|  |
| --- |
|  |

|  |
| --- |
| **Cahier des charges** |
| DataBear, EIP 2017. |
|  |
| Ce document est le cahier des charges concernant le projet DataBear, il contient l’ensemble des éléments fonctionnels du projet. Nous rappellerons dans un premier temps ce qu’est un EIP ainsi que le but principal de notre projet. Puis, nous présenterons les contraintes et exigences que nous nous sommes fixées. De plus, nous décrirons notre environnement de travail ainsi que les outils que nous utiliserons. Enfin, nous fournirons un descriptif des tests que nous effectuerons ainsi qu’un plan d’organisation du projet. |
|  |
| **02/03/2015** |
|  |

Description du document :

|  |  |
| --- | --- |
| **Titre** | 2017\_CDC\_DataBear |
| **Date** | 02/03/2015 |
| **Auteur**  **Responsable**  **E-mail**  **Sujet** | LAFARGUE Vincent |
| LAFARGUE Vincent |
| Lafarg\_v@epitech.eu |
| Cahier des charges, EIP DataBear |
| **Mots clés** | Eip, Twitter, CDC, UML, Epitech, DataBear |
| **Version** | 1.0 |

Tableau des révisions :

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Auteur** | **Section(s)** | **Commentaire** |
| 01/03/2015 | Bondot Etienne | I, VI, V |  |
| 01/03/2015 | Brignon Lucas | I, II-b |  |
| 01/03/2015 | Chapelot Lucas | Résumé, I, III, IV-a |  |
| 01/03/2015 | Lafargue Vincent | I, IV-b, IV-c |  |
| 01/03/2015 | Schuler Marine | I, VI |  |
| 01/03/2015 | Thiry Victor | I, IV-b, V |  |
| 01/03/2015 | Zajda Florent | I, II-a | Présentation des contraintes fonctionnelles, 1ere vérification orthographe |

Table des matières

[I. Rappel de l’EIP 1](#_Toc413580545)

[*a.* *Qu’est-ce qu’un EIP et Epitech* 1](#_Toc413580546)

[b. *Principe de base et système futur* 1](#_Toc413580550)

[II. Les Contraintes 1](#_Toc413580551)

[*a.* *Contraintes fonctionnelles* 1](#_Toc413580552)

[*b.* *Exigences non fonctionnelles* 2](#_Toc413580553)

[III. Description des différentes parties du programme à réaliser 2](#_Toc413580554)

[*a.* *Algorithme* 2](#_Toc413580556)

[*b.* *Site Vitrine* 3](#_Toc413580558)

[IV. Présentation de l’environnement de réalisation 3](#_Toc413580560)

[*a. Environnement de réalisation 3*](#_Toc413580561)

[*i.Outils 3*](#_Toc413580562)

[*ii.Norme 4*](#_Toc413580562)

[*iii.Contraintes 4*](#_Toc413580562)

[*b.* *Composants existants* 4](#_Toc413580563)

[*c.* *Points sensibles* 5](#_Toc413580566)

[V. Description des tests 5](#_Toc413580567)

[VI. Organisation projet 6](#_Toc413580569)

[VII. Annexe 7](#_Toc413580570)

1. Rappel de l’EIP
   1. *Qu’est-ce qu’un EIP et Epitech*

Epitech est une école d’expertise informatique en 5ans qui délivre un diplôme d’expert en technologie de l’information. L’école a su au fil du temps se démarquer des autres écoles dites classiques en proposant une formation atypique se basant principalement sur le fait d’apprendre soi-même, l’entraide et une pédagogie sous forme de projets. Fini les longues heures de cours théoriques et bienvenue dans l’école de la mise en pratique instantanée.

L’EIP ou « Epitech Innovative Project » c’est l’esprit même de l’école. Il s’agit d’un projet de fin d’étude à réaliser sur deux ans avec un groupe de 6 à 15 personnes (pour les plus gros groupes en moyenne). Ce projet a pour but de faire relever aux étudiants des défis techniques et humains à la manière d’un véritable projet entrepreneurial. En effet, les étudiants devront apprendre à gérer l’aspect humain à travers leur 4ème année à l’international où ils vont se retrouver dispersés sur le globe à des milliers de kilomètres mais où ils devront rester soudés.

L’EIP est l’occasion de se créer une carte de visite professionnelle et d’apprendre la gestion intégrale d’un projet de a à z : Cahier des charges, diagramme de Gantt, développement, communication …

* 1. *Principe de base et système futur*

DataBear a pour but de récupérer et analyser rapidement et efficacement des flux twitter relatifs à des séries, des films ou des émissions télé dans le but de créer des graphes pour modéliser les données.

