

RAPPORT DE STAGE

STAGE DEVELOPPEUR FULL-STACK

EL SAADAWY AHMED

TUTEUR EN ENTREPRISE :

BENHABIB Abdelkrim

TUTEUR ACADÉMIQUE :

BONINO Marc

**Du
22 Janvier**

**Au
22 Mars**



VifAuto
48 Rue de Beaumont,
95270 Noisy-sur-Oise

Université de la Sorbonne
Paris Nord
99 Av. Jean Baptiste Clément,
93430 Villetaneuse

UNIVERSITÉ
SORBONNE
PARIS NORD

1. Remerciements

Tout d'abord, je tiens à remercier mon tuteur de stage BENHABIB Abdelkrim pour mon stage au sein de son équipe, pour sa patience, son soutien, pour avoir suivi l'évolution du stage et pour avoir répondu à toutes mes questions.

Je remercie par ailleurs, Madame ZARGAYOUNA et Madame BENABOU, pour leurs disponibilités et pour m'avoir grandement aidé dans la recherche de stage, aussi bien moi que tous les étudiants de deuxième année.

Enfin, j'aimerais également remercier mon tuteur académique Monsieur BONINO pour son temps et pour avoir suivi l'évolution de mon stage.

2. Sommaire

1. Remerciements	1
2. Sommaire	2
3. Introduction	4
4. L'entreprise	5
4.1 Présentation de l'entreprise	5
4.1.1 Structure et organisation.	5
4.1.2 Mission et stratégie	5
4.1.3 Produits et/ou services	6
4.1.4 Clients/cibles	6
4.1.5 Fournisseurs	6
4.1.6 Projets informatiques	6
5. Sujet de stage	7
5.1 Contexte	7
5.2 Fondations du projet	7
5.2.1 Cahier des charges	7
5.2.2 Maquettes et base de données	8
5.3 Conception et développement	10
5.3.1 Choix technologiques	10
5.3.2 Structure de la base de données	12
5.3.3 Développement frontend	15
5.3.4 Développement backend	16
5.4 Problématique et tests	19
5.4.1 Problèmes rencontrés	19
5.4.2 Solutions	19
5.5 Bilan du projet	20
5.6 Suite du stage	21

6. Bilan Personnel	22
7. Conclusion	22
8. Lexique	23
9. Annexes	25

3. Introduction

Dans le contexte de mon stage de deuxième année à l'Université de la Sorbonne Paris Nord et afin de valider ma formation de BUT Informatique, j'ai eu l'opportunité de travailler chez Vif Auto, du 29 Janvier au 22 Mars de l'année 2024, à Noisy-sur-Oise (95270).

Vif Auto est un magasin spécialisé dans la vente de pièces automobiles et de carrosserie. Ils permettent à leurs clients de commander une pièce et la recevoir dans la journée.

Au sein de leur groupe, j'ai pu travailler sur la création d'un outil de gestion du stock de peinture. Cet outil facilitera l'utilisation et le réapprovisionnement de la peinture. L'entreprise a agi en tant que client et j'ai travaillé en autonomie tout en recevant l'aide nécessaire de mon tuteur de stage quand j'étais en difficulté. Cette manière de travailler m'a permis de me questionner sur comment les compétences acquises au cours de mon BUT Informatique pourraient me servir lors de projet professionnel.

Dans ce rapport, je commencerais par présenter l'entreprise VifAuto, ses partenaires, sa structure, ses missions et ses clients. Par la suite je présenterais mon poste, les technologies utilisées et enfin les missions que j'ai pu réaliser. Puis je finirais par les compétences et connaissances acquises tout en long de cette expérience et conclurai en faisant un bilan complet de cette expérience.

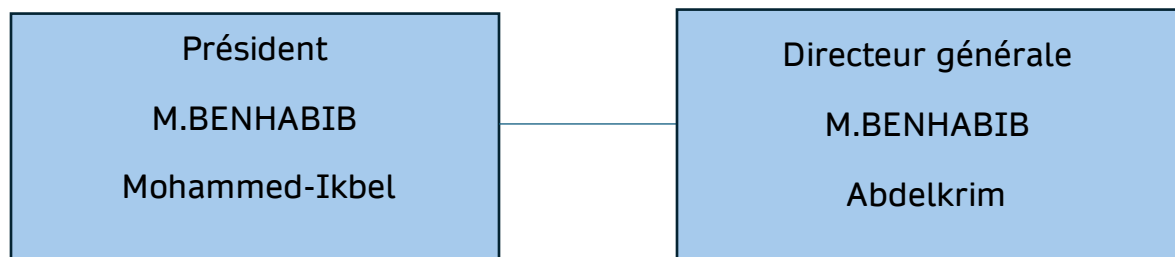
4. L'entreprise

4.1 Présentation de l'entreprise

L'entreprise Vif Auto, fondée en décembre 2022 par deux ingénieurs en informatique avec un capital de 15 000 €, est spécialisée dans la vente de pièces automobiles, mécaniques et carrosserie. Ils ont lancé simultanément la première version de leur site web et leur magasin physique. Leur bureau et leur magasin est situé à Noisy-sur-Oise. La société emploie des pratiques originales pour répondre aux besoins spécifiques de leurs clients. L'équipe est composée du président de l'entreprise et du directeur générale.

4.1.1 Structure et organisation

L'équipe est composée du président de l'entreprise, BENHABIB Mohammed-Ikbel et du directeur générale, BENHABI Abdelkrim, mon tuteur de stage, ancien étudiant en informatique.



4.1.2 Mission et stratégie

La mission principale de Vif Auto est de maximiser l'attraction et la fidélisation des clients en garantissant la livraison des pièces automobiles dans un délai de 24 heures mais aussi à fournir des produits d'origine pour assurer la qualité et la fiabilité. En se concentrant sur la rapidité et la fiabilité, Vif Auto cherche à établir une relation de long terme avec chaque client. Cette mission permet à Vif Auto d'avoir une stratégie qui se démarque de leurs concurrents et d'avoir une relation de confiance avec leurs clients qui sont souvent des mécaniciens.

4.1.3 Produits et/ou services

En plus de la vente de pièces automobiles, Vif Auto offre un service de préparation de mélanges de peintures pour les carrossiers. Ce service est fait sur mesure grâce à une machine et permet d'avoir la couleur précise, cela garantit des résultats parfaits et la satisfaction de leurs clients.

4.1.4 Clients/cibles

Vif Auto fournit leurs produits à la fois à des particuliers qui cherchent des pièces spécifiques pour leur véhicules mais aussi à des garagistes professionnels qui eux ont besoin d'un approvisionnement régulier en pièces et en peintures. Vif Auto fait aussi en sorte d'établir une bonne relation avec les garages de leur secteur pour continuer à s'étendre.

4.1.5 Fournisseurs

Vif Auto se fournit auprès d'entreprises telles que DropIDF, Procodis, OttoGo ou encore MecaDepot, cela leur permet d'étendre leur gamme de produits et de comparer les prix de chacun. Cette stratégie d'approvisionnement diversifiée assure une offre constante et concurrentiel.

