Service web de statistiques de lecture de textes – Rapport de projet

Salma LAMCHACHTI
François LY

1 PRÉSENTATION GÉNÉRALE DU PROJET ET DES OUTILS NÉCESSAIRES A SA RÉALISATION

« Scriffon » est un site communautaire de publication de texte.C'est une interface de lecture minimaliste centrée sur le texte, adaptée aussi bien aux ordinateurs qu'aux appareils mobiles. Il est ouvert à tous et à pour vocation d'être international. Il n'y a pas de contrainte quant au type d'écrit mais il se concentre sur la publication de texte que le lecteur peut lire en une seule fois. Scriffon présente des fonctionnalités telles que la traduction collaborative et le service de statistiques que nous allons développer et qui permettra aux auteurs d'avoir un aperçu sur le taux de lecture de leurs textes.

2 CAHIER DES CHARGES

L'objectif principal du projet est de développer une application web générant des statistiques sur le taux de lecture des écrits publiés sur http ://scriffon.com.

La figure 1 représente une synthèse du cahier des charges.

3 OUTILS UTILISÉS :

Les outils utilisés dans le cadre de ce projet nous ont été imposés par l'entreprise Scriffon. On retrouve ainsi :

- Javascript : Langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur[1]. C'est un langage orienté objet à prototype, c'est-à -dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés.
- Prototype : Bibliothèque Javascript qui a pour but de faciliter le développement des applications web dynamiques.

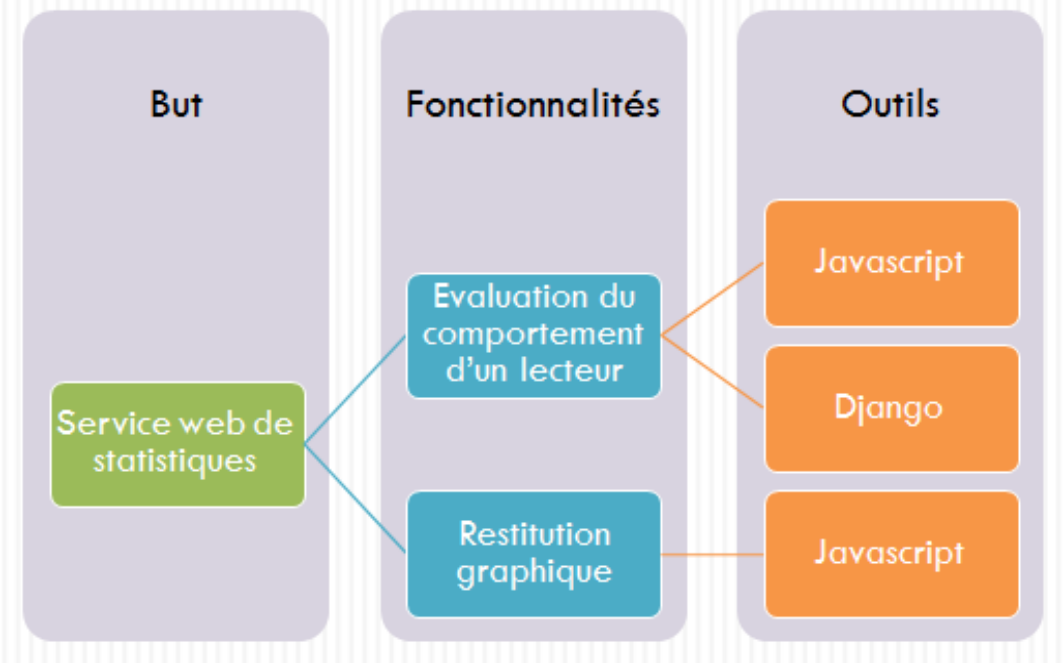


FIGURE 1: Cahier des charges

- Python : Langage de programmation multi-paradigme. Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire et d'un système de gestion d'exceptions.Il fonctionne sur la plupart des plates-formes informatiques, et est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.
- Django : Bibliothèque de développement web en Python. Il a pour but de rendre le développement web simple et rapide.

Pour le partage des fichiers et la gestion des différentes versions des codes, on a utilisé "Subversion" qui représente un système de gestion de version.

4 ASPECT TECHNIQUES

Notre service web se divise en 4 composantes, qui seront en interaction les unes avec les autres.

- Le composant A s'exécute au niveau du client, ses différentes fonctionnalités sont :
  - la demande d'autorisation de lancement d'une session de récolte de données.
  - la récupération de la position du lecteur dans le texte.

  - la communication des données au composant B.
  - la gestion des cookies.
- Le composant B sera lancé sur le serveur, il a à sa charge :
  - la mesure de la charge du serveur.
  - l'envoi d'une autorisation ou d'un refus pour le lancement d'une session de récolte.
  - la réception des données envoyées par le composant A, et le stockage de ces données dans la base de donnée.
- Le composant C sera lancé sur le serveur, il doit :
  - lire les données brutes stockées dans la base de données.
  - déterminer celles qui sont représentatives d'une lecture.
  - créer des données statistiques à partir de ces données (temps moyen de lecture, nombre de lecteurs s'arrêtant à une position donnée ...).
- Le composant D est le service de restitution graphique des statistiques.

