



# Cahier des charges fonctionnel et technique

---



PT3 : Réalisation d'un bot informatique

Rova R - Benoît T - Sébastien B

<b>Présentation générale du problème</b>	<b>2</b>
Projet	2
Contexte	2
Énoncé du besoin et des objectifs	3
Contraintes et limites du projet	3
<b>Expression fonctionnelle du besoin</b>	<b>5</b>
Fonctions de service et de contrainte	5
<b>Cadre de réponse ( critères d'appréciation pour chaque fonction )</b>	<b>9</b>
<b>Backlog de produit</b>	<b>11</b>
<b>Justification des choix techniques :</b>	<b>12</b>
Choix de la technologie :	12
Pourquoi Firestore ?	13
Fonctionnement du Bot	13
Choix des API	13
Perspectives	13

## Présentation générale du problème

### I. Projet

Ce petit projet informatique consiste à développer une application informatique. L'application informatique prendra la forme d'un bot informatique ( dont je vous renvoie à la définition plus précise ci-dessous ) qui se baladera sur internet, plus précisément les réseaux sociaux, en simulant les actions des internautes humains. Une autre fonctionnalité prévue, c'est la possibilité de l'utiliser en tant que compagnon, pour qu'il puisse gérer en toute autonomie nos comptes réseaux sociaux, selon les paramètres d'entrée que nous lui passons. Il pourra ainsi, par exemple, : publier automatiquement à la date désirée, suivre des comptes, chercher selon des des mot-clés, faire une synthèse des actualités, interagir, liker, etc. Bref la liste est longue, d'où il nous sera nécessaire de travailler en mode Agile.

Un bot sur internet : Un bot sur internet est un ensemble d'algorithmes conçus pour exécuter des tâches automatisées. Ces applications réalisent des tâches simples et répétitives de manière efficace. On peut faire l'analogie avec les robots réels sauf qu'il agissent dans un monde virtuel sans corps mécanique. La plupart des bots sont utilisés à des fins marketing par les entreprises. Les grands groupes comme Facebook, Google et Twitter eux s'appuient sur des bots Internet pour leur fonctionnement, notamment les chatbots, des bots sur les messageries, qui répondent aux utilisateurs par rapport à leurs besoins spécifiques. On peut aussi parler des robots d'indexations de google qui visitent des millions de pages à la seconde. Finalement, il y a des bots dirigés par une l'intelligence artificielle, c'est-à-dire qu'ils peuvent apprendre en autonomie. Cependant, certains bots simples et bêtes suivent juste des instructions périodiques ou ponctuelles.

### II. Contexte

Ce projet s'inscrit dans le cadre PT3. Nous devons donc, d'ici le mois de Janvier, rendre une application fonctionnelle. Ce cahier des charges fonctionnel regroupe les attendus et les besoins qui devront être comblés d'ici là. Comme dit précédemment, nous travaillerons en mode Agile, c'est-à-dire que les fonctionnalités seront développées petit à petit, et nous devons nous entretenir avec le tuteur à chaque fin des sprints. ( Pour voir les dates et jalons associés à ces phases de sprints, se rapporter à la partie planning et diagramme de Gantt. ). Il est nécessaire de préciser que cette partie ne comportera pas les spécifications plus techniques que vous retrouverez dans le cahier des charges techniques.

Il n'y a pas d'autres projets en parallèle donc nous devons nous focaliser entièrement sur celui-ci.

### III. Énoncé du besoin et des objectifs

Nous n'allons pas refaire une analyse des besoins. Cependant, il est nécessaire de resituer les chiffres car il sont très parlants.

Nos études préalables nous montrent que plus de 30% des utilisateurs des réseaux sociaux ( tranche d'âge 18 - 25 ans ) passent en moyenne plus de 4 heures de leur journée en ligne; plus de 10% d'entre eux y sont plus de 6 heures en moyenne. À la question " Pensez-vous que vos passez trop de temps sur les réseaux sociaux ?", plus de 30% des questionnés répondent oui et 30% restent mitigés.

De plus, 50% d'entre eux souhaitent qu'une application de ce genre les aident à mieux gérer leur vie sociale sur internet.

