

RAPPORT DE PJI:

Sujet 21

***Multicriteria : Une Web Extension
pour l'aide à la décision multicritère***

Réalisé par : Marine Deffontaine et Léane Texier

Master 1 informatique, parcours Informatique

Encadré par : Nadarajen Veerapen

Sommaire

Introduction	4
Étude du sujet et des technos utiles à sa réalisation	5
Gestion du projet	8
Développement de la web extension	10
La base : sans choix du site et des critères	10
Création du graphe	10
Ajout du graphe à l'extension	12
Récupération des données	12
Phase de test	14
Ajout des articles des autres pages sur le graphe et choix des axes	15
Ajout possible d'articles d'autres pages sur le graphe	15
Mémorisation des données	15
Ajout du choix des deux critères par l'utilisateur	17
Gestion des sites et données	18
Disponibilité de la web extension sur d'autres sites d'e-commerce	20
Améliorations visuelles et phase de test	22
Un peu de CSS	22
Points de Pareto	23
Phase de test	24
Pistes d'amélioration	25
Conclusion	26

Introduction

Depuis quelques années, les achats via l'internet sont de plus en plus nombreux. En effet, en 2017, l'e-commerce représentait 2300 milliards de dollars de vente. De plus, cela devrait encore accroître avec les années et atteindre 4500 milliards en 2021. De surcroît, en 2018, 45% des internautes dans le monde ont acheté en ligne. Cela représente 1,7 milliards de personnes.

Afin de satisfaire leurs clients, les sites proposent de plus en plus de choix et il est donc de plus en plus compliqué de trouver le meilleur article sans perdre de temps. De nombreux sites proposent des filtres afin de trier les résultats. Cependant, cela n'est pas toujours suffisant au vu du nombre impressionnant d'articles disponibles. De plus, un choix est généralement proposé au client afin de trier les résultats (exemple en fonction du prix croissant ou décroissant) mais cela n'est possible que suivant un critère dont le choix est restreint. Il serait donc très utile d'avoir un outil permettant de pouvoir choisir un article plus aisément grâce à différents critères que l'utilisateur trouve importants.

Une des idées serait de mettre en place une web extension. C'est cette solution qui nous a été présentée et sur laquelle nous travaillons. Cette web extension serait disponible sur différents sites afin de pouvoir satisfaire le maximum de personnes. L'utilisateur pourrait ainsi voir via la web extension un diagramme récapitulant tous les articles présents sur la page. Les marchandises seraient alors représentées sur le diagramme suivant deux critères. De plus, il y aurait la possibilité d'enlever du diagramme un article qui ne nous intéresse pas afin de s'y retrouver plus facilement. Une autre fonctionnalité serait de pouvoir voir les informations de l'article en le sélectionnant sur le diagramme. Souvent, tous les articles ne sont pas sur la même page, il serait quand même très utile de les voir apparaître sur le même diagramme. De plus, tous les utilisateurs n'ont pas les mêmes critères de recherche, il serait donc fortement intéressant de laisser la possibilité à l'utilisateur de choisir les deux critères qui sont les plus importants pour lui.

Étude du sujet et des technos utiles à sa réalisation

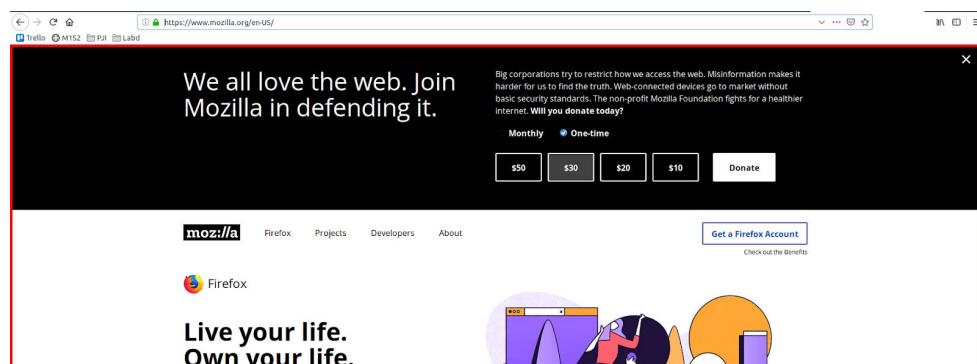
Avant de commencer à travailler sur le code, nous avons, tout d'abord, étudié le sujet ainsi que les technos à utiliser pour le réaliser.

L'étude du sujet nous a, tout d'abord, permis de voir toutes les fonctionnalités qui seraient intéressantes à mettre en place. Dans un second temps, nous avons, avec l'accord de notre encadrant, choisi les fonctionnalités que l'on souhaitait mettre en place en les priorisant.

Suite à l'étude du sujet, il a fallu nous familiariser avec les web extensions étant donné que nous n'avions aucune connaissance sur ce sujet. Afin de bien comprendre le fonctionnement, nous avons effectué deux tutoriels proposés par Mozilla.

Pour la première web extension proposée, nous avons dû créer un module permettant d'ajouter une bordure de couleur rouge sur toutes les pages chargées depuis le site "mozilla.org" et tous ses sous-domaines. Ce tutoriel nous a également permis de voir comment procéder afin de charger le module sur Firefox dans le but de pouvoir utiliser la web extension.

Voici le résultat obtenu :

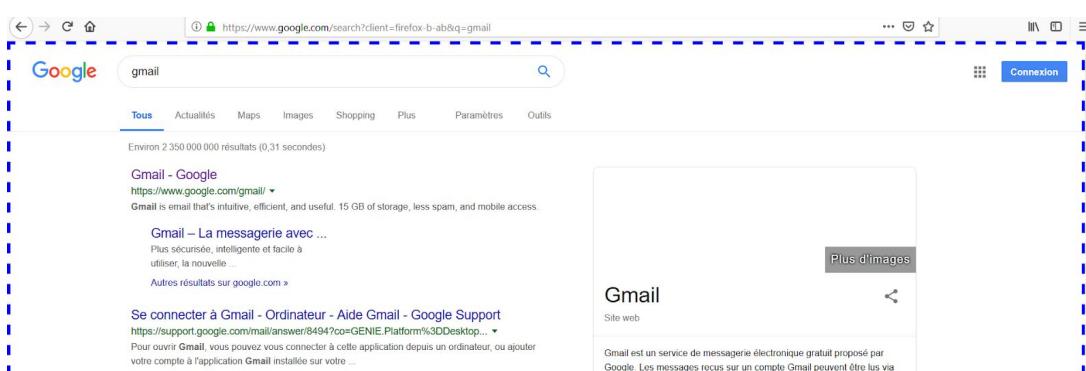


Résultat de la première web extension sur le site "mozilla.org"

Le tutoriel effectué est disponible à l'adresse suivante :

https://developer.mozilla.org/fr/docs/Mozilla/Add-ons/WebExtensions/Your_first_WebExtension

Suite à ce tutoriel, nous l'avons modifié afin de bien comprendre le fonctionnement de chaque fichier et de chacune de leurs sous-parties. Nous avons alors changé le CSS à appliquer et nous avons indiqué à la web extension que cela s'applique non plus sur les sites de "mozilla.org" mais sur tous les sites qui correspondent à la formule regex suivante "`*:/*/*`". Voici le résultat obtenu :



Résultat de la première web extension modifiée sur une page de recherche Google

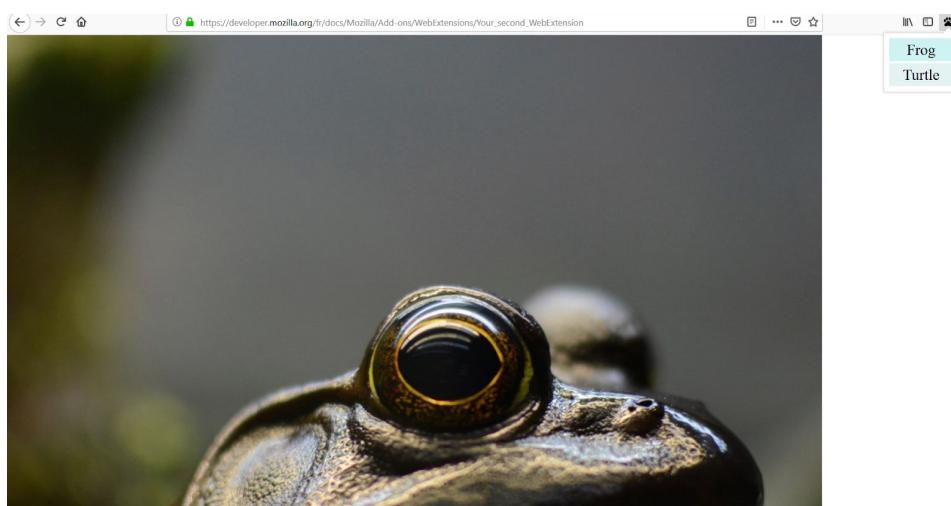
Nous avons ensuite effectué le second tutoriel proposé par Mozilla qui est accessible à l'adresse suivante :

https://developer.mozilla.org/fr/docs/Mozilla/Add-ons/WebExtensions/Your_second_WebExtension

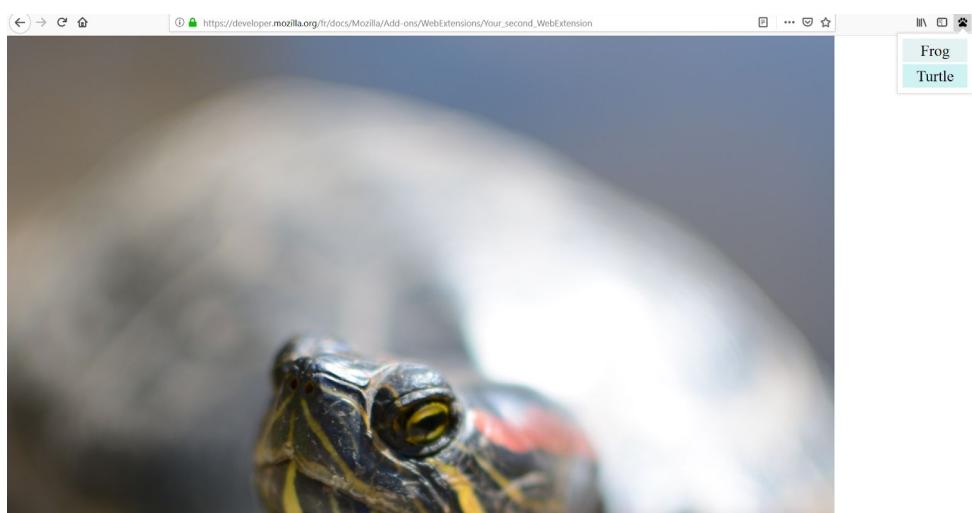
Ce deuxième module ajoute un nouveau bouton à la barre d'outils Firefox. Lorsque l'utilisateur clique sur ce bouton, une fenêtre s'affiche afin de lui permettre de choisir un animal parmi ceux proposés. Une fois l'animal choisi, le contenu de la page actuelle est remplacé par l'image de l'animal choisi. Voici le résultat obtenu :



Web extension avec le choix des animaux possibles



Résultat de la seconde web extension suite à un clic sur le bouton “Frog” de la web extension



Résultat de la seconde web extension suite à un clic sur le bouton “Turtle” de la web extension

Nous avons par la suite amélioré cette web extension en proposant à l'utilisateur de recharger ce qu'il y avait sur la page initialement en ajoutant un bouton "Clear".