4.1.6 Projets informatiques

Le directeur général de Vif Auto était, comme dit précédemment, un ancien étudiant en informatique, il s'est occupé de la conception du site web de l'entreprise avec un ancien camarade à lui. Le site a été développé en VueJs pour le frontend et NodeJs pour le backend, ce site permet de faire un devis pour la commande d'une pièce ou pour contacter l'entreprise. Ce site témoigne de l'engagement de l'entreprise dans la digitalisation de ses services et permet d'améliorer l'accessibilité et l'interaction avec leurs clients. Ils ont aussi un tableau Excel qui permet de calculer le niveau de stocks de peinture. Ce tableau utilisé par Vif Auto s'est vite avéré être limité pour la gestion du stock et c'est donc là où le besoin de créer un outil de gestion personnalisé c'est manifester. Ce besoin nécessaire à l'entreprise va devenir ma mission principale au sein de l'entreprise.

5. Sujet de stage

5.1 Contexte

Lors de ma première rencontre avec mon tuteur de stage, on m'a proposé quatre projets, un premier projet qui consistait à créer un site répertoriant les fournisseurs ainsi que les marques associées, un autre projet qui consistait à créer un outil de gestion du stock de peinture, un logiciel de facturation et enfin une extension google chrome à implémenter.

J'ai trouvé ce stage une semaine après la date initialement prévue pour le stage, donc lorsque je suis arrivé pour la première fois dans les locaux, l'autre stagiaire présent avec moi avait commencé le premier projet. Cela n'avait pas une grande importance car je voulais faire le deuxième projet. Ce projet se résume à créer un outil qui estime la quantité de peinture encore en stock en magasin selon les quantités utilisées. Ces dernières seront spécifiées dans un fichier XML à chaque mélange préparé.

5.2 Fondations du projet

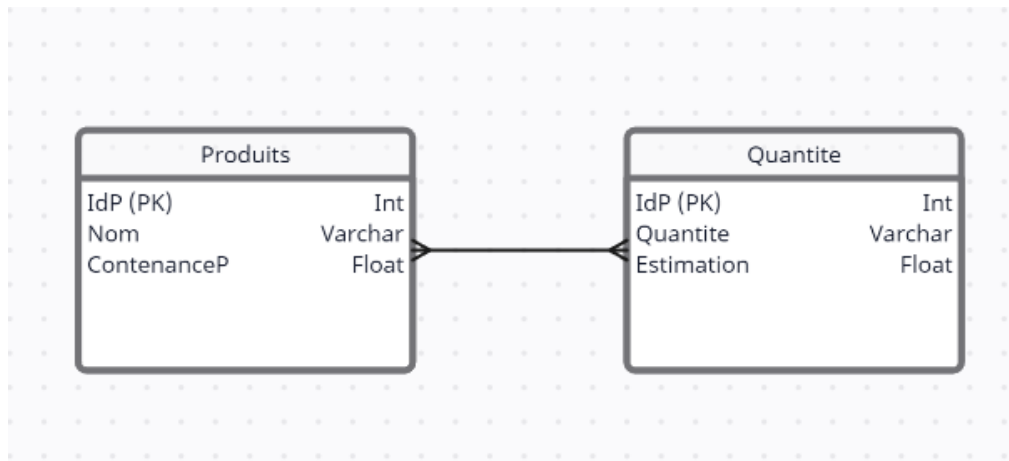
La première semaine du stage m'a permis de préparer avant le développement du site web, M.Benhbab m'a demandé de lui fournir un petit cahier des charges pour voir si j'avais bien compris le projet, une maquette du futur site web, préparer la structure de la base de données et enfin de créer un répertoire GitHub pour suivre l'évolution de l'outil.

5.2.1 Cahier des charges

Pour ce qui est du cahier des charges (voir annexe 1) je n'ai pas fait un grand cahier des charges car il servait seulement à vérifier si j'avais bien compris l'essentiel du projet. Pour faire ce cahier des charges j'ai repris le modèle de l'IUT (voir annexe 2) que j'ai trouvé complet car il permet d'aborder plusieurs aspects du projet tel que les limites du projet, les exigences fonctionnelles et non fonctionnelles et les risques du projet. Mais ce modèle est aussi parfait pour montrer à mon tuteur que j'avais compris ma mission.

5.2.2 Maquettes et base de données

La base de données n'est pas très grande mais doit être solide pour le bon fonctionnement du site web et pour pouvoir accueillir le maximum de données. Au cours du stage, j'ai développé trois versions de la base de données, chacune adaptée aux exigences évolutives du projet. La première version contenait seulement le nom d'un produit, la contenance de son pot, car il y'a des pots de 1L, de 3,5L et de 0,5L, le nombre de pot du produit et l'estimation total de la quantité restante du pot.



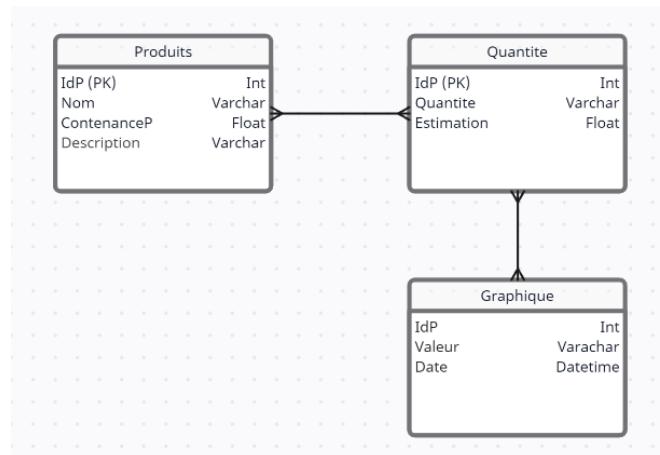
1. Première version de la base de données

Cette version de la base de données était suffisante pour entamer le développement du site web car il réunissait les infos présentes dans le tableau Excel.

	A	B	C
1	Produit	Contenance pot	Estimation quantité
2	300	1	1288,8
3	301	3,5	796,7
4	302	1	1334,1

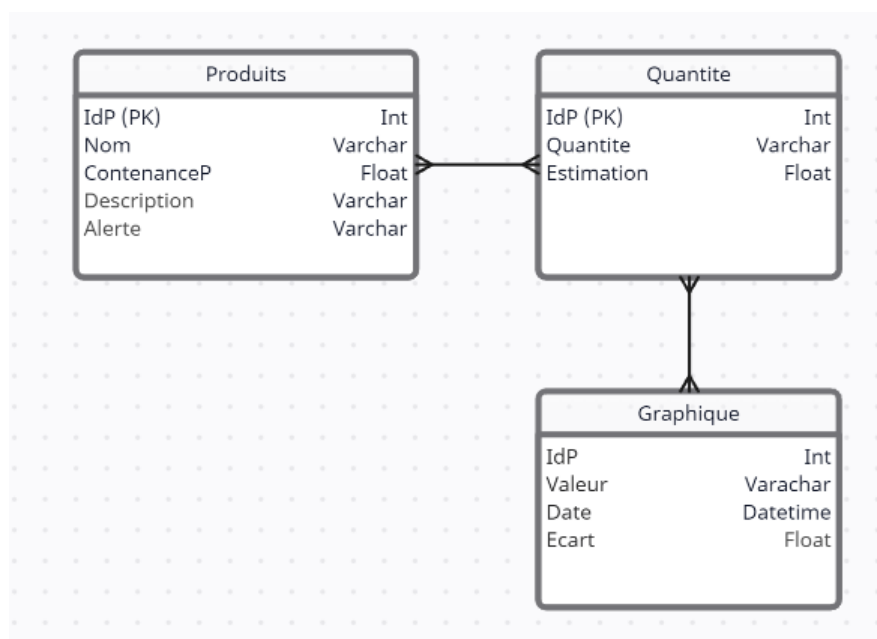
2. Visuel du ruban et de quelque donnée du tableau Excel

Par la suite mon tuteur de stage a validé ma base de données mais m'a suggéré d'ajouter une table pour la création du graphique, j'ai donc ajouté la table graphique et ajoute une colonne description dans la table Produits.