La figure 2 représente une synthèse de l'analyse fonctionnelle de notre projet et montre les liens qui existent entre les différents composants.

5 Analyse des différences entre spécifications et réalisations :

5.1 Composant A :

- Spécifications : Lors de la spécification technique, nous avions décidé d'utiliser les fonctions Javascript suivantes pour la détermination de la position de la barre de défilement et le temps moyen de lecture de chaque écrit :
  - scrolltop() : Renvoie le nombre de pixels cachés au-dessus de la partie visible de la page.
  - window.innerHeight() : Donne la hauteur de la zone de contenu visible en pixel.
  - getheight() : Renvoie la hauteur d'un élément en pixel.
  - Ajax.PeriodicalUpdater() : Issue du "framework" Prototype, elle permet un envoi de requêtes de façon périodique.
  - Utilisation des exceptions javascript "onMouseOut" et "onKeyDown".
  - document.cookie() : Pour gérer les cookies.
- Réalisations :
  - window.pageYOffset - document.getElementById('texte').offsetTop : Renvoie la position du plus haut pixel affiché par rapport au début du texte.
  - document.getElementById('id').offsetheight : Renvoie la hauteur d'un élément en pixel.
  - PeriodicalExecuter() : Exécute de façon périodique une fonction.
- Explications :
  - Pour déterminer le plus haut pixel du texte, au lieu d'utiliser scrolltop() qui nous renvoit juste le nombre de pixels cachés au-dessus de la partie visible de la page,

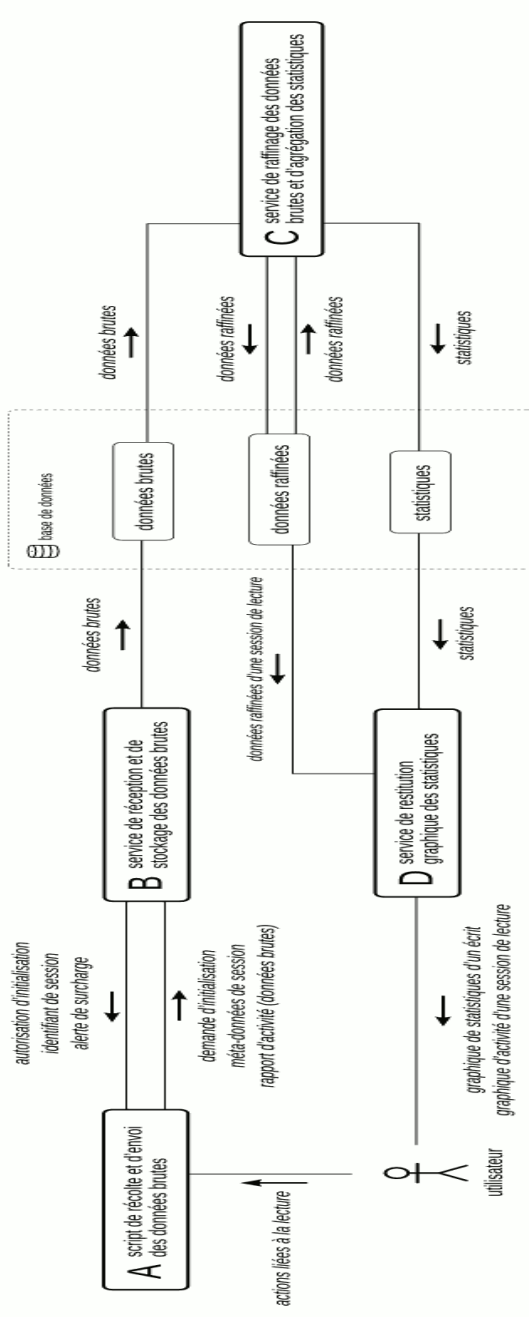


FIGURE 2: Diagramme des flux

nous avons utilisé "window.pageYOffset - document.getElementById('texte').offsetTop" qui nous permet d'avoir le pixel où débute la première ligne du texte sans prendre en compte le titre...
- Nous avons remplacé le "getheight()" par "document.getElementById('id').offsetheight" car il ne fonctionnait pas avec tous les navigateurs. Ainis, au lieu de rajouter des conditions sur le type de navigateur, nous avons préféré changé de méthode et en choisir une qui fonctionne avec tous les navigateurs.
- Nous avons supprimé l'utilisation des exceptions car nous avons trouvé une fonction Javascript qui permet de détecter la fermeture d'une page, donc une requête peut être envoyée lors de cette détection.
- Pour les cookies, après un courriel que nous a envoyé Monsieur Bouladour, il nous a suggéré de traiter les cookies avec Django. Nous n'avons pas eu le temps d'intégrer cette fonctionnalité à la fin du projet.
- Nous avions finalement besoin d'exécuter des calculs de façon périodique, et non plus seulement une requête ajax, c'est pourquoi nous avons utilisé PeriodicalExecuter plutôt que PeriodicalUpdater.

