

# S4-Développement d'un robot pour extraire des données à partir de Twitter

Donadoni Quentin  
Veynand Lucas  
Boussetta Nael

Tuteur:  
Abdessamad Imine

2020/2021



## Table des matières

<b>Table des matières</b>	<b>1</b>
<b>Introduction:</b>	<b>2</b>
Objet du document	2
Présentation du projet	2
Une description détaillée du sujet en mettant l'accent sur les missions attendues par le projet	2
Une étude de l'existant (recherche web, étude comparative, ...)	3
Une étude technique sur l'existence de solutions permettant de mener à bien le projet	5
Qui va acheter le produit ? Qui est la cible ?	5
À quels besoins le produit va-t-il répondre ?	5
Quelles sont les fonctionnalités critiques pour répondre aux besoins de façon à avoir un produit réussi ?	5
Comment le produit se situe-t-il par rapport aux produits existants sur le marché, (parts de marché, points de ventes...) ?	6
Présentation de l'équipe et rôles de chacun	6
Lucas Veynand (Chef de projet)	6
Quentin Donadoni	7
Nael Boussetta	8
Planning de déroulement du projet	8
<b>Analyse:</b>	<b>10</b>
Les fonctionnalités	10
Fonctionnalités principales	10
Fonctionnalités secondaires	10
Diagrammes UML	11
Diagramme de cas d'utilisation	11
Diagramme de classe	12
Diagramme d'activité	13
Différentes évolutions:	14
<b>Réalisation:</b>	<b>14</b>
Difficultés rencontrées:	14
<b>Conclusion:</b>	<b>16</b>

## Introduction:

### 1. Objet du document

Ce document a pour but de présenter l'intégralité de notre projet en détail. Nous allons dans un premier temps vous expliquer en quoi il consiste tout en faisant une étude de l'existant. Pour continuer, vous pourrez prendre connaissance du rôle de chacun dans notre projet et ensuite, vous pourrez consulter le déroulement de notre projet effectué selon une méthode agile grâce à l'outil Trello. De plus, il y aura une partie analyse ou il y aura les différentes fonctions de notre projet ainsi que différents diagrammes UML (diagramme de cas d'utilisation, diagramme de classe et diagramme d'activité). Dans cette partie figurera aussi notre évolution par rapport à notre vision du projet en décembre. Pour finir, il y aura une partie réalisation dans laquelle figurera nos différentes méthodes pour tester le bon fonctionnement de notre projet, mais aussi les différentes difficultés que l'on a pu rencontrer durant ce projet.

### 2. Présentation du projet

#### a. Une description détaillée du sujet en mettant l'accent sur les missions attendues par le projet

Nous allons devoir créer un robot qui aura un compte Twitter. Celui-ci ira consulter d'autres profils, et grâce à l'outil sélénium, le bot récupérera des informations. Dans un premier temps, les tests ne seront faits que sur nos comptes personnels. Ces informations seront récupérées et stockées dans une base de données. Le robot devra donc créer des tables de bases de données et y stocker les informations récupérées. Une fois la récolte de données finie sur un profil, le bot doit mettre en file d'attente tous les followers et les comptes que follow le compte étudié afin d'analyser ces comptes par la suite. L'objectif final est de pouvoir faire des requêtes sur ces bases de données afin de pouvoir en faire des statistiques.

### b. Une étude de l'existant (recherche web, étude comparative, ...)

Il existe une multitude de bots Twitter, en effet, il y a de nombreux tutoriels qui expliquent comment créer des bots pour des applications diverses. Ceux-ci sont souvent utilisés pour tweeter automatiquement ou pour envoyer des messages automatiques sur-mesure.

Parmi les bots Twitter existants permettant de répondre au mieux à nos problèmes, il y a:



- Twexlist qui sert à extraire et sauvegarder n'importe quelles informations sur Twitter (tweets, abonnés, amis, mentions, membres d'une liste, retweets, favoris, résultats de recherche, messages privés). Il stocke les informations dans un fichier Excel.