3. Deuxième version de la base de données

Pour ce qui est de la version finale de ma base de données, elle consistait à ajouter deux colonnes. Une première colonne 'alerte' dans la table produit qui permet d'alerter l'utilisateur que la quantité d'un produit est presque fini et une deuxième colonne 'écart' qui calcule l'écart de chaque estimation de produit. Il faut savoir que ces deux colonnes ont été ajoutés vers la fin du stage, soit à la sixième semaine donc mon code s'est beaucoup plus basé sur la deuxième version.



4. Troisième et dernière version de la base de données

Pour la conception de la maquette, je me suis inspiré du site web de Vif Auto (voir annexe 3), qui a comme couleurs principales le jaune et le noir. J'ai fait en sorte que la maquette soit le plus claire possible pour encore une fois vérifier si j'ai bien compris ma mission. La maquette montre que le site est

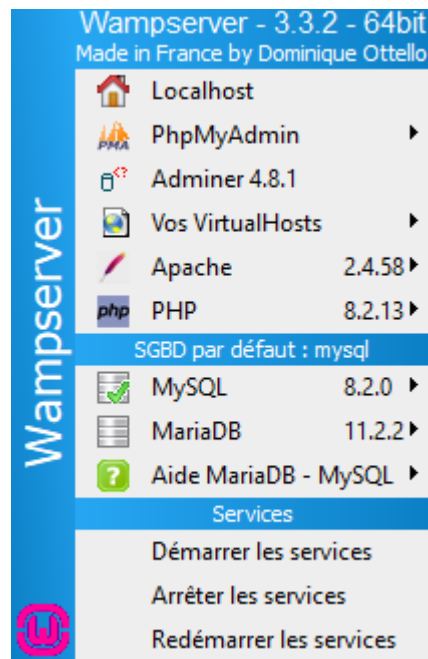
divisé en trois parties (voir annexe 4 pour la maquette complète), une première partie 'Importer' qui permet de modifier la quantité d'un produit, une deuxième partie avec la liste de tous les produits et leurs informations. Et enfin le graphique qui s'affiche selon le produit sélectionné.

5.3 Conception et développement

Après avoir effectué les préparatifs nécessaires pour la conception de l'outil, je suis prêt à passer à la programmation.

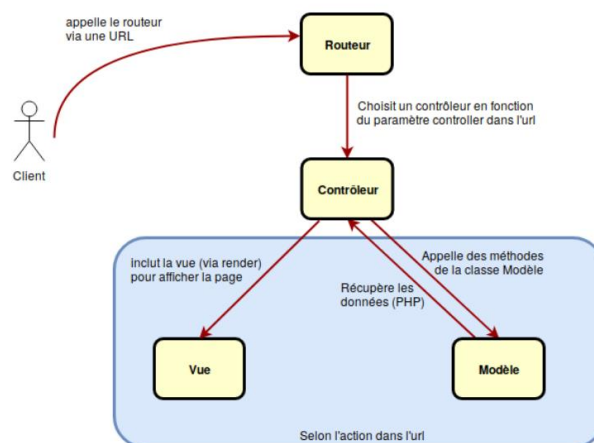
5.3.1 Choix technologiques

Comme dit précédemment le site web de Vif Auto a été programmé en VueJs et en NodeJs, mon tuteur de stage m'a demandé de m'informer sur ces langages pour savoir si je pourrai coder l'outil avec ces langages. Donc ma première semaine m'a servi de faire les préparatifs pour le projet mais aussi de m'exercer sur ces langages. J'avais en tête de faire le site en PHP qui est un langage que je maîtrise très bien, notamment grâce aux cours de développement web à l'IUT, j'en ai donc parlé à M.Benhabib qui a accepté que je code en PHP pour le frontend et le backend. Pour ce qui est de la base de données, j'ai opté pour une base MySQL car il supporte de grandes bases de données, répond très vite et est stable. Tous ces points forts de MySQL sont ce que je recherche pour le bon fonctionnement de l'outil. Pour accéder à MySQL tout en codant en PHP j'installe et j'utilise WampServer, un serveur local qui permet de voir les modifications apportées au site web du côté visuel et du côté machine.



5. Interface d'utilisation de WampServer

Pour optimiser l'organisation lors de la création du site, je choisis de coder en utilisant la structure MVC (Modèle-Vue-Contrôleur) afin de faciliter la maintenance et les mises à jour du site. Cette architecture est un modèle de conception issue de l'expérience des programmeurs. Il est très utilisé car il possède de nombreux avantages comme le fait de structurer son code de façon claire et logique mais en contrepartie, cette structure demande un temps d'apprentissage et nécessite une grande rigueur lors du développement. Le modèle s'occupe de la partie données (coté backend), la vue présente les données à l'utilisateur (coté frontend) et le contrôleur fait le lien entre les deux.



6. Interactions Modèle/Vue/Contrôleur

5.3.2 Développement Frontend

J'ai commencé le code par la partie frontend, donc en html/css dans la partie Vue de la structure MVC. Le site contient une page. Dans le [head html](#) j'ai ajouté une description, une icône et un titre, mais aussi la même police présente sur le site de Vif Auto et j'ai appelé un script [JS](#) pour la création du graphique.