Web extension avec le choix des animaux possibles et le bouton permettant de recharger la page initiale

Suite à ces différents tutoriels, nous avons dû nous renseigner sur les technologies utiles à la réalisation de cette web extension. Pour cela, notre encadrant nous a conseillé de faire le développement principalement en JavaScript, HTML et CSS. Ces technologies nous sont assez familières étant donné que nous avons dû les utiliser plusieurs fois au cours de notre licence. Il a également fallu choisir une bibliothèque afin de pouvoir visualiser les données sur un graphe. Pour cela, notre professeur référent nous a conseillé de nous renseigner sur "D3.js" (D3 = Data-Driven Documents) qui est une librairie graphique JavaScript permettant d'afficher des données dans des graphes de manière dynamique. Afin de bien comprendre son fonctionnement, nous avons étudié différents exemples présents sur Internet. Cela nous a été très utile lors de la mise en place de cette librairie dans notre web extension car nous avons pu modifier ces exemples afin de les adapter à notre projet. Afin de pouvoir récupérer les données des articles de la page web à afficher sur le graphe, nous avons dû utiliser des XPaths ce qui était nouveau pour l'une d'entre nous. Cependant, grâce à l'étude de quelques exemples, cela n'a pas posé de problème par la suite.

Gestion du projet

Afin de gérer au mieux notre projet, nous avons créé un diagramme de Gantt que nous avons mis à jour et adapté au fur et à mesure de notre avancement. Ce diagramme nous a permis d'estimer la durée des tâches, de mettre un ordre sur les différentes tâches à effectuer et de voir où l'on en sera à une date souhaitée. Vous trouverez en Annexe 1 le premier diagramme de Gantt que nous avons fait lors de la mise en place du projet. Celui-ci a évolué au fil du temps pour donner une version finale disponible en Annexe 2.

Sur ce diagramme, vous pouvez remarquer que les premières semaines nous ont permis de découvrir les web extensions et de nous familiariser avec les différents outils utiles par la suite. L'estimation des premières semaines s'est révélée bonne notamment dû au fait que nous n'avons pas rencontré de difficultés particulières et que c'était une estimation à court terme.

A partir de mi-février, nous avons commencé à afficher les données sur un diagramme. Cette étape nous a pris un peu plus de temps que prévu car nous avons rencontré des problèmes qui seront expliqués ultérieurement. De plus, nous avons décidé que tous les développements ne seront plus testés et faits pour le site "Amazon" mais pour le site de "La Centrale" car c'est un site où la récupération des données est plus simple étant donné que les XPaths sont plus faciles et qu'il y a peu de pages différentes pour visualiser les listes d'articles. Le retard dû à cette étape a pu être rattrapé pendant la phase de test qui a été plus rapide que prévu.

Suite à cela, nous avons commencé à travailler sur l'ajout des données des pages suivantes. Nous avions initialement estimé cette tâche à 2 semaines. Cependant, cela n'a pas été aussi simple que nous pouvions nous imaginer. Malgré de nombreuses solutions testées nous avons dû trouver une alternative car aucune n'a été concluante. Cette étape a donc eu un impact sur l'avancement du projet et nous a fortement retardé dans ce que nous avions prévu à l'origine.

Afin d'avancer tout de même dans le projet, nous avons travaillé en même temps sur l'ajout du choix des axes par l'utilisateur sur le site de "La Centrale" d'une façon simplifiée car l'idée initiale souhaitée n'était pas réalisable dans le temps qu'il nous restait. En effet, l'idée originale était de proposer à l'utilisateur de choisir successivement les données qu'il souhaitait afficher sur le graphe en lui faisant sélectionner pour chaque type de données, quelques informations similaires afin qu'on puisse en déduire le XPath associé à la récupération des données. Afin de pouvoir quand même adapter le graphe en fonction de ce que l'utilisateur souhaite comparer, nous lui proposons une liste possible de données dont les informations sont numériques (exemples : prix, kilomètres) et le graphe est alors adapté en fonction.

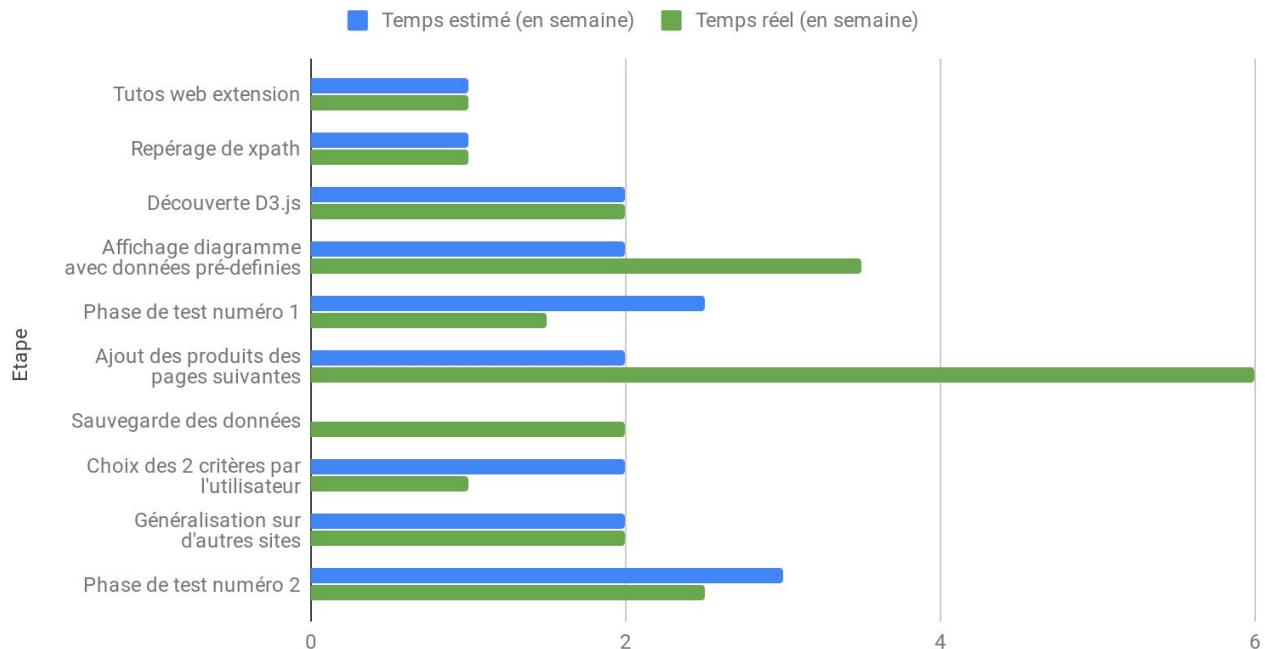
De plus, nous n'avions pas pensé à la sauvegarde des données que nous avons donc dû mettre en place car celles-ci sont perdues à chaque fermeture de la web extension.

Ensuite, nous avons préféré résoudre tous les bugs, nettoyer et documenter le code avant de continuer les développements afin d'avoir à ce stade une version viable du projet.

Puis, nous avons travaillé sur le déploiement de la web extension sur d'autres sites. Pour cela, nous avons dû mettre à jour notre code afin de pouvoir traiter n'importe quel site, ajouté dans le fichier de configuration, de la même manière. L'estimation de temps pour cette phase a été respectée.

Pour terminer, nous avons effectué une deuxième phase de test afin d'avoir le retour d'utilisateurs.

Diagramme représentant les temps réels et estimés des grandes étapes de développement de l'extension

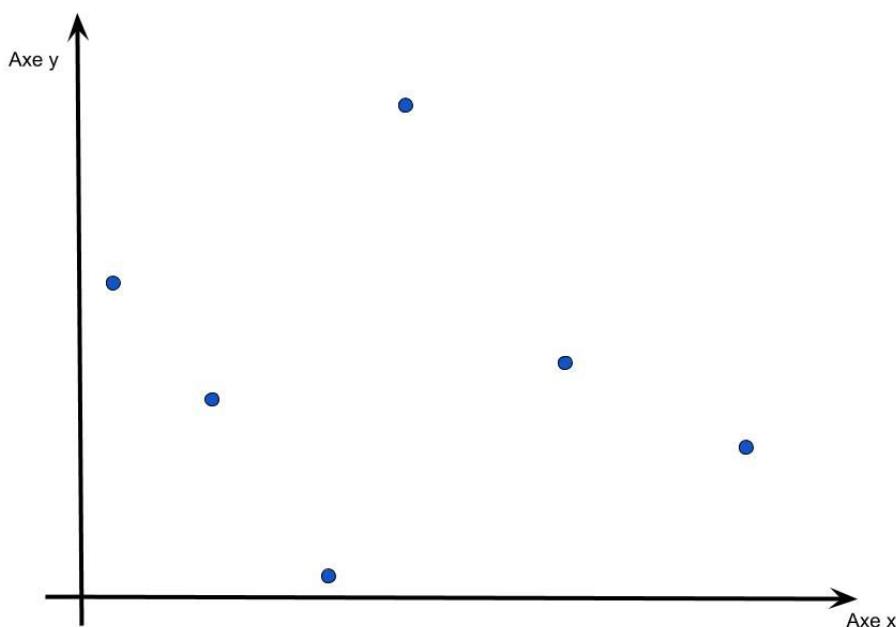


Développement de la web extension

La base : sans choix du site et des critères

Création du graphe

Avant de commencer la phase de développement, nous avons tout d'abord fait un croquis afin de déterminer le visuel souhaité pour la première phase de notre web extension. Voici le croquis :

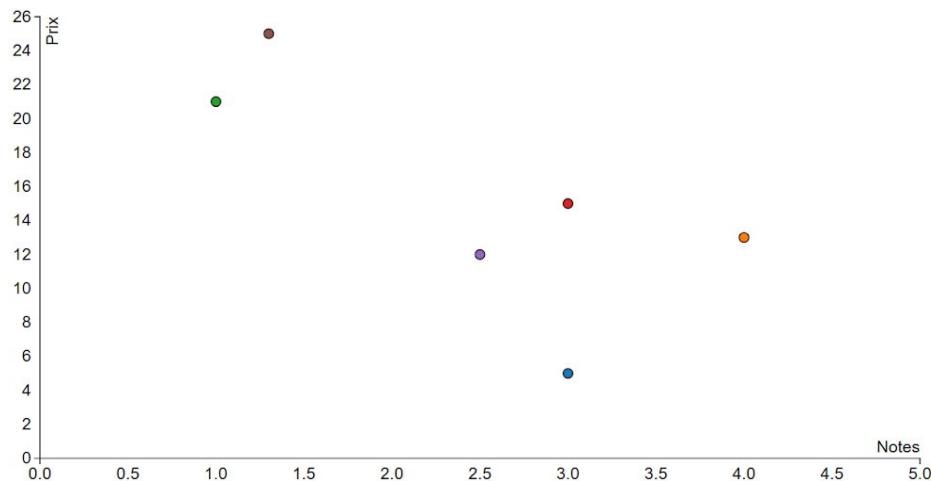


Quand on passe sur un point, les données relatives à celui-ci s'affichent.
Quand on clique dessus, le point est supprimé.

