

RAPPORT DE STAGE  
Concepteur Développeur d'Application

---



Regamey Manuel

---

---

<b>1. Introduction</b>	<b>3</b>
<b>2. Présentation de l'entreprise Adjustem</b>	<b>5</b>
<b>3. Contexte</b>	<b>7</b>
<b>4. Présentation de l'outil de développement</b>	<b>9</b>
<b>5. Phase de recherche et conception</b>	<b>11</b>
<b>6. Présentation des fonctionnalités</b>	<b>21</b>
<b>7. Développement avec CodeIgniter 4</b>	<b>24</b>
<b>8. Résultats et Intégration dans l' Adjustem Web</b>	<b>32</b>
<b>9. Déploiements</b>	<b>33</b>
<b>10. Conclusion</b>	<b>34</b>
<b>11. Annexes</b>	<b>35</b>

---

## 1. Introduction

### **Remerciement**

Je tiens tout d'abord à remercier mon tuteur de Stage Donatien Célia et son associé, Jérôme Gallot cogérant de la société Adjustem pour m'avoir accepté et permis d'effectuer ce stage dans des conditions particulièrement favorables.

Un grand merci à Rémi, Cédric, Gauthier, Kevin, Lee-Mickaël et Lætitia pour leur accueil, leur bonne humeur et l'ambiance paisible et bienveillante qui règne au sein de la société Adjustem.

Je remercie ensuite l'équipe pédagogique de l'Afpa d'Amiens et tout particulièrement notre formateur, Germain Sipière pour sa patience et les conseils prodigues et les autres formatrices, formateurs nous ayant épaulés pendant ces mois de formation.

Et pour finir merci à tous les stagiaires croisés ici à l'Afpa d'Amiens pour cette parenthèse de vie professionnelle où nous travaillons tous pour un meilleur avenir.

Merci à Arlette, Nabi, Dylan, Benjamin, Stéphane, Maxime, Anne, Herbert, Souleymane, Nathan, Scotty, Mathieu, Mohamed, Léa et tant d'autres !

---

## **Présentation de l'auteur**

Initialement Tourneur-Fraiseur de formation, j'ai eu l'occasion de travailler comme développeur Delphi entre 2007 et 2009 à la suite d'une formation de développeur industriel et embarqué autour du langage C en 2005. Intégrateur et chargé de maintenance dans la même société pendant sept années par la suite, j'ai fait le choix de me reconvertir comme ferronnier d'art à l'AFPA du Havre en 2015 avec joie et succès pour retourner au travail du métal et pouvoir quitter la région parisienne. J'ai pu donc pratiquer une activité professionnelle comme ferronnier/métallier pendant près de sept années.

Malheureusement quelques soucis de santé m'ayant rendu inapte au travail d'atelier, j'ai décidé de recommencer une carrière de développeur afin de pouvoir continuer à travailler sans handicap .

Mon passé de développeur aidant j'ai pu intégrer le centre AFPA d'Amiens un un premier MS DIW fin Mars 2022, au cours duquel j'ai pu me familiariser avec les technologies du web que je ne connaissais finalement que de manière superficielle.

---

## **2. Présentation de l'entreprise Adjustem**

Adjustem est une société éditrice de logiciels de gestion commerciale spécialisée dans le chiffrage/devis de menuiserie, destiné aux professionnels de la menuiserie : Négociants, Fabricants, Installateurs.

### **Histoire de la société**

Le logiciel Adjustem était au départ le projet de fin d'étude de Rémi Théron jeune ingénieur informatique au milieu des années 1990.

En démarrant une première exploitation commerciale de la solution Adjustem en 1998 Monsieur Théron sera l'un des précurseurs de la numérisation, du traitement de données et du chiffrage de devis dans un secteur où les catalogues papiers sont la seule référence.

La société Adjustem actuelle voit le jour en 2006 date d'association de Rémi Théron, créateur du logiciel Adjustem, et de Jérôme Gallot.

A la suite au décès de Monsieur Théron en 2012 Jérôme Gallot s' associe à Donatien Célia afin de faire perdurer la suite logicielle Adjustem.

### **Composition de l'entreprise**

L'entreprise co-dirigée par Donatien Célia et Jérôme Gallot.

L'équipe de développement se compose de Rémi développeur C++, Cédric et Lee-Mickaël tous deux développeur web et Donatien qui supervise l'équipe et développe lui-même sous C++ Builder.

Jérôme quant à lui dirige la partie intégration/hotline avec l'aide de Laétitia, Gauthier et Kévin.

### **Activité économique et environnement de l'entreprise**

Devant répondre à une variété de client vaste et diversifiée l'entreprise Adjustem à su adapter son offre aux contraintes technique de l'artisan menuisier jusqu'au grands groupes tels que BigMat, Union-Matériaux ou Castorama.

---

Ainsi tout naturellement l'une des contraintes de la société Adjustem est de maintenir un haut niveau de rétrocompatibilité.

Tout en continuant de maintenir et adapter Adjustem à la perpétuelle évolution des normes du bâtiment, l'équipe d'Adjustem travaille sur une nouvelle suite logicielle : Adjustem Web.

Dans le cadre de cette évolution Donatien Célia m'a confié pour mission le développement d'une version web d'un module de mise en situation de menuiserie. L'actuelle version de ce module est une application desktop développée sous l'IDE C++ Borland Builder permettant de dessiner une menuiserie sur une photographie de bâtiment existant.

L'existence de ce module a permis de définir clairement les attentes minimales pour les résultats de mon travail.

Par un hasard de calendrier, l'entreprise Adjustem s'est retrouvée, lors de ma période de stage, prise dans une refonte d'une partie complexe de sa principale suite logicielle de chiffrage lié à une évolution de la fiscalité sur l'éco-participation des éléments d'ameublement. Cela m'a permis de m'immerger dans l'activité de cette PME en pleine effervescence tout en jouissant d'une totale liberté de recherche pour mon travail.

Le présent rapport fera bien entendu la présentation du travail accompli de la réception du cahier des charges de l'application à l'intégration finale de mon travail dans l'environnement Adjustem Web à la toute fin de ma période en entreprise.

---

### **3. Contexte**

#### **Présentation du domaine d'application du logiciel de Mise en Situation et des besoins pour lesquels il a été développé.**

Le module de mise en situation original étant actuellement en production et utilisé régulièrement par les clients, mon travail devra répondre aux mêmes besoins et apporter quelques avancées

##### **a) cahier des charges :**

La Mise en Situation est un outil permettant de visualiser une menuiserie (porte, fenêtre ou porte de garage) sur un bâtiment. La Mise en Situation doit pouvoir être utilisée par des commerciaux afin de présenter un produit au client ou illustrer un devis.

-Les fonctionnalités de l'ancien module doivent être reproduites dans la nouvelle version soit :

- Prise en charge des formats d'image JPEG et PNG
- Ajout suppression de photos d'un produit/menuiserie
- Ajout suppression de photos d'une photo de bâtiment
- Placement de la menuiserie en mode Proportionnel
- Placement de la menuiserie en mode Personnalité (ratio libre)
- Ajout d'une mention « Photo non contractuelle » en noir ou blanc.
- Export de l'image obtenue via enregistrement en local

De plus les points suivants sont attendus dans la nouvelle version de la Mise en Situation

-Les images de menuiseries devront à présent pouvoir être mise en perspective  
-Le module de mise en situation pour Adjustem Web doit être développé sur une base Php

- Le module doit être responsive
- Le module doit respecter la charte graphique de l'Adjustem\_Web basé sur bootstrap 4.6

-La mention doit pouvoir être plus paramétrable  
-Le module doit pouvoir s'exécuter sur Microsoft Edge, Chrome et Firefox  
-Le module doit pouvoir s'exécuter sur un poste desktop classique, un terminal tactile et une tablette  
-Utilisation des variables de session possible mais pas de cookies  
-Le module doit pouvoir gérer et exporter les images au format base64  
  
-Le module peut fonctionner en stand-alone ou être intégré dans la suite Adjustem Web

---

### b) L'environnement : Adjustem\_Web

Le logiciel Adjustem Web actuellement développé sous le framework Php CodeIgniter 4 est le successeur de la suite Adjustem. Le projet démarré en 2022 est encore totalement en développement. La Mise en Situation est destinée à y être intégrée.

---

## 4. Présentation de l'outil de développement

Pour le développement de la Mise en Situation j'ai décidé d'utiliser une bonne partie des outils avec lesquels j'ai pu être familiarisé lors de ma formation au centre AFPA d'Amiens.

