**TABLE DES MATIERES**

[Introduction 1](#_Toc67174740)

[Chapitre 1 : PRESENTATION de la 3w ACADEMY 2](#_Toc67174741)

[I. Qui somme-nous : 3](#_Toc67174742)

[II. Nos programmes 3](#_Toc67174743)

[III. Notre equipe 3](#_Toc67174744)

[Chapitre 2 : CONCEPTS théoriques 4](#_Toc67174745)

[I. Micro service 5](#_Toc67174746)

[II. RTMP 5](#_Toc67174747)

[III. Hls 5](#_Toc67174748)

[IV. Transmuxing/Transcoding 5](#_Toc67174749)

[Chapitre 3 : ANALYSE ET SPECIFICATION des besoins 6](#_Toc67174750)

[I. Capture des besoins globaux 7](#_Toc67174751)

[1.1. Définition des acteurs 7](#_Toc67174752)

[1.2. Analyse des besoins globaux 7](#_Toc67174753)

[II. Spécification des besoins fonctionnels globaux 11](#_Toc67174754)

[III. Division en micro services 11](#_Toc67174755)

[IV. L’architecture de l’application 12](#_Toc67174756)

[Conclusion 12](#_Toc67174757)

[Chapitre 4: SPRINTS 13](#_Toc67174758)

[I. Sprint 1 : 14](#_Toc67174759)

[1.1. Sprint Backlog 14](#_Toc67174760)

[1.2. Diagrammes de cas d’utilisation Global 15](#_Toc67174761)

[1.3. Diagramme D’ACTIVITE GLOBAL 15](#_Toc67174762)

[1.4. Conception 15](#_Toc67174763)

[1.5. Réalisation 16](#_Toc67174764)

[1.6. Scalabilité 17](#_Toc67174765)

[1.7. Architecture Finale 17](#_Toc67174766)

[II. Sprint 2 18](#_Toc67174767)

[2.1. Sprint Backlog 18](#_Toc67174768)

[2.2. System de chat 18](#_Toc67174769)

[2.3. Système Vidéo 19](#_Toc67174770)

[2.4. Intégration 19](#_Toc67174771)

# Introduction

# 

# Chapitre 1 : PRESENTATION de la 3w ACADEMY

## Qui somme-nous :

La 3W Academy, l’école de référence dans les métiers du digital, a ouvert ses portes au Maroc en 2018. Labellisée “Grande École du Numérique”, l’école délivre des certifications reconnue par l’Etat Français.

## Nos programmes

A ameliorer

Notre formation phare est le Bootcamp, une formation innovante, courte et intensive, qui s’adresse aux étudiants, salariés et chercheurs d’emploi souhaitant se transformer en Développeurs WEB en seulement 4 mois.

## Notre equipe

La 3W Academy a choisi parmis c’est laureat les meilleures candidats pour prendre part a cette competition , notemment :

* ---
* ---
* ---
* ---

# Chapitre 2 : CONCEPTS théoriques

Afin d’atteindre les objectifs du projet et répondre aux différents besoin établis, l’étude des concepts théoriques qui lui sont relatifs et des différents moyens mis en œuvre est une étape primordiale.

## Micro service

## RTMP

## Hls

## Transmuxing/Transcoding

# Chapitre 3 : ANALYSE ET SPECIFICATION des besoins

Le présent chapitre représente le sprint de démarrage au cours duquel sera spécifié les exigences des différents utilisateurs. Une étude des besoins globaux de ces acteurs est alors nécessaire. Ensuite, sera exposé le backlog du produit ainsi que les besoins non fonctionnels, pour finir avec proposée.

## Capture des besoins globaux

L’étape de l’analyse des besoins est très importante puisque la réussite de toute application dépend de la qualité de son étude. Il faut donc bien déterminer les fonctionnalités attendues du système pour les différents acteurs.