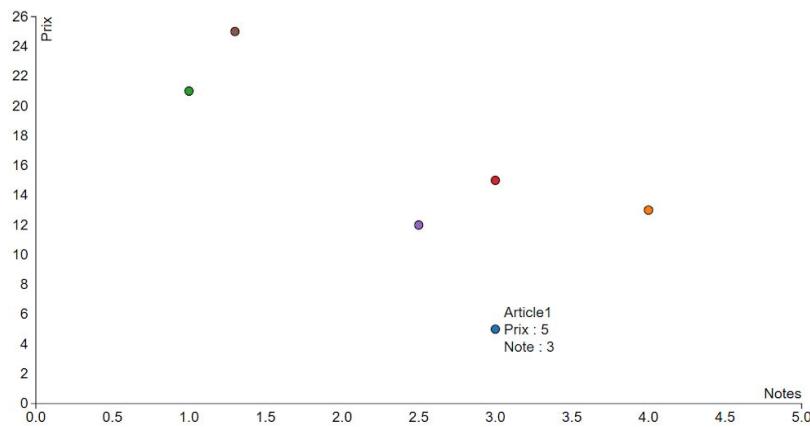
Premier croquis de notre web extension

Nous avons fait un graphe assez simple car, pour cette première phase, aucune fonctionnalité n'est proposée à l'utilisateur.

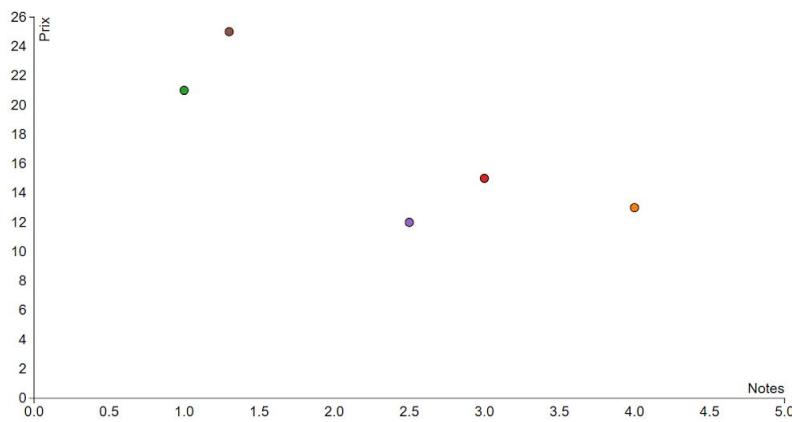
Une fois le croquis effectué, nous avons réalisé un diagramme directement dans une page HTML afin de nous familiariser avec la librairie “D3.js” et pour, par la suite, pouvoir l'inclure dans notre web extension. Pour cela, nous nous sommes aidés des exemples que nous avions étudiés au préalable. Afin de le réaliser, nous avons utilisé un fichier CSV qui contient des valeurs en dur de données d'articles pour nous permettre d'afficher des points sur notre graphique avec les fonctionnalités que l'on souhaitait mettre en place. Sur ce graphe, nous avons donc décidé de faire apparaître des informations complémentaires de l'article dans une bulle d'informations lorsque l'utilisateur survole le point lié à l'article. Nous avons également ajouté la possibilité de supprimer un point en cliquant dessus afin de pouvoir mieux s'y retrouver parmi les articles qui intéressent réellement l'utilisateur. Voici les différents résultats obtenus pour cette phase :



Graphe réalisé avec D3.js dans notre page HTML



Affichage des informations de l'article 1 au survol du point lié



Suppression de l'article 1 à la suite d'un clique sur le point lié à cet article

Une fois le graphique réalisé dans une page HTML, nous l'avons inclus dans une web extension que nous avons créé. Lors de cette étape, nous avons rencontré des difficultés.

Ajout du graphe à l'extension

Tout d'abord, lorsque nous avons créé notre page HTML, nous chargions le script de D3 directement depuis le site, ce qui ne posait pas de problème, grâce à la balise suivante :

```
<script src="https://d3js.org/d3.v3.min.js"></script>"
```

Cependant, afin de pouvoir utiliser la librairie dans notre web extension, nous avons dû télécharger la version de D3.js en local et l'ajouter à nos fichiers.

Nous avons également été confrontées à des problèmes de sécurité, puisque les extensions ont une politique de sécurité de contenu qui leur est appliquée par défaut. Cette sécurité bloque le chargement des différents scripts et empêche leur utilisation. De ce fait, nous n'arrivions pas à utiliser la librairie graphique puisque le chargement de celle-ci était bloquée. Pour résoudre ce problème, nous nous sommes renseignées sur la Content Security Policy (CSP)¹ et nous avons ajouté une clé "content_security_policy" dans notre manifest afin de permettre à l'extension d'exécuter les scripts souhaités. A la suite de cette étape, notre graphique s'affichait correctement dans notre extension avec les valeurs en dures présentes dans le fichier CSV.

Récupération des données

Une fois notre graphique avec les données en dur réalisé, nous avons travaillé sur la récupération des données présentes sur la page web sur laquelle l'utilisateur est.

Nous avons alors utilisé les XPaths afin de récupérer les informations sur la page. Nous avons choisi deux sites : Amazon et La Centrale. Sur Amazon, nous récupérons le titre des articles d'une page, les prix, les avis et les nombres d'avis. Tandis que sur La Centrale nous récupérons les marques et versions des voitures, les prix, les années, les kilomètres, ainsi que le nombre d'années de garantie.

Les requêtes étant plus simple à gérer sur le site de La Centrale, nous avons décidé de changer de site de test afin de choisir celui-ci. Nous avons ainsi pris la décision de commencer par nous focaliser sur ce site, en ayant pour objectif de généraliser le code par la suite sur d'autres sites tels qu'Amazon. En effet, voici les XPaths permettant sur chaque site de récupérer les prix des articles :

- Amazon : //div[@class="s-item-container"]/div[5]/div/a/span[contains(@class, "a-color-price s-price")]
- La Centrale : //div[contains(@class, "fieldPrice")]/nobr/span[2]

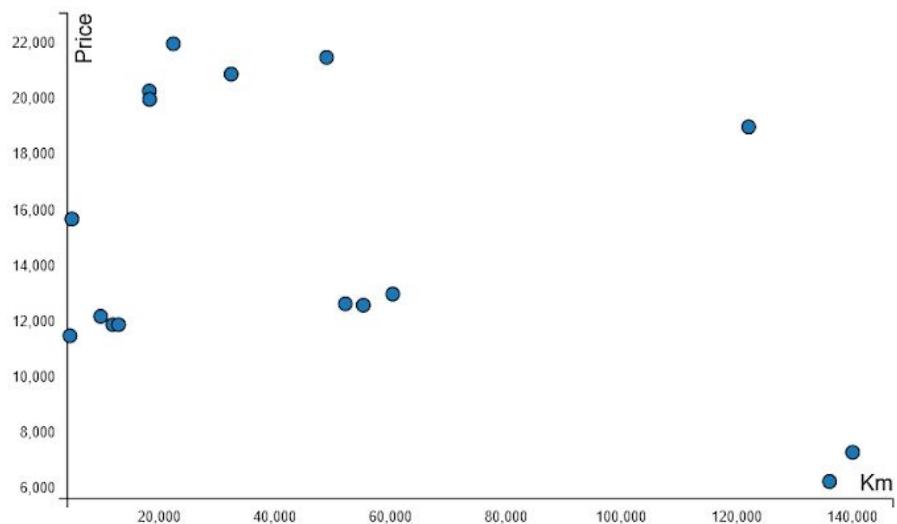
Nous avons mis en place la récupération des données dans notre code. Cependant, nous ne pouvions pas faire directement nos requêtes dans notre code JavaScript car celui-ci était lié à notre pop-up de web extension et n'avait donc pas accès aux données de la page. Nous avons donc dû déléguer ce travail à un autre fichier JavaScript qui était lié à la page web sur laquelle l'utilisateur est. Pour cela, il a donc fallu faire communiquer les deux fichiers entre eux à l'aide de listeners. Nous avons utilisé les fonctions "runtime.sendMessage" et "runtime.onMessage" qui permettent d'envoyer et d'écouter les messages.

¹ CSP : <https://developer.mozilla.org/fr/docs/Web/HTTP/CSP>

Marine Deffontaine et Léane Texier

Une fois les informations récupérées nous les avons traitées avec un format JSON, qui était selon nous et avec l'avis notre encadrant, la meilleure méthode afin de manipuler les données. Ainsi, il nous a fallu changer notre façon de dessiner notre graphe car celui-ci prenait en entrée un fichier CSV et non des données sous le format JSON. De plus, nous avons mis en place une portion de code qui permet aux axes de s'adapter à l'intervalle des données récupérées.

Pour le premier test sur La Centrale, nous avons choisi de faire un diagramme qui affiche par défaut le prix des voitures en fonction de leurs kilomètres.



Graphe de notre web extension avec les données de la page courante pour le prix des voitures en fonction de leurs kilomètres

Nous affichons au survol des points, les données qui nous semblent importantes pour l'utilisateur tel que son prix, son année, son nombre de kilomètres, son nombre d'années de garantie, sa marque et sa version.



Survol d'un point du graphe et affichage des informations de la voiture

La suppression d'un point est également possible sur ce graphe comme c'était le cas avec les données en dur.

Phase de test

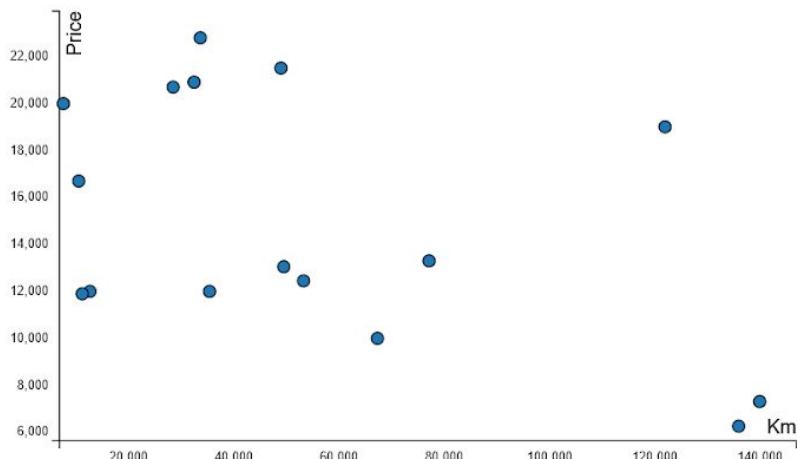
A ce stade du projet, notre web extension était opérationnelle sur le site de La Centrale. Cependant, l'utilisateur ne pouvait pas choisir ses critères et la page courante devait contenir une liste de véhicules. Nous avons décidé de mettre en place une première phase de test afin de savoir si le graphique de notre web extension était suffisamment clair pour les utilisateurs et si nous pouvions l'améliorer. Nous avons alors interrogé neuf personnes pour cette première phase de test certaines ayant un lien avec l'Informatique mais également avec des personnes novices dans ce domaine. La moitié des personnes avaient entre 18 et 25 ans. La seconde moitié avait entre 45 et 55 ans.

Pour cette première phase, nous avons décidé de ne pas poser de questions à l'utilisateur et de ne lui donner aucune indication afin de ne pas biaiser son jugement. Pour cela, nous ouvrions la web extension pour le testeur après l'avoir placé sur une page où la récupération des données était possible car à ce stade nous ne gérions pas le cas où l'utilisateur ouvrirait l'extension sur un autre site ou sur une page ne disposant pas de données à récupérer.