- L'API Search Tweets qui sert à extraire des publications selon une recherche particulière (un ou plusieurs mots-clés, des hashtags ou des noms d'utilisateurs ou encore une période donnée). L'API renvoie alors un JSON (format de données textuelles dérivé du langage JavaScript). Il permet de représenter de l'information structurée comme le permet XML).



- L'API Get Tweets Timelines permet notamment de récupérer le fil d'actualités basé sur nos abonnements ou de directement récupérer les 20 derniers tweets d'un utilisateur donné.







- L'API TwitterR ou R Based Twitter Client donne accès à l'API Twitter. La plupart des fonctionnalités de l'API sont prises en charge, avec une préférence pour les appels à l'API qui sont plus utiles pour l'analyse des données que pour l'interaction quotidienne. Cette API est l'API officielle de Twitter.



- Twitter Analytics fournit toutes les données analytiques de base afin de comprendre comment vos tweets impactent votre communauté. Cela inclut une vue d'ensemble du nombre d'impressions, un classement des meilleurs tweets, l'évolution de votre communauté et le taux d'engagement.

Twitter Analytics

	 <b><u>Tweexlist</u></b>	 <b><u>Search Tweets</u></b>	 <b><u>Get Tweets Timelines</u></b>	 <b>TwitterR</b>	 <b>Twitter Analytics</b>
<b>+</b>	Beaucoup d'informations extraites	Base de Données utilisée Beaucoup d'informations extraites	Base de Données utilisée	Toutes les fonctionnalités de twitter Beaucoup d'autres fonctionnalités	Très graphique et donc simple d'utilisation
<b>-</b>	Base de Données utilisée Payant	Version complète payante	Limite d'informations extraites		Incomplet en version gratuite

Si on les compare, on remarque que les 2 meilleurs sont Search Tweets et Twitter Analytics. Le premier est plus complexe, mais nous donne énormément d'informations ce qui est utile si on veut étudier beaucoup de données. Le second est beaucoup plus graphique et actuellement le plus utilisé de tous car donne des informations de manière simple et graphique ce qui permet de facilement comprendre les données, mais il reste incomplet en version gratuite. Si l'on veut effectuer des recherches plus poussées, il faudra utiliser la version payante.



c. Une étude technique sur l'existence de solutions permettant de mener à bien le projet

L'outil sélénium permet de récupérer des informations sur des pages HTML. Vis-à-vis de notre projet nous allons utiliser cet outil pour scanner des comptes Twitter sous format HTML afin d'y récolter toutes les données possibles.

Puis par la suite, nous allons stocker ces données dans des tables de notre base de données créée par le bot, pour ce faire les tables seront créées grâce à Java et notamment le JDBC.

d. Qui va acheter le produit ? Qui est la cible ?

La cible de ce produit va être notre professeur de projet tutoré qui nous donnera un compte Twitter dont on va extraire les données et sur lequel on va faire des requêtes, s'il ne donne pas de compte, on utilisera une liste des suiveurs.

La cible est le fait de pouvoir faire des statistiques sur une base de données.

e. À quels besoins le produit va-t-il répondre ?

Dans un souci de vouloir extraire de plus en plus de données de plus en plus vite et de mieux les organiser, nous allons créer ce robot qui devra répondre à tous ces besoins. Afin de pouvoir faire de bonnes statistiques sur notre base de données, il faut que le robot soit concis, rapide et efficace.

f. Quelles sont les fonctionnalités critiques pour répondre aux besoins de façon à avoir un produit réussi ?

Le bot doit extraire les données à partir de l'identifiant d'une personne initiale sur Twitter en mémorisant ces données dans la base de données. Le robot va ensuite regarder chaque follower et chaque personne suivie par le modèle et extraire ses données et les stocker dans une file d'attente pour pouvoir les étudier plus tard et ainsi de suite.

g. Comment le produit se situe-t-il par rapport aux produits existants sur le marché, (parts de marché, points de ventes...) ?