### Définition des acteurs

En plus de l’administrateur dont la présence permanente durant toutes les phases du projet garantie la réussite et l’immensité de l’application, trois acteurs principaux sont identifiés :

* Utilisateur public
* Streamer
* Viewers

### Analyse des besoins globaux

Les besoins sont divisés en deux catégories, à savoir les besoins fonctionnels et les besoins non fonctionnels.

#### Besoins fonctionnels

Ce sont les actions et les réactions que le système doit accomplir suite à une demande d’un acteur principal. Tenant compte de la nature de l’application, on distingue les besoins par acteurs :

**Administrateur :**

* Authentification
* Choisir la langue du site Web
* Gestion des utilisateurs ;
* Gestion des Rôles ;
* Étudier les interactions des utilisateurs
* Gestion des Tournois
* Gestion des articles du blog
* Monitoring de la bonne marche du system
* Accompagner les utilisateurs et améliorer son expérience.

**Streamer/ Viewers :**

* Authentification ;
* Choix de la langue de l’application
* Gestion du profile
* Gestion de la chaine
* S’abonner à une chaine
* Communiquer avec un autre utilisateur
* Être notifier des derniers messages reçus
* Être notifier des promotions
* Rechercher un utilisateur
* Renseigner son calendrier de Stream
* Streamer
* Afficher la liste d’amis
* Afficher les demandes d’amis
* Afficher les chaines auxquelles il est abonné
* Naviguez sur les lives par catégories
* Visualiser un live
* Chatter sur le Live
* Afficher tous les lives selon des critères
* Participer à un tournoi
* Afficher les Tournois en cours
* Donner et recevoir des donations

**Utilisateur publique :**

* Choix de la langue de l’application
* Être notifier des promotions
* Rechercher un utilisateur
* Visualiser un live
* Afficher tous les lives selon des critères
* Afficher les Tournois en cours

#### Les besoins non fonctionnels

**L’extensibilité** : L’architecture de l’application permettra l’évolution et la maintenance (ajout ou suppression ou mise à jour) au niveau de ses différents modules d’une manière flexible.

**La scalabilité** : l’application devra gérer un flux énorme de données en temps réel (Vidéo-Chat-Interactions de l’utilisateur, données de l’utilisateur).

**La sécurité** : L’accès aux informations protégées n’est possible qu’après vérification des privilèges et des droits d’accès. Ainsi tout utilisateur passera par une phase d’authentification pour pouvoir consulter les services offerts par l’application.

**L’ergonomie et la convivialité** : L’application fournira une interface conviviale et simple à utiliser et qui ne requiert aucun prérequis, elle pourra être exploitable par tous les types d’utilisateurs.

#### Backlog de produit

Après avoir cité les besoins de l’application, cette partie décrira le backlog du produit, qui représente une liste de taches exprimées sous forme de besoins.