Le site web présentera quelques exemples de graphiques dynamiques sur les séries du moment mais en nombre limité ainsi que quelques top10 des séries qui sont en vogue. Cette partie du site sera accessible à tous. L’idée est de montrer à l’utilisateur que l’algorithme est fonctionnel et les résultats fiables, en limitant l’accès aux données.

L’objectif principal de l’EIP est, à terme, de pouvoir vendre des données à des entreprises tout en mettant en avant Twitter. Les personnes intéressées pourront nous contacter via le site pour bénéficier d’un accès privilégié et pouvoir faire des recherches approfondies sur une liste de séries plus importante.

1. Les Contraintes
   1. *Contraintes fonctionnelles*

* Anonymisation des tweets :

Nous récupérons les tweets que les gens publient à propos de séries, films, etc… afin de les analyser. Nous stockons ensuite le résultat de cette analyse, mais ne gardons aucune trace des tweets ou de leurs auteurs.

* Dépendance d’API externe (API twitter) :
  + Il est possible que notre algorithme analyse les tweets plus rapidement que l’API twitter les récupère.
  + Peu probable, mais si twitter ou l’API venaient à disparaître, nous ne serions plus en mesure de réaliser notre EIP.
* Technologies avec mises à jour constantes :

Vu le nombre de technologies différentes utilisées, nous allons être confrontés à de nombreuses mises à jour. Nous prévoyons de ne pas toutes les suivre, à moins que l’un de nos outils soit rendu obsolète par une mise à jour importante.

* Site responsive (s’adaptant à la taille de l’écran) :

Le site web devra être consultable sur mobile comme sur ordinateur. Afin qu’il soit agréable à regarder sur l’un comme sur l’autre, il faut que son design s’adapte à ces deux types d’écrans.

* Compatibilité tous navigateurs :

Nous prévoyons de rendre le site web accessible aux navigateurs internet les plus répandus : Firefox, Google Chrome, Internet Explorer 8 et versions supérieures, Opera, Safari.

* Rapport professionnel format PDF daté.

Les données pourront être consultées directement sur le site web, ou envoyées sous forme d’un rapport sur demande du client, formaté selon ses spécifications.

* 1. *Exigences non fonctionnelles*
* 100 Tweets  / seconde : C’est une performance que nous nous sommes imposée mais qui pourra changer au cours du projet car ce n’est pour l’instant qu’une donnée arbitraire.
* Sécurité (protection de l’espace professionnel, de la base de données) : Une protection accordée à nos clients pour leurs informations personnelles (nom de l’entreprise, adresse mail, mot de passe, …) ainsi que pour leurs données, acquises via nos services (graphiques et statistiques).
* Marge d’erreur décroissante en fonction de la version de l’algorithme : Il y aura forcément des erreurs dans l’algorithme, que nous nous efforcerons de réduire au fur et à mesure que le projet et donc l’algorithme avance, jusqu’à ce que, idéalement, il n’y ait plus d’erreurs.

1. Description des différentes parties du programme à réaliser

Notre projet peut se décomposer en deux parties, un algorithme de récupération et de traitement de Tweets et un site d’accès aux données.

* 1. *Algorithme*

Le programme contenant l’algorithme se décompose en trois étapes :

1. Récupération des tweets
2. Analyse des tweets
3. Stockage des résultats dans la base de données

Ce programme interagira avec l’API Twitter ainsi qu’avec la base de données de notre serveur.

* 1. *Site Vitrine*

Nous disposerons d’un site vitrine. Un écran de login permettra d’accéder à des sous-domaines disposants de fonctionnalités supplémentaires

1. Présentation du projet DataBear et de L’Équipe
2. Graphes d’exemples présentant quelques résultats au hasard
3. Demande pour avoir un devis pour passer premium
4. Écran de login
   1. Sous-Domaines
5. Fonction de recherche avec filtres (Nom de la série/Film, N° épisode, Période, Origine de la série/Film, Pays, Top 10 …)
6. Présentation des résultats de recherche : tracé de graphes et affichage résultats (X tweets, X positifs, X négatifs ...)
7. Demande de rapports

Les sous-domaines interagiront avec la base de données du serveur.

En finalité, nous disposerons de plusieurs éléments qui constitueront DataBear. Ces éléments seront :

* Un programme de récupération de tweets couplé à un algorithme d’analyse syntaxique
* Un site vitrine
* Un sous-domaine par client premium

Voir en annexe, diagrammes UML.