Le produit va permettre d'extraire en continue ce qui est une nouveauté puisqu'aucun bot sur le marché ne permet encore de le faire. En effet, d'autres bots existent, mais ces derniers ne permettent d'analyser les données que d'une seule personne et de donner des statistiques dessus.

### 3. Présentation de l'équipe et rôles de chacun

a. Lucas Veynand (Chef de projet)

Lucas a été désigné comme chef de projet par le groupe. Dès lors, il a mis en place certains outils afin de garantir une avancée homogène et contrôlée du projet. Un groupe de discussion a notamment été mis en place en plus d'un Excel qui a pour but de planifier les tâches à effectuer. Cet Excel a été automatisé pour un suivi de tâche plus facile (notamment grâce à un code couleur).

A CHAQUE MODIFICATION/AJOUT IL FAUT PENSER A METTRE A JOUR TRELLO ET GITHUB								
Nael	Etat	Commentaire	Quentin	Etat	Commentaire	Lucas	Etat	Commentaire
installer Selenium	Fait	reussir a ouvrir une page avec selenium	installer Selenium	Fait	reussir a ouvrir une page avec selenium	installer Selenium	Fait	essaye de faire fonctionner avec chrome
ajouter aux documents vision du projet la cible	Fait	Par exemple le prof	ajouter les nouvelles fonctionnalités a propos de la demande de requête	Fait	on a la plupart des fonctionnalités sur l'extraction mais pas celle sur la demande de requête	reflechir a la conception de la base de données (organisation)	Fait	les relations et les différents éléments stockés dans chaque table
ajouter aux documents la fait qu'on va demander une requête	Fait	on avait tout écrit en fonction de l'extraction mais pas avec la demande de requête	changer diagramme d'activité	Fait	avec les nouvelles fonctionnalités ajoutés	ajouter les fonctionnalités	Fait	(nombre max de comptes a analyser)
changer diagramme de cas d'utilisation	Fait	avec les nouvelles fonctionnalités ajoutés	Faire le diagramme de classe de la base de données	Fait	Une fois la conception de la base de données faite, faire le diagramme de classe	Créer des comptes twitter	Fait	Créer 3 comptes twitter
Créer des comptes twitter	Fait	Créer 4 comptes twitter	Créer des comptes twitter	Fait	Créer 3 comptes twitter	créer image profil SherloBot	Fait	En rapport avec un bot chercheur
		Créer le compte	Faire la classe qui gere		dans le même style que la classe attente qui était a la base un			

Il a créé de plus une bannière et image de profil pour le robot qui lui est personnel et lui permet d'être unique.



Pour la partie analyse :

- La connexion au compte Twitter
- La récupérations des différentes données
- Le passage d'un compte à un autre
- L'analyse récursive
- L'affichage dans la console
- L'ajout dans la BDD

Pour la partie requête :

- Coder les requêtes des 4 boutons
- Gérer les différentes erreurs lors des requêtes personnalisées

Autre :

- Participation aux diaporamas

## b. Quentin Donadoni

Pour la partie analyse :

- Coder quelques tables de la BDD

Autre :

- Coder l'interface Analyse/Requêtes
- Participation aux diaporamas
- Créer les différents diagrammes





### c. Nael Boussetta

Pour la partie analyse :

- La connexion à la base de données
- Coder quelques table de la BDD

Pour la partie requête :

- Coder l'interface des requêtes

Autre :


- Conception en majeure partie des diaporamas

## 4. Planning de déroulement du projet

Au début du projet, on a commencé par se demander quelles étaient les différentes contraintes imposées par le projet notamment en faisant le test de l'ascenseur. Nous avons ensuite essayé de nous familiariser avec selenium pour voir les limites de cet API. Ainsi, ce début nous a permis d'analyser comment nous pourrions réaliser notre projet à savoir créer un robot pouvant récolter des données sur Twitter.