Grâce à cette phase de test, nous avons pu nous rendre compte que le fait de survoler un point pour avoir les informations n'était pas intuitif et nous avons donc ajouté une indication en haut de notre web extension pour préciser cette fonctionnalité à l'utilisateur. De même, le fait de pouvoir supprimer un article du graphe en cliquant sur le point correspondant n'était pas indiqué, de ce fait l'utilisateur ne connaissait pas l'existence de cette fonctionnalité. Nous avons donc ajouté une autre indication pour l'expliquer à l'usager. Nous nous sommes également rendu compte que notre graphique n'avait pas de titre, nous l'avons donc ajouté.

To display informations, hover on the point. To delete a point, click on it.

Graph representing the Km depending on the Price



Graphe de notre web extension suite à la phase de test

Ajout des articles des autres pages sur le graphe et choix des axes

Ajout possible d'articles d'autres pages sur le graphe

Nous avons ensuite voulu que la web extension n'affiche plus seulement les données de la page courante mais également les informations des x pages suivantes. Pour cela, nous avons testé différentes approches que nous allons expliquer en indiquant pourquoi elles n'ont alors pas fonctionnées pour nous.

Tout d'abord, nous avons essayé la méthode du clic sur la page suivante afin de récupérer les données. L'idée était que la web extension récupérait les informations de la page courante, cliquait sur le bouton afin de passer à la page suivante où l'extension récupérait alors les informations et cela un nombre de fois x prédéfini. Cependant, lors de l'exécution, nous ne récupérions que les données de la première page. En effet, ce qui se passait réellement, c'était que l'extension récupérait les données de la page courante x fois et cliquait seulement après sur les pages suivantes car cette fonctionnalité est asynchrone.

Nous avons donc eu une autre idée qui consistait à non plus cliquer directement sur la page mais à changer l'URL de la page dans notre script. Nous avons testé différentes méthodes proposées par JavaScript afin de faire cela. Cependant, nous avions le même problème que précédemment étant donné que toutes les méthodes proposées étaient également asynchrones.

Suite à des recherches sur Internet, nous avons remarqué que des "Promise" existent afin de gérer ce problème de synchronisation. Nous avons donc essayé de les utiliser afin de rendre synchrone l'une des deux méthodes décrites précédemment. Malgré de nombreux tests, nous n'avons pas réussi à rendre synchrone l'une ou l'autre méthode.

Nous avons alors parlé de ce problème avec notre professeur référent afin de savoir s'il connaissait des solutions. Il nous a alors parlé de XMLHttpRequest après avoir vu une web extension de repagination sur Internet. Nous avons essayé de récupérer les nouvelles pages grâce à un appel à XMLHttpRequest qui propose de récupérer les données de manière synchrone ou asynchrone selon les options choisies. Cependant, nous n'avons pas réussi à récupérer les pages souhaitées, nous récupérions des réponses dans un format invalide et nous n'arrivions pas à les traiter.

Etant donné qu'aucune des solutions testées n'a été concluante et avec tout le retard pris, nous avons alors décidé de laisser de côté cette fonctionnalité d'ajout automatique des pages suivantes. Cependant, il nous semblait nécessaire de proposer à l'utilisateur la possibilité d'ajouter sur le même graphe les données de différentes pages afin qu'il puisse bien comparer.

Mémorisation des données

Afin de mettre en place cette fonctionnalité, nous avons rencontré une nouvelle difficulté à laquelle nous n'avions pas pensé. En effet, le problème était que lorsque l'on quittait la web extension et qu'on revenait dessus même si nous n'avions pas changé de page les points précédemment supprimés réapparaissaient car le code était relancé. De même, si nous étions sur une autre page, les points précédents n'étaient alors plus présents, il y avait seulement ceux relatifs à la page courante. Il nous a

donc fallu trouver une solution afin de garder en mémoire les différentes données (axes et données des articles).

Pour faire cela, nous avons utilisé le mécanisme de stockage basé sur l'API Web Storage². Celui-ci permet aux extensions de récupérer des données en les stockant. Nous avons décidé d'utiliser le stockage local. Les éléments souhaités sont ainsi enregistrés localement sur la machine sur laquelle l'extension est installée.

Nous avons donc choisi d'enregistrer dans ce stockage local, les différents axes du graphe ainsi que toutes les données à afficher dessus lors de l'ouverture de l'extension.

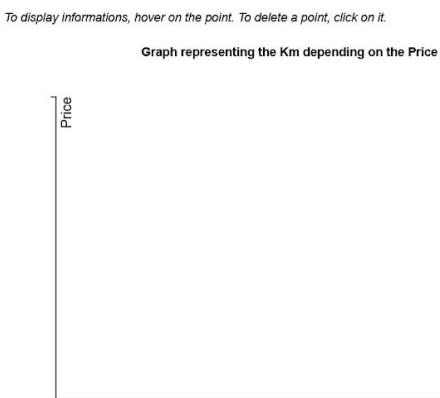
Grâce à la mémorisation de ces données, nous avons pu proposer différentes nouvelles fonctionnalités à l'utilisateur. Désormais, lorsqu'il ouvre la web extension, deux possibilités s'offrent à lui. Il peut ainsi soit :

- reset toutes les données qui ont éventuellement été enregistrées avant et charger ainsi seulement les données de la page courante.
- afficher les données précédemment enregistrées.



Choix proposés à l'utilisateur lors de l'ouverture de la web extension

Lors du clic sur l'un des deux boutons (peu importe lequel), un diagramme est alors affiché et de nouveaux boutons sont alors disponibles. Dans le cas où l'usager décide d'afficher les données stockées mais qu'aucune donnée n'a été préalablement enregistrée un graphe vide est alors affiché.



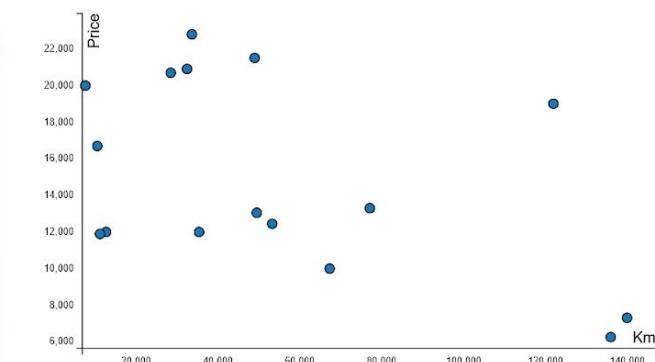
Graphe affiché dans le cas où aucune donnée n'est stockée et que l'utilisateur souhaite les afficher

² API Web Storage : https://developer.mozilla.org/fr/docs/Web/API/Web_Storage_API
Marine Deffontaine et Léane Texier

[Reset and load page's data](#) [Add page data](#) [Save changes](#) [Change options](#)

To display informations, hover on the point. To delete a point, click on it.

Graph representing the Km depending on the Price

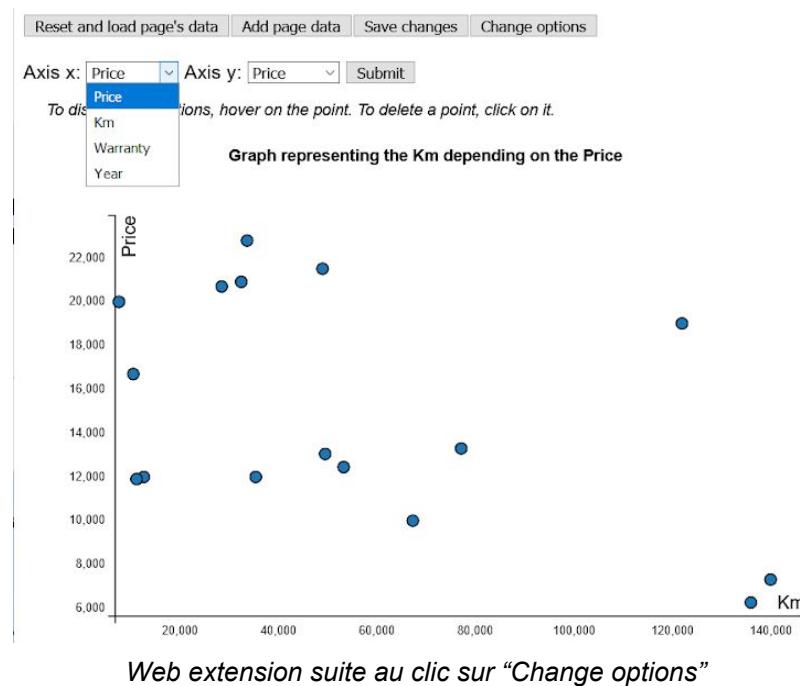


Web extension suite au clic sur l'un des boutons précédents

Le bouton permettant de reset toutes les données qui ont été enregistrées au préalable et d'afficher seulement les données de la page courante est toujours présent. L'utilisateur a également la possibilité d'ajouter les données de la page courante (bouton “Add page data”). Lors du clic sur celui-ci, les nouvelles données sont alors ajoutés sur la graphe en plus de celles déjà présentes et le graphe est alors adapté au besoin. Les données stockées sont alors automatiquement mises à jour. Cela signifie que si l'utilisateur ferme la web extension et la réouvre en cliquant sur “Display previous data” les données précédemment ajoutées seront alors bien présentes. Etant donné que l'utilisateur a la possibilité de supprimer des articles du graphe, il a fallu lui proposer le fait de pouvoir mémoriser cela afin que ces points ne réapparaissent pas lors de la réouverture de l'extension avec les anciennes données. Nous aurions pu le faire de manière automatique lors de la suppression d'un point. Cependant, nous avons étudié le cas où l'utilisateur supprimerait par erreur un point, il devrait alors refaire toutes les manipulations (par exemple “Add page data” sur différentes pages, suppressions de plusieurs points, ...) qu'il avait faites précédemment afin d'afficher les articles souhaités sur le graphe. Le bouton “Save changes” permet ainsi à l'usager de ne pas tout perdre en cas de mauvais clic.

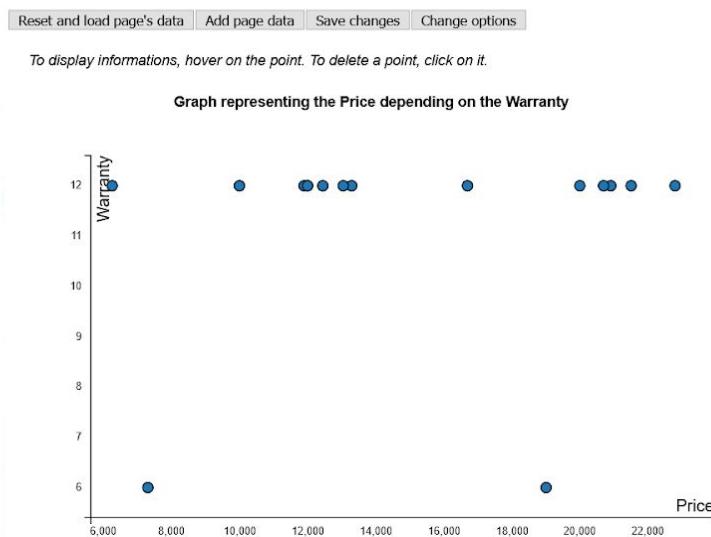
Ajout du choix des deux critères par l'utilisateur

Un dernier bouton (“Change options”) est proposé à l'utilisateur. Celui-ci lui permet de changer les différents axes. Lors d'un clic sur ce bouton, deux listes déroulantes sont alors proposées afin que l'usager choisisse ce qu'il veut afficher sur l'axe des abscisses et des ordonnées. Dans cette liste, seulement les données de type numériques sont proposées.