De ces chiffres ressortent deux objectifs principaux qu'il faudra que cette application atteigne :


- 1. Cette application permettra à l'utilisateur d'économiser au moins 60% de son temps en faisant des rapports périodiques sur les sujets (publications drôles, famille, professionnel, etc. ) que l'utilisateur définit en amont.**
- 2. Cette application pourra gérer ses comptes réseaux sociaux ( des différentes plateformes ) en autonomie. Elle pourra par exemple gérer les demandes d'amis, les likes et différentes interactions, publier périodiquement.**

### IV. Contraintes et limites du projet

Contraintes internes: L'interface doit être claire et compréhensible pour les nouveaux utilisateurs.

Contraintes temporelles: La recette informatique devra se faire en décembre, c'est-à-dire que les fonctionnalités voulues soient développées et testées d'ici janvier ( se rapporter à la partie planification ). Les sprints devront être respectés et des rendez-vous périodiques avec le tuteurs organisés. Travailler en mode Agile nous assurera que les besoins formulés seront comblés.

Contraintes sur les ressources humaines: Nous sommes 3 développeurs - concepteurs dans le groupe. Nous devons donc nous organiser pour distribuer au mieux les tâches. En plus du développement, nous devons faire nous même la conception ce qui n'est pas une tâche aisée. Nous devons donc diviser les différentes tâches selon les capacités de chacun afin d'avancer au plus vite. Comme nous ne travaillerons pas forcément sur les mêmes points, des problèmes pourront émerger. Nous prendrons donc le temps entre chaque sprint de voir ce que chacun a fait et essayerons de trouver des solutions afin d'aider celui qui bloquera.



Contraintes de développement: Le langage de développement, la plateforme ainsi que des objectifs intermédiaires seront imposés par le tuteur de projet. Nous aurons donc l'obligation de les respecter même s'il nous incombe de trouver par nous même comment procéder. Évidemment, ces obligations techniques seront plus détaillées dans le cahier des charges techniques mais il est important de les citer ici car le tuteur est implicitement celui qui formule les besoins; Concrètement, les consignes sont les suivantes :

- Le bot devra posséder et gérer ses propres comptes gmail, twitter, facebook, instagram et Dropbox.
- Il doit être capable d'envoyer des notifications et ses sms.
- Il devra posséder une base de données locale et en ligne.
- Son activité sur le net sera guidée de manière semi-automatique par un module d'IA
- Il devra utiliser les bibliothèques existantes et des API.

Contraintes budgétaires: Ce projet n'est pas soumis à des contraintes budgétaires.

Contrainte externe: Nous devons respecter les lois en vigueur face à la manipulation des données à caractère personnel. Ce point est critique car notre application aura un accès total aux comptes des utilisateurs et leur vie privée. Nous devons ainsi garantir une sécurité infaillible pour protéger ces données critiques.

Limites du projet: Nous ne développerons pas une IA totalement autonome car ceci requiert des compétences très poussées que nous ne possédons pas.

Nous ne réutiliserons en aucun cas et sous aucune forme les données personnelles des utilisateurs de l'application.

Nous répondrons aux besoins formulés en évitant toute fonctionnalité superflue ou inutile.

## Expression fonctionnelle du besoin

### V. Fonctions de service et de contrainte

Étant donné que nous voulons réaliser un projet, on doit d'abord se poser des questions fondamentales pour savoir exactement dans quel but on fait cela.

À partir des besoins formulés, nous pouvons déduire grâce à la méthode APTE ce diagramme "bête à cornes" assez simple:

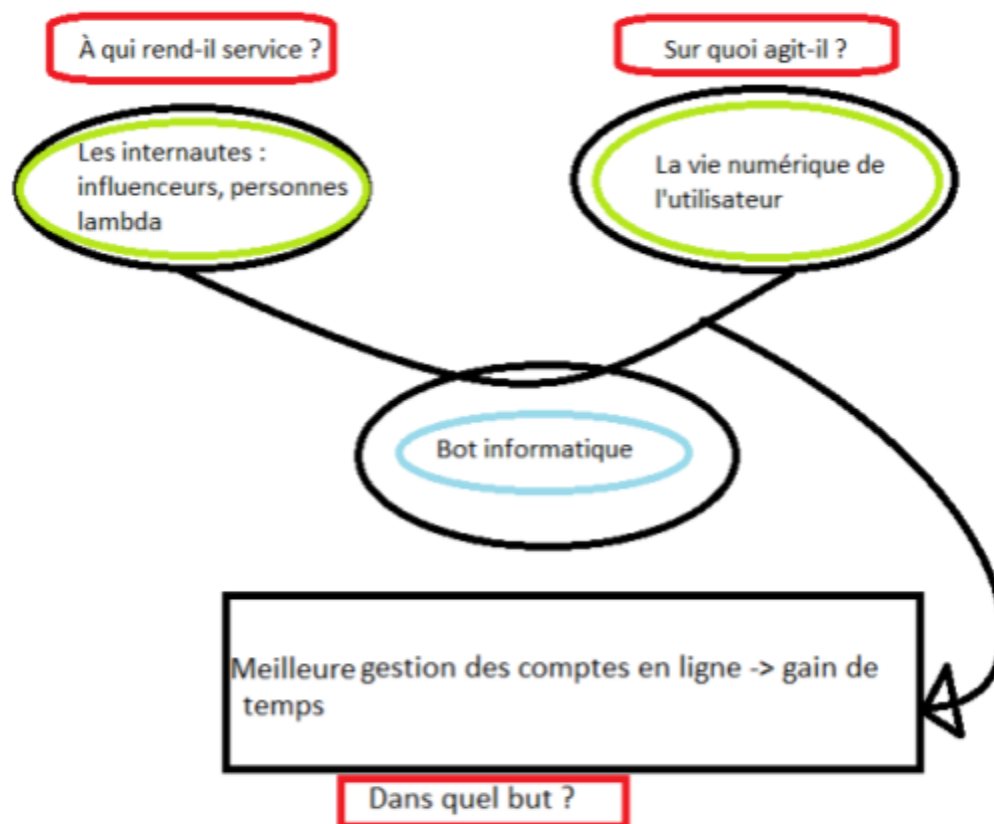
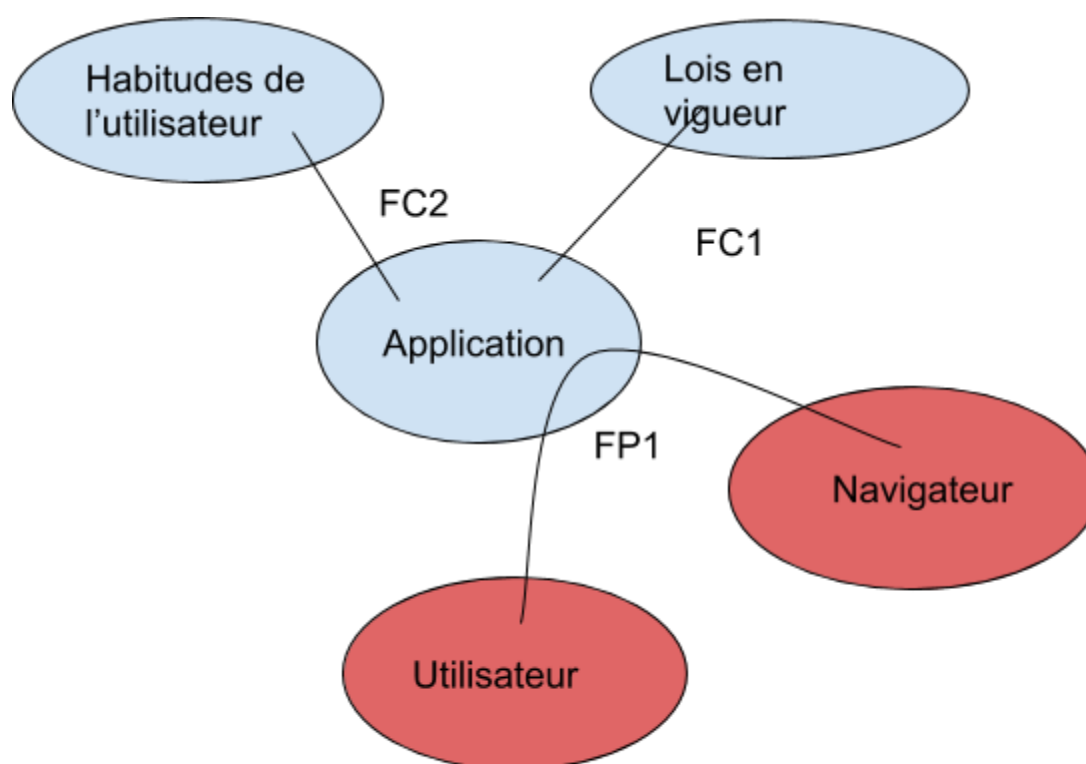