En parallèle, nous faisons les différents diagrammes afin d'établir une ébauche de comment nous allons faire ce projet. Ces diagrammes notamment ceux de classes ont évolué durant le projet pour s'adapter au mieux à ce que l'on devait faire.

Nous avons ensuite, durant la première itération, commencé la programmation du projet. On a donc commencé par la récolte et le stockage des identifiants des followers et followes d'un compte donné dans la base de données et ainsi de suite jusqu'à ce que tous les comptes aient été ajoutés.



Lors de la deuxième itération, nous avons récupéré le pseudo, l'identifiant, la biographie, le lieu et la date de création du compte donné et des followers et followes du compte et ainsi de suite. Nous avons par ailleurs mis une option de nombre de comptes à analyser pour limiter l'analyse et qu'elle ne dure pas trop longtemps. Nous avons aussi créé une interface graphique de requête pour permettre d'effectuer les requêtes sur la base. Par ailleurs, l'interface de requêtes a été reliée à l'interface d'analyse par l'intermédiaire d'un menu qui permet la sélection entre les 2 catégories. Nous avons aussi rajouté le fait que le menu requête soit grisé et inaccessible si les tables de la base sont vides.

Pour finir pendant la troisième itération, nous avons fait que le robot récupère tous les tweets jusqu'à une certaine limite de temps; de même pour les likes. Enfin, nous avons créé des requêtes personnalisées disponibles dans le menu et nous avons permis d'écrire nos propres requêtes.

## Analyse:

### 1. Les fonctionnalités

#### a. Fonctionnalités principales

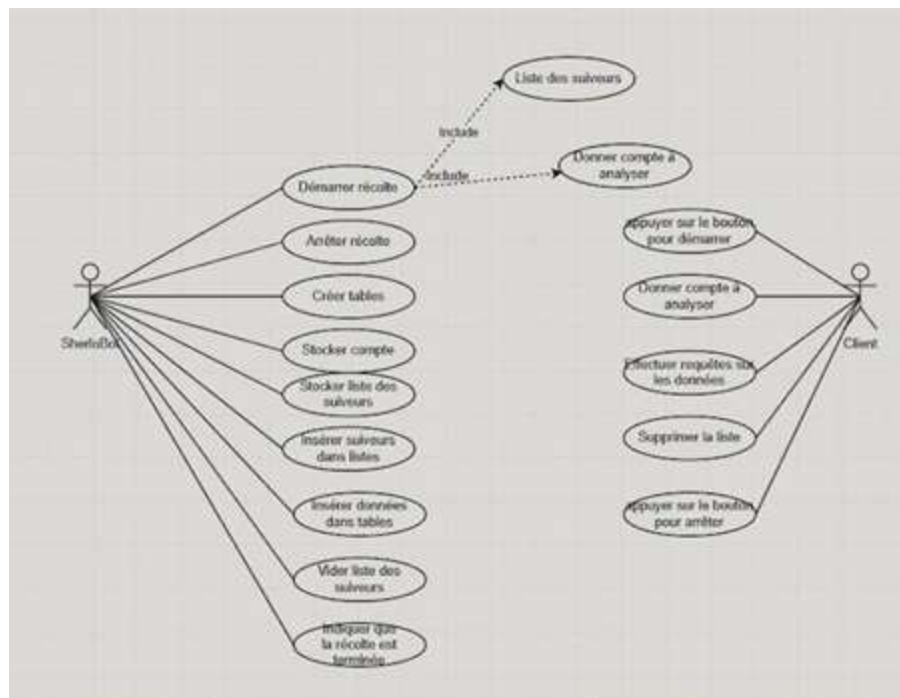
- Aller sur un compte donné
- Récupérer les followers et followes du compte et les stocker dans la base de données respectivement dans les tables ListeFollow et ListeFollowes
- Naviguer sur les followers et followes
- Récupérer la description, la date et le lieu du compte en train d'être analysé et les stocker dans la base de données dans la table Compte
- Récupérer les tweets et retweet et stocker les informations dans la base de données dans la table Tweet
- Récupérer les likes du compte en train d'être analysé et les stocker dans la base de données dans la table Lik
- Mise en place d'un menu pour choisir soit l'analyse soit les requêtes
- Possibilité de faire différentes requêtes sur la base de données