*pour le prototypage et la conception*

- Figma
- PlantUML
- carbon.now.sh
- GoogleDoc

*pour le développement*

- Visual Studio Code
- Notepad++
- winmerge (application de comparaison de fichiers)
- Dbeaver
- Github

Pour la communication au sein de l'équipe d'Adjustem l'entreprise a choisi d'utiliser un serveur Discord depuis 2020.

Pour la sauvegarde des données Adjustem travaille avec un système de sauvegarde en local hérité de près de trente années d'historique qui fonctionne en parallèle d'outils plus modernes comme github.

La mise en situation part sur les impératifs de développement suivants :

- Développement sur un poste de travail sous Windows 7 pro 64 bits
- Le module devra s'intégrer dans l'Adjustem-Web qui est développé sous le framework PHP CodeIgniter 4
- La base de données **adjustem\_web** est une base MySQL dont le moteur de chaque table est MyISAM.
- La version de php actuellement utilisée dans l'application Adjustem-Web est la 7.4.33
- Pour la charte graphique l'utilisation de Bootstrap 4.6 est nécessaire

Pour la réalisation de la compatibilité du modules sur les navigateurs cibles de l'Adjustem-Web mise à disposition d'un second poste desktop équipé d'un écran tactile

---

Les différents tests tout au long du développement ont été réalisés sur les navigateurs suivant :

- Microsoft Edge
- Chrome
- Opera
- Firefox

Lancé sous trois systèmes différents : Windows 7 x64, Windows 10 x64, et Linux Mint( sauf pour Microsoft Edge)

Enfin j'ai pu prendre en main les outils  
phpMyAdmin et Wampserver 3.3.0 et fais mes test sur le mini server PHP spark

---

## 5. Phase de recherche et conception

Sur le premier mois de ma période d'application en entreprise le mode de lancement de l'application n'est pas encore totalement arrêtée et je dois donc réfléchir sur un fonctionnement global que l'application fonctionne en plug in de la suite logicielle Adjustem-Web ( qui est en développement) ou qu'elle fonctionne en standalone.

Il m'est demandé de faire transiter les paramètres d'utilisations ou les données (photos) par le biais des variables de session PHP. L'application doit donc à chaque démarrage ou recharge ment aller puiser les informations nécessaires à son bon fonctionnement dans les variables de sessions à l'aide de PHP. Afficher ces informations à l'aide d'une interface HTML et effectuer tout traitement à l'aide de JavaScript. Les changements d'états ou de paramètres seront envoyés dans les variables de session pour qu'elles soient prises en compte à chaque recharge ment de l'application.

A ce stade il m'est demandé de prévoir le téléversement du résultat de la Mise en Situation directement au sein de la base **adjustem\_web** (base de donné) sous forme d'une image en base64. Après avoir eu une version du script SQL de la création de la base adjustem\_web, j'isole les tables avec lesquelles je dois d'interagir :

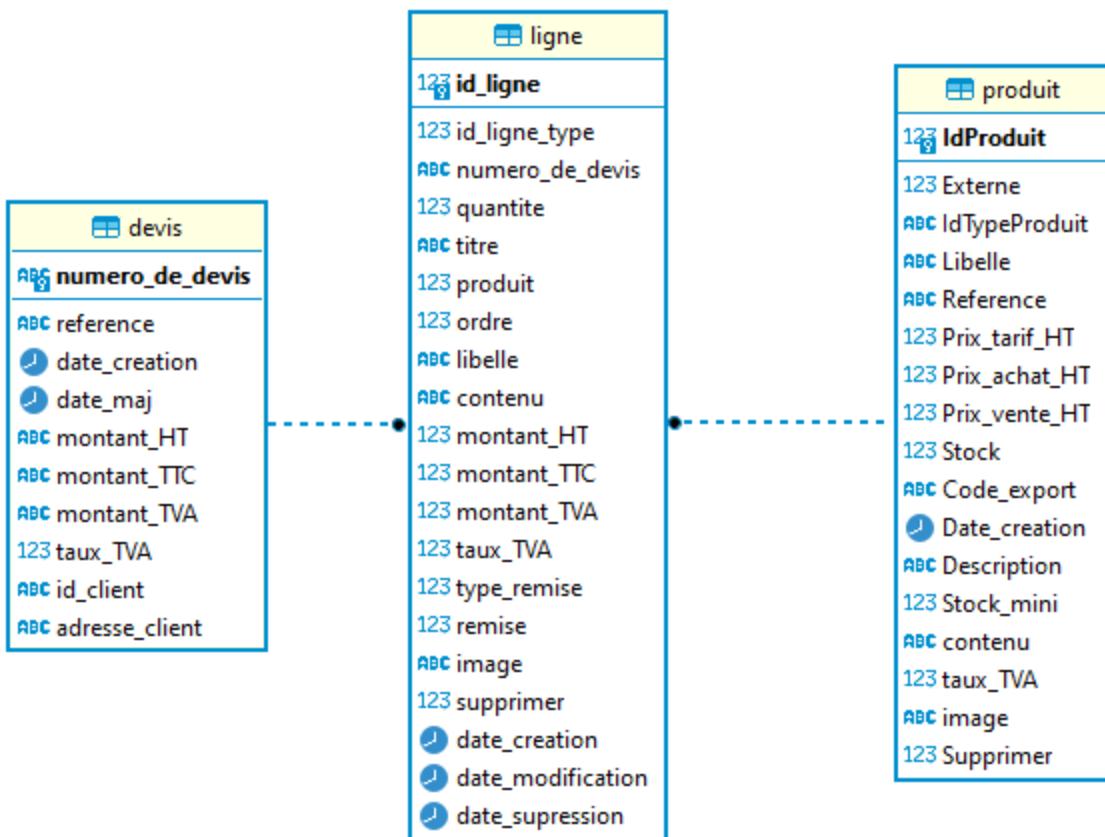


fig 1 : tables Devis, Ligne et Produit adjustem\_web

Le champ **Produit.image** doit contenir l'image de la menuiserie(l'Objet) à placer sur la Vue. Le champ **Ligne.image** est le champ destiné à stocker le résultat de la Mise en Situation.

Je peux donc modéliser le fonctionnement global du module et de son environnement par ce diagramme de séquence :