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, viewport-fit=cover">
  <meta name="description" content="Outils qui permet gerer son stock de peinture">
  <title>Vif Auto</title>
  <link rel="stylesheet" href="Content/css/style.css"/>
  <link rel="icon" href="Content/image/vifauto.png" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
```

7. Code du head html

Par la suite, j'ai séparé mon body html en plusieurs balises, telles que la balises 'header', 'main' et 'footer' pour avoir une structure logique lors de la programmation. Les balises 'header' et 'footer' contiennent l'en-tête et le pied de page du site.

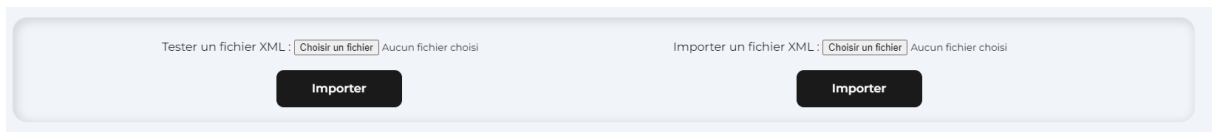


8. Visuel de l'en-tête(header)



9. Visuel du pied de page/footer)

Après que cela a été fait, je me suis occupé de la balise 'main', qui contient tous ce qui a été demandé par M.Benhbab, soit la liste des produits, le graphique mais aussi l'import de données sur les produits. La partie visuelle de l'importation manuel des données est très simple à comprendre cependant pour l'importation XML, mon tuteur avait besoin de quelques explications.

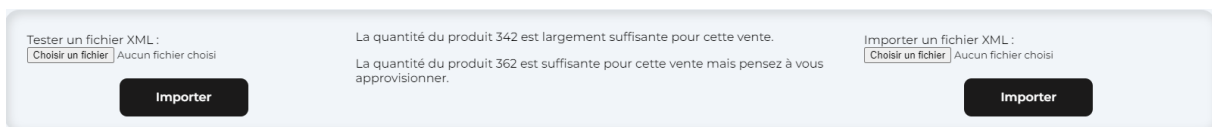


Tester un fichier XML : Aucun fichier choisi

Importer un fichier XML : Aucun fichier choisi

10. Visuel de l'importation XML

Comme nous pouvons le voir il y'a ici deux bouton importer, le premier permet de vérifier si la quantité demandée pour le mélange de peinture est suffisante pour la vente. Après avoir importé un message s'affiche pour savoir si le mélange peut avoir lieu ou pas.



Tester un fichier XML : Aucun fichier choisi

La quantité du produit 342 est largement suffisante pour cette vente.

La quantité du produit 362 est suffisante pour cette vente mais pensez à vous approvisionner.

Importer un fichier XML : Aucun fichier choisi

11. Visuel de l'importation XML après avoir testé un mélange

Dans ce cas-là le mélange peut avoir lieu mais il faudra penser à réapprovisionner le produit 362.

Cependant lorsque j'import un fichier XML dans le second bouton 'Importer', les valeurs présentes dans le XML soustraient l'estimation actuelle de la peinture.



Retrouver un produit :

Estimation <= :

[Réinitialiser la recherche](#)

12. Outil de filtrage des produits

Pour ce qui est de la liste des produits, on retrouve un outil de filtrage si l'utilisateur cherche un produit précis par son nom ou par son estimation.

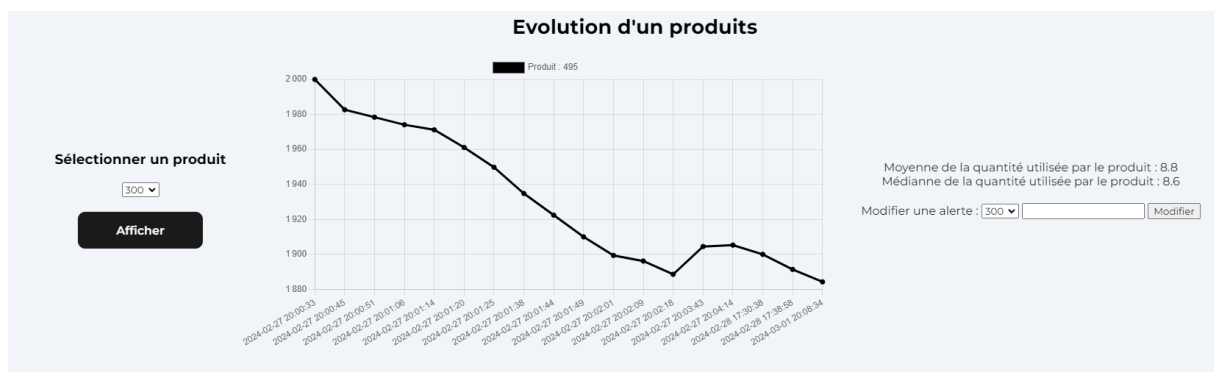
La liste des produits est sous forme de tableau avec un ruban simple qui récupéré le nom des colonnes de la base de données puis affiche toutes les infos de chaque produit comme le nom du produit, sa contenance, sa quantité, ...

Produit	Contenance par pot(mL)	Quantité	Estimation(mL)	Description	Alerte
300	1000	3 +	2187.9		500
301	3500	2 +	4111.9		1000
302	1000	2 +	1334.1		500

13. Liste des produits

Vous pouvez remarquer plusieurs choses ici tels que l'estimation qui a une couleur de fond verte mais qui peut aussi être orange lorsque l'estimation est inférieure à la colonne alerte et rouge lorsque le produit est complètement épuisé. Il y'a aussi un logo 'poubelle' à droite qui permet de supprimer un produit, bien évidemment une confirmation est demandée avant. Puis pour finir nous pouvons voir un bouton '+' qui ajoute un pot donc +1 pour la colonne quantité mais cela ajoute aussi la contenance du pot dans la colonne estimation, donc si appuie sur le '+' pour un produit qui a pot de 1000mL et a une estimation de 1200mL, l'estimation passera à 2200mL.

Pour terminer sur la partie visuelle du site on a la section du graphique qui montre l'évolution de l'estimation d'un produit choisi avec comme ordonnée l'estimation et comme abscisses la date à laquelle l'estimation a été modifié.



14. Graphique du produit 495

Ici nous pouvons sélectionner un produit à gauche puis le graphique s'affiche au milieu, enfin à droite nous pouvons voir la moyenne et la médiane de la quantité utilisées par le produit.

Voilà pour ce qui est de la partie visuelle du site web passons maintenant à la partie backend.

5.3.3 Structure de la base de données

Avant de passer au code du site web parlons encore un peu de la base de données, de la connexion entre le site et la base de données et de ses déclencheurs.

Pour lier la base de données au site web on utilise un fichier 'credentials.php' dans lequel on renseigne le login, le mot de passe et le nom de la base de données qui aura un accès total à la base de données.

```
<?php
//Information pour la connexion à la bdd
$dsn = 'mysql:host=localhost;dbname=db_vifauto';
$login = 'root';
$mdp = '';
```

15. Fichier credentials.php pour la connexion à la bdd
















Ce fichier est ensuite appelé par la commande :

