

**CAHIER DE CHARGES**  
**CONCEPTION ET REALISATION D'UN**  
**AGREGATEUR DE FLUX D'INFORMATION**  
**MIAGE 2020-2021**

## SUIVI ET REFERENCES

DOCUMENT REDIGE PAR :			
NOM & PRENOM	MATRICULE	ADRESSE MAIL	NUMERO
ABOUBAKAR OUSMANE		<a href="mailto:aboousmane@gmail.com">aboousmane@gmail.com</a>	699 04 44 27
ATONFACK FOKEU			
BASSON NDJIB			
CHRISTIAN FOMEKONG		<a href="mailto:christian_fomekong@yahoo.fr">christian_fomekong@yahoo.fr</a>	690 89 34 71
TSOUMOU HEVY		<a href="mailto:hevyfrancelly@gmail.com">hevyfrancelly@gmail.com</a>	655 53 70 14
TESSE WAMBA		<a href="mailto:wambacynthie@gmail.com">wambacynthie@gmail.com</a>	690 46 95 51

LISTE DE DIFFUSION		
NOM & PRENOM	ADRESSE MAIL	NUMERO
Dr DJIKEN		
ETUDIANT DE MIAGE 1		

REFERENCES		
TITRES DU DOCUMENT	DESCRIPTION	VERSION
Cahiers des charges	Présentation global du projet et spécifications fonctionnelles et techniques	1.0.0

SUIVI DES VERSIONS DU DOCUMENT			
VERSION	DATE	STATUT	NATURE DE LA MODIFICATION
1.0.0	23/01/2021	CREATION	Aucune

## SOMMAIRE

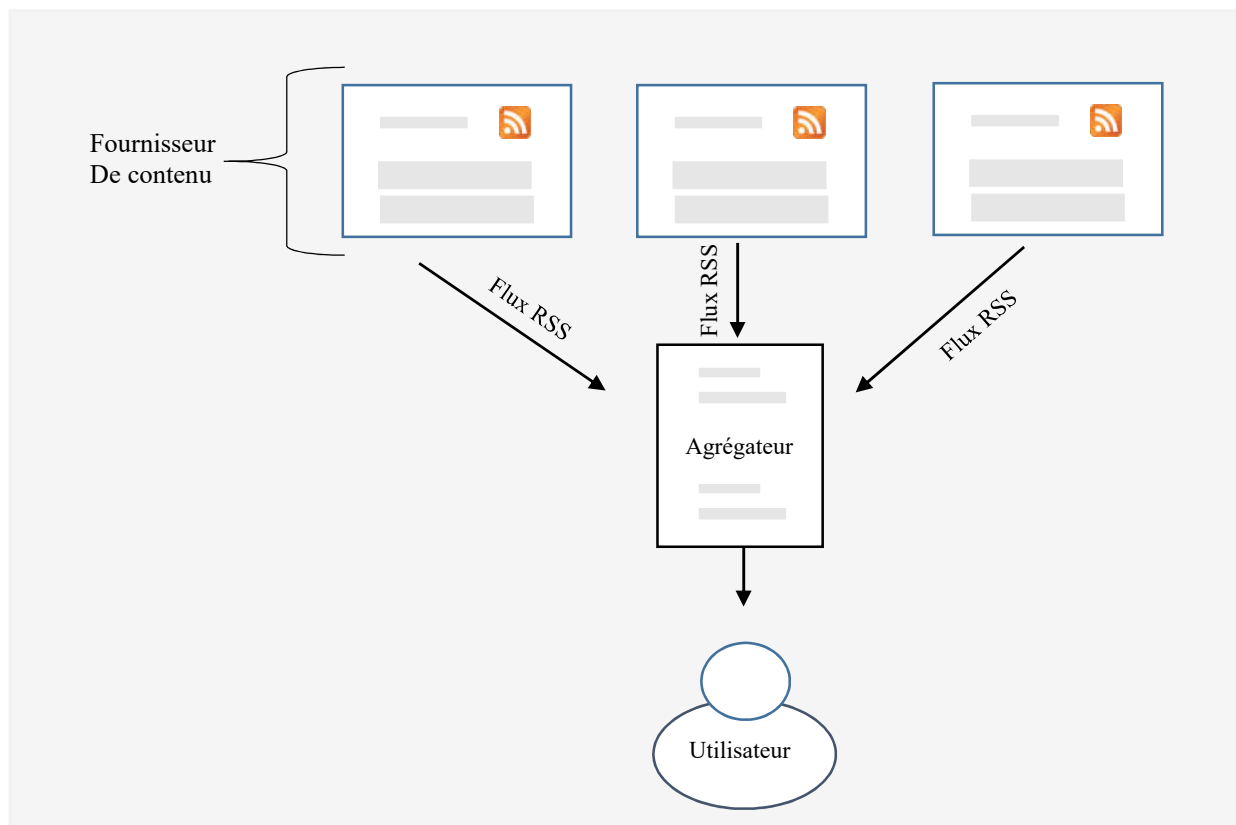
SUIVI ET REFERENCES .....	2
<b>I- PRESENTATION DU PROJET .....</b>	<b>4</b>
1- Présentation de la technologie RSS 2.0.....	4
2- Quelques agrégateurs: .....	6
3- L'apport de notre application par rapport à d'autres.....	7
4- Présentation des règles de gestion.....	7
<b>II- ANALYSE DU PROJET .....</b>	<b>7</b>
1- Méthode d'analyse.....	7
2- Spécification des besoins fonctionnels .....	8
3- Spécification des besoins non fonctionnels.....	8
3-1- Exigences ergonomiques .....	8
3-2- Sécurité.....	8
<b>III- MODELISATION .....</b>	<b>9</b>
1- Diagramme de cas d'utilisation.....	9
2- Diagramme de classe .....	10
3- Diagramme de séquence .....	11
<b>IV- TECHNOLOGIES UTILISEES .....</b>	<b>11</b>