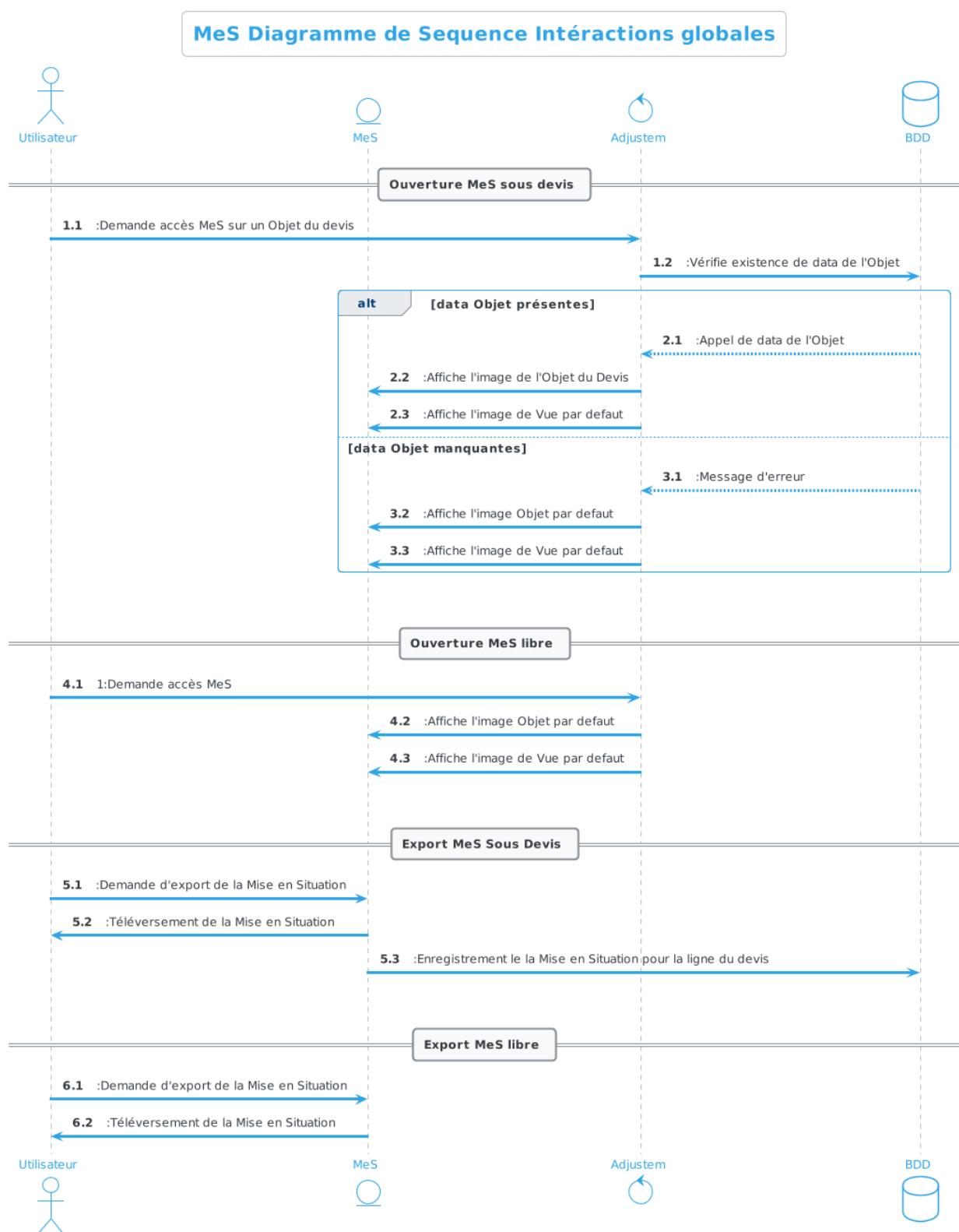


fig 2 : Diagramme de séquence intéractions BDD

## Prototype Figma

Sur la version précédente de la Mise en Situation la vue se trouve sur toute la partie gauche de l'écran et le menu des fonctionnalités caisse à outils



fig 3 : Ancienne version Mise en Situation

J'ai donc créé un prototype sous Figma reprenant cette organisation graphique tout en intégrant une palette de couleur se rapprochant de celle utilisée dans l'Adjustem-Web.



fig 4 : Prototype Figma

Sur la demande de mon tuteur de stage j'ai commencé à explorer les possibilités du PHP sur le traitement des images.

Encore peu rodés aux possibilités du PHP j'ai commencé ainsi ma période d'application en entreprise sur une véritable recherche des différentes technos que je devais pouvoir mettre en œuvre pour la réussite de mon projet.

Dans un premier temps, la bibliothèque libre GD écrite en C et supportée par PHP s'est naturellement imposée lors d'essais de traitement d'images.

En effet GD est un outil complet développé depuis plus de vingt-cinq années par Pierre Joye pouvant manipuler les images au format JPEG et PNG (entre autres) qui sont les formats stipulés dans le cahier des charges du module de Mise en Situation.



*fig 5 Exemple de traitement d'image avec la bibliothèque GD en PHP*

Après la validation de la capacité technique de l'outil à effectuer divers traitements sur les images qui me permettrait de redimensionner les images, les déformer mais aussi gérer la transparence (nécessaire pour la mise en perspective illustrée sur la Fig5) est apparu un problème de taille.

En restant sur un traitement d'image à l'aide de PHP l'application risquait de pâtrir d'une lourdeur architecturale des plus conséquente. En restant sur un processus basé sur GD l'application cliente devrait perpétuellement renvoyer tous les paramètres de modifications voulu par l'utilisateur vers le serveur pour que ce dernier traite l'image avant de la renvoyer à l'application cliente.

Après échange avec mon tuteur de stage Mr Celia, il m'était devenu évident que

---

l'entreprise ne souhaitait pas de développement basé sur une technologie non native qui nécessiterait un support accru chez le client. Ainsi je me suis donc tout naturellement tourné vers l'utilisation de l'API canvas sous JavaScript utilisant l'élément canvas de la spécification HTML5.

Au fur et à mesure de l'avancement du développement et de la modification de la demande initiale, certaines fonctionnalités sont venues se greffer au cahier des charges initial.

#### Gestion de la mention :

Le texte de la mention doit être personnalisable. De plus, après tests et retours des intégrateurs de l'entreprise Adjustem et selon la nature de l'image de la Vue, il s'avère qu'une mention avec une couleur blanche ou noire et une position fixe peut s'avérer inesthétique voire peu lisible. Il sera donc possible de positionner la mention aux quatre coins de l'image et de choisir de manière simple la couleur de la police d'écriture parmi une palette prédéfinie.

#### La transparence :

La Mise en Situation ne permet pas pour l'heure de travailler avec des images ajourées (grilles, fenêtres, clôtures ... ). Le fond blanc des images de produits dans les divers catalogues masquent totalement l'image de la vue en arrière plan. En transformant les fonds unis en zone de transparence il sera tout à fait possible d'élargir l'éventail des possibilités du module de Mise en Situation.

#### Le détourage d'image :

Le logiciel Adjustem intégrant les catalogues de dizaines de fabricants, l'ensemble d'éléments de menuiseries potentiellement utilisable dans le module de Mise en Situation représente un peu plus de 15000 images de portes, fenêtre, grilles, garde-corps, portails, portes de garage et autres clôtures. Toutes ces photos sont au format JPEG. Une immense majorité sur fond blanc et bien souvent entouré d'une zone neutre de blanc. Cela présente un énorme inconvénient lors de leur utilisation sur une Mise en Situation. En effet, même en transformant les zones neutres en zones de transparence la taille de l'image Objet sera bien supérieure à la partie utile de l'image. En rognant les parties neutres entourant les images, la partie utile coïncidera avec la

---

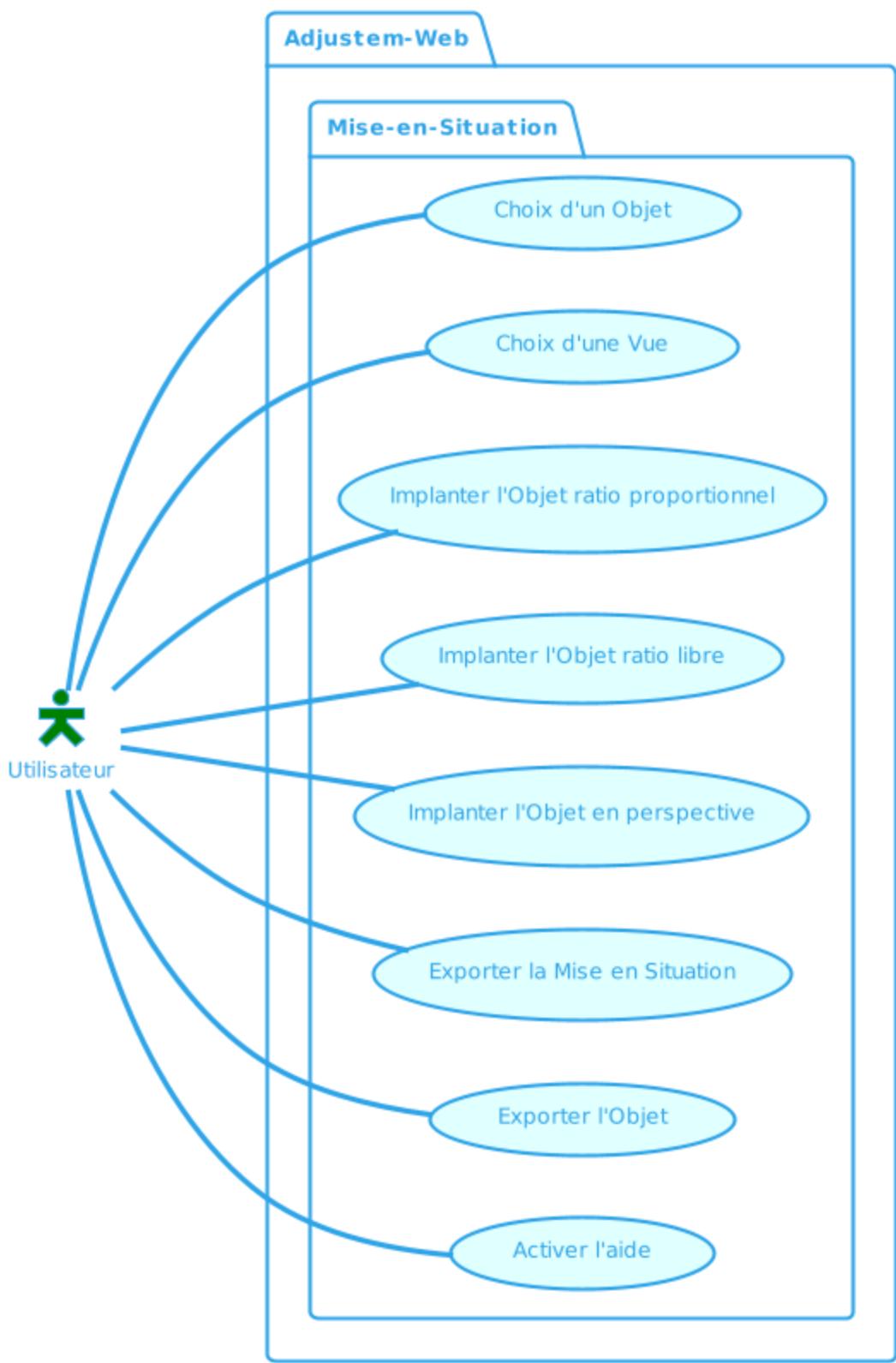
zone de sélection en mode proportionnel et personnalisé ou le quadrilatère utile lors du dessin par point.