1. Présentation de l’environnement de réalisation
   1. *Environnement de réalisation*
      1. *Outils*

* L’algorithme sera développé en python 2.7.9. Nous nous appuierons également sur le Framework Django dans sa version 1.7.1.
* Le site et ses sous-domaines seront développés dans les langages habituels HTML5, CSS3, PHP 5. Nous utiliserons également Bootstrap dans sa version 3.3.2 pour faciliter la mise en page et rendre le site responsive.
* La base de données sera accessible sur les Systèmes de Gestion de Base de Données Relationnelle SQLite ainsi que sur PostgreSQL.
* L’algorithme utilisera l’API twitter 1.1.
* Le moteur de recherche Elasticsearch nous permettra une gestion asynchrone
* Les graphiques seront générés via Graphos, un Framework pour Django
  + 1. *Norme*

Nous ne suivrons pas de norme, néanmoins il sera nécessaire de maintenir un code propre. Les noms de fonctions et d'objets seront explicites. Une structure de contrôle sera toujours suivie d’un retour à la ligne. Dans la mesure du possible, rendre le code clair et éviter les fonctions trop longues qui s’éloigneraient de leur but principal. De plus, nous factoriserons au maximum le code pour rendre les commentaires superflus.

* + 1. *Contraintes*

Seules deux contraintes régiront notre environnement de travail :

* En cas d’update d’un de nos outils vers une version majeure, nous passerons à cette nouvelle version.
* Nous nous cantonnerons à des technologies Open-Source.
  1. *Composants existants*

DataBear se base principalement sur l’avis des utilisateurs en utilisant le réseau social Twitter. De ce fait, nous allons utiliser **l’API mise en place par Twitter**. Son principal atout est de permettre la récupération de tous les composants liés à un tweet. Exemple: les re-tweets, les ID des personnes, la mise en favoris ou encore la géolocalisation. Concernant son cycle de vie, on peut aisément dire qu’il sera assez long, en effet nous en sommes à la version 1.1 et le réseau social Twitter est encore très jeune, créé en 2006, il connaît depuis 2009 une réelle expansion et voit son nombre de comptes augmenter de jour en jour. L’API est donc d’une importance vitale pour notre projet, car elle va nous permettre la récupération du message pour l'intégrer au sein de notre algorithme et ainsi déterminer la nature positive ou négative du message.

Le deuxième outil qui va être utilisé dans le projet DataBear est **Django** un framework web python. Il est rapide, efficace et facilite le développement web. Son point fort est son interface avec la base de données qui permet un développement bien plus simplifié qu’avec une technologie plus ancienne. Son cycle de vie est comme Twitter assez long. Technologie datant de juillet 2005, Django connaît des mises à jour chaque année dont la dernière date du 2 septembre 2014. Il s’agit d’un outil qui a pour but de perdurer encore dans le temps. Sa place dans le projet est importante, puisque Django va constituer l’architecture et le développement de notre site web ainsi que l’envoi de requêtes sur la base de données. Il comprendra aussi en dernier lieu l’algorithme de traitement.

Autre outil mise en place : **Bootstrap**. Un framework d’interface où une collection d’outils facilite la mise en place de sites internet. Il est compatible avec les dernières versions des grands navigateurs. Depuis sa version 2, Bootstrap permet la création de sites adaptatifs. D’un point de vue technique, cette technologie est un agrégat d’HTML et de CSS. Nous sommes actuellement en version 3.3.0, avec une grosse mise à jour par an. On a donc une version par an.

Le dernier outil est Elasticsearch. Il va permettre de traiter la partie asynchrone de récupération et d’analyse des tweets. Il est actuellement en version 1.2 et a vu le jour en Février 2010. L’entreprise responsable de Elasticsearch a réussi une levée de fond de 10 millions de dollars et vaut aujourd’hui 700 millions de dollars ce qui en fait une technologie sure et fiable pour l’avenir.

* 1. *Points sensibles*

Nous disions donc dans la partie précédente que DataBear se basait sur le réseau social Twitter. L’un des points sensibles est donc évidemment l'obsolescence, le déclin ou la fermeture du réseau social. Si ce dernier venait à disparaître alors tout le projet serait court-circuité. Cependant, il n’est pas le seul réseau social au monde, Facebook, par exemple, pourrait être une roue de secours. De plus, si Twitter disparaissait, cela laisserait présager l’apparition d’un nouveau concurrent et ce dernier pourrait permettre de maintenir DataBear à flot.