## I- PRESENTATION DU PROJET

### 1- Présentation de la technologie RSS 2.0

Le réseau internet est une mine d'informations, au quotidien les internautes consultent plusieurs sites d'information aux thématiques aussi diverses que variées. L'habitude la plus courante est d'épingler le site en favoris si celui-ci offre des articles intéressants mais, au fil du temps, revisiter ces signets est devenue une tâche fastidieuse pour les utilisateurs d'autant plus que le nombre de signets grandit. Une meilleure technique pour les utilisateurs est de recevoir automatiquement des informations sur les mises à jour de leurs sites Web favoris.

De nombreux sites Web ont reconnu ce problème et ont commencé à fournir des informations dans un format commun, le format RSS (Really Simple Syndication), le format RSS apparu en 2002 est une famille de formats de données utilisé pour la syndication de contenu Web. RSS permet aux sites Web de distribuer leur contenu en dehors d'un navigateur Web ; Et s'avère être un moyen simple de tenir l'internaute à l'affût des mises à jour automatiquement sur les sites Web qu'il visite régulièrement. Il s'agit pour les sites web de mettre à disposition des internautes un format de données standardisés généralement en XML et qui permettra à ces derniers au travers d'un logiciel appelé agrégateur de recevoir les informations de ses sites préférés ; l'internaute consulte des sites web et identifie le site offrant les informations en RSS puis les ajoute à son agrégateur et il n'a plus besoin prochainement de revenir sur le site car les informations sont centralisées autour de l'agrégateur qui se chargera de la mise à jour constante des informations par rapport aux nouveautés. En gros l'internaute ne va pas plus à la recherche de l'information c'est plutôt celle-ci qui vient vers lui. Schématiquement on aura :



Le principe de l'agrégateur de contenu est de vous épargner du temps. Vous n'avez plus besoin de vous connecter à chaque site à surveiller, car il le fait pour vous, rapatrie les nouveaux contenus et les présente dans un Dashboard. Il ne vous reste plus qu'à lancer l'application le matin ou durant votre pause, et à lire les contenus qui vous intéressent.