Web extension suite au clic sur “Change options”

Une fois son choix effectué, il peut alors le valider. Le formulaire disparaît et le nouveau graphe est alors affiché suivant les axes choisis.



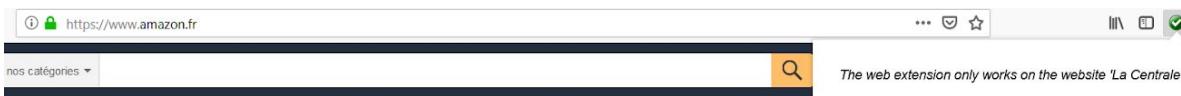
Web extension suite aux changements des axes

Les axes étant enregistrés dans le stockage local, lorsque nous quittons l'extension et que nous la réouvrons, les axes sont les mêmes que lorsque nous avons fermé la web extension.

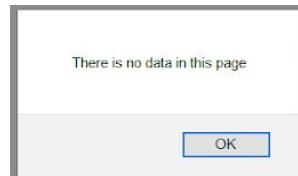
Gestion des sites et données

A ce stade, l'extension ne gérait pas le fait de savoir si nous étions bien sur le site de La Centrale et si nous étions sur une page où des données étaient bien présentes lors de l'ajout des informations de celle-ci.

Pour mettre en place un traitement, il nous a tout d'abord fallu récupérer l'URL sur laquelle nous nous trouvions. Cela nous a demandé un peu de réflexion étant donné que le même problème que lors de la récupération des données se posait (problème de partage d'informations entre la page web et l'extension). Pour trouver l'information souhaitée, nous avons mis en place une requête qui se lance automatiquement lors de l'ouverture de l'extension. Grâce à l'URL ainsi trouvé, nous avons pu faire un traitement dessus afin de savoir si nous nous trouvions sur le site de La Centrale ou non. Si c'est le cas alors tout se passe comme précédemment. Dans le cas contraire, étant donné qu'à ce stade la web extension n'était disponible que sur ce site, nous affichions un message le lui indiquant.



Lorsque l'utilisateur est bien sur le site de La Centrale et qu'il souhaite ajouter les informations de la page web courante mais qu'aucune donnée n'est disponible, un message est alors affiché afin qu'il sache et comprenne pourquoi aucun point n'est alors ajouté au graphe.



Alerte si jamais aucune donnée ne peut être ajoutée au graphe

Nous avons pu remarquer un nouveau problème avec cette alerte. En effet, à chaque ouverture de l'extension, le script utile à la récupération des données est chargé. Ainsi lors de la x-ème ouverture de l'extension, il y aura x scripts lancés. A l'envoi du message, chaque script répond et donc il y avait x alertes dans le cas où aucun article n'était présent sur la page. Afin de régler ce problème, nous avons utilisé un booléen qui indique si nous sommes en attente d'un message (= d'une réponse) ou non. Ainsi, quand on envoie une requête, ce booléen est mis à vrai. A la réception d'un message, si le booléen est à vrai, nous le mettons à faux et nous traitons la réponse. Dans le cas contraire, aucune action n'est effectuée.

Nous avons aussi vu que si nous appuyons plusieurs fois sur "Add page data" alors que nous sommes sur la même page, les données sont alors présentes plusieurs fois. Nous avons alors décidé de supprimer les points qui étaient en double.

Disponibilité de la web extension sur d'autres sites d'e-commerce

Afin de pouvoir développer notre extension, il a fallu trouver un moyen de la rendre utilisable sur d'autres sites. Pour cela, nous avons dû trouver une solution qui permet d'ajouter un site facilement et rapidement une fois les XPaths nécessaires repérés sur le site. Nous avons donc décidé de mettre en place un dictionnaire par site dans le fichier "multicriteres.js". L'ajout d'un nouveau site est expliqué en détail dans le readme afin que tout le monde sache comment faire. Voici un exemple de dictionnaire (pour le site "Auchan drive") :

```
var auchandrive = {  
    "Articles" : "//article[@class='product-item']",  
    "DecimalSeparator" : "coma",  
    "Name" : "./dt[@class='product-item__title']/span",  
    "Price" : "@data-price",  
    "Average rating" : "./div[@class='rating-average']",  
    "Quantity rating" : "./div[@class='rating-quantity']",  
    "Price per kilo" : "./span[@class='product-item__price-per']",  
    "Maximize" : ["Average rating", "Quantity rating"],  
    "Minimize" : ["Price", "Price per kilo"]  
}
```

Certains champs ("Articles", "DecimalSeparator", "Maximize" et "Minimize") spécifiés dans le readme sont obligatoires. Ils permettent le bon traitement des données. De plus, si ces champs ne sont pas indiqués ou mal renseignés, lors du lancement de la web extension sur le site souhaité, un message apparaît afin qu'il connaisse la nature du problème.

The extension is not available on this site. Check the configuration file and its requirements (dict 'Articles' undefined).

Exemple de message de l'extension en cas de problème dans le dictionnaire défini

Le champ "Articles" permet de récupérer individuellement chaque article. Grâce à ce XPath, nous pouvons alors avoir la liste de tous les articles présents sur la page. Tous les autres champs (autres que les champs obligatoires) que l'utilisateur indique dans le dictionnaire doivent être des chemins relatifs par rapport à "Articles".

Le champ "DecimalSeparator" a été ajouté afin de traiter le cas des nombres décimaux suivants les sites. En effet, les sites anglais utilisent généralement des points pour séparer la partie entière de la partie décimale d'un nom et des virgules pour séparer les milliers. Tandis que les sites français utilisent la virgule comme séparateur de la partie entière et de la partie décimale. Le champ "DecimalSeparator" est obligatoirement "dot" (= point) ou "comma" (= virgule) et permet de traiter les nombres décimaux.

Les champs "Maximize" et "Minimize" permettent d'indiquer si les données (numériques) sont à maximiser ou à minimiser. Cela permet ainsi d'afficher les meilleurs points d'une manière différente. Nous développerons cette partie un peu plus tard (cf Points de Pareto). Afin de pouvoir afficher un graphe qui dépend de deux valeurs, l'utilisateur doit indiquer au minimum deux valeurs à maximiser ou minimiser.

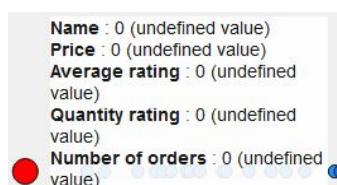
Suite à la mise en place de ces dictionnaires, nous avons ajouté “Auchan drive” à la liste des pages Internet sur lesquelles l’extension est disponible. Puis, nous avons décidé de traiter le site de “La Centrale” comme n’importe quel autre site afin de notamment simplifier le traitement et de ne plus différencier deux cas (cas de “La Centrale” et cas des autres pages).

Initialement, nous avions prévu de généraliser l’extension sur “Amazon”. Cependant, la liste des articles n’est pas toujours affichée de la même manière. Les XPaths pour récupérer les articles sont donc différents ce qui complique le traitement.

Nous avons également regardé pour déployer l’extension sur “CDiscount” mais nous n’avons pas pu car les notes sont des images. Nous n’avons donc pas la possibilité de récupérer la note de l’article par simple XPath.

En déployant la web extension sur “Auchan drive”, nous avons remarqué que parfois certains champs ne sont pas renseignés (exemple : note). Cela bloquait alors l’extension et aucune donnée n’était alors affichée car aucune information n’était transmise à l’extension. Nous avons donc décidé que les données non renseignées sur le site ont comme valeur “0 (undefined value)” dans ce cas.

Cette phase nous a également permis de nous rendre compte de la complexité à trouver des XPaths satisfaisants. Par exemple, sur “Auchan drive”, nous avons rencontré le cas où le XPath “Articles” sélectionnait une donnée qui ne correspondait pas à un article (notamment lors de présence de pub sur la page). Nous avons alors pu régler le problème en modifiant le XPath. Cependant, ce dernier persistait parfois sur certains sites et le changement de XPath ne pouvait pas le résoudre. Nous avions alors des points où toutes les données (numériques ou non) étaient à “0 (undefined value)”. Afin de régler cela, nous vérifions que tous les champs ne sont pas vides. S’ils sont tous vides, l’article n’est pas ajouté à la liste des résultats à afficher sur le graphe.

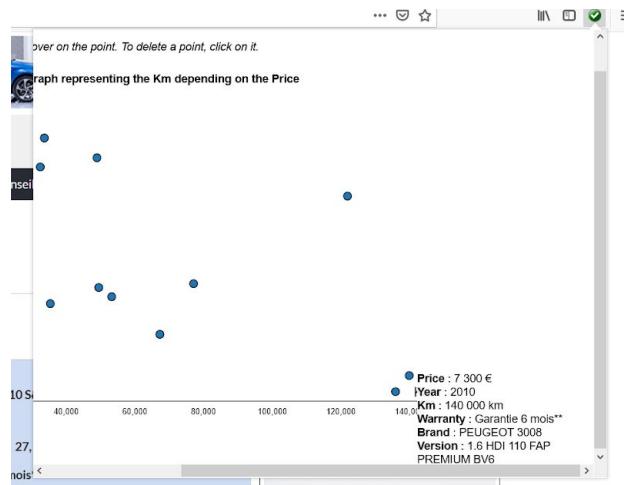


Info-bulle d’une donnée sélectionnée à tort

Améliorations visuelles et phase de test

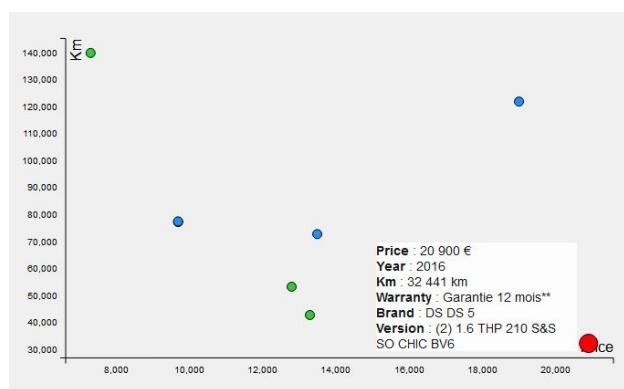
Un peu de CSS

Une fois notre extension opérationnelle, nous avons pris un peu de temps afin de la rendre plus attractive. Pour cela, nous avons fait un peu de CSS. De plus, nous avons, lors de cette phase, réglé un problème d'affichage. En effet, lorsque nous survolions un point, ces données étaient affichées dans une info-bulle en bas à droite de ce point peu importe sa localisation. Cela posait alors un problème pour les points situés trop en bas ou trop à droite car on devait alors scroller afin de voir les informations.