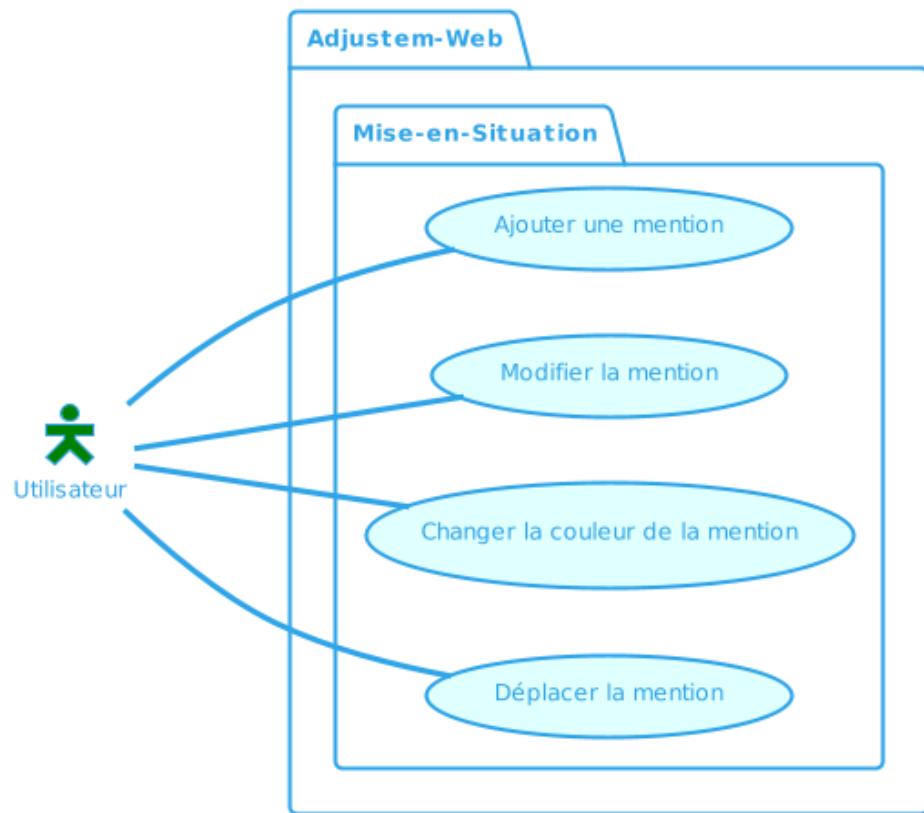
*fig 6 : exemple photo sans et avec détourage*

En schématisant les fonctionnalités de l'application sous forme de diagrammes de cas d'utilisation UML généré sous PlantUML, Nous obtiendrons trois diagrammes (par soucis de place et de lisibilité) :

## Mise en Situation Cas d'Utilisations Général



## Mise en Situation Cas d'Utilisations Partie Mention



## Mise en Situation Cas d'Utilisations Partie Objet

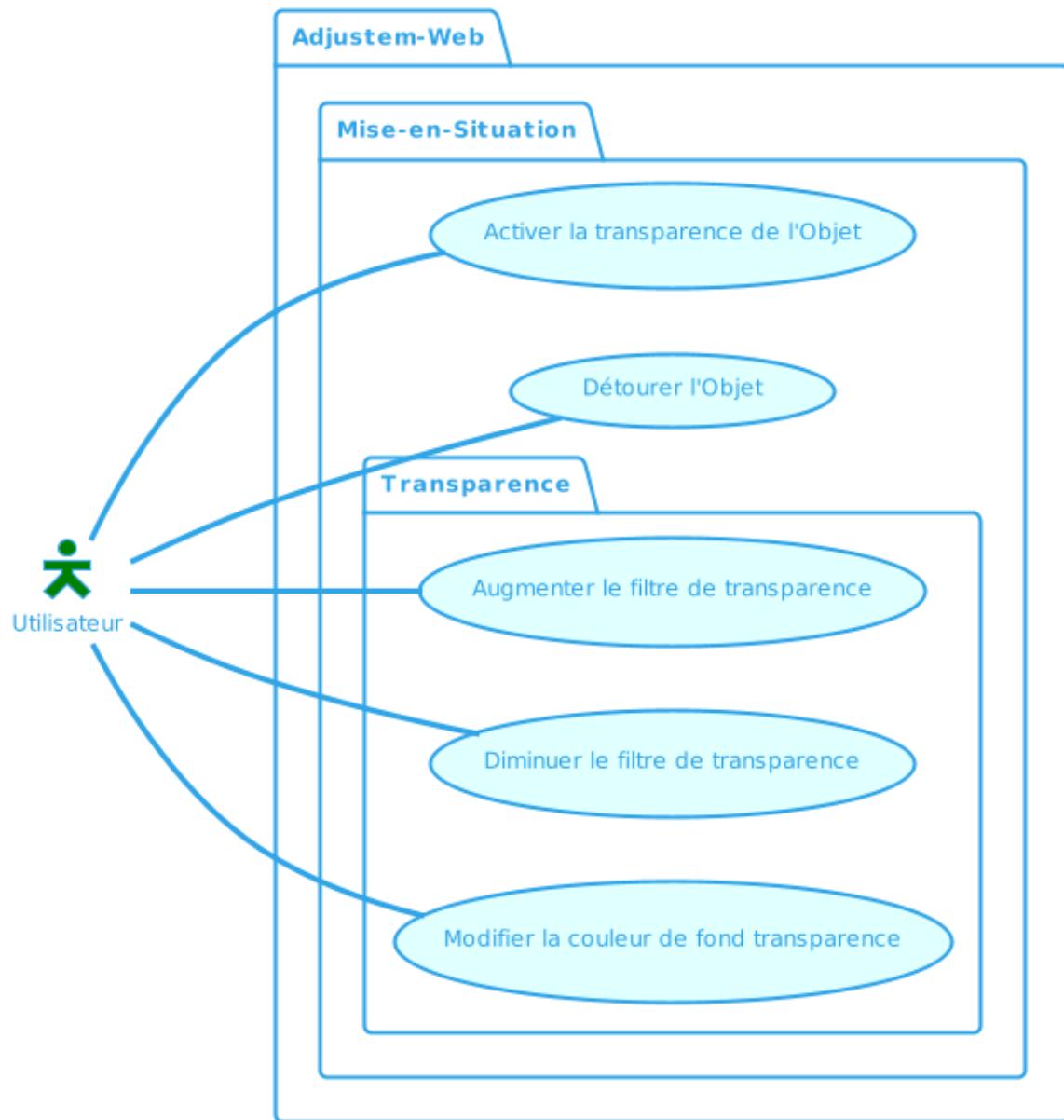


fig 7, 8 et 9 : Diagrammes de cas d'utilisation des fonctionnalités de la Mise en Situation

---

## 6. Présentation des fonctionnalités de la Mise en Situation

### A ) Chargement des données

Au départ il était prévu de fournir à l'application Mise en Situation les Ids du devis et de la ligne de devis. Le but était de pouvoir enregistrer en base une version base 64 de l'image obtenue avec le module de Mise en Situation lorsque celui-ci est lancé depuis une ligne de devis dans l'Adjustem-Web. Des problèmes évidents de l'impact du stockage des images en base64 (alourdissant en moyenne de 30% la taille d'une image) ont poussé la société Adjustem à abandonner cette fonctionnalité.

La Mise en Situation charge, au démarrage de l'application, deux images : une menuiserie ou Objet et un bâtiment ou Vue. Sachant que l'utilisation de cookies est proscrite. J'ai mis en place un système de gestion des données par le biais des variables de session PHP.