5.2 Composant B :

- Spécifications :
  - L'autorisation de session de récolte est déterminée par la charge du CPU ou le nombre de connexion.
  - La réception et le stockage des données se feront grâce au "framework" Django avec les méthodes "POST" et "save".
- Réalisations :
  - lecture de /proc/stat : permet de connaître la charge CPU. Au final nous n'avons pas pris en compte le nombre de connexions dans l'autorisation de session de récoltes.
  - json.dumps : création de réponses json aux requêtes ajax.
  - utilisation de l'url pour l'envoi d'autorisation.
  - request.POST.get : permet de récupérer les données envoyées au format json.
- Explications :
  - Afin de connaitre la charge CPU, nous ne pouvions pas utiliser la méthode getloadavg() de la classe os, car elle retourne uniquement le nombre de processus moyen dans la file du système depuis un temps donné. Il a donc fallu aller chercher l'information directement dans les statistiques systèmes, notre méthode ne fonctionne à priori qu'avec un système de type UNIX.
  - L'utilisation de json.dumps nous a permis de créer facilement des réponses au format json. Cela nous permet de donner au client un id de session, ou bien un refus de session de récolte.

- Lors de l'envoi d'autorisation, nous souhaitons récupérer le texte lu qui est concerné par cette session, afin de pouvoir lier dans la base de données session et texte lu. Cela permettra également par la suite de réaliser un système d'autorisation de sessions de récoltes plus poussé, en prenant en compte le nombre de données déjà disponibles pour un texte donné dans l'algorithme qui détermine s'il est possible d'effectuer une session de récolte ou non.

5.3 Composant C

- Spécifications :
  - Utilisation du "framework" Django.
  - Calcul des données statistiques sur les données brutes stockées dans la base de données.
  - Comparaison du nombre de mots lu entre deux envois par rapport à la vitesse moyenne de lecture.
- Réalisations :
  - Parcours de la base de données et récupération de toutes les données brutes concernant une session donnée dans un ordre temporel croissant.
  - On détermine la pertinance d'un enregistrement en comparant les données concernant la position du plus haut pixel affiché, mais aussi en prenant en compte la vitesse de lecture.
- Explications :
  - Pour nos données statistiques, nous avons déterminé la portion de texte affichée la plus éloignée du début du texte. Pour cela, on compare la position du plus haut pixel affiché des différents enregistrements. En considérant qu'un texte est presque uniforme, on peut approcher la vitesse de lecture sur une portion de texte, car on fournit dans le nombre de mots du texte dans l'url de la requête ajax. Celui-ci est connu, car on a rajouté un autour du nombre de mots d'un texte.

5.4 Composant D

- Spécifications :
  - Restitution graphique dynamique
  - Utilisation du même canevas pour tous les textes, les informations proviennent de la base de données.
  - Exemple d'aperçu rapide de ce à quoi devrait ressembler notre page web une fois terminée est donné figure 3.
- Réalisations :
  - On avait prévu de représenter les données sous la forme d'un histogramme (Voir l'exemple d'aperçu). Hors on a décidé de les représenter sous la forme

d'un histogramme et sous la forme de nuage de points.

– Explications :
  – La figure 4 représente un aperçu de la restitution graphique finale.