Le second point sensible peut-être réuni en un sous-ensemble et concerne les technologies utilisées: Django, Bootstrap ou encore l’API twitter. En effet, si l’un de ces composants devenait obsolète, DataBear serait quasiment hors-service car le projet dépend de chacune de ces technologies et chacune dépend quasiment l’une de l’autre. Sans Django, le site web s’effondre. Le Bootstrap permet un contenu responsive sur mobile et/ou tablette et l’absence de l’API compliquerait et augmenterait le temps de développement. Il existe évidemment différents moyens pour palier à ce genre de désagrément : le site web peut se développer de façon plus conventionnelle, le contenu responsive peut être géré avec le CSS de manière plus classique et si l’API Twitter disparaissait alors cela voudrait dire que Twitter lui-même disparaîtrait, ce qui nous forcerait à nous rabattre sur un autre réseau social.

1. Description des tests

C’est principalement durant la construction de l’API que les différents tests unitaires seront mis en place afin de tester les méthodes d’obtention des tweets ainsi que les méthodes de parsing durant l’analyse syntaxique des données.

Les tests unitaires des algorithmes Python seront effectués via l’utilisation des outils de tests délivrés par Django au moyen des modules unittest et django.test qui permettent :

* + de créer un client de test agissant comme un navigateur afin de tester les différentes vues et méthodes implémentées
  + de simuler des requêtes POST / GET
  + de tester les rendus de templates et d’observer la transition des données pour chaque contexte

Après chaque implémentation, qu’il s’agisse d’une méthode ou d’une fonction propre à l’algorithme, cette implémentation sera suivi immédiatement du développement d’un test unitaire afin de tester la fonctionnalité.

Une base de donnée vierge de test est mise en place par les mécanismes de Django afin de servir aux tests unitaires. Cette base de donnée de test est détruite une fois l’ensemble des tests effectués.

Pour certains tests d’analyse syntaxiques, nous mettrons en place une série de tests humains (50 tweets analysé par un homme puis par la machine) afin de définir un ratio d’erreur qui devra tendre à diminuer au fil des versions de l’application.

Processus :

* définition d’une fonctionnalité pour l’application en fonction des besoins
* implémentation de la fonctionnalité
* implémentation des tests unitaires propre à la fonctionnalité
* exécution des tests
* si les tests sont concluants, implémentation de la fonctionnalité suivante
* si les tests sont non concluants, correction de la fonctionnalité

Dans les tests de l’analyse syntaxique, le ratio d’erreur devra impérativement diminuer au fil des améliorations.

1. Organisation projet

Développement du site vitrine (méthode AGILE) :

* [page] Accueil : présentation, identité du projet, possibilité de demander un devis
* [page] Béta : page de test avec quelques fonctionnalités de l’application, possibilité de faire une recherche de tendance par mot clé ou en sélectionnant une série / un film et d’obtenir, suivant le choix de l’utilisateur, un graphique des données analysées.
  + choix des filtres
  + choix du graphique de sortie : utilisation de Graphos (https://github.com/agiliq/django-graphos)
  + possibilité de partager le résultat sur les réseaux sociaux
* [page] Espace professionnel : sous domaine réservé aux professionnels souhaitant avoir une visibilité sur les flux de données et les analyses de tendances en temps réel. Cet espace sera réservé aux clients ayant souscris à une des offres proposées par DataBear et n’ayant pas commandé de devis spécifique.

Personnes : **Etienne Bondot, Marine Schuler**

Planning : première version pour décembre 2015, version finale pour janvier 2017

Mise en place de la base de données :

* création des modèles
* création des schémas relationnels
* tests

Personnes : **Vincent Lafargue, Florent Zajda**

Planning : première version décembre 2015, version finale pour mai 2016

Algorithme pur (méthode AGILE) :

* récupération de données brutes (v0.0.1)
* analyse syntaxique
* définition d’une tendance positive ou négative

Personnes : **Lucas Chapelot, Lucas Brignon, Victor Thiry**

Planning : première version pour décembre 2015, deuxième version pour mai 2016 avec tests sur la base de données, version finale pour janvier 2017

Communication:

Mise en place de toute la partie communication autour de notre EIP au niveau de l’identité visuelle et des réseaux sociaux.

* [page] Facebook
* [page] twitter
* [page] Google+
* identité visuelle : logo, charte graphique, punch line

Nous sommes actuellement à la recherche d’un graphiste.