Le format simple d'un flux RSS 2.0 est le suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>Mon site</title>
    <description>Ceci est un exemple de flux RSS 2.0</description>
    <lastBuildDate>Sat, 07 Sep 2002 00:00:01 GMT</lastBuildDate>
    <link>http://www.example.org</link>
    <item>
      <title>Actualité N°1</title>
      <description>Ceci est ma première actualité</description>
      <pubDate>Sat, 07 Sep 2002 00:00:01 GMT</pubDate>
      <link>http://www.example.org/actu1</link>
    </item>
    <item>
      <title>Actualité N°2</title>
      <description>Ceci est ma seconde actualité</description>
      <pubDate>Sat, 07 Sep 2002 00:00:01 GMT</pubDate>
      <link>http://www.example.org/actu2</link>
    </item>
  </channel>
</rss>
```

Le contenu d'un document RSS est encadré par deux balises <rss> et </rss> dont l'attribut version (obligatoire) spécifie la version du standard à laquelle il se conforme.

Au niveau suivant, une balise <channel> unique contient pour chaque article un groupe de métadonnées qui le décrivent et son contenu proprement dit.

### A- Les métadonnées

Trois métadonnées sont obligatoires dans un channe (canal ):

- <title> : Définit le titre du flux ;
- <description> : Décrit succinctement le flux ;
- <link> : Définit l'URL du site correspondant au flux.

D'autres sont optionnelles :

- `<pubDate>` : Définit la date de publication du flux ;
- `<lastBuildDate>` : Définit la dernière date de modification du flux ; (RFC 822)
- `<image>` : Permet d'insérer une image dans le flux ;
- `<language>` : Définit la langue du flux ;
- `<enclosure>` : Permet d'insérer un média (vidéo, etc.) dans le flux.

## **B- Contenu : Description de chaque article**

À chaque article correspond une balise `<item>`.

Dans cette balise se trouvent les données correspondantes à l'actualité sous forme de balise. Les balises les plus courantes sont :

- `<title>` : Définit le titre de l'actualité ;
- `<link>` : Définit l'URL du flux correspondant à l'actualité ;
- `<pubDate>` : Définit la date de l'actualité (RFC 822) ;
- `<description>` : Définit une description succincte de l'actualité ;