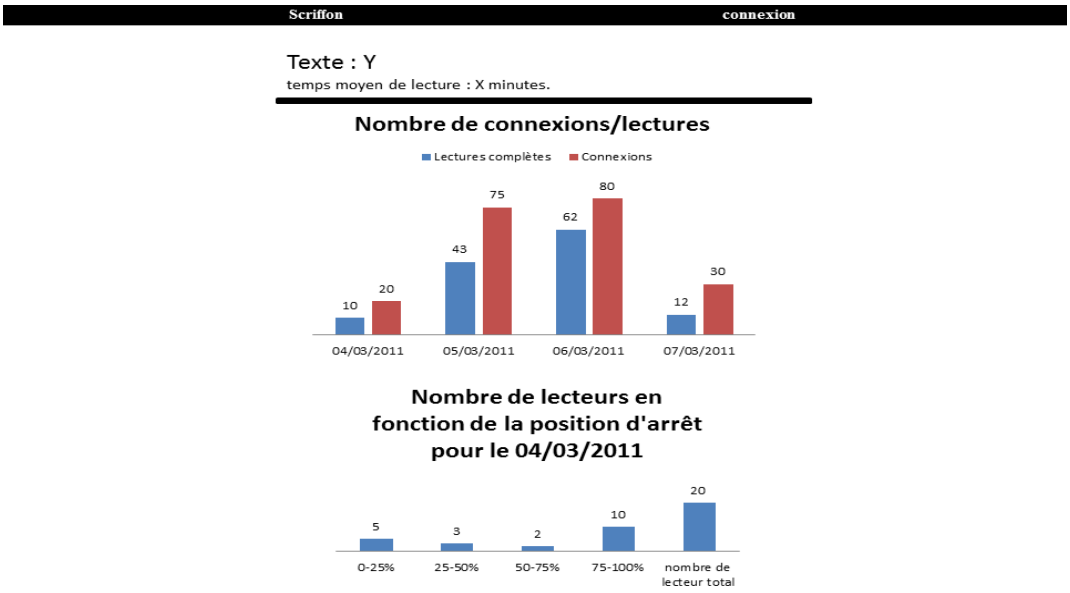


FIGURE 3: Exemple de la restitution graphique

6 RÉSULTATS DES EXPÉRIMENTATIONS ET TESTS DE VALIDATION :

6.1 Test pertinence position de fin de lecture pour texte long

– Protocole du test :
  Pour évaluer la pertinence des données sur la position d'arrêt de lecture fournit par le service web de statistiques, nous avons réalisés des tests grâce à six de nos camarades de classes. Pour cela, nous avons préparé un questionnaire 5 et leur avons demandé de lire quelques textes longs, de répondre au questionnaire et de noter en détail l'attitude qu'ils ont adopté. Nous avons pu réaliser 26 tests sur deux textes différents. Nous avons ensuite comparé ces tests avec les données brutes stockées dans la base de données.
– Analyse des résultats :
  Les données stockées dans la base de données correspondent aux données des tests effectués. Si le lecteur fait juste défiler la page sans faire de lecture,

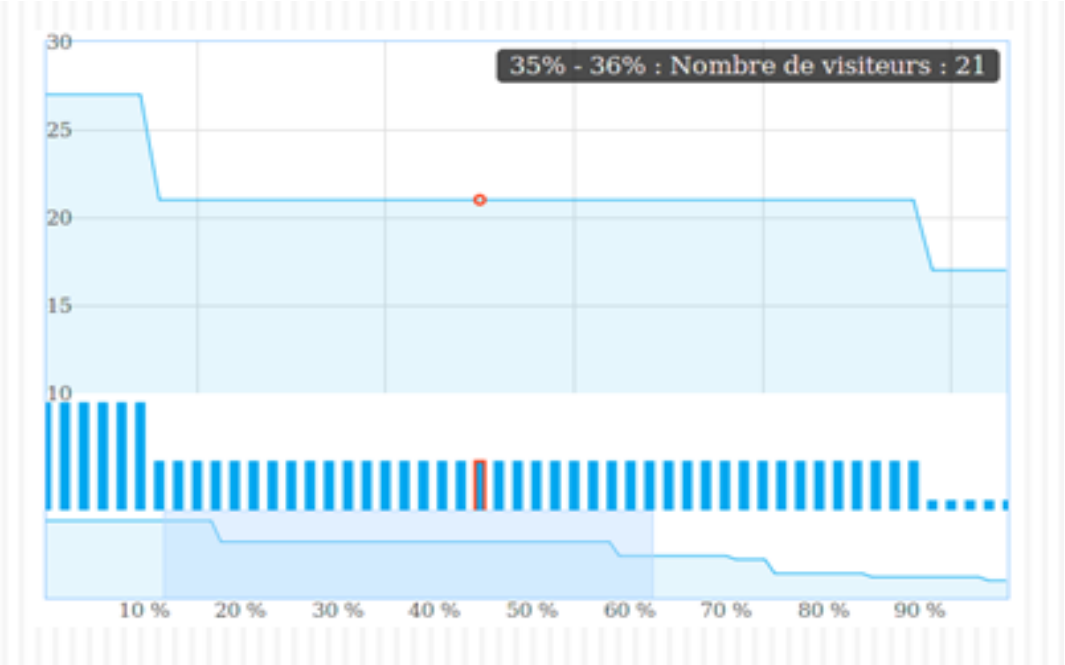


FIGURE 4: Aperçu final

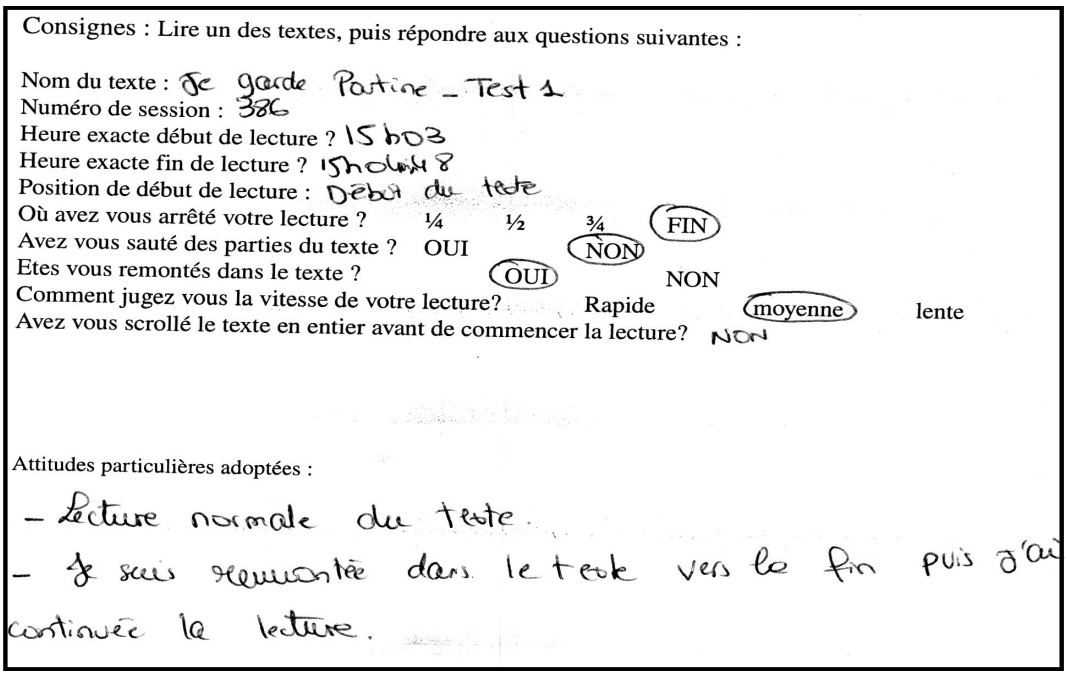
les données ne sont pas prise en compte pour le raffinage des données et la réalisation des données statistiques.