Illustration 1 : Diagramme bête à cornes

En résumé, notre projet a pour élément central le développement d'une application informatique pour rendre service aux internautes. Ceci lui permettra de mieux gérer sa vie numérique dans le but de lui éviter une utilisation abusive et non-productive.

Cependant, nous devons définir les fonctions de services attendues et générées par l'utilisation du produit. Pour cela, nous vous proposons le diagramme pieuvre afin de mieux détailler les besoins et les contraintes liés à l'environnement du produit et à sa vie

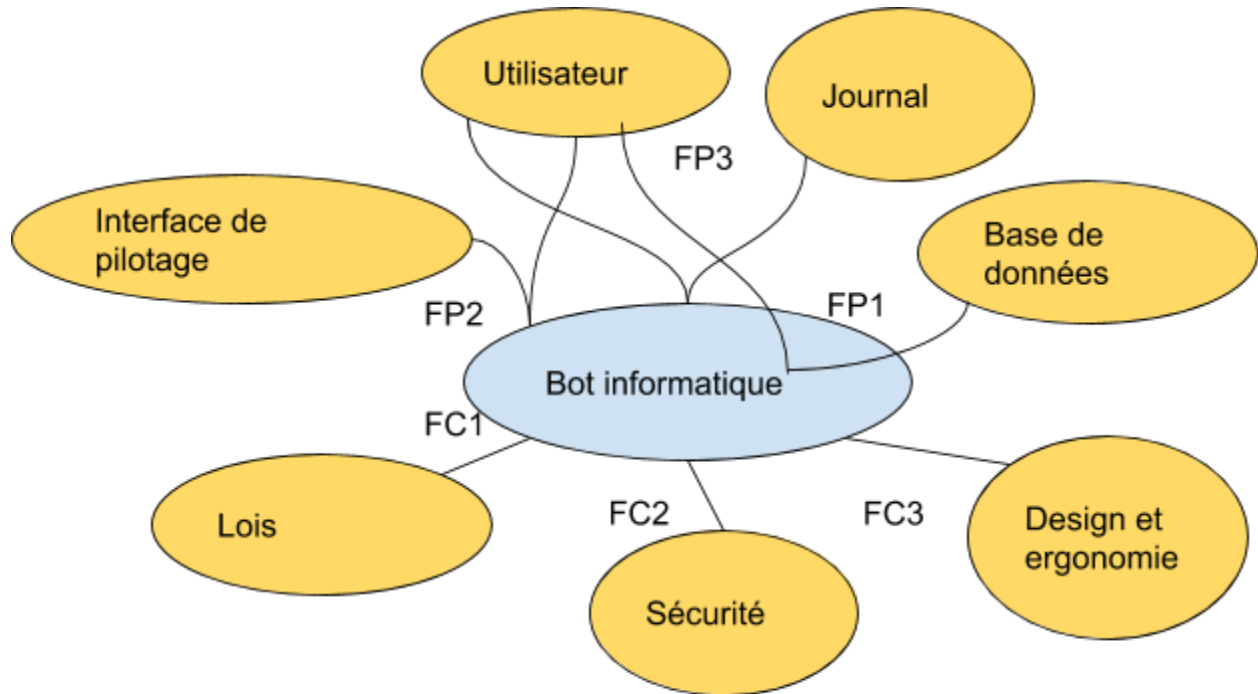
#### Diagramme Pieuvre - Installation :