- `<guid>` : Définit de manière unique l'actualité.

Selon la DTD RSS 2.0, il doit y avoir au moins un `<title>` ou une `<description>` dans un `item` et le reste des balises est optionnel.

Il existe encore d'autres balises comme :

- `<author>` : Définit l'adresse électronique (mail) de l'auteur ;
- `<category>` : Associe l'item à une catégorie ;
- `<comments>` : Définit l'URL d'une page de commentaire en rapport avec l'item.

Il peut également arriver que l'on trouve d'autres balises dans un flux RSS, grâce aux espaces de noms (namespaces). On peut ainsi y inclure de nouvelles fonctions comme cela a été fait pour iTunes par exemple. Voir concernant ce sujet la spécification RSS 2.

## **2- Quelques agrégateurs:**

### **- Old Reader**

C'est un agrégateur de flux RSS qui permet de centraliser la veille. Très proche de Google Reader, il fut une des meilleures alternatives lorsque ce service Google a fermé. Vous pouvez exporter des flux pour les lire dans une seule interface.

### **- NewsBlur**

C'est un agrégateur de flux RSS simple et facile à utiliser. Un peu moins pratique que d'autres, il faut prendre le temps de s'y intéresser pour vraiment exploiter toutes ses capacités. Les sources peuvent être ajoutées manuellement pour enrichir votre veille ou par le biais d'un fichier OPML. Le menu latéral permet une gestion facile des dossiers (qu'on peut créer d'un simple clic) classés par thèmes.

- **Inoreader**

Comme tous les outils que nous venons de voir, Inoreader vous permet de lister les sites à surveiller. Son objectif est de vous faire gagner un maximum de temps en vous proposant de classer automatiquement les meilleurs contenus pour vous. Vous définissez des mots-clés et l'outil vous affiche les articles contenant les termes prédéfinis en premier.

- **Etc ...**

