click();
 catch (error) {
 alert(' veuillez réessayer');
```

# Annexe 19

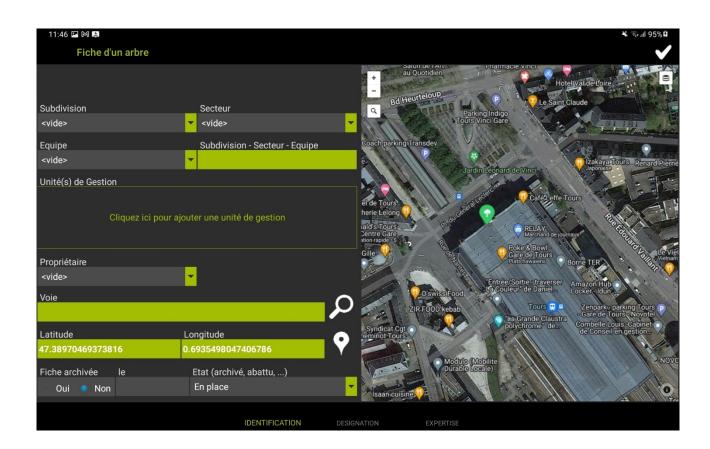
« Ajout d'un arbre selon sa position GPS et différents critères (la postion par défault étant place Jean Jaurès) »



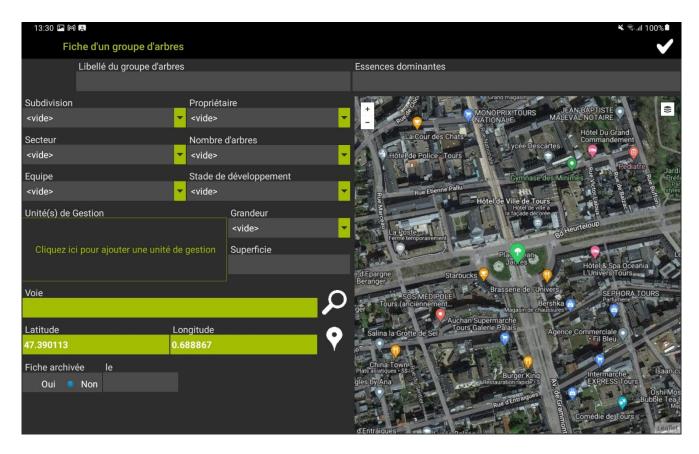
« Exemple d'utilisation du marqueur et sa position GPS »

Exemple sur le lycée Paul-Louis-Courier :





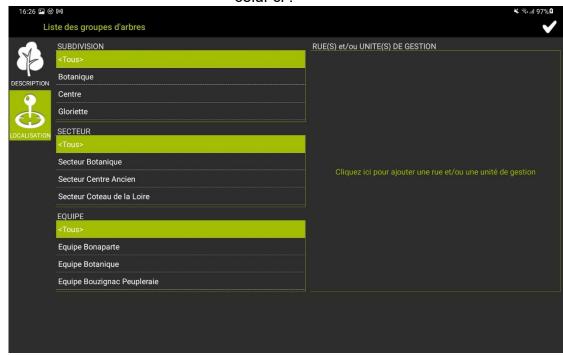
« Exemple de fiche d'un groupe d'arbre à renseigner »



# Annexe 22

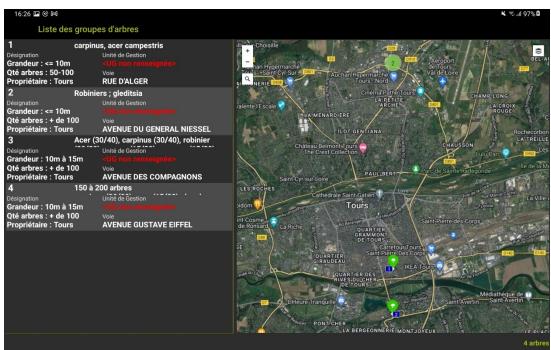
« les différents onglets pour la recherche de groupe d'arbre »

Onglet qui permet de localiser un groupe d'arbre via une adresse ou via la description de celui-ci :

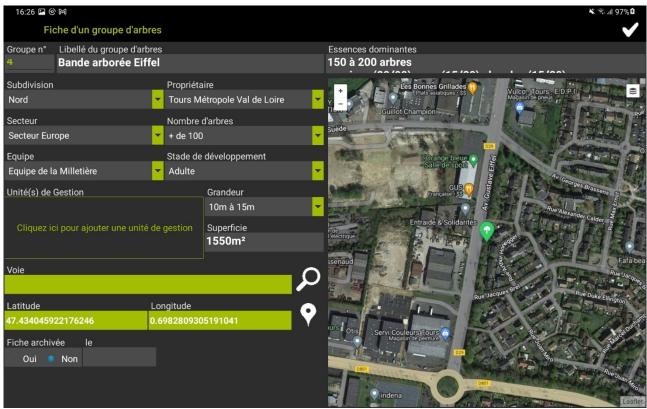


35/

Tous les groupes d'arbres actuellement dans entré dans Aristée Gestion version mobile :



Onglet apparaissant lorsque l'on sélectionne un groupe d'arbre (il s'agit de l'onglet qui permet de créer un groupe d'arbre, ici ce sont les informations qui ont étaient préalablement entré, on peut donc les modifier ) :



Fonction permettant à chaque fin de mouvement d'appeler la fonction « gps » qui vient mettre à jour les coordonnées du marqueur dans windev :