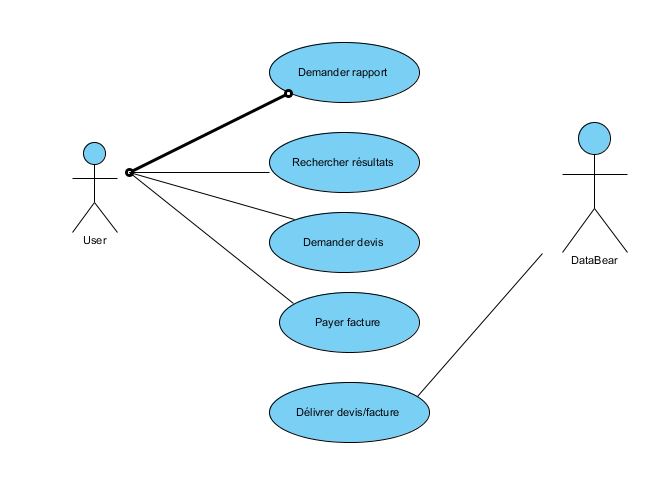
* démarchage de partenariat

Recherche d’entreprise susceptible d’être intéressée par les données générées par notre EIP.

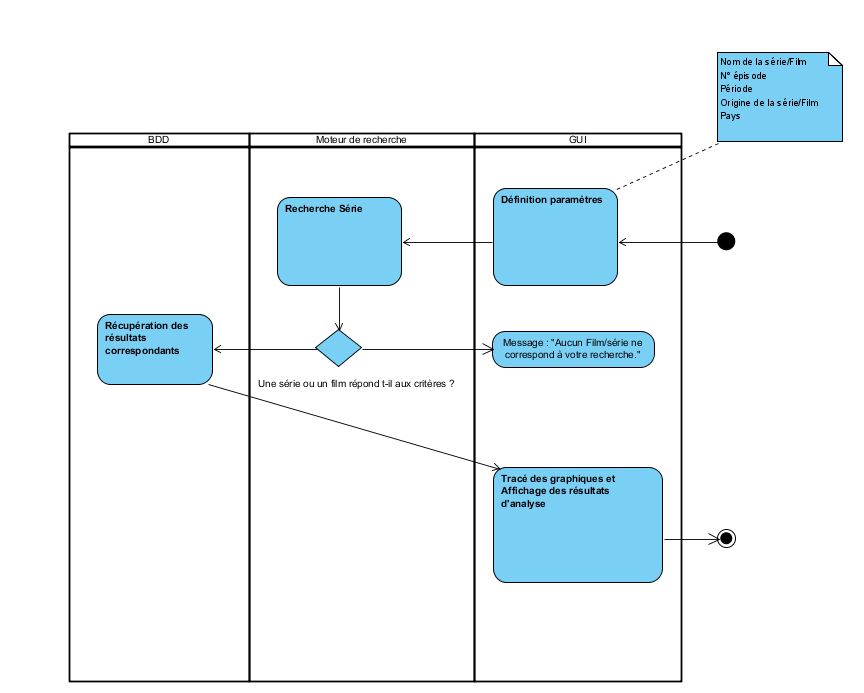
Personnes : **L’Équipe complète**

Planning : dès octobre 2015 si le logo et la charte graphique sont définis

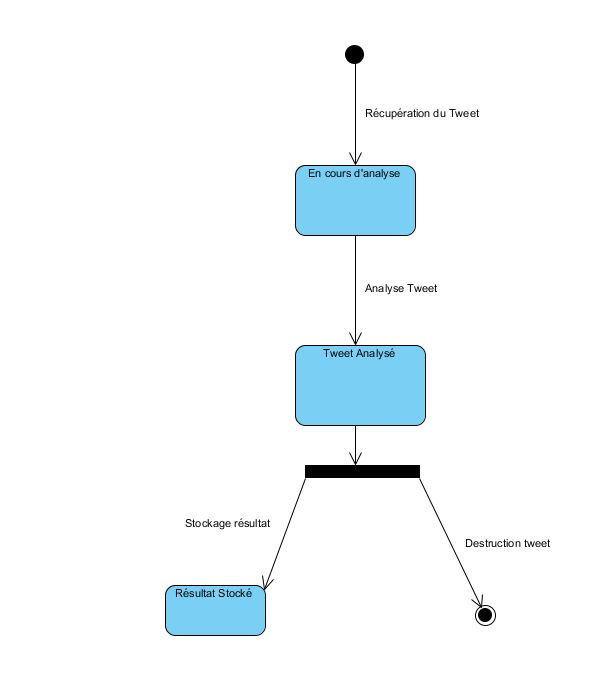
1. Annexe



1 User Case



2 Diagramme d'activité



3 Diagramme d'automate