Ancien affichage des informations d'un point en bas à droite

Le problème était que la bulle n'a pas de hauteur définie car cela dépend des informations présentes dedans, c'est pourquoi cela nous a posé des difficultés à mettre en place. Le fait d'afficher l'info-bulle à droite ou à gauche n'a pas posé de problème car une largeur est spécifiée. La première idée pour le règlement de la hauteur de l'info-bulle a donc été de regarder combien d'informations était à afficher (en considérant qu'une information prenait en général une ligne dans l'info-bulle) et de calculer la hauteur en fonction. Cela fonctionnait globalement. Cependant, par la suite, nous sommes tombés sur des informations qui s'affichaient sur plus d'une ligne. Nous avons donc dû revoir notre code afin d'estimer plus précisément la taille de l'info-bulle en fonction du nombre de caractères de chaque information.



Affichage actuel des informations d'un point en bas à droite

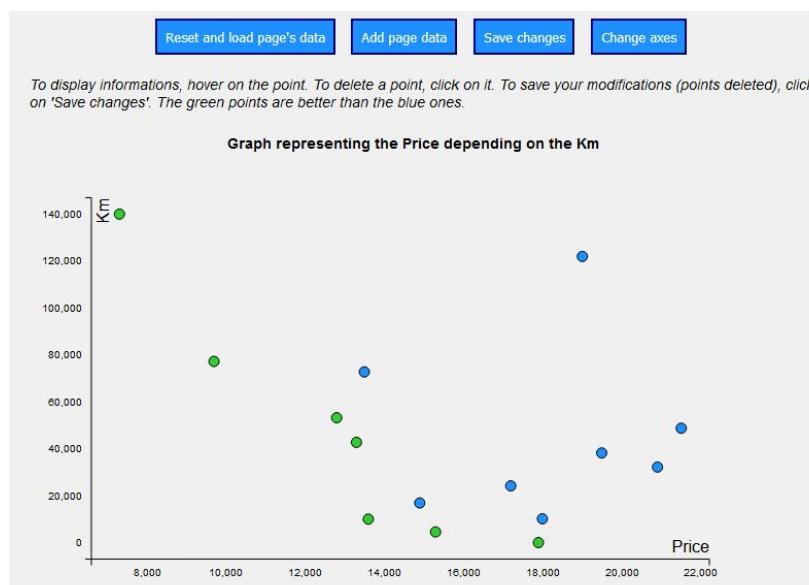
Points de Pareto

Afin de spécifier à l'utilisateur les points les plus intéressants pour lui, nous devons trouver les optimums de Pareto³. Pour cela, nous avons besoin de savoir si les axes sont à maximiser ou minimiser. C'est pourquoi l'utilisateur doit spécifier cela dans le dictionnaire lors de l'ajout d'un site. Quatre cas différents sont alors à traiter :

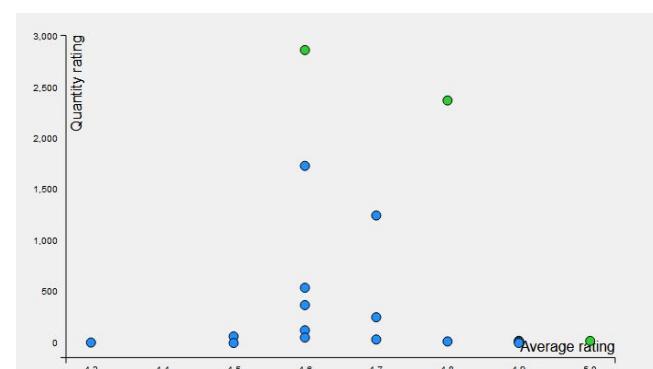
- les deux axes sont à maximiser
- les deux axes sont à minimiser
- l'axe des x est à maximiser et l'axe des y est à minimiser
- l'axe des x est à minimiser et l'axe des y est à maximiser

La phase de codage a été précédée d'une phase de compréhension de ces points ainsi que d'une phase de réflexion afin de trouver une solution dans le but d'avoir ces points.

Avant de dessiner le graphe, nous recherchons les points de Pareto suivant le cas où nous sommes. Puis, nous les enregistrons dans une variable globale. Ainsi, lors de la phase de coloration des points, nous regardons dans cette liste si le point est un optimum de Pareto. Si c'est le cas, il est alors coloré en vert, sinon il est en bleu.



Graphe avec les optimums de pareto (les deux axes sont minimisés)



Graphe avec les optimums de pareto (les deux axes sont maximisés)

³ https://fr.wikipedia.org/wiki/Optimum_de_Pareto

Phase de test

Afin d'avoir des avis et de régler des problèmes potentiels nous avons mis en place une seconde phase de test.

Lors de cette phase, nous avons fait tester notre web extension par 16 personnes. Certaines d'entre elles avaient déjà testé notre extension lors de la première phase. Elles ont ainsi pu la tester à nouveau afin de voir les améliorations. Dans un premier temps, nous avons laissé les personnes tester notre extension. Puis, en plus de l'échange que nous avons eu avec eux, nous leur avons envoyé un formulaire à remplir afin d'avoir leur avis. Le formulaire (Annexe 3) était composé de trois sections. La première permettait d'en savoir plus sur la personne. La deuxième avait pour but d'en savoir un peu plus sur leurs achats internet. La dernière partie était des questions sur la web extension. Les réponses détaillées sont en Annexe 4.

La plupart des personnes testeuses sont assez jeunes 18-40 ans et sont soit étudiants en Informatique soit travaillent dans ce domaine. Cependant, nous avons aussi pu la faire tester par quelques personnes entre 40-60 ans et qui ont des connaissances limitées en Informatique. Notre public était mixte. Cela a ainsi permis ainsi d'avoir différents points de vue.

La majorité des testeurs ont tendance à acheter fréquemment sur Internet (au moins une fois par mois). De plus, environ 80% des gens interrogés ont des difficultés à trouver l'article qu'ils souhaitent acheter. Cela est notamment dû au fait qu'il y a trop de choix qui ne les intéresse pas forcément et qu'il y a trop de sites.

Les gens trouvent l'extension utile (note moyenne de 8/10). L'extension est pour la plupart des gens relativement facile d'utilisation (note moyenne de 7/10). Les problèmes souvent reprochés étaient dus au fait que les gens ne lisaien pas forcément les phrases présentes au-dessus du graphe et donc, par exemple, supprimait des points sans comprendre pourquoi. Il faudrait donc éventuellement trouver un moyen de s'assurer que l'utilisateur ait bien lu les informations importantes lors de la première utilisation. La majorité des utilisateurs pensent que l'extension leur serait utile pour leurs futurs achats et ils la recommanderaient à quelqu'un ayant du mal à choisir ses articles sur Internet. Certains nous ont même demandés si elle était disponible en téléchargement. Cependant, à ce jour, ce n'est pas encore le cas. Pour finir, ils considèrent le graphe plutôt parlant (note moyenne de 7,75/10).

La grande majorité des gens aimeraient que l'extension soit déployée sur d'autres sites comme "Leclerc drive", "Leboncoin", "Fnac", "Darty", "Boulanger" et surtout "Amazon". Cependant, ce dernier reste compliqué à ajouter à la web extension comme nous avons pu l'expliquer précédemment.

Nous avons également demandé à une personne d'ajouter un site dans le fichier "multicriteres.js" afin de savoir si notre readme était complet et compréhensible. Cette personne a alors ajouté le site "Ali Express" qui a donc pu être utilisé par les autres testeurs. A la suite de ce test et aux remarques de la personne, nous avons mis à jour le readme en ajoutant quelques précisions.

Pistes d'amélioration

Notre web extension est actuellement fonctionnelle. Cependant, différentes choses que nous n'avions pas mises en place pourraient être intéressantes à faire. Certaines de ces pistes nous ont été proposées lors de la phase de test. Nous allons énumérer celles qui nous semblent les plus pertinentes.

L'extension n'est actuellement disponible que sur Firefox. Il pourrait donc être intéressant de la déployer sur d'autres navigateurs comme Chrome. Nous n'avons pas eu le temps de nous pencher sur cela. Nous ne savons pas donc pas juger de la difficulté et du temps nécessaire à sa réalisation.

Même si l'ajout des données d'autres pages est possible en allant sur la page et en cliquant sur le bouton "Add page data", il serait intéressant que l'utilisateur puisse également avoir la possibilité d'ajouter les x pages suivantes de manière automatique. Malgré de nombreux tests, nous n'avons pas réussi à mettre cela en place, nous estimons donc que cette tâche est complexe.

Actuellement, un seul mode de graphe est proposé à l'utilisateur. Une amélioration serait donc de pouvoir lui en proposer d'autres comme, par exemple, pouvoir avoir un graphe suivant trois données ou plus et non plus deux. Cela aurait donc un impact sur le code afin de traiter les différents cas.

Une autre fonctionnalité intéressante à mettre en place serait de pouvoir proposer à l'utilisateur de choisir lui-même les données qu'il souhaite afficher sans qu'il y ait besoin d'un fichier de configuration. L'idée serait que l'usager sélectionne un ensemble de données indiquant par exemple le prix de chaque article, la web extension pourrait ainsi en extraire le XPath correspondant. Ensuite, elle pourrait mettre en évidence tous les éléments correspondants afin que l'utilisateur puisse confirmer ou non si cela correspond bien aux données qu'il souhaite. Cela pose de nombreux problèmes. Nous pensons donc que c'est une fonctionnalité compliquée à mettre en place et qui nécessite du temps. En effet, il est déjà difficile de trouver "manuellement" les XPaths. Les trouver de manière automatique nous semble alors compliqué. De plus, nous pouvons alors nous interroger sur le nom du champ (par défaut, saisi par l'utilisateur,...). Il faudrait également trouver un moyen de savoir si la donnée est à maximiser ou à minimiser. De plus, il serait nécessaire non seulement qu'il choisisse les axes qu'il souhaite mais aussi les autres informations qu'il désire afficher dans l'info-bulle.

Il serait également utile de pouvoir trouver facilement le produit lié au point sur la page internet. Le problème est notamment la communication entre l'extension et la page qui doit se faire grâce à des "Listeners". De plus, étant donné que nous pouvons afficher les données de plusieurs pages sur le même graphe, nous pouvons nous interroger sur comment cela se passerait pour un article non présent sur la page actuelle. Une idée serait de mettre en évidence l'article (exemple : surlignage) s'il est sur la page courante. Une autre idée serait d'ajouter dans les données du point un lien qui nous mène directement à la page de l'article. Nous pensons que c'est une fonctionnalité intéressante mais qui prend du temps à mettre en place.

L'extension est actuellement en anglais. Cela limite donc le nombre de personnes pouvant l'utiliser. Une option de langue pourrait être ajoutée afin de pouvoir toucher encore plus de personnes. Cela ne nous semble pas compliqué à mettre en place. Il nécessite un changement de code avec des fichiers pour chaque langue.

Une fois Internet éteins les données sont alors perdus. Une fonctionnalité permettant de télécharger un récapitulatif des produits présents sur le graphe serait alors utile. On peut se questionner sur la façon de les afficher. La difficulté est alors liée à cette interrogation.