*Illustration 2: Diagramme pieuvre - Installation*

Fonctions principales
<b>FP1: Permettre à l'utilisateur d'utiliser l'application sur n'importe quel navigateur.</b>
Fonctions contraintes
<b>FC1: Respecter les lois par rapport aux données à caractère personnel</b>
<b>FC2: S'adapter aux habitudes des utilisateurs</b>

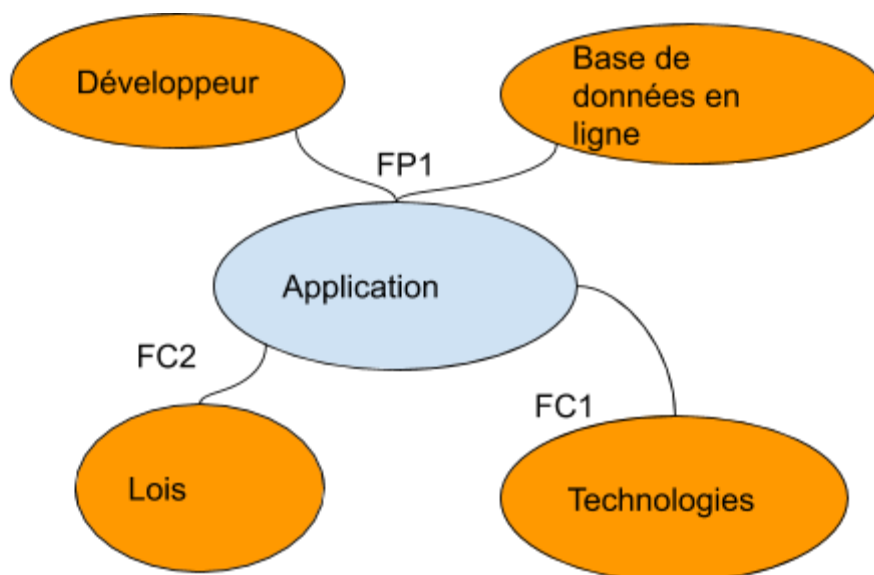
### Diagramme Pieuvre - Utilisation :



*Illustration 3: diagramme pieuvre - utilisation*

Fonctions principales
<b>FP1: Permettre à l'utilisateur de connecter et enregistrer ses comptes principaux</b>
<b>FP2: Permettre à l'utilisateur de paramétrer le bot</b>
<b>FP3: Permettre à l'utilisateur de connaître l'utilisation faite de ses comptes (journal)</b>
Fonctions contraintes
<b>FC1: Prendre compte des lois en vigueur</b>
<b>FC2: Utiliser un cryptage pour sauvegarder les identifiants</b>
<b>FC3: Assurer une bonne ergonomie et fournir une documentation détaillée</b>



**Diagramme Pieuvre - Maintenance :***Illustration 4: diagramme pieuvre - maintenance*

Fonctions principales
<b>FP1: Permettre les sauvegardes distantes pour éviter la perte des données</b>
Fonctions contraintes
<b>FC1: Suivre l'évolution des technologies (supports)</b>
<b>FC2: Suivre l'évolutions des lois</b>