**3- L'apport de notre application par rapport à d'autres**

Le facteur social est devenu très important dans le développement et la pérennité des applications. Avec l'avènement des réseaux sociaux, les internautes aiment explorer, découvrir et partager du contenu. Dans le prolongement de ceci, nous avons intégré le facteur social dans notre application afin que l'utilisateur ne puisse perdre ses habitudes sociales si importantes, notre application permettra non seulement de souscrire à des flux d'informations les consulter mais aussi de constituer une communauté d'amis afin de les partager.

**4- Présentation des règles de gestion**

- si on souscrit à un flux pour la première fois le système télécharge les 10 publications du flux
- chaque fois que l'on consulte un flux le système télécharge les nouvelles publications de ce flux
- l'ensemble des publications partagées par un utilisateur forment un nouveau flux appelé « flux de l'utilisateur » ayant partagé ces publications.
- Un autre utilisateur peut souscrire à ce nouveau « flux de l'utilisateur » tout comme pour un autre flux
- La demande d'amitié se fait en rentrant l'email de l'ami (Y) par l'utilisateur (X) et l'ami (Y) recevra une notification d'amitié qu'il peut accepter ou refuser
- Si (X) est ami avec (Y) alors, (Y) est aussi ami avec (X)
- Un utilisateur ne peut faire qu'un seul commentaire pour une seule publication partagée par son ami
- Les commentaires ne sont visibles que par les amis de la personne ayant partagé la publication.

**II- ANALYSE DU PROJET**

**1- Méthode d'analyse**

UML (Unified Modeling Language), est un langage de modélisation des systèmes standard, qui utilise des diagrammes pour représenter chaque aspect d'un système : statique et dynamique, en s'appuyant sur la notion d'objet qui est un véritable atout pour ce langage. Les "méthodologues" disent qu'une méthode, pour être opérationnelle, doit avoir les composantes suivantes :

- Une démarche (les étapes, phases et tâches de mise en œuvre)

- Des formalismes (les modélisations et les techniques de transformation)
- Une organisation et des moyens de mise en œuvre

UML se positionne exclusivement comme un ensemble de formalismes ou il faut y associer une démarche et une organisation pour constituer une méthode. UML est un langage de modélisation et non une méthode. UML propose une boîte à outils (à dessins) pour faire de la modélisation de tout type de système d'information. (On peut utiliser un diagramme d'activité pour modéliser un processus métier qui ne sera pas forcément informatisé).