```
include_once 'Utils/credentials.php';
```

dans le fichier index.php qui contient toutes les informations du site.

Un déclencheur est une procédure qui s'exécute automatiquement en réponse à certains événements sur une table ou dans une base de données. Ces événements peuvent être des opérations d'insertion, de mise à jour ou de suppression.

Dans le cas de notre site web plusieurs déclencheurs sont présent, cinq au total qui ont un nom un temps, soit avant ou après l'opération, et une opération d'ajout ou de modification.

Nom	Table	Temps	Évènement	
ajuster_quantite_avant_insert	quantite	BEFORE	INSERT	 Éditer  Exporter  Supprimer
ajuster_quantite_avant_update	quantite	BEFORE	UPDATE	 Éditer  Exporter  Supprimer
apres_mise_a_jour_estimation	quantite	AFTER	UPDATE	 Éditer  Exporter  Supprimer
calculer_ecart	graphique	BEFORE	INSERT	 Éditer  Exporter  Supprimer
estimation_non_negative	quantite	BEFORE	UPDATE	 Éditer  Exporter  Supprimer

16. Liste des déclencheurs dans PhpMyAdmin

Les deux premiers déclencheurs permettent de calculer automatiquement la colonne quantité avec la valeur de contenanceP et estimation. Le premier déclencheur agit avant l'insertion (BEFORE INSERT) et le second avant la mise à jour (BEFORE UPDATE). Ils sélectionnent d'abord la contenanceP (contenance par produit) du produit en cours d'insertion ou de mise à jour. Si cette contenance est supérieure à zéro, ils ajustent la quantité en arrondissant à l'entier supérieur le résultat de la division de l'estimation par la contenance.

Le troisième déclencheur est utilisé pour le graphique lorsqu'une estimation est modifiée. Si la nouvelle estimation diffère de l'ancienne, le déclencheur insère un nouvel enregistrement dans la table graphique. Cet enregistrement contient l'ID du produit (idP), la nouvelle estimation de quantité et la date et heure actuelle (CURRENT_TIMESTAMP). Ce mécanisme permet de suivre l'évolution de chaque produit.

Le quatrième est aussi nécessaire à la table graphique, il est configuré pour s'exécuter avant l'insertion de données dans la table graphique. Il calcule la différence absolue entre la nouvelle valeur à insérer et la dernière valeur enregistrée pour le même produit. Cette différence est ensuite assignée à la nouvelle colonne écart de la table graphique.

Et pour terminer le déclencheur 'estimation_non_negative' est conçu pour s'assurer que les estimations de quantités dans la table quantité restent non négatives lors des mises à jour. Avant chaque mise à jour (BEFORE UPDATE), si la nouvelle valeur estimée est inférieure à zéro, le déclencheur ajuste cette estimation à zéro. Cela garantit l'intégrité des données en évitant des estimations de quantité négatives. (Voir annexe 5 pour le code complet de ces déclencheurs.)

5.3.4 Développement Backend

Parlons maintenant du côté non visible par l'utilisateur, le backend. Le backend comprend la partie modèle et contrôleur de la structure MVC. Mais avant de parler de ces deux parties je vais vous parler du fichier index.php qui relie la vue, le modèle et le contrôleur. Ce fichier agit comme un point central du site web. Il initialise l'environnement en démarrant une session et en configurant l'affichage des erreurs. Ce code gère la sélection du contrôleur et de la vue appropriée. Dans un cas où le contrôleur demandé n'existe pas ou n'est pas trouvé, le site génère une erreur 404.

Commençons par la partie modèle, qui gère les interactions avec la base de données. Il regroupe la connexion à la base de données et l'exécution des requête SQL. Cela permet de faire des opérations CRUD(créer, lire, mettre à jour, supprimer) sur les données, comme ajouter un produit, vérifier l'existence d'un produit, modifier une estimation et plus encore. Le modèle fonctionne de manière simple, on créer la classe Model, on lie la classe a la base de données, on créer une fonction getModel qui permet d'appeler le model dans le contrôleur, puis on créer les fonctions nécessaires au site web.

```
<?php

class Model {
    private $db;
    private static $instance = null;

    /**
     * Méthode permettant de se connecter a la bdd
     */
    private function __construct()
    {
        include "Utils/credentials.php";
        $this->db = new PDO($dsn, $login, $mdp);
        $this->db -> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }

    /**
     * Méthode permettant de récupérer un modèle car le constructeur est privé
     */
    public static function getModel() {
        if (self::$instance === null) {
            self::$instance = new self();
        }
        return self::$instance;
    }
}
```

17. Code du fichier model.php

Toutes les fonctions présentes dans ce fichier sont accompagnées de commentaires pour faciliter les futurs mise à jour du site s'il doit en avoir.

La partie contrôleur contient deux fichiers, un premier qui se nomme 'controller.php' et qui sert de base pour tous les contrôleurs dans l'architecture MVC car il contient un constructeur qui détermine et exécute les actions demandées via l'URL. Ce fichier contient aussi une action par défaut si aucune action n'est spécifié, une méthode 'render' qui permet d'afficher la vue correspondante avec les données nécessaires et une action 'error' qui affiche une page d'erreur si la vue est inexistante. Ce fichier, une fois créer,

n'est plus modifier car les modifications vont être faites dans le second fichier.

Ce second fichier, qui se nomme 'controller_home.php' et qui hérite de 'controller.php', inclut toutes les méthodes nécessaires pour manipuler les données dans la base de données comme le calcul de la moyenne et de la médiane, pour gérer l'importation des fichiers XML et envoyer les données traitées à la vue.

Parlons maintenant du code. Je vais expliquer deux parties du code, la première partie parlera de l'importation XML et la deuxième du graphique car le . L'importation XML implique la lecture d'un fichier XML pour mettre à jour la base de données. Le contrôleur extrait les données nécessaires telles que le nom et la quantité d'un produit à l'aide d'une fonction déjà implémenté en PHP qui se nomme 'simplexml_load_file'.

```
//Importer le fichier XML et convertir les information dans la bdd
if (!empty($_FILES["fichierXML"])) {
    $tmpPath = $_FILES['fichierXML']['tmp_name'];
    $xmldata = simplexml_load_file($tmpPath) or die("Failed to load");
    foreach($xmldata->Session->Formulation->Mix as $mix) {
        $mixName = preg_replace('/\D/', '', $mix->Mix_name);
        $Mixxml = $mix->Mix_ml;
        $m->updateFromXML($mixName, $Mixxml);
    }
    header('Location: index.php');
}
```

18. Code de l'importation XML

Une fois les informations récoltées, elles sont envoyées dans la base de données à l'aide de la fonction 'updateFromXML' qui provient du modèle. Une fois cela fait, on rafraichit la page avec la fonction 'header'.

Pour ce qui est du graphique, le code gère les informations sur un produit sélectionné pour visualiser sa progression, calculer la moyenne et la médiane de ces valeurs, ces données sont ensuite affichées sous forme graphique.