Ici plusieurs cas possibles

I ) Ces deux images (ou juste l'une des deux) peuvent être passées en paramètre par l'application ayant lancé le processus (suite logicielle Adjustem-Web). Dans ce cas l'image de la vue et/ou de l'objet doivent être enregistrés sous forme d'images en base64 dans un conteneur spécialement créé dans les variables de session PHP par l'Adjustem-Web. Cela concerne une menuiserie contenue dans un devis déjà établi et la photo de bâtiment du client pour pouvoir simuler visuellement la pose de la menuiserie.

II ) L'une ou les deux images sont issues du chargement d'une image par le biais de l'utilisation du module (cas du chargement du module après avoir changé la Vue ou l'Objet avec la fonction appropriée). Cette ou ces images sont stockées sous forme d'images en base64 dans un autre conteneur des variables de session PHP

III ) Si l'une ou les deux images sont manquantes, la Mise en Situation charge les images par défaut contenues dans les ressources de l'application.

La Mise en Situation devait pouvoir charger également les variables d'utilisation en mémoire (précédent modes utilisés, couleurs personnalisées, position des fenêtres ... ). Mais la conservation de ces informations n'a pas été souhaitée.

---

```
// bloc de chargement de l'image de l'Objet
if (isset($_SESSION["ObjetDevis"])) {
    $_SESSION["fichierObj"] = $_SESSION["ObjetDevis"];
} else {
    if (isset($_FILES["fichierObj"])) {
        $_SESSION["fichierObj"] = $_FILES["fichierObj"];
    }
}

if (isset($_SESSION["fichierObj"])) {
    $path0 = $_SESSION["fichierObj"]['name'];
    $type0 = pathinfo($path0, PATHINFO_EXTENSION);
    $path0 = $_SESSION["fichierObj"]['tmp_name'];
    if (file_exists($path0)) {
        $data0 = file_get_contents($path0);
        $img64Obj = 'data:image/' . $type0 . ';base64,' . base64_encode($data0);
        $_SESSION["ObjCourante"] = $img64Obj;
    } else {
        if (isset($_SESSION["ObjCourante"])) {
            $img64Obj = $_SESSION["ObjCourante"];
        }
    }
}

$data = [
    'img64Obj' => $img64Obj,
    'img64Vue' => $img64Vue,
    'mode' => $mode,
    'flip' => $flip
]
return view('pages/mise_en_situation/mise_en_situation', $data);
```

---

## B ) Analyse du système et Chargement de l'application

La Mise en Situation recherche le type de terminal sur lequel va être exécutée l'application. En effet la Mise en Situation est une application gérant l'interaction tactile. Le dessin est possible au doigt mais les procédures de dessin seront alors différentes de celles utilisées pour le dessin à la souris.

Le navigateur web utilisé pour exécuter l'application est analysé. La Mise en Situation a été développée pour être utilisée sous Chrome, Edge et Firefox. Le fonctionnement est possible sous d'autres navigateurs (comme Opera par exemple) mais tous n'ont pas pu être testé faute de temps. De plus, certains navigateurs plus anciens ne permettent pas l'utilisation correcte de l'élément canvas absolument nécessaire au bon fonctionnement de l'application. Une partie non négligeable de la clientèle de la société Adjustem utilise encore Internet Explorer par exemple, l'information sur le navigateur utilisé est vitale pour le bon déploiement de l'application.

La Mise en Situation analyse la taille de la fenêtre du navigateur. Étant une application de dessin, la Mise en Situation doit pouvoir bénéficier de la plus grande part d'écran disponible pour afficher avec le plus de fidélité possible les images. Il faut toujours respecter les proportions de l'image de Vue pour conserver la plus grande part de réalisme dans le rendu de la simulation. Ainsi la taille de la fenêtre du navigateur détermine la taille d'affichage de la Vue.

La partie responsive de l'application, gérée par Bootstrap, est programmée pour toujours attribuer un maximum d'espace à la Vue. Lorsque la largeur de fenêtre du navigateur se réduit à une valeur inférieure à 1200 pixels, l'application s'affiche en mode colonne pour que la Vue profite du maximum d'espace possible. Pour éviter tout désagrément à un éventuel redimensionnement effectué par l'utilisateur, la Mise en Situation se rechargera à chaque changement de taille.

Chaque recharge de l'application affiche donc une Vue vierge de tout dessin et de toute option.

---

## C ) Fonctionnalités du module de Mise en Situation

L'utilisateur peut donc :

- Activer/désactiver l'aide contextuelle
- Changer l'image de la menuiserie/Objet
- Changer l'image du bâtiment/Vue
- Implanter l'Objet sur la Vue en mode Proportionnel à la souris ou en tactile
- Implanter l'Objet sur la Vue de manière libre sans contrainte de ratio à la souris ou en tactile
- Implanter l'Objet sur la Vue en l'incluant dans un quadrilatère dont les quatre points sont définis par l'utilisateur
- Exporter l'image de l'Objet seul
- Exporter la vue de la Mise en Situation
  
- Effectuer un détourage de l'image de l'Objet
- Passer l'image de l'Objet en mode transparence
- Augmenter du filtre de la transparence
- Diminuer du filtre de la transparence
- Changement de la couleur de fond dans la miniature de l'objet
  
- Ajouter une mention
- Personnaliser cette mention
- Changer la mention de place
- Changer la couleur de la mention

---

## 7. Développement avec CodeIgniter 4

CodeIgniter est un Framework PHP simple, léger et performant, sur le modèle de développement Model Vue Controller pouvant fonctionner avec le logiciel de gestion de dépendance PHP Composer. C'est sur cette architecture MVC que le projet Mise en Situation est bâti de manière simple.

### Model : SituationModel.php

Le Model est la partie du programme qui prend en charge la communication avec la base de données. C'est par les fonctionnalités développées dans ce fichiers source que la Mise en Situation met à jour le champ **Ligne.image** à l'aide de cette fonction PHP :

```
public function ajouterImageLigne($id_ligne,$image) {  
    $db = Database::connect();  
    $db->table('ligne')  
        ->set('image', (!empty($image) ? $image : ''))  
        ->where('id_ligne',$id_ligne)  
        ->update();  
    $query = $db->insertID();  
    return $query;  
}
```

Cette partie est malheureusement non implémentée dans la version de l'application présentée dans le rapport. Le code fonctionnel mis à disposition dans le SituationModel.php sera repris en temps voulu par l'entreprise Adjustem.

### View : mise\_en\_situation.php

La partie View est l'Interface-Homme-Machine de l'application. Elle contient le code HTML5 et fait appel au script JavaScript, au fichier CSS et Bootstrap pour la mise en forme et le responsive de la page.

Afin de faciliter le développement, la génération d'un squelette HTML a été une première étape permettant de jeter les bases de l'application. Cela m'a permis de présenter les fonctionnalités au fur et à mesure de leur développement. Et donc d'avoir régulièrement des retours de l'équipe d'intégration d'Adjustem.



*fig 10 : maquette.html qui a servi de base au développement de la View*

### **Controller : Situation.php**

C'est le fichier sources qui va piloter l'ensemble de l'application. Son action sera d'interroger les variables de session à chaque reload, de charger la View en lui passant les paramètres récoltés précédemment et au final de charger en base le résultat de la mise en situation par le biais du SituationModel.php.

### **Ressources : Assets**

Afin de fonctionner correctement la Mise en Situation va puiser dans le dossier Assets : Les images par défaut et logo, un fichier CSS, et un fichier JS.

L'immense majorité du travail de l'application est contenue dans le fichier JavaScript qui est le véritable moteur de toute action commandée par l'utilisateur.

---

## Point remarquable du fonctionnement de la View

Sécurisation automatique de l'application: chaque changement de l'image de l'Objet ou de la Vue se fait par le biais d'un <input> HTML de type file dont la soumission du formulaire se fait avec une méthode **post**.

CodeIgniter 4 impose l'usage de la méthode csrf\_field() pour l'utilisation de formulaire HTML afin de prévenir tout contournement de systèmes d'authentification (attaque de type Cross-site request forgery)

```
<form id='changePicture' action="" method="post" enctype="multipart/form-data"
hidden>
    <?= csrf_field() ?>
    <input id="inpVueSet" class="input-file" type="file" name="fichierVue"
accept="image/*">
    <input id="inpObjSet" class="input-file" type="file" name="fichierObj"
accept="image/*">
</form>
```

## Points remarquables du dessin et interaction avec la Vue

La Mise en Situation est avant tout une application de dessin.

Pour les modes de dessin proportionnel et personnalisé :

L'application permet de dessiner une menuiserie sur l'image d'un bâtiment au moyen du click et resize avec le curseur de la souris. Pour pouvoir visualiser en temps réel la taille de la menuiserie à dessiner selon son emplacement sur l'image de la vue, un rectangle de sélection sera dessiné. Les dimensions du rectangle seront mises à jour à chaque changement de position du curseur de la souris.