6.2 Test pertinence position fin de lecture pour texte court

- Protocole du test :
  On a suivi le même protocole que pour les textes longs.
- Analyse des résultats :
  Le serveur web de statistiques nous génère pour tous les tests que le texte en entier a été lu.En effet,l'évaluation de la position de fin de lecture repose sur la position de la barre de défilement. Or pour les textes courts, cette barre n'existe pas , donc les résultats sont faux.

6.3 Test temps de lecture d'un écrit

- Protocole du test :
  Dans le questionnaire qu'on a délivré à nos camarades, on leur demande de saisir l'identifiant de session ainsi que l'heure précise de début et de fin de lecture. Nous comparons ensuite leurs données avec les données qui ont été


Consignes : Lire un des textes, puis répondre aux questions suivantes :

Nom du texte : Je garde Patine _ Test 1
Numéro de session : 386
Heure exacte début de lecture ? 15h03
Heure exacte fin de lecture ? 15h04m18
Position de début de lecture : Début du texte
Où avez vous arrêté votre lecture ? ¼ ½ ¾ FIN
Avez vous sauté des parties du texte ? OUI NON
Etes vous remontés dans le texte ? OUI NON
Comment jugez vous la vitesse de votre lecture? Rapide moyenne lente
Avez vous scrollé le texte en entier avant de commencer la lecture? NON

Attitudes particulières adoptées :
- Lecture normale du texte.
- Je suis remontée dans le texte vers la fin puis j'ai continuée la lecture.

FIGURE 5: Questionnaire

stockées dans la base de données.
- Analyse des résultats : Lors de la première comparaison que nous avons effectuée, le temps de lecture correspondait bien aux tests effectuées, néanmoins il y avait un décalge de deux heures entre l'heure saisie par nos camarades et celle que nous avions enregistrées sur la base de données. Cela est dû au fait que nous utilisons l'heure Posix qui crée donc un décalage de deux heures. Cela a été corrigé en ajoutant deux heures à nos données.

6.4 Test du nombre de visiteurs pour chaque écrit

- Protocole du test :
  Pour chaque texte, nous avions demandé à nos six camarades de le lire. Nous avons alors ragardé le nombre d'utilisateurs dans la base de données.
- Analyse des résultats :
  Le nombre d'utilisateurs dans la base de données correspondait bien à six lecteurs à chaque fois.

6.5 Test envoi d'autorisation lorsque le CPU est surchargé

- Protocole du test :
  On surcharge le CPU, puis on lit un texte plusieurs fois. On regarde alors si les données ont été enregistrées dans la base de données.
- Analyse des résultats :
  Si le CPU est surchargé de plus de 75%, on remarque que l'identifiant de session qui donne l'autorisation de récolter des données n'est pas envoyé, et aucune donnée n'est stockée dans la base de données.

6.6 Tests lecture multi-tables

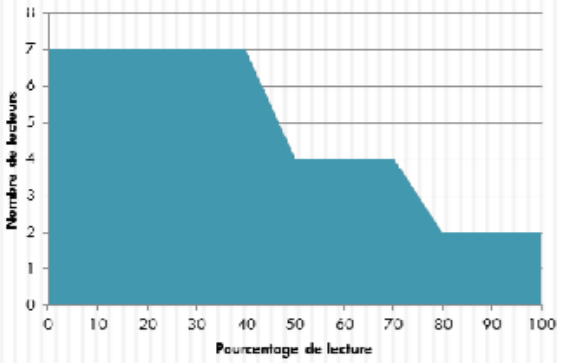
- Protocole du test :
  On ouvre plusieurs textes sur différents onglets, puis on ne lit qu'un seul texte. On regarde alors dans la base de données si des données relatives aux textes non lu ont été enregistrées dans la base de données.
- Analyse des résultats :
  On remarque que pour tous les textes qui n'ont pas été lu, notre programme considère que le début du texte a été lu. Cela est dû au fait qu'on ne détecte pas l'onglet qui est actif, et on envoi une autorisation pour tous les textes.

6.7 Test lecture multi-sessions

- Protocole du test :
  On lit un texte qui n'a jamais été lu auparavant en utilisant le même ordinateur et le même navigateur. On regrade ensuite si on est considéré comme étant un seul lecteur ou plusieurs lecteurs.
- Analyse des résultats :
  On remarque que le programme nous considère comme étant des lecteurs différents. Cela est dû au fait que nous n'avons pas gérer les cookies qui permettent d'identifier si on est le même lecteur.