#### b. Fonctionnalités secondaires

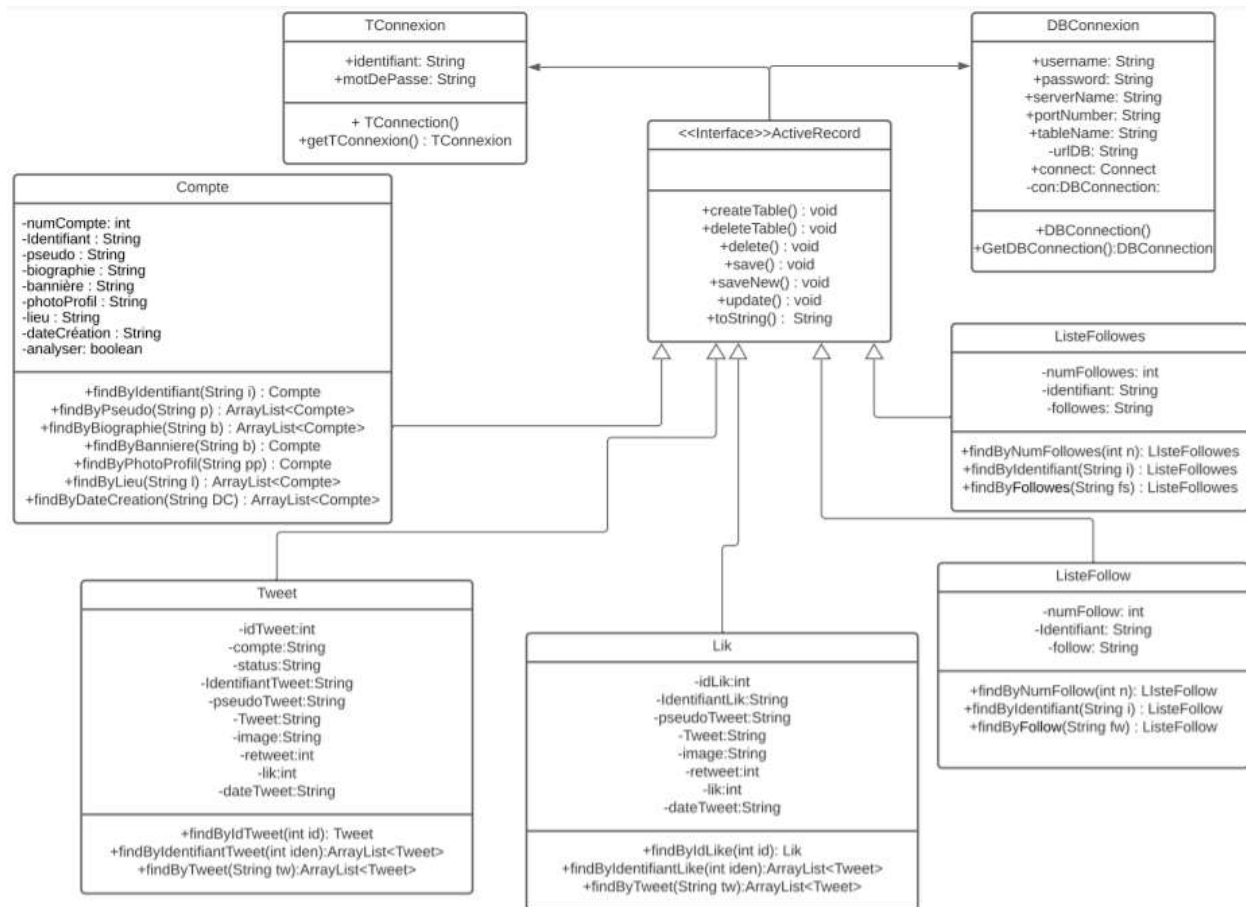
- Affiche l'analyse en direct dans l'invite de commande
- Si aucun compte n'a été donné, le robot ira automatiquement sur un compte prédéfini
- Analyser un nombre de compte avec un nombre défini (initialement 10) de compte que l'on peut changer avant de lancer l'analyse
- Empêche l'accès aux requêtes si la base de données est vide (la case apparaît alors grisée)
- Mise en place de 4 boutons faisant apparaître des requêtes prédéfinies qui apparaissent chacune dans la barre en dessous de ces derniers. On peut par ailleurs écrire sa propre requête personnalisée en langage MySql dans cette barre.