```
marker.on('moveend', function () {
    let Marker_Dep = {
        "Lat": this._latlng.lat,
        "Long": this._latlng.lng
    }:
    gps[JSON.stringify(Marker_Dep)]
```

Fonction permettant à chaque fin demouvement d'appeler la fonction « gps » qui vient mettre à jour les coordonnées du marqueur dans windev :

```
map.on("click", function (event) {
    marker.setLatLng(event.latlng)
    let Marker_Dep = {
        "Lat": marker._latlng.lat,
        "Long": marker._latlng.lng
    };

gps(JSON.stringify(Marker_Dep))
});
```

Fonction gps qui récupère le json générait auparavant et le passe en paramètre de la fonction WL.Execute() qui elle-même appelle la procédure WLangage « setLatLng » :

```
function gps(e) {
    try {
        WL.Execute("setLatLng", e);
    }
    catch (error) {
        alert(error);
    }
}
```

procédure setLatLng qui à chaque fin de mouvement du marqueur met à jour les coordonnées :

```
PROCÉDURE setLatLng(lvTest)

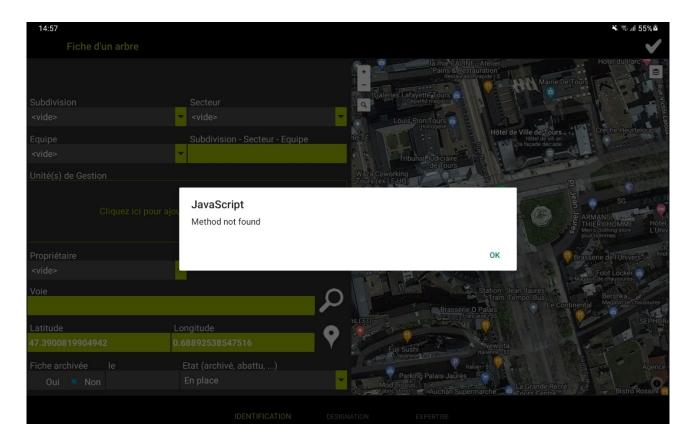
coord est un JSON = VariantVersJSON(lvTest)

SAI_Longitude = coord.Long
SAI_Latitude = coord.Lat
```

#### Entrée mise à jour :

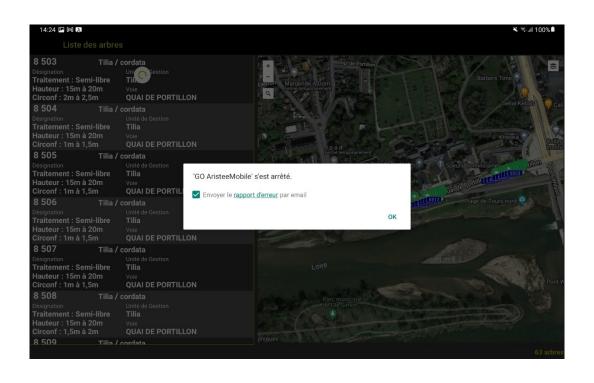
Latitude	Longitude
47.390113	0.688867
CONTRACT OF STREET	

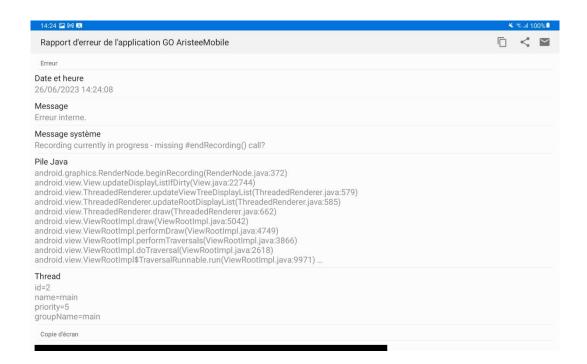
« Erreur lors de l'exécution de la méthode « WL. Execute »



# Annexe 25

Rapport d'erreur





#### Précision sur le rapport d'erreur

# Annexe 26

Fonction permettant de mettre à jour la couleur des marqueurs

Ancienne fonction:

```
let selectedMarker = null;
function highlightMarker(marker, oldIcon, newIcon) {
    if (selectedMarker) {
        // Remettre à l'état de base le marqueur précédemment sélectionné
        selectedMarker.setIcon(oldIcon);
        selectedMarker.setZIndexOffset(900);
    }

    // Mettre en surbrillance le marqueur sélectionné
    marker.setIcon(newIcon);
    selectedMarker.setZIndexOffset(1000);
}
```

Nouvelle fonction:

```
let selectedMarker = null;
let oldMarkerColor;
function highlightMarker(marker, oldIcon, newIcon) {
    if (selectedMarker) {
        // Remettre à l'état de base le marqueur précédemment sélectionné
        selectedMarker.setIcon(oldMarkerColor);
        selectedMarker.setZIndexOffset(900);
    }

    // Mettre en surbrillance le marqueur sélectionné
    oldMarkerColor = oldIcon;
    marker.setIcon(newIcon);
    selectedMarker = marker;
    selectedMarker.setZIndexOffset(1000);
}
```