Conclusion

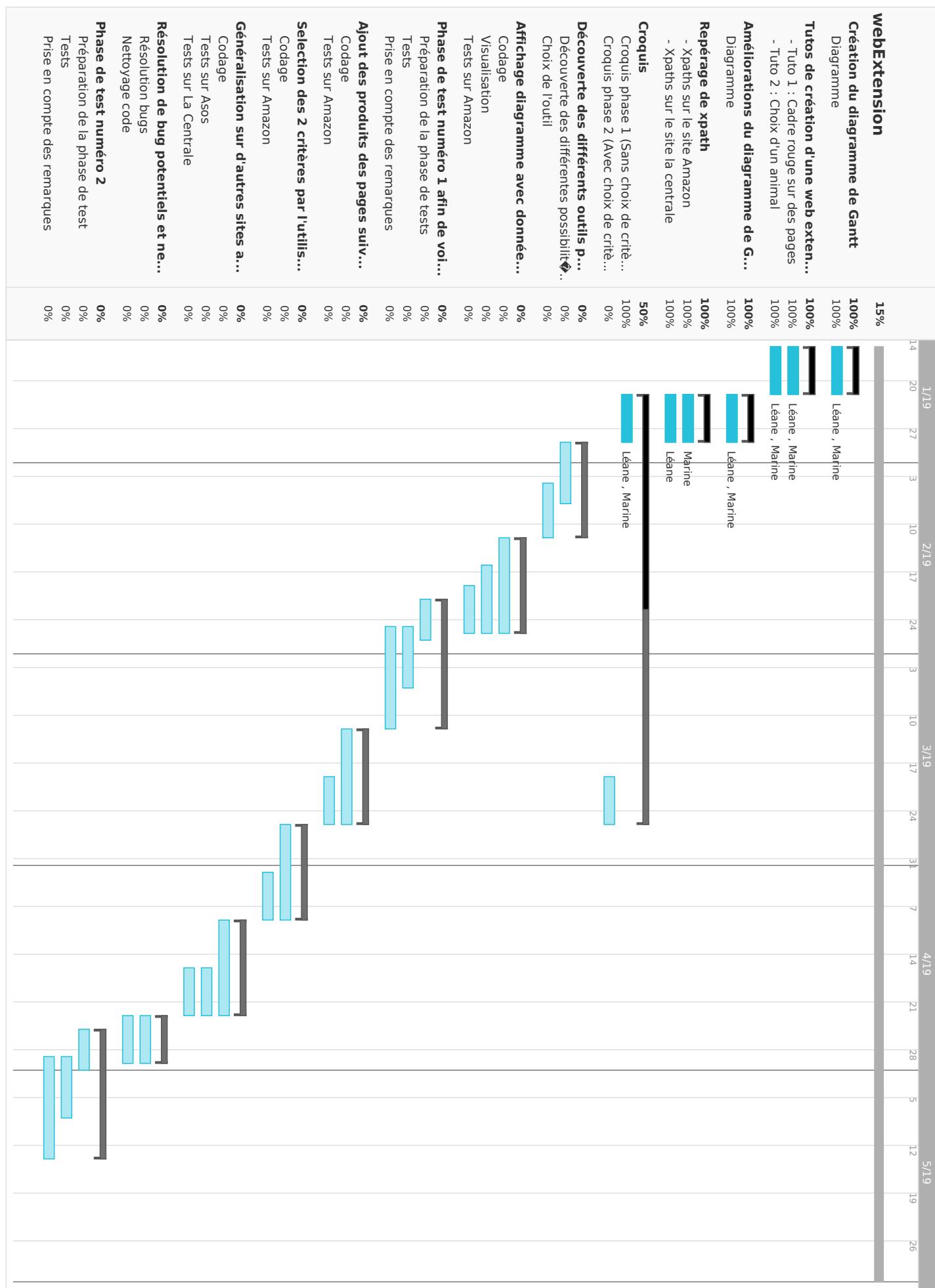
Notre web extension est actuellement utilisable. Les objectifs que nous nous étions fixés au début du projet avec notre encadrant ont été remplis. Ce sujet nous a permis de découvrir les web extensions. Ainsi, nous avons pu nous rendre compte que cela n'était pas aussi simple que ce à quoi nous pouvions nous attendre au début notamment à cause des sécurités et des méthodes asynchrones.

Afin de pouvoir améliorer l'extension par la suite, par nous-mêmes ou par des gens extérieurs, nous avons pris le temps de documenter notre code ainsi que de faire un readme afin de s'y retrouver plus facilement.

Ce projet a été enrichissant car il nous a apporté de nouvelles compétences. Il nous a permis d'apprendre à travailler d'une manière différente de ce qu'on a l'habitude de faire à la fac. En effet, nous avons pour habitude de travailler seul ou en binôme. Cependant, en règle générale, nous avons un temps hebdomadaire de travail obligatoire en salle informatique avec présence du professeur et avec un sujet précis à respecter. Ce sujet nous a alors permis d'apprendre à travailler avec moins de suivi et plus de libertés. Ce sujet nous a aussi donné la possibilité de mettre en place des idées que nous avons eu. De plus, nous avons pu utiliser nos connaissances dans un projet concret sur lequel nous étions les seules à travailler.

Le retour des gens a été gratifiant pour nous. En effet, les gens la trouvent très utile. De plus, plusieurs personnes ont montré un réel intérêt pour notre travail. Cela nous a donc montré que notre travail n'a pas été vain et qu'il pourra éventuellement être utilisé par des gens si nous la rendons disponible au téléchargement. Cela nous a énormément plus car nous aimons voir l'intérêt dans ce que nous faisons et cela est le cas étant donné que le résultat est concret et satisfaisant.

Annexe 1

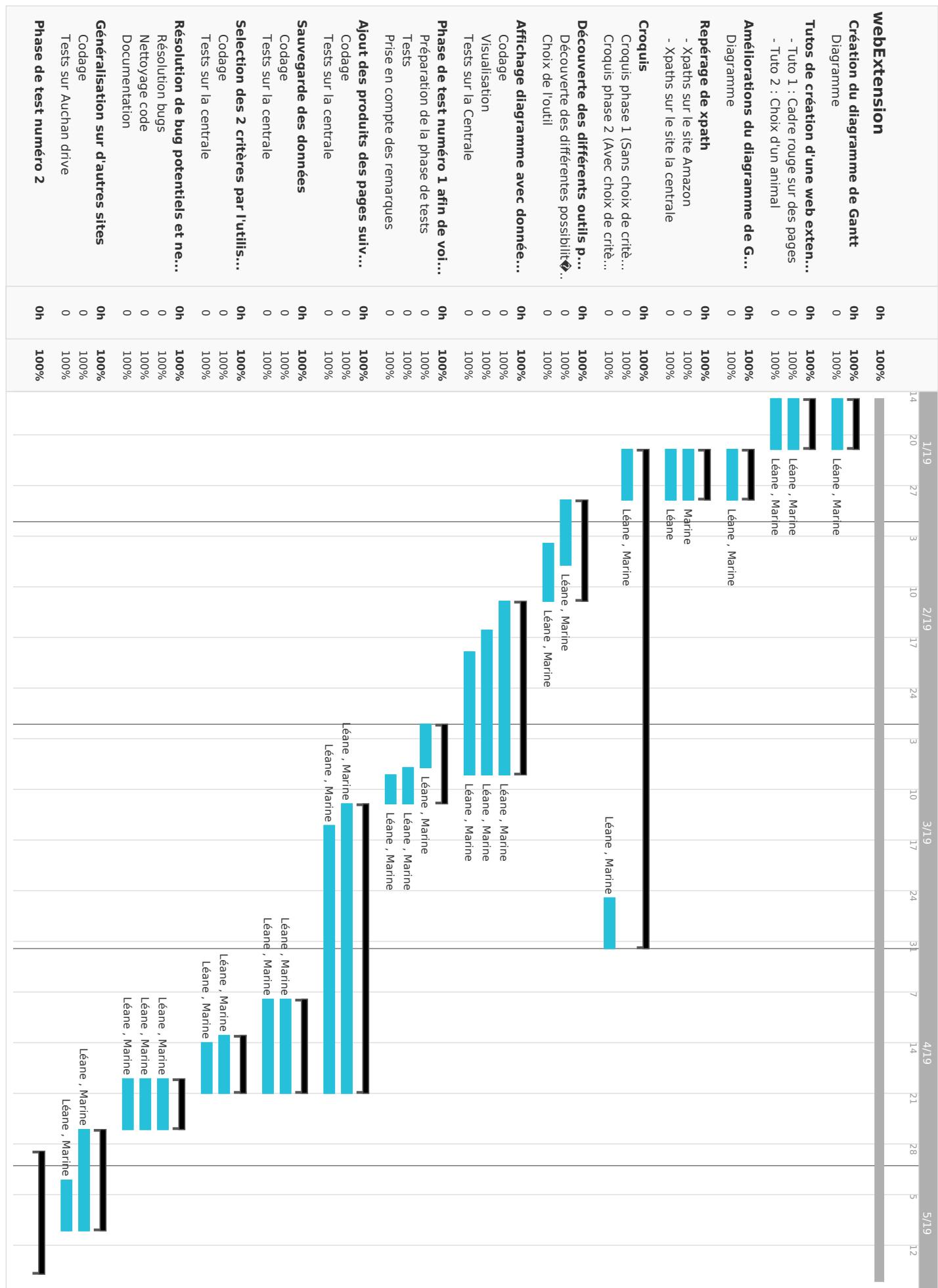


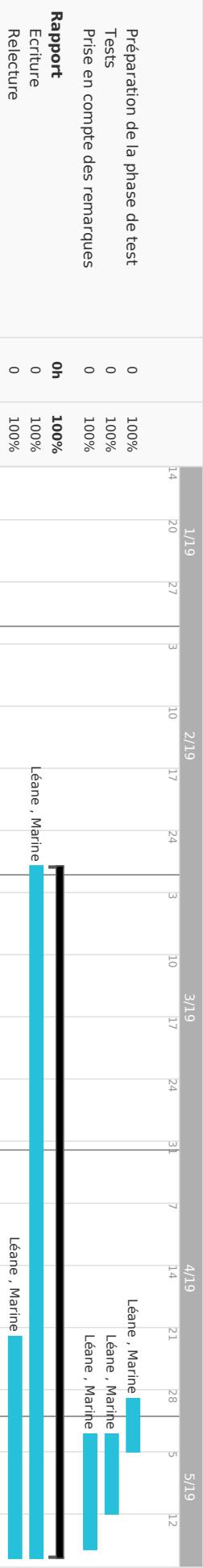
Rapport
0%
Ecriture
0%
Relecture
0%

1/19 2/19 3/19 4/19 5/19
14 20 27 3 10 17 24 3 10 17 24 31 7 14 21 28 5 12 19 26



Annexe 2





Annexe 3

You

*Obligatoire

Quelle est votre tranche d'âge ?*

- de 18 ans
- 18-25 ans
- 25-40 ans
- 40-60 ans
- + de 60 ans

Etes vous un homme ou une femme ?*

- Homme
- Femme

Quelle est votre connaissance de l'informatique ?*

- Étudiant en Informatique
- Travail dans l'informatique
- Connaissances limitées

SUIVANT

Vos achats sur internet

A quelle fréquence effectuez vous des achats via Internet (Vente en ligne, Drive, ...) ? *

- Au moins une fois par jour
- Au moins une fois par semaine
- Au moins une fois par mois
- Tous les 3 mois environ
- Tous les 6 mois environ
- 1 fois par an
- Moins d'une fois par an
- Jamais

Avez-vous parfois des difficultés à trouver l'article que vous souhaitez acheter ? *

- Oui
- Non
- Non concerné car je n'achète jamais sur internet

Si vous avez répondu oui à la question précédente, quelles sont les difficultés ?