## Cadre de réponse ( critères d'appréciation pour chaque fonction )

Fonction	Objectifs	Critères d'appréciation	Niveau de priorité
Installation			
<b>FP1: Permettre à l'utilisateur d'utiliser l'application sur n'importe quel navigateur.</b>	-L'interface principale sera disponible via un navigateur simple	-N'importe quel navigateur fera l'affaire, que ce soit sur mobile ou sur ordinateur	<b>-Priorité haute</b> : il s'agit de l'objectif du premier sprint, pouvoir afficher une interface
<b>FC1: Respecter les lois par rapport aux données à caractère personnel</b>	-L'application doit être conforme aux lois sur les Données personnelles	-Les usages devront respecter les normes que l'on retrouvera plus facilement sur le site de la CNIL <sup>1</sup>	<b>-Priorité haute</b> : nous manipulons des données personnelles critiques
<b>FC2: S'adapter aux habitudes des utilisateurs</b>	-L'utilisation doit être familière à l'utilisateur	-L'utilisateur ne rencontre aucun problème au niveau du paramétrage	<b>-Priorité moyenne</b>
Utilisation			
<b>FP1: Permettre à l'utilisateur de connecter et enregistrer ses comptes principaux</b>	-Création d'une interface de connexion	-Les comptes ont bel et bien été enregistrés	<b>-Priorité haute</b> : il s'agit de l'objectif du premier sprint, pouvoir enregistrer les comptes
<b>FP2: Permettre à l'utilisateur de paramétrer le bot</b>	-Création d'une interface de paramétrage	-Les paramètres sont compréhensibles par l'utilisateur	<b>-Priorité moyenne</b> : au début, imposer des paramètres
<b>FP3: Permettre à l'utilisateur de connaître l'utilisation faite</b>	-Création d'un journal des activités du bot	-Le journal doit être clair, concis et pertinent. Il doit être exhaustif.	<b>-Priorité haute</b> : une première version doit être développée au terme du deuxième

<sup>1</sup> <https://www.cnil.fr/>

de ses comptes (journal)			sprint
<b>FC1: Prendre compte des lois en vigueur</b>	-L'application doit être conforme aux lois sur les Données personnelles	-L'utilisation en temps réel doit venir <b>d'un consentement réel et justifiable</b> de l'utilisateur	<b>-Priorité haute</b>
<b>FC2: Utiliser un cryptage pour sauvegarder les identifiants</b>	-Crypter les identifiants, les paramètres	-Cryptage assez puissant pour être infaillible (ex: SHA-256)	<b>-Priorité moyenne</b> : Google Firebase propose déjà ces fonctionnalités donc nous allons juste les réutiliser
<b>FC3: Assurer une bonne ergonomie et fournir une documentation détaillée</b>	-Interface facile d'utilisation -Documentation	-L'utilisateur peut se débrouiller.	<b>-Priorité basse:</b> nous accorderons un créneau pour faire ces tâches
Maintenance			
<b>FP1: Permettre les sauvegardes distantes pour éviter la perte des données</b>	-Sauvegarder le journal et les données sur le cloud	-Remise en route rapide en cas de bug	<b>-Priorité moyenne</b>
<b>FC1: Suivre l'évolution des technologies (supports)</b>	-Faire une veille technologique	-Application à la pointe de la technologie en termes de sécurité et de fonctionnalités	<b>-Priorité moyenne</b>
<b>FC2: Suivre l'évolutions des lois</b>	-Adapter selon les lois	-À jour sur les lois en vigueur	<b>-Priorité haute</b>

## Backlog de produit

En tant que	je souhaite	afin que /de	Critères d'acceptation	Estimation arbitraire de la difficulté	Status	Pour le
Administrateur	visualiser le contenu d'une base de données, et le modifier	le bot puisse stocker des données	Les données et paramètres sont bien enregistrés dans la base de données	5	Fait	15/10/2019
Utilisateur	voir la liste de mes comptes enregistrés	je puisse en rajouter ou les supprimer	Les éléments modifiés doivent être pris en compte dans la base de données et affichés correctement après actualisation	8	Fait	30/10/2019
Utilisateur	pouvoir enregistrer ses propres comptes	je puisse décider quel réseau le bot doit gérer	Les informations sont bien entrés dans la/les base de données	5	Fait	25/10/2019
Utilisateur	définir des actions automatiques	je puisse gagner du temps / ne pas oublier	Je reçois des notifications	21 (à découper en sous tâches)	À faire	20/11/2019
Utilisateur	être en sécurité	préserver mes données personnelles	Les données des comptes sont cryptés et difficile à décoder	5	À faire	30/11/2019
Administrateur	devoir m'identifier	limiter l'accès aux pages de paramètres et de modification du comportement du bot	Un utilisateur non identifié ne doit pas pouvoir accéder aux pages autorisant une modification de données	5	Fait	30/11/2019
Utilisateur	recevoir des sms / notifications	je soit au courant des actions du bot	le délais de réception est court	8	À faire	10/12/2019
Administrateur	pouvoir personnaliser la charte graphique du site	optimiser l'ergonomie du site	templates html /css bien séparés par dossiers et sous-dossiers	8	Faire	10/12/2019
Administrateur / Utilisateur	avoir un fichier log ( journal ) à disposition	je soit au courant des actions du bot	tout y est écrit, avec la date et l'heure	13 (à découper en sous tâches)	À faire	10/11/2019