La gestion de la souris ou du touch ont été des étapes importantes de la phase de conception et la partie spécifique du rectangle de sélection a été un premier challenge à relever.

Voici trois diagrammes d'activités

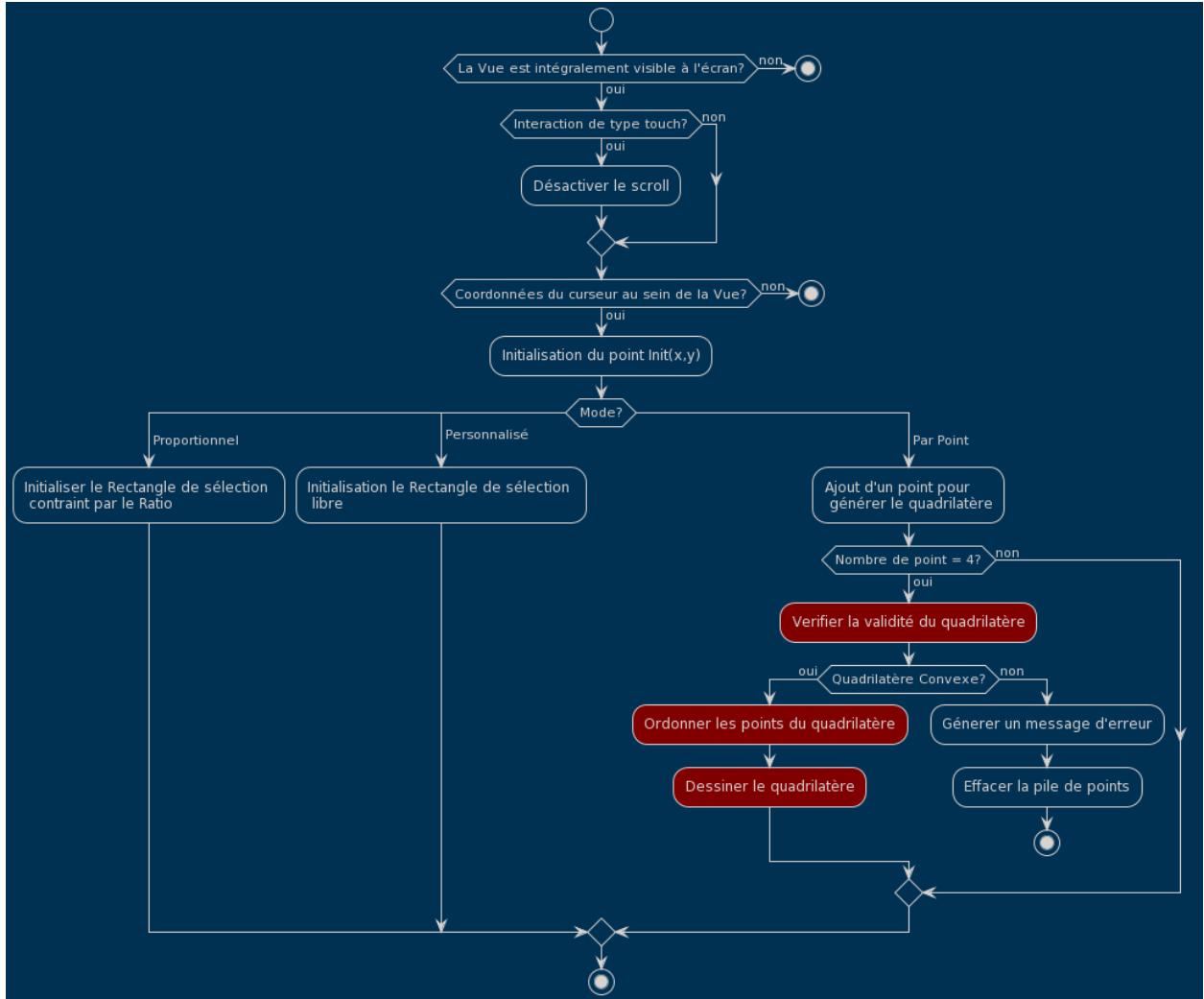


fig 11 : Fonctionnement du mousedown pour la partie JavaScript

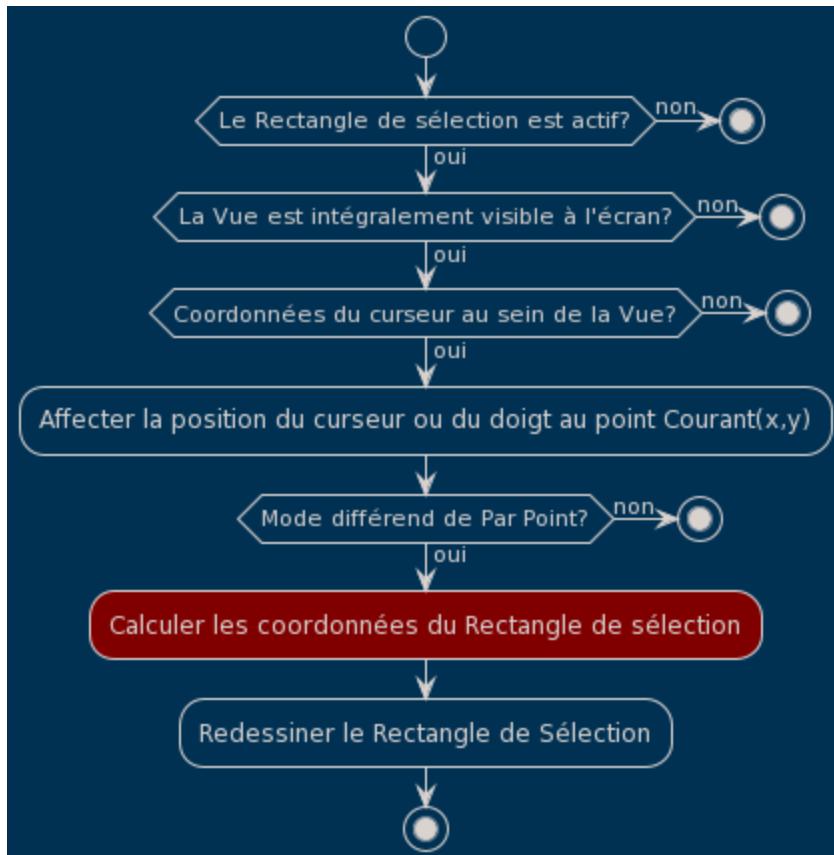


fig 12 : Fonctionnement du mousemove pour la partie JavaScript

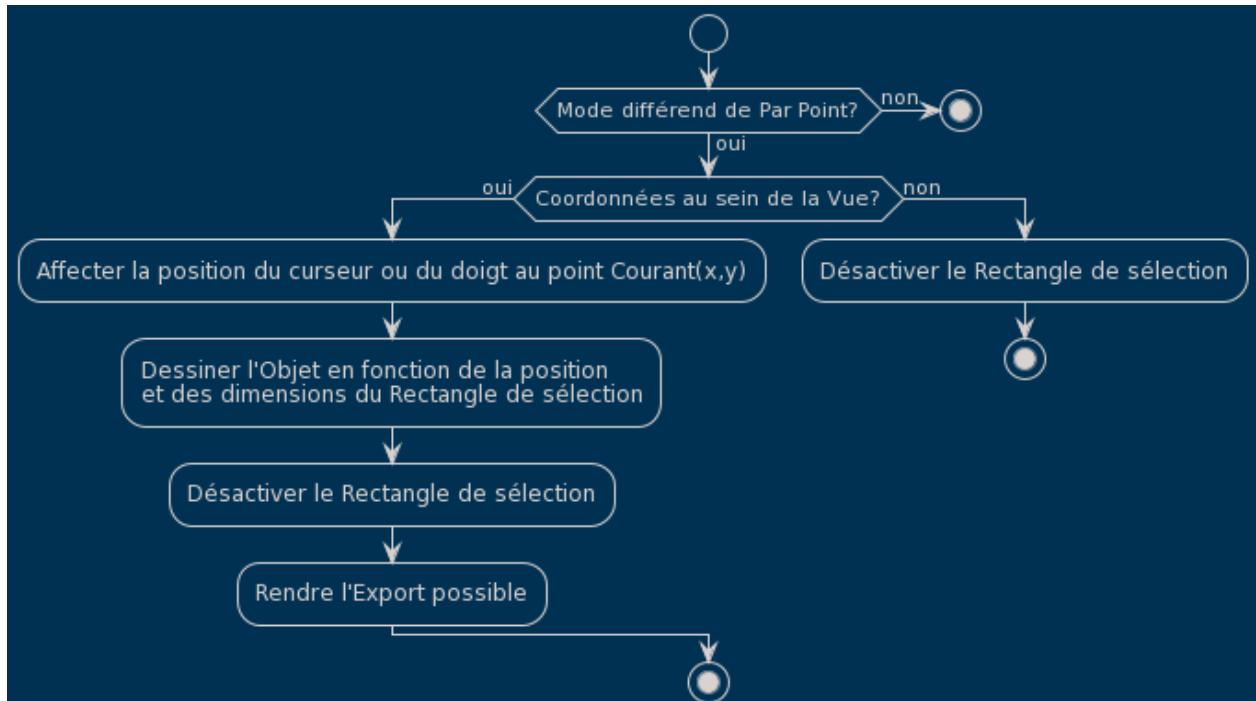


fig 13 : Fonctionnement du mouseup pour la partie JavaScript

---

Ces diagrammes couvrent toute la partie interaction entre la souris et la zone de dessins de manière non exhaustive.