6.8 Test restitution graphique

- Protocole du test :
  Grâce aux tests effectués par nos camarades de classes, nous avons pu tracer avec un tableau excel un graphique théorique représenté sur la figure 6.
  Après avoir tracé ce graphique théorique, nous avons généré avec notre programme le graphique expérimental relatif à ces tests, la figure 7 nous en donne un aperçu :
- Analyse des résultats :



Graphique théorique

Nb utilisateurs	% de lecture
7	0%
7	10%
7	20%
7	30%
7	40%
4	50%
4	60%
4	70%
2	80%
2	90%
2	100%

FIGURE 6: Graphique théorique

Pour le texte choisi, nous observons que les deux graphiques ont la même allure et donc que la restitution graphiques correspond bien aux données brutes stockées, néanmoins on remarque quelques petites différences sur la position de fin de lecture. Cela est dû au fait que les personnes qui ont réalisé les tests ne peuvent évaluer précisément la position de fin de lecture, si ils se sont arrêtés dans l'intervalle [20%-30%] ou plutôt dans l'intervalle [30%-40%] par exemple, alors que le service web nous fournit des données précises.

6.9 Bilan des résultats :

Le tableau 8 représente une synthèse des tests de validation et des résultats obtenus :

7 GESTION DE PROJET :

7.1 Découpage en tâches :

Lors de la réalisation de la spécification technique, nous avons découpé notre projet en quatre grandes tâches :

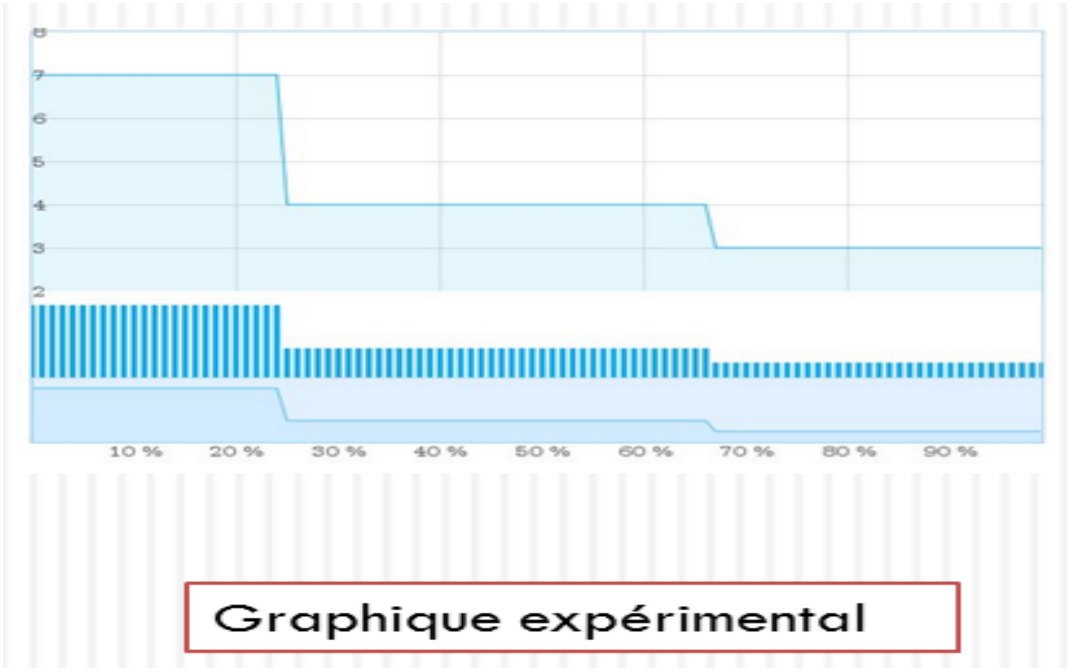


FIGURE 7: Graphique expérimental

– Le travail executé par le client.
– La gestion de la base de données.
– Le traitement des données.
– La réalisation de l'interface graphique.

Suite à ce découpage puis à un découpage plus fin, nous nous sommes répartis les différentens tâches entre nous. Nous avons ensuite réalisé le digramme de Gantt initial ci-dessous 9
Au cours du projet, nous avons pris du retard sur quelques tâches , nous avons décidé alors de travailler en binôme pour le combler et s'adapter à la nouvelle situation, pour qu'ensuite on puisse reprendre les tâches selon le diagramme de Gantt initial. La figure 10 est un aperçu du diagramme de Gantt réalisé :

7.2 S'organiser pour mieux communiquer :

– Organisaton interne :
  – Nous nous réunissions tous les lundis et les mardis après-midi pour discuter de l'état d'avancement de chacun d'entre nous, si nous n'étions pas en retard par rapport au diagramme de Gantt. On regardait ensemble ce qu'il nous fallait faire pour la semaine d'après, et si il fallait qu'on travaille en dehors des

Test	*Validation*
Détection lecture ou défilement de la page	√
Position de fin de lecture pour texte long	√
Position de fin de lecture pour texte court	✗
Temps de lecture d'un écrit	√
Nombre de visiteurs pour chaque texte	√
Envoi d'autorisation seulement si serveur non surchargé	√
Lecture multi-sessions	✗
Lecture multi-tables	✗