```
//Affichage du graphique
$nom = 1;
if (isset($_POST["produit6"])) {
    $nom          = htmlspecialchars($_POST["produit6"]);
    $values        = $m->infoP($nom);
    $median        = round($this->calculMedian($values), 1);
    $moyenne       = round($this->calculMoyenne($values), 1);
    $result        = $m->graphique($nom);
    $donneesJson   = json_encode($result);
    $nomProduit    = $m->getProductNameById($nom);
    echo '<script type="text/javascript">';
    echo 'window.location.href="' . $_SERVER['PHP_SELF'] . '#graph"';
    echo '</script>';
} else {
    $result = [];
    $donneesJson = json_encode($result);
    $median      = 0;
    $moyenne     = 0;
    $nomProduit  = "";
}
}
```

19. Code de l'affichage graphique

5.4 Problème et tests

5.4.1 Problèmes rencontrés

Au cours du projet, j'ai rencontré plusieurs problèmes, le premier est que je ne savais pas ce qu'était un fichier XML, à quoi cela servait, comment est-il généré mais surtout comment le manipuler. J'ai aussi remarqué que le site avait une navigation compliquée car il y'avait trop de produit dans la base de données, donc pour accéder au graphique, qui est en bas de page, il fallait beaucoup scroller vers le bas. Ensuite, il y'avait aussi le fait que je n'avais jamais construit un graphique en Javascript. Après cela, il y'a aussi eu le fait que toutes les données du tableau Excel devaient être importées. Tous ces problèmes m'ont ralenti dans mon projet mais j'ai, pour tous ces problèmes, trouvé une solution.

5.4.2 Solutions

Vu que je ne savais pas à quoi ressemblait un fichier XML j'ai demandé à mon tuteur de m'envoyer un exemple de fichier qui peut être généré par

son logiciel pour mieux visualiser ma mission (Voir annexe 6), il m'avait notamment expliqué que c'était un fichier lisible par les ordinateurs. Une fois le fichier reçu et analysé, je suis parti m'informer sur comment l'intégrer dans un code php, puis je suis tombé sur une vidéo qui explique très bien l'intégration d'un fichier XML dans un site (voir annexe 7). Pour ce qui est de la longue liste de produit, j'ai ajouté un système de pagination qui affiche 25 produits par page, pour ce faire j'ai repris le système d'un ancien projet fait en cours de développement web. Pour ce qui est du graphique, j'en ai déjà code mais en python, j'ai donc codé un graphique en python qui récupère les données d'un tableau. Le problème est qu'il n'est pas possible d'afficher un code python en php, j'ai donc essayé de convertir mon code python en un code javascript mais cela n'a était qu'une succession d'échec, j'ai donc décider d'envoyer mon python à une IA qu'il a transformé en code javascript fonctionnel. Et pour finir, vu qu'il y'avait un très grand nombre de données dans le tableau Excel j'ai décidé de créer un script python, qui insère toutes les données dans la base de données.

```
qtt = [  
    1600, 2900, 1680, 500, 390, 940, 780, 370, 500, 330,  
    1000, 330, 860, 570, 1580, 850, 800, 1000, 500, 660,  
    920, 1790, 520, 850, 1410, 680, 1000, 6100, 1000, 6200,  
    290, 960, 1000, 3500, 920, 1000, 1000, 460, 5700, 980,  
    2800, 3500, 3500, 3500, 1430, 1000, 3100, 6500, 880, 1730,  
    690, 1260, 310, 1000, 1910, 800, 650, 450, 870, 710,  
    800, 400, 590, 930, 1000, 870, 360, 740, 1000, 390,  
    500, 500, 970, 500, 810, 420, 500, 650  
]  
  