## Justification des choix techniques :

### Choix de la technologie :

Tout d'abord, deux choix se sont proposés à nous :

- Réaliser une application pour mobile (apk application native)
- Rendre notre application accessible via n'importe quel navigateur internet (sur PC, mobile, ...) en réalisant une application web.

Nous nous sommes rendus compte que la réalisation d'une application mobile était trop contraignante et complexe, nous avons donc choisi la deuxième solution.

Il est vrai qu'une application native est rapide tout en permettant d'utiliser des fonctionnalités avancées (caméra, notifications push, etc.). Cependant, une application web est plus pratique dans un projet de petite taille comme le nôtre. Le déploiement est plus rapide et plus facile. De plus, il n'y aura aucun problème de compatibilité avec les systèmes d'exploitation du mobile. Dans notre cas, nous avons choisi **d'utiliser Flask** qui est un web framework open-source de développement en Python. Son but principal est d'être léger, afin de garder la souplesse de la programmation Python, associé à un système de templates.

### Fonctionnalités de Flask :

- Contient un serveur de développement et un debugger
- Supporte les tests unitaires
- Utilise le moteur de template Jinja2
- Supporte les cookies sécurisés (session)
- Entièrement compatible avec WSGI 1.0
- Se base sur l'Unicode
- Dispose d'une documentation complète
- Compatible avec Google App Engine
- Il est possible de créer des extensions

## Pourquoi Firestore ?

Cloud Firestore est une base de données NoSQL construite pour des applications en général. Il est basé sur le type Documents. Son objectif est de fournir une expérience utilisateur exceptionnelle et de simplifier le développement d'applications avec la synchronisation en direct, la prise en charge hors ligne et les transactions ACID sur des centaines de documents et de collections

Pour la sécurité : la gestion des identités et l'authentification Firebase sont intégrées pour garantir la sécurité de votre application et de ses données;

Enfin, cette solution est l'une des plus rapides et efficaces du marché en termes de solutions cloud ( prend en charge la modification de 500 collections et documents en une seule transaction ).

## Fonctionnement du Bot

Principe de durée de vie :

Lorsqu'on va donner au bot une tâche à faire, une durée de vie ou il sera actif est calculée. Une boucle va aussi le maintenir en vie jusqu'à ce qu'il ait effectué cette action. Si on ajoute une deuxième action pour un temps plus lointain, cette durée de vie va être augmentée en conséquence.

## Choix des API

Nous avons deux possibilités pour le choix des API qui vont servir à effectuer automatiquement des actions et récupérer des données :

- Utiliser des API "primaires" : Sélénium pour les tests automatisés et BeautifulSoup pour le web Scraping, et créer les fonctionnalités nous même (se connecter, ...)
- Utiliser des API fournis par les réseaux sociaux : plus simple à utiliser, mais plus contraignants

Après étude des Twitter (API Twython et API Tweepy), nous avons choisi de principalement utiliser les API "primaires", car ceux-ci ne sont pas limités. En effet, les API Twitter ne nous permettaient pas d'effectuer des actions pour un compte autre que le compte développeur.

Nous avons ensuite d'utiliser d'autres API pour des fonctionnalités spécifique : l'API Gmail pour l'envoi et la réception de mail, et l'API pyFCM pour l'envoi de notifications.

## Perspectives

Nous pouvons dans le cadre de l'amélioration du projet (PT4) nous intéresser à la transformation de notre application en **application hybride**, encapsulée dans une application native sur Android. Nous pensons notamment à Ionic ou Cordova. Ces "bridges" nous permettront de raccorder l'appli à une grande majorité des fonctions du téléphone tout en gardant la facilité de développement en HTML5/CSS.