## 2. Diagrammes UML

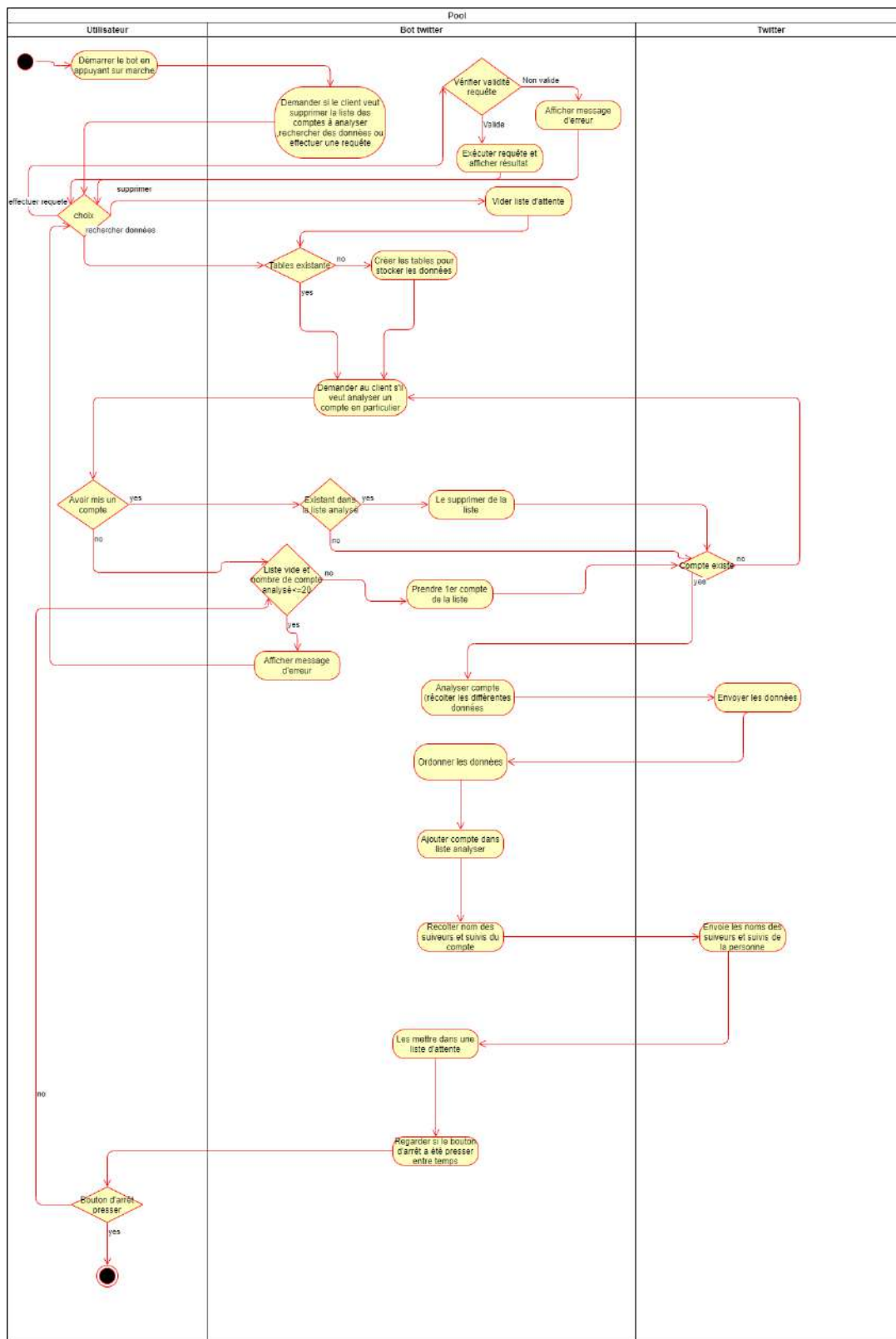
### a. Diagramme de cas d'utilisation



## b. Diagramme de classe



## c. Diagramme d'activité



### Différentes évolutions:

Nous avons beaucoup avancé depuis l'étude préalable, car nous étions seulement en train de créer la base de données et de tester comment marchait l'api sélénium. Nous avons donc terminé la base de données ainsi que ses schémas UML, ensuite nous avons créé une interface pour associer notre base à notre programme à l'aide de JDBC, après, nous avons créé une interface pour pouvoir faire la récolte de données à l'aide de la partie de programme dédiée à sélénium qui nous permet d'automatiser notre navigateur. Puis, nous avons décidé d'associer les deux interfaces à deux boutons qui sont dans la page d'accueil ce qui nous permet de choisir quoi faire. Nous avons aussi apporté une amélioration qui permet de griser le bouton des requêtes quand la base de données est vide, ce qui permet de ne pas avoir à vérifier si on a des données sur quoi faire des analyses ou pas. Ensuite, nous avons décidé de rendre plus personnalisable l'application de requête sur les bases de données en laissant à l'utilisateur le choix d'écrire lui-même sa requête. Car avant, nous utilisions des requêtes prédéfinies, ce qui aurait été embêtant, car la base n'aurait pas pu être exploitée à 100 %. Nous avons réussi à terminer notre projet, qui est pour nous un succès.

### Réalisation:


#### Tests de validation:

Nous avons effectué de nombreux tests pour vérifier la bonne récupération des données ainsi que l'insertion dans la base de données ? Un test spécifique pour chaque donnée que l'on récupère a été effectué.