FIGURE 8: Synthèse des tests de validation

séances de projet pour combler un retard si il y en avait un.
  - Nous mettions en commun nos codes par Subversion, pour que chacun de nous puisse consulter les fichiers créés par l'autre et aussi pour garder toutes les versions de l'avancement du projet.
- Organisation externe :
  Monsieur Bouladour a été très présent pour nous lors de ce projet, on pouvait le contacter par courriel ou en l'appelant directement, nous avons ainsi fixé avec lui trois rencontres :
  - Une première rencontre pour mieux cibler les objectifs du projet à atteindre.
  - Une deuxième rencontre qui s'est déroulée le 24/04/2011 afin de réaliser une démonstration intermédiaire du projet, évaluer avec lui les tâches qui restaient à faire et les améliorations possibles.
  - Une troisième rencontre qui s'est déroulée le 09/05/2011 afin de réaliser une démonstration finale du projet et voir ce que nous pouvions améliorer durant la semaine qui restait.

8 FEUILLE DE ROUTE POUR LES SUIVANTS

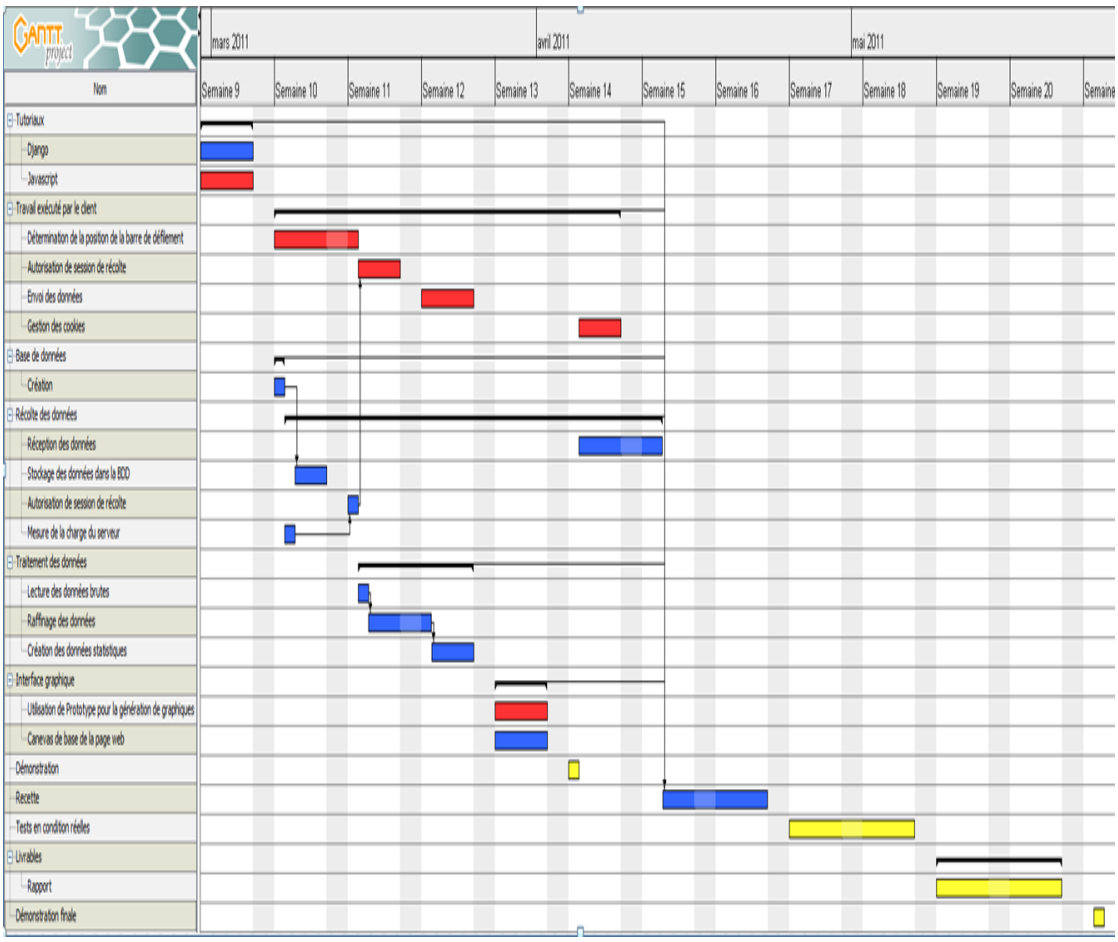


FIGURE 9: Diagramme de Gantt initial

8.1 Les compétences qu'il faut acquérir pour ce projet :

- Programmation en Python
- Programmation Javascript
- Conception d'une base de données Sql

8.2 Les documents de référence :

- www.prototypejs.org
- http ://www.djangoproject.com/
- www.humblesoftware.com/finance/index