## **2- Spécification des besoins fonctionnels**

Les besoins fonctionnels ou besoins métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Selon les règles de gestion de notre futur système, nous avons identifié les besoins fonctionnels suivant :

- S'inscrire
- Enregistrer les flux d'information
- Consulter ses flux d'information ou celui de ses amis
- Consulter les publications d'un flux
- Lire les publications
- Voir plus sur la publication (il sera redirigé vers le site)
- Faire une demande d'amitié
- Consulter mes demandes d'amitié
- Accepter ou rejeter une demande
- Partager les publications de ses flux et commenter celle de ses amis
- Commenter une publication

## **3- Spécification des besoins non fonctionnels**

Les besoins non fonctionnels représentent les exigences internes au système et cachées aux utilisateurs :

### **3-1- Exigences ergonomiques**

- Une interface conviviale, lisible et facile à utiliser.
- Rapidité du temps de réponse car il joue un rôle fondamental. Il doit être court.
- L'application devra être toujours disponible et ouvert à tous sans restriction

### **3-2- Sécurité**

- Traçabilité des mouvements et des opérations effectuées
- Le système doit être sécurisé avec l'obligation pour chaque utilisateur de saisir un mot de passe et un login.
- L'authentification : Elle est assurée par un module d'authentification et d'autorisation, il gère l'accès entre les utilisateurs et l'application par un login et un mot de passe.

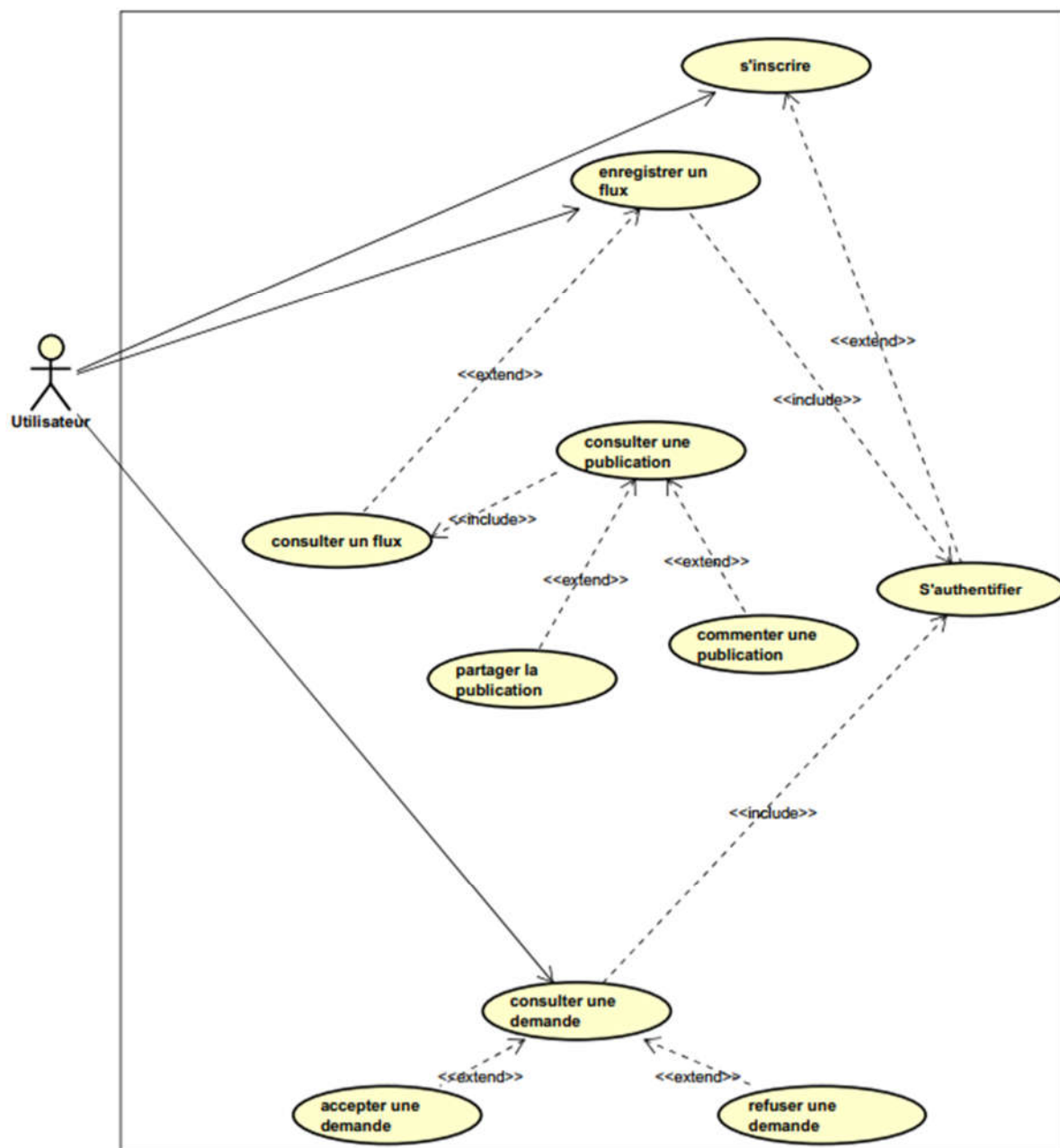


### III- MODELISATION

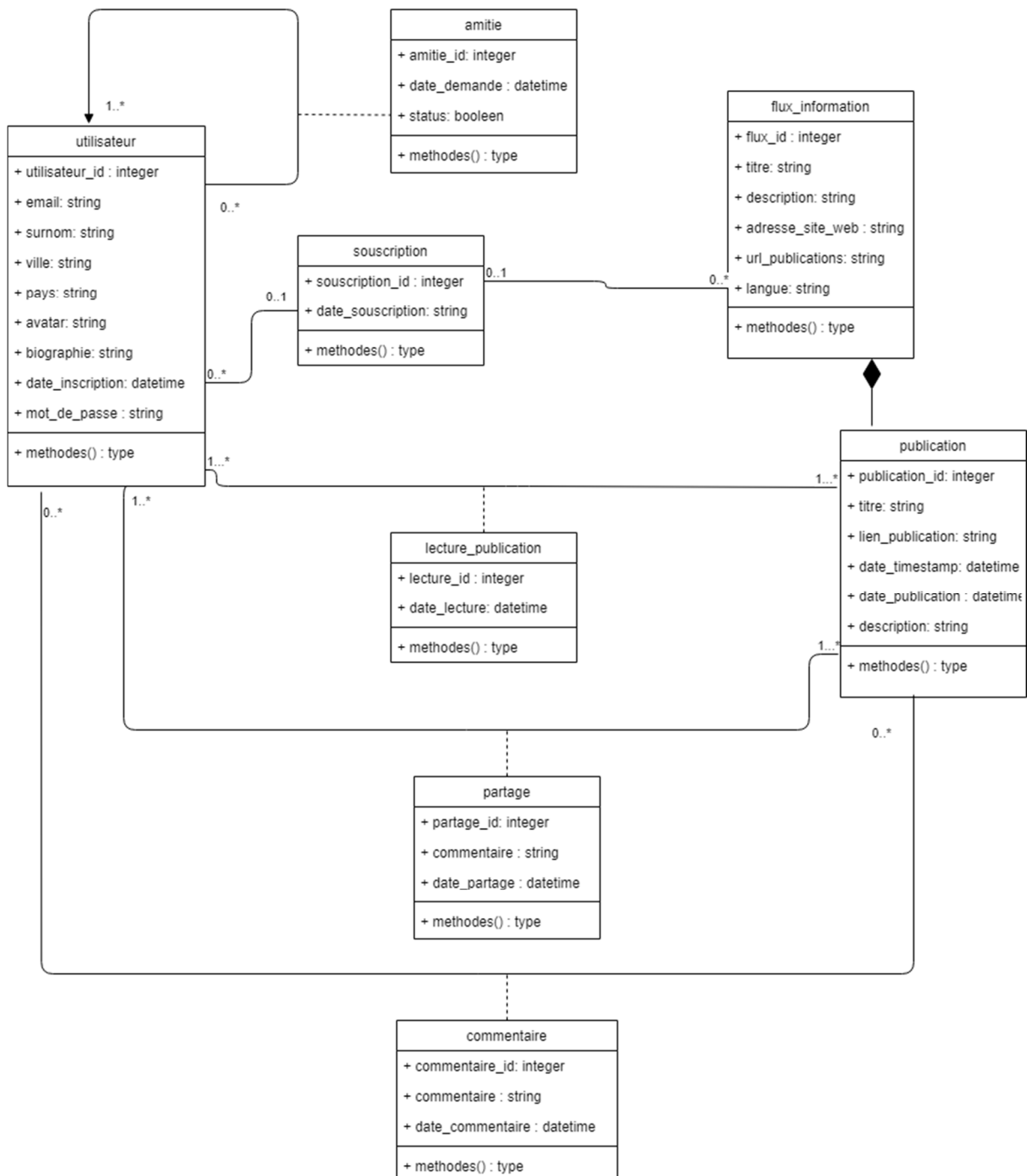
#### 1- Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique. En d'autres termes un cas d'utilisation modélise un service rendu par le système sans imposer le mode de réalisation.

Partant de ce bref aperçu, les cas d'utilisation de notre futur système se rapportent donc aux besoins fonctionnels ci-dessus énumérés (voir II-2) et représentés comme suit.