Nous avons de plus effectué des tests sur la connexion du bot à son compte Twitter, car c'est une fonctionnalité indispensable qui se doit de marcher sans soucis.

#### Difficultés rencontrées:

Nous avons rencontré quelques difficultés notamment liées à Twitter. En effet, nous utilisons des chemins d'accès qui nous permettent de retrouver un ou plusieurs éléments sur la page web Twitter. Ces chemins recherchent en fait un certain type de



CSS et Twitter change beaucoup(souvent le 1er du mois) le nom ainsi que leur chemin CSS, ce qui nous oblige à retrouver et changer ces chemins pour que le bot soit toujours opérationnel. Nous avons facilité ce changement en mettant en place des variables statiques qui sont toutes regroupées au même endroit, mais cela a été pendant quelque temps une énorme perte de temps de devoir à chaque fois retrouver et changer ces variables à chaque endroit où elles étaient utilisées.



## **Conclusion:**

Pour conclure, ce projet nous a vraiment été bénéfique, il nous a appris à travailler en groupe sur un projet assez conséquent et à bien planifier nos tâches pour ne pas se laisser submerger par les choses à faire. De plus, ce projet nous a permis d'acquérir une certaine autonomie puisque nous étions assez libres de nos choix pour parvenir à finaliser le projet ce qui nous a donc poussés à faire différentes recherches afin de déterminer le meilleur moyen de le concrétiser.

Ensuite, nous sommes assez fiers d'avoir réussi ce projet, car en plus de pouvoir l'intégrer à notre CV et de pouvoir en parler durant les entretiens, cela nous a permis de nous rendre compte que nous étions capables de réussir des projets d'envergure plus conséquente.