- Trop de choix
- Pas assez de choix
- Trop de sites
- Trop d'articles qui ne m'intéressent pas
- Trop d'articles qui ne rentrent pas dans mes critères
- Autre : _____

La web extension

Trouvez-vous la web extension utile? *

0 1 2 3 4 5 6 7 8 9 10

Pas du tout

○ ○ ○ ○ ○ ○ ○ ○ ○ ○

Totallement

Trouvez-vous l'extension facile d'utilisation ? *

0 1 2 3 4 5 6 7 8 9 10

Ras du tout

.....

Total amount

Pensez-vous l'utiliser pour vos futurs achats ? *

0 1 2 3 4 5 6 7 8 9 10

Aucune chance

0000000000

Sans hésiter

La recommanderiez-vous à quelqu'un qui a du mal à choisir ses articles sur internet ? *

0 1 2 3 4 5 6 7 8 9 10

Rac du tout

0000000000000000

Sans hésiter

Le graphe est-il parlant? *

0 1 2 3 4 5 6 7 8 9 10

Pas du tout

○○○○○○○○○○

Total amount

Sur quels sites aimerez-vous que l'extension soit déployée ?

Votre réponse

Avez-vous des pistes d'améliorations ou remarques à faire?

Votre réponse

Annexe 4

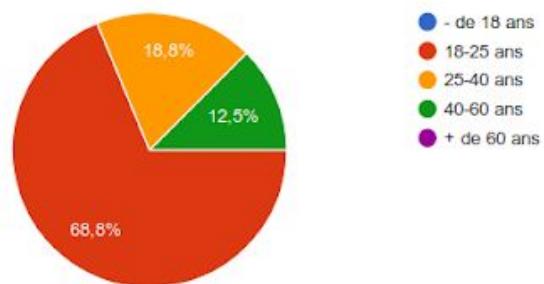
Vous

16 réponses

[Publier les données analytiques](#)

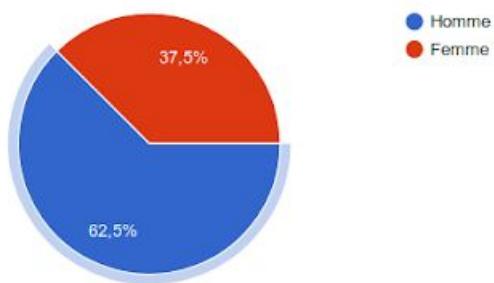
Quelle est votre tranche d'âge ?

16 réponses



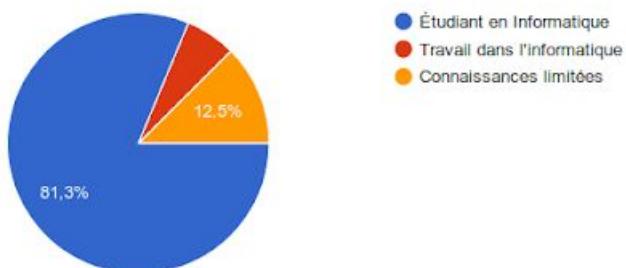
Êtes vous un homme ou une femme ?

16 réponses



Quelle est votre connaissance de l'informatique ?

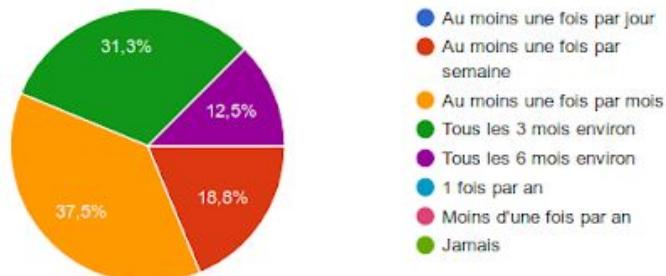
16 réponses



Vos achats sur internet

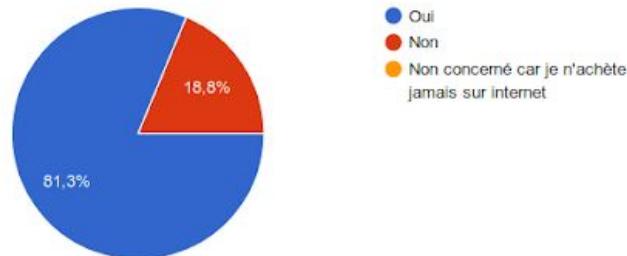
A quelle fréquence effectuez vous des achats via Internet
(Vente en ligne, Drive, ...) ?

16 réponses



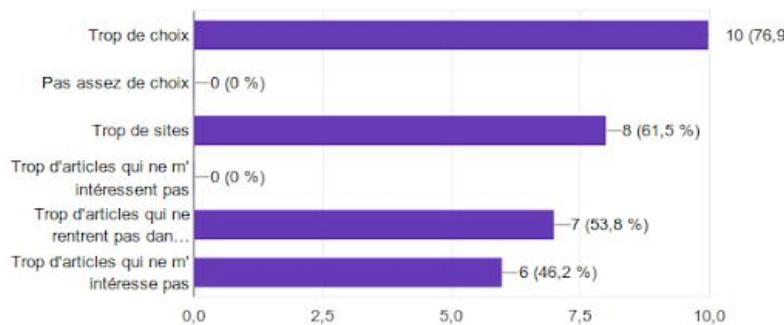
Avez-vous parfois des difficultés à trouver l'article que vous souhaitez acheter ?

16 réponses



Si vous avez répondu oui à la question précédente, quelles sont les difficultés ?

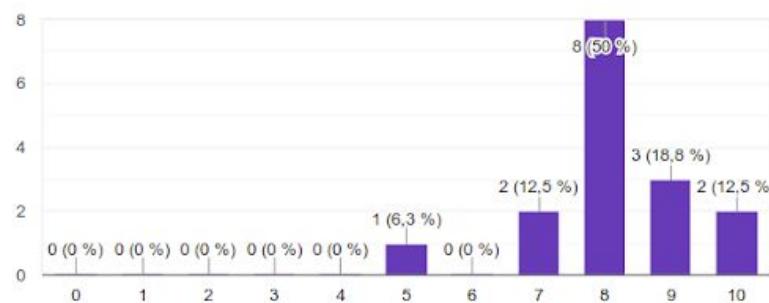
13 réponses



La web extension

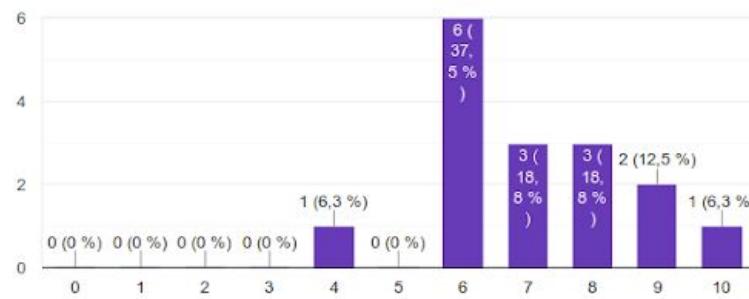
Trouvez-vous la web extension utile?

16 réponses



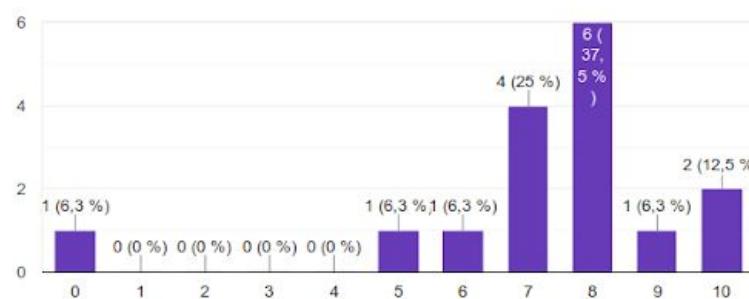
Trouvez-vous l'extension facile d'utilisation ?

16 réponses



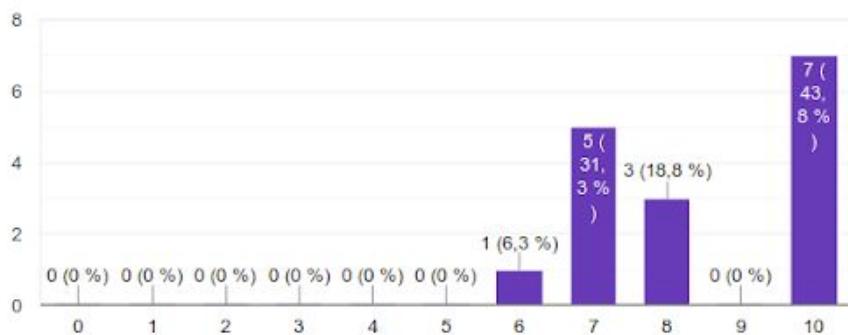
Pensez-vous l'utiliser pour vos futurs achats ?

16 réponses



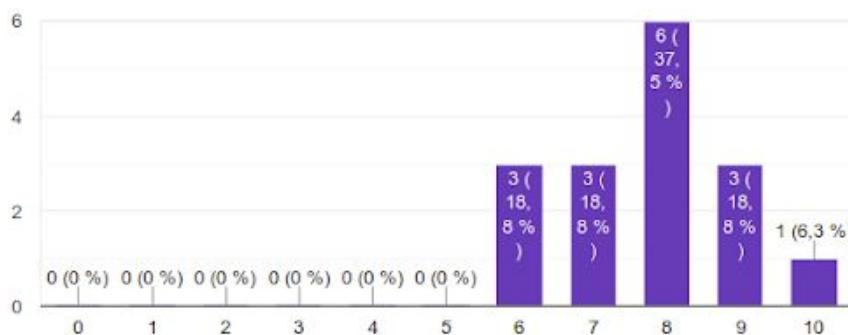
La recommanderiez-vous à quelqu'un qui a du mal à choisir ses articles sur internet ?

16 réponses



Le graphe est-il parlant?

16 réponses



Sur quels sites aimeriez-vous que l'extension soit déployée ?

12 réponses

Amazon

amazon

Amazon

leclerc drive

TOUS

lacentrale, leboncoin (site avec un même produit vendu par plusieurs vendeurs)

Auchan drive, AliExpress

Amazon, Auchan drive, Boulanger

amazon :p

Amazo, FNAC, Darty

Avez-vous des pistes d'améliorations ou remarques à faire?

11 réponses

ajouter nos propres critères

quand je cherche du thé il y a du café... déçue

sinon peut être pas assez intuitif genre j'ai pas lu le texte mais j'avais envie de cliquer sur les points (peut être bloquer ou griser le graphe tant qu'on a pas lu le texte, puis le débloquer une fois lu)

Ajouter des images aux informations ou pouvoir recharger la page web en fonction du graphe

Si possible, pouvoir distinguer les produits si on affiche des produits différents (possibilité : parser le nom du produit pour identifier sa catégorie).

Explications plus claires sur les boutons (tutoriel ?), pouvoir accéder au produit depuis le graphe, avoir toujours un point de sélectionner sur le graphe (le plus proche ?)

Imprimer le graphe

Pouvoir faire une liste de courses, et l'appli sort que les prix optimaux pour chaque articles

choisir sa langue

validation de la sauvegarde

Lien vers le produit

Lorsque l'on survole un point du graphe, que l'article soit mis en évidence sur la page si il y est présent.