for i in range(len(qtt)):  
    print(f"INSERT INTO Quantite VALUES ({i + 1},0, {qtt[i]});")
```

20. Code python pour l'insertion des données

5.5 Bilan du projet

Ce projet a été pour moi une très bonne expérience qui m'a permis de me confronter à plusieurs défis techniques tels que la manipulation de fichiers XML ou bien la gestion d'une grande base de données. Ce projet m'a permis de voir comment les compétences développées au cours de ma formation peuvent être directement appliquées dans un contexte professionnel et les

difficultés rencontrés encourage l'innovation et l'autonomie. L'outil est fonctionnel et est actuellement utilisé par l'entreprise, cet outil va leur permettre de gérer leurs stocks de manière plus efficace dans le futur.

5.6 Suite du stage

Lorsque j'ai terminé ce projet, il me restait encore deux semaines de stages, mon tuteur a donc décidé de m'inclure dans un travail de groupe avec les deux autres stagiaires. Ce travail consistait à créer un bot qui permettait de récupérer des informations sur un site web. Ce bot devait récupérer des infos sur les sites web des fournisseurs de Vif Auto, pour faciliter le choix de Vif Auto lors de ses achats. Le bot lui a déjà été créé lorsque j'ai rejoint le groupe, il a fallu le réajuster pour chaque site de chaque fournisseur. Ce travail était très intéressant car il s'attaquait à quelque chose de complètement inconnu pour moi, mais malheureusement je n'ai pas eu le temps de faire quelque chose de concret car il restait deux semaines avant la fin de mon stage. Je me suis donc seulement informé et exercé sur le scraping.

6. Bilan Personnel

Mon stage chez Vif Auto m'a beaucoup appris, notamment à être plus organisé et rigoureux. J'ai aussi découvert de nouveaux outils comme le travail avec les fichiers XML et amélioré ma maîtrise de PHP et MySQL. Cette expérience m'a conforté dans mon envie de travailler dans des entreprises innovantes dans le domaine du développement web. Elle a légèrement orienté mon projet professionnel vers des postes mêlant technique et gestion de projet. En bref, ce stage a été une étape importante pour moi, m'ouvrant à de nouvelles possibilités pour mon avenir professionnel.

7. Conclusion

Je suis très satisfait de cette première expérience en entreprise parce qu'elle m'a permis d'acquérir une solide expérience professionnelle, de développer des compétences techniques et de nouer des relations humaines enrichissantes. Le projet auquel j'ai contribué a été intégré dans les pratiques de l'entreprise, ce qui témoigne de son utilité et de sa pertinence. Les méthodes de travail et les outils que j'ai appris et perfectionnés pendant ce stage, notamment autour de la gestion des bases de données et du développement web, sont directement applicables dans le domaine professionnel. Sur le plan personnel, j'ai développé des compétences en gestion de projet et en travail d'équipe, apprenant à naviguer dans les délais et à répondre aux attentes dans un environnement professionnel. Cette expérience a confirmé mon désir de poursuivre dans ce secteur et m'a ouvert des perspectives pour mon projet professionnel futur.

8. Lexique

Backend : La partie d'une application ou d'un site web qui gère les fonctionnalités invisibles pour l'utilisateur

VueJs : Un framework JavaScript populaire utilisé pour construire des interfaces utilisateur interactives et dynamiques dans les applications web.

NodeJs : Un environnement d'exécution JavaScript côté serveur, utilisé pour développer des applications web et des serveurs.

XML : Extensible Markup Language, un langage de balisage utilisé pour stocker et transporter des données de manière lisible par les machines.

IUT : Institut Universitaire de Technologie

MySQL : Un système de gestion de base de données relationnelle open source, largement utilisé dans le développement web.

WampServer : Un ensemble d'outils open source pour le développement web sur des plateformes Windows, comprenant Apache, MySQL et PHP.

JS : JavaScript, un langage de programmation utilisé pour rendre les pages web interactives et dynamiques.

Head HTML : La partie d'un document HTML qui contient des informations sur le document lui-même, comme son titre et ses métadonnées.

Body HTML : La partie d'un document HTML qui contient le contenu visible de la page web, comme le texte, les images et les liens.

Balise HTML : Une instruction utilisée dans le langage HTML pour définir la structure et le contenu d'une page web.

CURRENT TIMESTAMP : Un indicateur de date et d'heure actuelle dans une base de données

SQL : Structured Query Language, un langage de programmation utilisé pour gérer et manipuler des bases de données relationnelles.

CRUD : Acronyme pour Create, Read, Update, Delete, qui sont les opérations de base effectuées sur les données dans une base de données.

URL : Uniform Resource Locator, une adresse utilisée pour localiser une ressource sur internet, comme une page web ou un fichier.

PHP : Hypertext Preprocessor, un langage de programmation utilisé principalement pour le développement web côté serveur.

Scraping : Le processus d'extraction automatique de données à partir de sites web, généralement à l'aide de scripts ou de programmes.

Frontend : La partie d'une application ou d'un site web avec laquelle l'utilisateur interagit directement.

9. Annexes

Annexe 1, Cahier des charges

CAHIER DES CHARGES - PROJET VIF AUTO

Environnement du projet

Ce projet consiste à mettre en place une application web, intégrant des fonctionnalités de gestion de base de données dans un environnement de stage.

Contexte

L'objectif est de créer un outil qui permet d'estimer la quantité de peinture restante en magasin en fonction des quantités utilisées, enregistrées dans un fichier XML à chaque mélange préparé.

Limites

Le projet se concentre sur le développement technique et la gestion de projet, sans s'étendre aux aspects commerciaux tels que le marketing et la vente.

Exigences Fonctionnelles

- Capacité à lire et interpréter les données depuis un fichier XML.
- Calcul de la quantité de peinture restante après chaque mélange.
- Ajout ou suppression manuel de la quantité de peinture.
- Affichage d'un graphique qui montre l'utilisation d'un produit.
- Intégration avec une base de données.
- User Stories :
 - En tant que vendeur, je veux pouvoir entrer les quantités de peinture utilisées pour chaque mélange, afin de mettre à jour le stock en temps réel.
 - En tant que vendeur, je veux voir une liste des produits disponibles et leurs quantités restantes, afin de planifier les approvisionnements.
 - En tant que vendeur, je veux pouvoir importer les données de consommation de peinture depuis un fichier XML, afin de faciliter la mise à jour du stock.
 - En tant que vendeur, je veux recevoir des alertes lorsque le stock d'un produit atteint un seuil critique, afin d'éviter les ruptures de stock.

Exigences Non Fonctionnels

Technique: Compatibilité avec les systèmes d'exploitation courants, facilité d'intégration avec les systèmes existants.

Performance: Réponses rapides lors des calculs et des mises à jour de stock.

Ergonomie: Interface simple et intuitive.

Risques

- Risque de non-adéquation avec les besoins réels du client.
- Risque technique lié à l'importation des données CSV.

Principaux jalons du Projet

Début du Projet : 30/01/2024

Conception et Planification : A voir

Développement et Tests : A voir

Présentation Finale et Livraison : Fin Mars

Annexe 2, Modèle de cahier des charges

CAHIER DES CHARGES - PROJET XXXXXXXX
SOCIETE / STARTUP xxxxxx
Version 1.xx **du** <date xx/xx/xxxx>

NOTES :
 1°) ce document n'est qu'un modèle qu'il faudra adapter en fonction du contexte des projets
 2°) le nom du fichier Cahier de Charges doit respecter le format suivant :
CdC <nom du groupe projet> <Version x.x> <date version>
 Il doit y avoir autant d'occurrence de fichier cahier des charges que de version. Dès que vous modifiez quelques choses dans le cahier des charges, cela donne lieu à une nouvelle version. Exemple de version : 0.1 , 0.2 etc ... puis 1.0 , 1.1 , etc...

Mettre les noms des rédacteurs

Environnement
Il s'agit d'expliquer ici quel est l'environnement du projet : entreprise, association ... en détaillant les activités de la société/startup, c'est à dire le « client ».

Interlocuteur(s) du projet
*Nomment les personnes chez le « client » (fonctions, rôles) qui pourront apporter des précisions sur le cahier des charges.
 Dans le cas de cette SAE, le « client » se fait aider de l'équipe projet pour la rédaction du cahier des charges.
 Dans la réalité, c'est aussi parfois le cas.*

Contexte
*Il s'agit d'expliquer ici les raisons, l'origine, la temporalité (pourquoi le faire maintenant) de ce projet, pour qui et pourquoi ce projet doit être développé.
 Présenter de façon générale et courte le contexte du projet
 Expliquer, positionner, si possible, le niveau de complexité, les difficultés apparentes lors de la rédaction du cahier des charges, et celles qui pourraient apparaître.*

Limites
Préciser les limites du projet, ce qui n'est PAS compris dans le projet afin d'éviter toute ambiguïté

Exigences fonctionnelles
*Lister et préciser les fonctions attendues (création, consultation, calculs ...) à partir des besoins exprimés par des phrases courtes, claires et non ambiguës. L'objectif est de comprendre ce qu'il y a à développer et se faire une idée précise de l'ampleur du projet.
 Préciser si une connaissance particulière du domaine fonctionnelle est nécessaire.*

Exigences NON fonctionnelles

Technique
Préciser l'environnement technique du projet : Operating System, environnement de développement, langage(s) de développement, navigateur(s), base de données, CMS, progiciel ou composant à (ré)utiliser, type de serveur ...

Performance
Préciser, si nécessaire, les contraintes de performances telles que : taux occupation UC, temps de réponse demandé ou exigé par le client

Ergonomie
Préciser, si nécessaire, les contraintes d'ergonomie en fonction des utilisateurs, de l'interactivité attendue

Modèle général cahier des charges /

(incorporation de vidéos, images, dessins, photos, ...). Y a-t-il une charte graphique fournie par le « client » ?

Volumétrie

Préciser si nécessaire, la volumétrie en terme de données (Go, nombre Données, nombre Tables, etc...), en terme de nombre d'utilisateurs (au total si connu et aussi en connexion potentielle simultanée) ...

Sécurité

Préciser, si nécessaire, les contraintes de sécurité – exemple : paiement en ligne - en fonction de l'utilisation et du niveau de confidentialité des données (données soumises à déclaration CNIL ...)

Développement durable

Préciser, si nécessaire, les contraintes de respect de l'environnement, tant sur le plan éthique qu'écologique et économique. En informatique, le « green IT » est appelé à se développer considérablement

Autres

Préciser, si nécessaire, les autres contraintes telles que : organisation, lieu géographique d'implantation, limites d'utilisation, commercialisation envisagée (propriété)

Risque(s) à ne pas faire le projet

Si l'on ne faisait pas ce projet, que se passerait-il ? Tous les projets doivent se justifier par un « apport » économique ou non pour l'entreprise

Risque(s) à faire le projet (lié à un échec potentiel du projet)

Lister les risques qui feraient que le projet n'aboutisse pas ou soit en échec.

Une fois les risques identifiés, prendre les mesures nécessaires afin que ces risques soient limités voire annulés

Principaux jalons


S'ils sont connus, indiquer ici les principales dates clés :

date de disponibilité du cahier des charges, date de mise en place de la planification du projet, différentes dates de mise à disposition des fonctionnalités, etc ... et la date de fin !

Précision :

il faudra impérativement faire une planification précise du projet (GANTT de préférence) : tâches à réaliser, tâches imbriquées et interdépendantes, membre assigné à une tâche, temps prévu pour chaque tâche

Annexe 3, Site web VifAuto.fr



[Nouveau devis](#)
[Historique](#)
[Contact](#)

Vif Auto

Magasin spécialisé en vente de pièce automobile et carrosserie.

Vous pouvez demander votre pièce en magasin, par téléphone ou directement sur notre site. Nos experts cherchent la référence exacte tout en consultant les disponibilités chez nos fournisseurs afin de vous proposer un devis. Après validation de notre devis votre commande sera prête en magasin le jour même ou le lendemain selon l'heure de la commande.

Le site est encore à ses débuts, nous travaillons continuellement à son amélioration.

Actions rapides

Créer un compte

Je m'inscris

Nouveau devis

Je demande un devis


Informations

Horaires


Lundi	8h30 - 18h30
Mardi	8h30 - 18h30
Mercredi	8h30 - 18h30
Jeudi	8h30 - 18h30
Vendredi	8h30 - 18h30

Numéro

09 73 05 10 05



Annexe 4, Maquette de L'outil



Produits

Importer

Graphique

Ajouter un produit

Reference du produit :

Contenance du produit (L) :

Description :

Ajouter

Augmenter la quantité d'un produit

Produit :

Quantité :

Augmenter

Diminuer la quantité d'un produit

Produit :

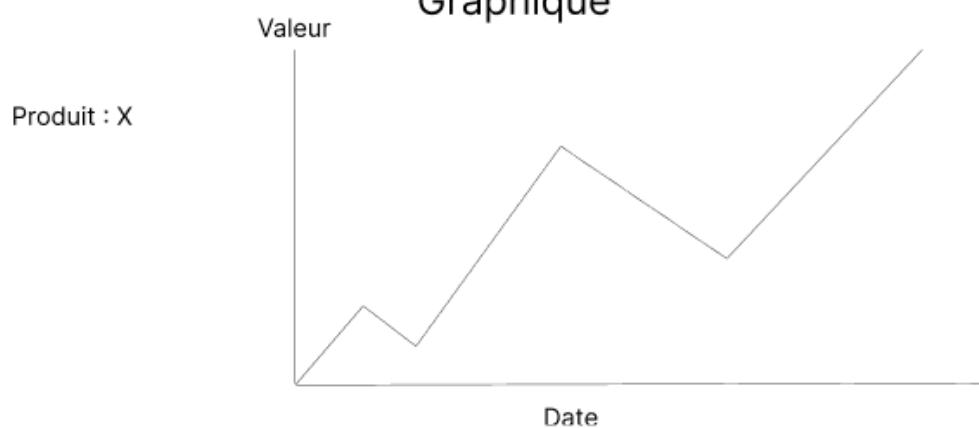
Quantité :

Diminuer

Liste des Produits

Produit	Contenance par pot(mL)	Quantité	Estimation(mL)	Description
300	1000	2	1200	Noir

Graphique



Annexe 5, Code des déclencheurs