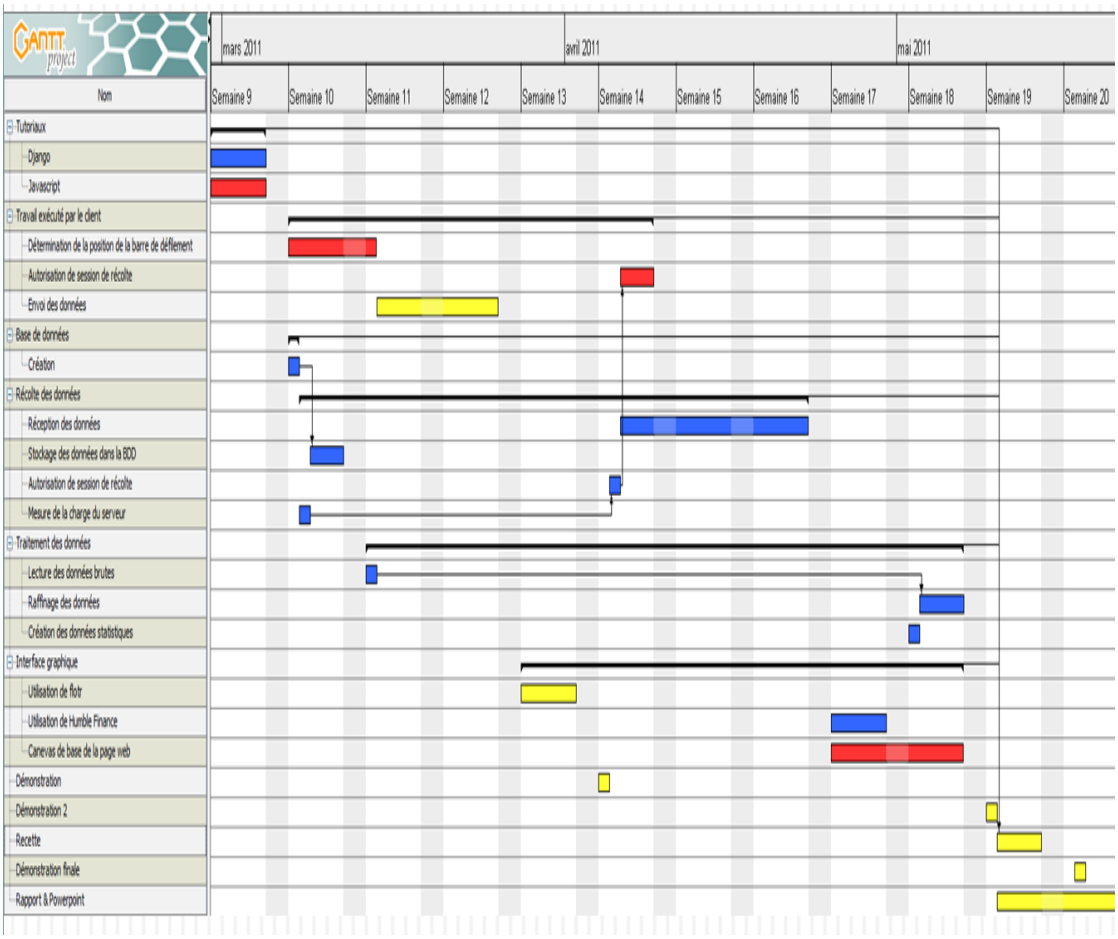


FIGURE 10: Diagramme de Gantt final

8.3 Les points critiques à prendre en compte :

- Le premier point critique de ce projet est de comprendre le fonctionnement des requêtes ajax, et et le fonctionnement de l'interaction client/serveur.
- Lorsqu'on utilise le serveur de développement de django, il est très important de spécifier toutes les urls des pages statiques dans le fichier urls.py. L'utilisation d'expressions régulières est un plus, car cela diminue le nombre d'urls à spécifier. Il en est de même pour les fichiers javascript.
- Lorsqu'on lance le serveur de développement de Django, chaque requête effectuée est affichée dans le terminal, ce qui peut être très utile pour découvrir des bugs très rapidement.

– L'utilisation de l'extension Firebug est un atout considérable pour vérifier que chaque requête effectuée fonctionne. On peut également récupérer les codes-erreurs liés à des erreurs interne du serveur.

8.4 Les points à valider pour le bon déroulement du projet :

Plusieurs points sont à valider, si on souhaite que le projet se déroule bien :
– Compréhension des requêtes ajax. Une fois que les requêtes ajax sont « maîtrisées », on peut effectuer des interactions entre le client et le serveur, et observer ce que les actions du lecteur implique sur la base de données.
– Interaction entre la base de données et le framework django.
– Données brutes cohérentes. En effet, il est essentiel de s'assurer de la cohérence des données récoltés, sans quoi il n'est pas possible d'avoir des données statistiques correctes.

9 Conculsion

Nous avons pu à la fin des 12 semaines consacrées au projet, réaliser un service web de statistiques qui donne des résultats satisfaisant pour des lectures linéaires. Ce service web peut être utilisé dans plusieurs applications telles que le taux de lecture des articles d'un journal.
Pour l'entreprise "Scriffon", ce projet valide le concept de pouvoir faire des statistiques sur la position d'arrêt de lecture des écrits et constitue une bonne base de travail qui peut être améliorée pour traiter une plus grande majorité des cas de lecture.