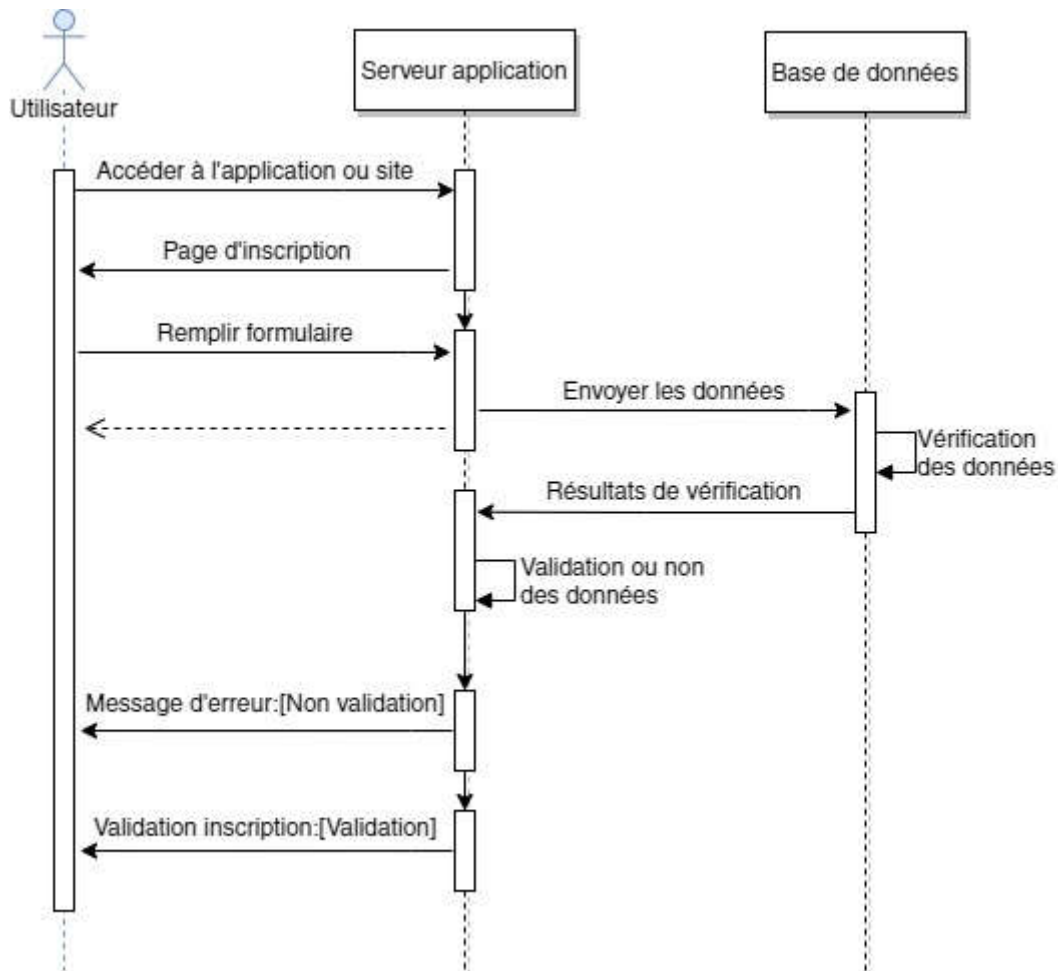


## 2- Diagramme de classe



### 3- Diagramme de séquence

#### Cas d'inscription



### IV- TECHNOLOGIES UTILISEES

#### a. Flask

Flask est un micro Framework open-source de développement web en Python. Il est classé comme micro Framework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités. Il est distribué sous licence BSD4.

Flask se base sur deux modules `werkzeug` et `jinja` pour proposer plusieurs des fonctionnalités suivantes :

- Serveur de développement et debugger
- Simplifie l'écriture de tests unitaires
- Moteur de Template pour le rendu HTML (Jinja)
- Supporte les cookies sécurisés (session)

- Entièrement compatible avec WSGI 1.0
- Se base sur l'Unicode
- Documentation complète
- Déploiement aisé sur plusieurs hébergeurs
- Ajout de fonctionnalités via les extensions (**WTForms** pour la validation des formulaires, **SQLAlchemy** qui peut faire office d'ORM, **Flask-Migrate** pour les migrations de la base de données, **FLASK-login** pour la gestion de l'authentification etc...)

## **b. MySQL**

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multi-utilisateur. C'est un logiciel libre, open source, développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire.

## **c. Le HTML et CSS et Javascript avec le Framework bootstrap css**

Bootstrap est un ensemble de fichiers CSS et JavaScript fonctionnant ensemble et qu'on va pouvoir utiliser pour créer des design complexes de manière relativement simple.

Le framework Bootstrap est donc un ensemble de fichiers CSS et JavaScript qui contiennent des règles prédéfinies et qui définissent des composants. Ces ensembles de règles sont enfermés dans des classes et nous n'aurons donc qu'à utiliser les classes qui nous intéressent afin d'appliquer un ensemble de styles à tel ou tel élément HTML.

Quelques atouts du framework Bootstrap Css:

- ✓ Un gain de temps de développement qui peut être conséquent ;
- ✓ Une certaine robustesse dans l'architecture globale du code ;
- ✓ Un framework appartenant à une grande société (Twitter).