Il est à noter que les actions sur fond rouge ont représenté un important travail de conception et développement

**Calculer les coordonnées du Rectangle de sélection**  
qui sera géré par la fonction JavaScript `setRectangle()`.

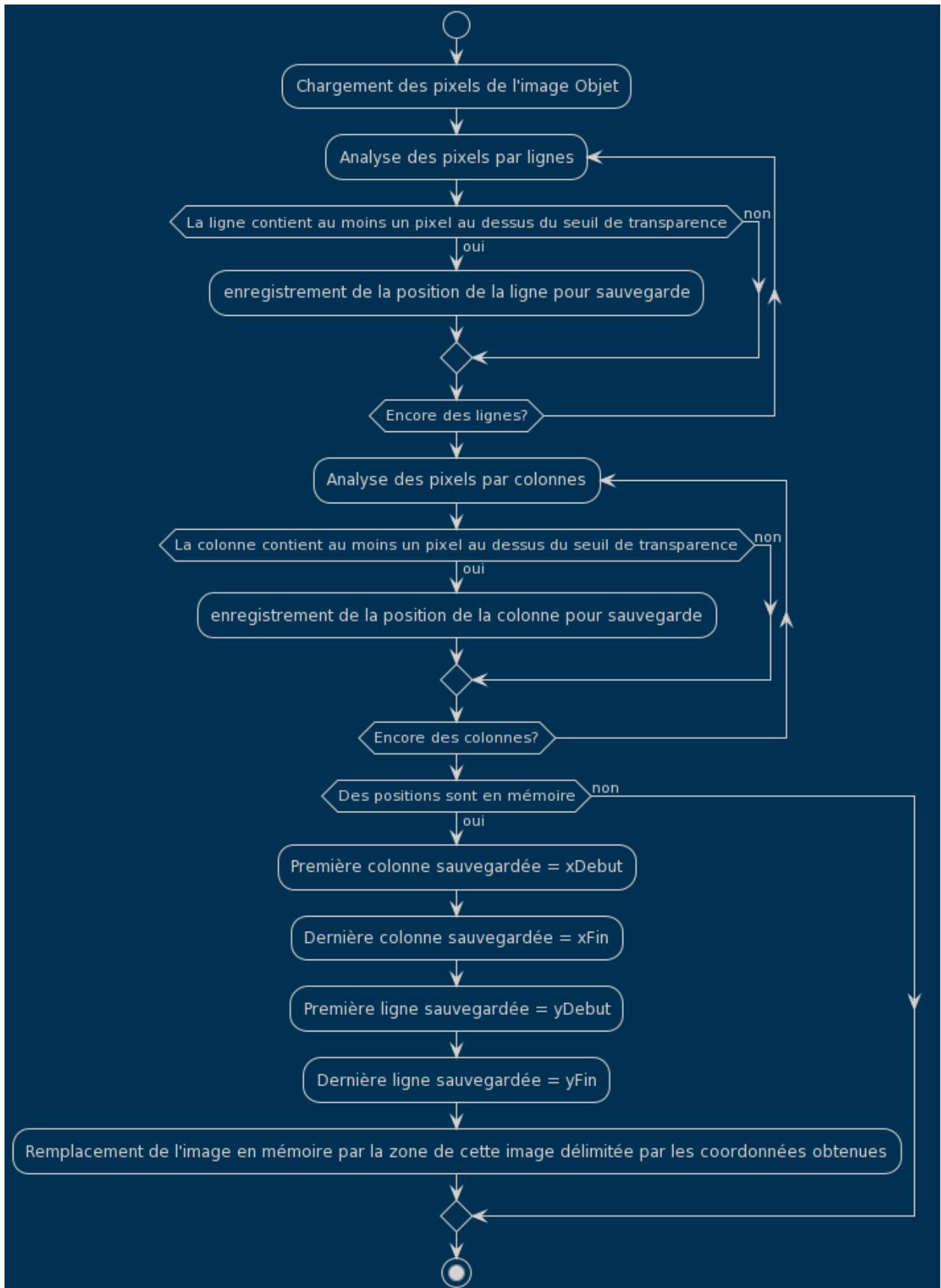
**Vérifier la validité du quadrilatère et Ordonner les points du quadrilatère**  
qui seront gérés par la fonction JavaScript `setQuadPerspective()`

**Dessiner le quadrilatère**  
qui sera géré par la fonction JavaScript `drawImageInPerspective()`

**Gestion de la transparence**  
qui sera géré par la fonction JavaScript `transparence()`  
La dernière fonctionnalité implémentée dans le Module de Mise en Situation fût la gestion de la transparence. En utilisant la gestion des pixels du canvas par le biais de son objet `ImageData`. J'ai mis en place une fonction pour rendre transparent tout pixel dont la moyenne des valeurs des pixels RGB était supérieur à un seuil passé en paramètre.

Cette transparence sera aussi utilisée par la fonction `detourage()` afin de supprimer toute partie non utile autour d'une image de menuiserie. Dont je publie le diagramme de séquence ici.