```

DROP TRIGGER IF EXISTS `ajuster_quantite_avant_insert`;
DELIMITER $$
CREATE TRIGGER `ajuster_quantite_avant_insert` BEFORE INSERT ON `quantite` FOR EACH ROW BEGIN
    DECLARE contenance FLOAT;

    SELECT contenanceP INTO contenance FROM Produits WHERE idP = NEW.idP;

    IF contenance > 0 THEN
        SET NEW.quantite = CEIL(NEW.estimated / contenance);
    END IF;
END
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `ajuster_quantite_avant_update`;
DELIMITER $$
CREATE TRIGGER `ajuster_quantite_avant_update` BEFORE UPDATE ON `quantite` FOR EACH ROW BEGIN
    DECLARE contenance FLOAT;

    SELECT contenanceP INTO contenance FROM Produits WHERE idP = NEW.idP;

    IF contenance > 0 THEN
        SET NEW.quantite = CEIL(NEW.estimated / contenance);
    END IF;
END
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `apres_mise_a_jour_estimation`;
DELIMITER $$
CREATE TRIGGER `apres_mise_a_jour_estimation` AFTER UPDATE ON `quantite` FOR EACH ROW BEGIN
    IF NEW.estimated <> OLD.estimated THEN
        INSERT INTO graphique (idP, valeur, date)
        VALUES (NEW.idP, NEW.estimated, CURRENT_TIMESTAMP);
    END IF;
END
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `estimation_non_negative`;
DELIMITER $$
CREATE TRIGGER `estimation_non_negative` BEFORE UPDATE ON `quantite` FOR EACH ROW BEGIN
    IF NEW.estimated < 0 THEN
        SET NEW.estimated = 0;
    END IF;
END
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `calculer_ecart`;
DELIMITER $$
CREATE TRIGGER calculer_ecart
BEFORE INSERT ON graphique
FOR EACH ROW
BEGIN
    DECLARE derniereValeur FLOAT;

    SELECT valeur INTO derniereValeur
    FROM graphique
    WHERE idP = NEW.idP
    ORDER BY date DESC
    LIMIT 1;

    IF derniereValeur IS NOT NULL THEN
        SET NEW.ecart = ABS(NEW.valeur - derniereValeur);
    END IF;
END
$$
DELIMITER ;
COMMIT;

```


Annexe 6, exemple d'un fichier XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Mobihel export, V1.0, 2016-07-14 -->
3 <XmlData xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
5   <Session>
6     <Session_time>2024-01-29T10:59:48</Session_time>
7     <Color_data>
8       <Manufacturer>PEUGEOT</Manufacturer>
9       <Code>KNC</Code>
10      <Color_name>ADMIRAL BLUE (2C)</Color_name>
11      <Base_name>BASE</Base_name>
12      <Layer>1/2</Layer>
13      <ID_3C>81558</ID_3C>
14      <Formulation_ID>81558</Formulation_ID>
15      <Chip>502766</Chip>
16      <Density>0.92</Density>
17      <Mobicode>24-86360</Mobicode>
18    </Color_data>
19    <Formulation>
20      <Quantity>0.400</Quantity>
21      <Unit>L</Unit>
22      <RTU_formulation>0</RTU_formulation>
23      <Sales_price>25.41</Sales_price>
24      <Sales_price_L>63.52</Sales_price_L>
25      <Mix>
26        <Mix>MIX342</Mix>
27        <Mix_name>MIX342</Mix_name>
28        <Mix_qty>199.5</Mix_qty>
29        <Total_qty>199.5</Total_qty>
30        <Actual_qty>0.0</Actual_qty>
31        <Mix_ml>15</Mix_ml>
32      </Mix>
33      <Mix>MIX362</Mix>
34      <Mix_name>MIX362</Mix_name>
35      <Mix_qty>132.2</Mix_qty>
36      <Total_qty>331.7</Total_qty>
37      <Actual_qty>0.0</Actual_qty>
38      <Mix_ml>143.5</Mix_ml>
39    </Mix>
40  </Formulation>
41 </Session>
42 </XmlData>

```

Annexe 7, tutoriel de manipulation d'un fichier XML

<https://www.youtube.com/watch?v=I6rSfSCautU>