|  |  |  |
| --- | --- | --- |
| Id | User stories | Priorité |
| 1 | Autant qu’administrateur je dois pouvoir m’authentifier sur une plateforme d’administration. | Moyenne |
|  | Autant qu’administrateur je dois pouvoir gérer les utilisateurs de la plateforme. |  |
|  | Autant qu’administrateur je dois pouvoir gérer les rôles des utilisateurs |  |
|  | Autant qu’administrateur je dois pouvoir avoir les outils nécessaires pour étudier les interactions des utilisateurs. |  |
|  | Autant qu’administrateur je dois pouvoir gérer les tournois. |  |
|  | Autant qu’administrateur je dois pouvoir gérer le blog de la plateforme. |  |
|  | Autant qu’administrateur je dois pouvoir avoir les outils nécessaires pour surveiller le système. |  |
|  | Autant que streamer/viewer je dois pouvoir m’authentifier. |  |
|  | Autant que streamer/viewer je dois pouvoir gérer mon profil. |  |
|  | Autant que streamer/viewer je dois pouvoir gérer ma chaine. |  |
|  | Autant que streamer/viewer je dois pouvoir m’abonner à une chaine. |  |
|  | Autant que streamer/viewer je dois pouvoir communiquer avec un autre utilisateur. |  |
|  | Autant que streamer/viewer je dois pouvoir etre notifier de chaque message que je reçois. |  |
|  | Autant que streamer/viewer je dois pouvoir chercher un utilisateur ou une chaine. |  |
|  | Autant que streamer/viewer je dois renseigner mon calendrier d’évènements. |  |
|  | Autant que streamer/viewer je dois pouvoir streamer. |  |
|  | Autant que streamer/viewer je dois pouvoir afficher ma liste d’amis. |  |
|  | Autant que streamer/viewer je dois pouvoir gérer les demandes d’amis. |  |
|  | Autant que streamer/viewer je dois pouvoir gérer mes abonnements. |  |
|  | Autant que streamer/viewer je dois pouvoir naviguer sur les lives. |  |
|  | Autant que streamer/viewer je dois pouvoir visualiser un live. |  |
|  | Autant que streamer/viewer je dois pouvoir chatter sur un live. |  |
|  | Autant que streamer/viewer je dois pouvoir participer à un tournoi. |  |
|  | Autant que streamer/viewer je dois pouvoir naviguez sur les tournois. |  |
|  | Autant que streamer/viewer je dois pouvoir donner et recevoir des donations. |  |
|  | Autant que streamer/viewer je dois pouvoir navig donner et recevoir des donations. |  |
|  | Autant qu’utilisateur publique je dois pouvoir Rechercher un utilisateur |  |
|  | Autant qu’utilisateur publique je dois pouvoir visualiser un live. |  |
|  | Autant qu’utilisateur publique je dois pouvoir naviguer sur lives. |  |
|  | Autant qu’utilisateur publique je dois pouvoir naviguer sur les tournois. |  |

**Tableau 1: BackLog du Produit**

## Spécification des besoins fonctionnels globaux

La plateforme offre diverses fonctionnalités. Le diagramme de cas d’utilisation global représenté par la figure ci-dessous, modélise les fonctionnalités des modules cadrées par le projet. Les chapitres suivants détaillent les cas d’utilisation réalisés.

**Figure 5: Use Case Global**

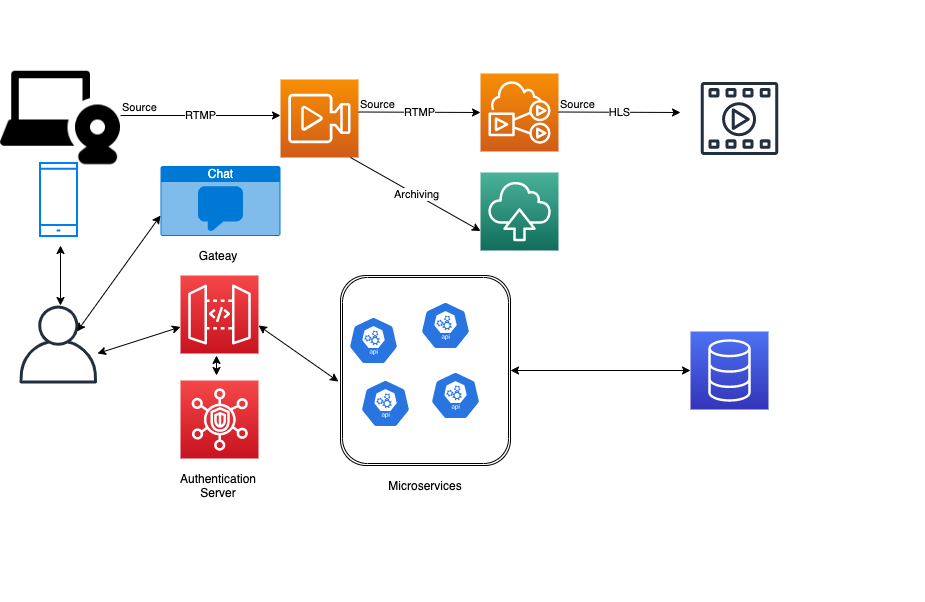
## Division en micro services

En se basant sur le Backlog du produit et sur la décomposition de notre projet en micro services, nous divisons Release en 4 sprints comme le montre le tableau ci-dessous.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sprint 1 : Plateforme Management | Sprint 2 : Chat System /Video Streaming | Sprint 4 : Tournement Management |
| User Story ID |  |  |  |