*ci dessous : fig 14 : Fonctionnement du detourage pour la partie JavaScript*

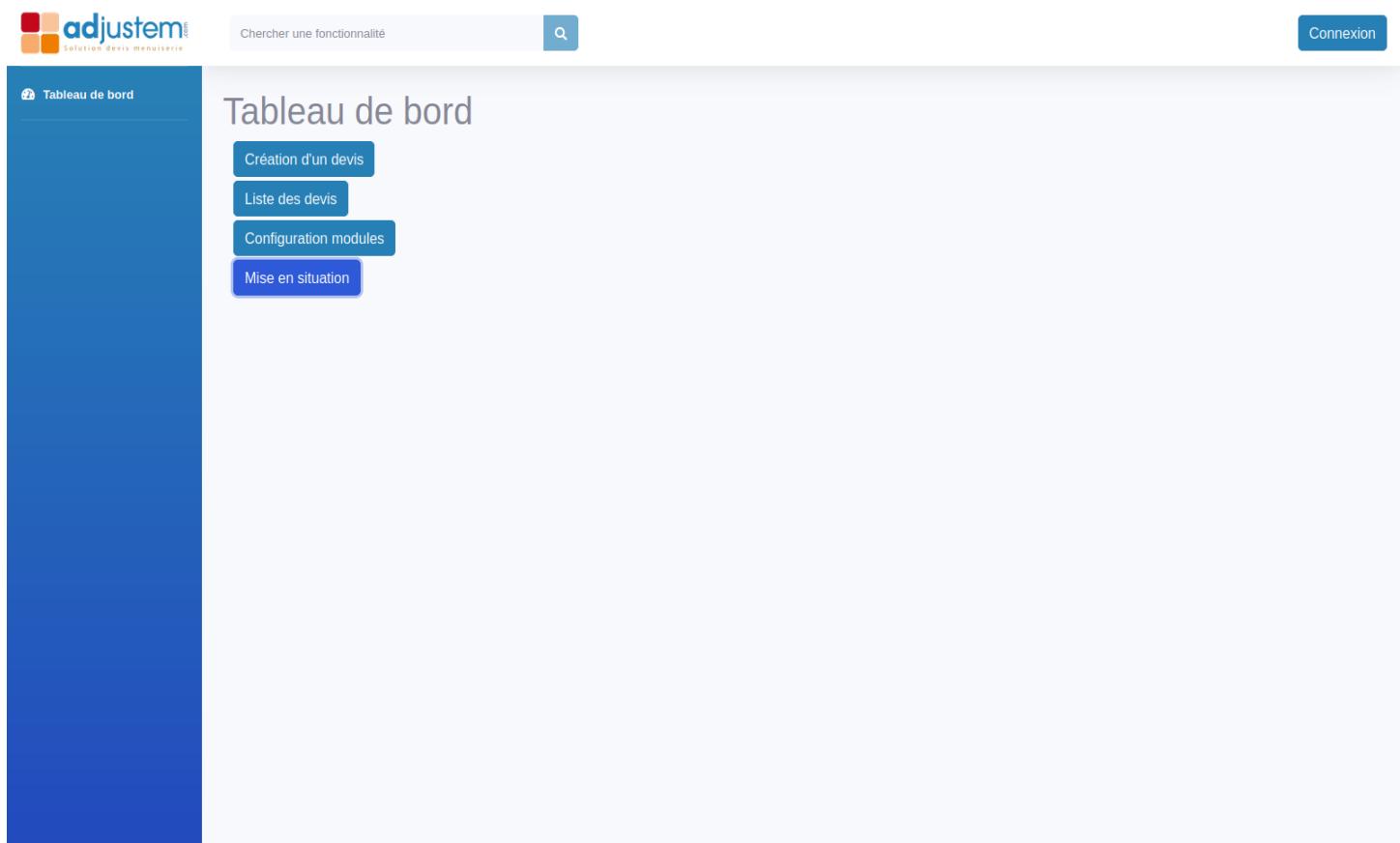


---

## 8. Résultats et Intégration dans l'Adjustem Web

La Mise en Situation est actuellement intégrée dans la dernière version de l'Adjustem Web en mode libre. L'utilisation sous Devis est conditionnée à un finalisation de la gestion des Devis. Toutefois il est à noter que le module de Mise en Situation remplit tout à fait le cahier des charges (avec ses ajouts ultérieurs). Le déploiement du mode sous Devis est près.

Voici la page d'accueil de l'Adjustem Web. La Mise en Situation y est accessible en mode libre.



The screenshot shows the Adjustem Web dashboard. At the top, there is a header with the Adjustem logo, a search bar labeled "Chercher une fonctionnalité", and a "Connexion" button. On the left, a blue sidebar contains a "Tableau de bord" section with four buttons: "Création d'un devis", "Liste des devis", "Configuration modules", and "Mise en situation". The main content area is currently empty.

---

## 9. Déploiements

Pour tester mon module je l'ai déployé avec succès en premier lieu sur le réseau local d'Adjustem sur serveur Wamp tournant sur un système Windows 7 64b.

Puis j'ai pu déployer le module sur un hébergement personnel à domicile sur un raspberry-pi avec serveur Apache afin de tester les fonctionnalités sur tablettes. Mais soucieux de tester les fonctionnalités sur une variété plus conséquente de terminaux j'ai déployé sur un hébergement web personnel le fruit de mon travail.

Les premiers (et derniers jusqu'à maintenant) problèmes rencontrés étaient liés au poids des images traité par le module. Très exactement au moment de la conversion en base64 d'images dont le poids dépasse les 8mb. Le problème a été résolu en augmentant les valeurs du fichier de configuration PHP.ini sur serveur :

```
POST_MAX_SIZE = 16M  
OUTPUT_BUFFERING = 16384
```

L'adresse <https://abmanilio.fr> permet l'accès libre au module de Mise en Situation qui permettra d'illustrer correctement ce rapport de stage.

---

## 10. Conclusion

Dans ce rapport j'ai essayé de présenter de manière chronologique le développement d'une application de Mise en Situation développée en Php, HTML et Javascript. Je peux ici déclarer avoir eu beaucoup de chance d'avoir eu à fournir ce travail. J'ai vraiment beaucoup apprécié le travail de recherche conséquent qui fût nécessaire à l'obtention du résultat et j'ai eu la preuve que JavaScript est une plateforme de développement réellement immense et ses possibilités ne le sont pas moins. J'ai beaucoup de fierté d'avoir pu terminer le travail qui m'a été demandé dans les temps. Toutefois j'ai conscience que le travail de recherche technique a pris le pas sur la qualité de la conception. Être plus aguerris au langage JavaScript m'aurait sans doute permis de mieux structurer mon code. L'interaction en temps réel de la Mise en Situation m'a poussé tout naturellement vers mes vieux mécanismes de développeur C et j'ai cette impression d'être passé à côté de nombreuses possibilités d'arriver à mes fins.

Mais ce qui finalement est le plus important, après onze mois de reconversion professionnelle, c'est d'avoir fini ce projet avec succès. J'ai pu donner satisfaction à une entreprise à l'aide de mon travail de concepteur et de développeur en respectant les termes d'un cahier des charges.

## 11. Annexes

Fonction transparence() en js

```
***** Fonction transparence *****

function transparence(monImage) {
    /* le but est de transformer la couleur de fond en fond transparent */

    var tmpCanvas = document.createElement("canvas"); //creation d'un canvas
    tmpCanvas.width = monImage.naturalWidth; //definition de la largeur
    tmpCanvas.height = monImage.naturalHeight; //definition de la hauteur
    var tmpContext = tmpCanvas.getContext("2d"); //recuperation du contexte
    tmpContext.drawImage(monImage, 0, 0); //dessin de l'image

    var tmpMap = tmpContext.getImageData(0, 0, tmpCanvas.width, tmpCanvas.height); //recuperation du
    Image Data
    var tmpImgData = tmpMap.data; //creation d'un raccourci vers les datas

    for (var i = 0; i < tmpImgData.length; i += 4) {

        if (tmpImgData[i] >= transValue ) {
            if (tmpImgData[i + 1] >= transValue ) {
                if (tmpImgData[i + 2] >= transValue ) {
                    tmpImgData[i] = 0;
                    tmpImgData[i + 1] = 0;
                    tmpImgData[i + 2] = 0;
                    tmpImgData[i + 3] = 0;
                }
            }
        }
    }

    ctxT.putImageData(tmpMap, 0, 0);
}
```

Fonction detourage en js

```

● ● ●

*****      Fonction detourage      *****

function detourage(monImage) {
    /* le but est de recuperer la partie utile de l'image dans un rectangle */

    var tmpCanvas = document.createElement("canvas"); //creation d'un canvas
    tmpCanvas.width = monImage.naturalWidth; //definition de la largeur
    tmpCanvas.height = monImage.naturalHeight; //definition de la hauteur
    var tmpContext = tmpCanvas.getContext("2d"); //recuperation du contexte
    tmpContext.drawImage(monImage, 0, 0); //dessin de l'image

    var tmpMap = tmpContext.getImageData(0, 0, tmpCanvas.width, tmpCanvas.height); //recuperation du
    Image Data
    var tmpImgData = tmpMap.data; //creation d'un raccourci vers les datas

    var whiteLine;
    var whiteCol = true;
    var nbLigne = 0;
    var nbColonne = 0;
    var i,j,k;
    var meslignes = [];
    var mescolonnes = [];

    for ( i = 0; i < tmpImgData.length; i += monImage.naturalWidth * 4) { // cette boucle permet de
    parcourir chaque ligne de pixels de l'image
        nbLigne++;
        whiteLine = true;

        for ( j = 0; j < (monImage.naturalWidth * 4); j += 4) {
            k = i + j;
            if (tmpImgData[k] <= transValue) { whiteLine =false;}
            if (tmpImgData[k + 1] <= transValue) {whiteLine =false;}
            if (tmpImgData[k + 2] <= transValue) {whiteLine =false;}
        }

        if (!whiteLine) {meslignes.push(nbLigne);}

    }

    for (i = 0; i < monImage.naturalWidth * 4; i += 4) { // cette boucle permet de parcourir chaque
    colonne de pixels de l'image
        nbColonne++;
        whiteCol = true;

        for (j = 0; j < tmpImgData.length; j += monImage.naturalWidth * 4) {
            k = i+j;
            if (tmpImgData[k] <= transValue) { whiteCol =false;}
            if (tmpImgData[k + 1] <= transValue) {whiteCol =false;}
            if (tmpImgData[k + 2] <= transValue) {whiteCol =false;}
        }

        if (!whiteCol) {mescolonnes.push(nbColonne);}

    }

    canvasT.width = mescolonnes.length ;
    canvasT.height = meslignes.length ;
    tmpMap = tmpContext.getImageData(mescolonnes[0] -1, meslignes[0] -1, mescolonnes.length + 20,
    meslignes.length + 20);
    ctxT.putImageData(tmpMap, 0, 0);

    var image = new Image();
    image.src = canvasT.toDataURL();
    return image;

}

```