**Tableau 2: Répartition du backlog en Sprints**

## L’architecture de l’application

****

**Figure 7: Architecture de l’Application**

## Conclusion

Tout au long de ce chapitre les besoins spécifiés ont permis d’avoir une vision claire et profonde sur le projet. Aussi la définition du backlog produit et la répartition en micro services ont cadrés la planification des sprints. Et ce, suivant l’architecture globale mise en place. Le chapitre suivant abordera les différents sprints un par un.

# Chapitre 4: SPRINTS

## Sprint 1 :

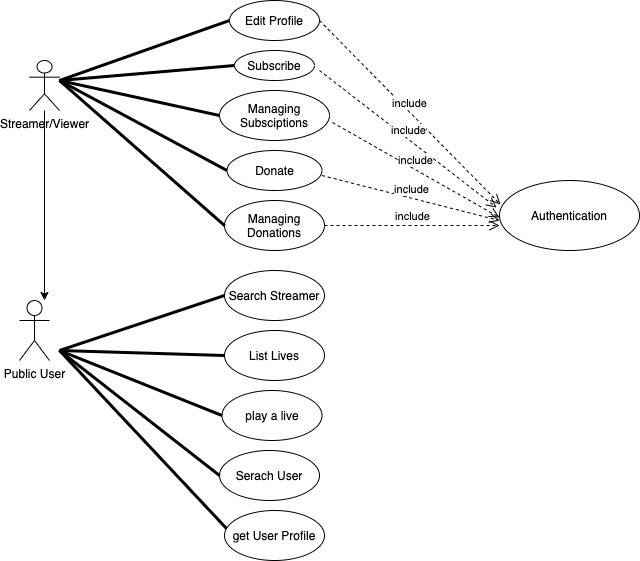
Le premier sprint devra permettre aux utilisateurs de la plateforme de naviguer sur la plateforme, chercher un streamer par son identité, chercher des Stream et gérer leurs abonnements.

### Sprint Backlog

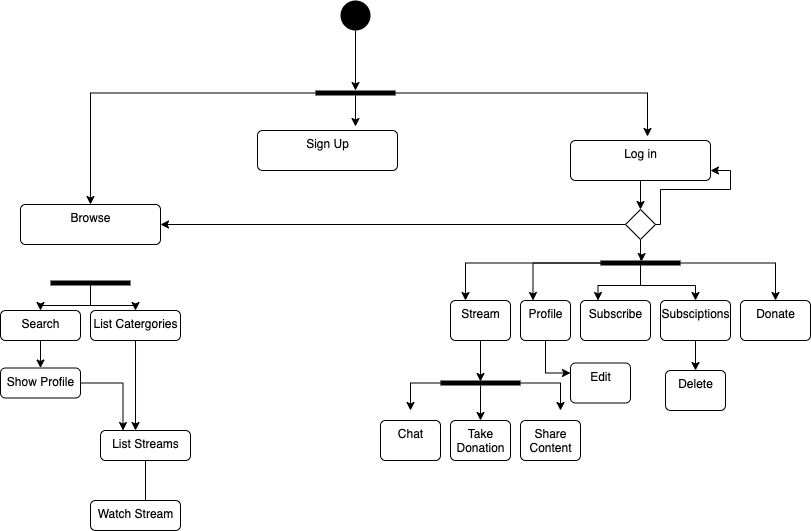
|  |  |
| --- | --- |
| **Exigence** | **Sous taches** |
| **1.1** | Authentification |
| **1.2** | Gestion du profile Profil |
| **1.3** | Gestion des abonnements |
| **1.4** | Gestion des Stream |
| **1.5** | Gestion des revenus |

**Tableau 3 : Sprint 1 Backlog**

### Diagrammes de cas d’utilisation Global



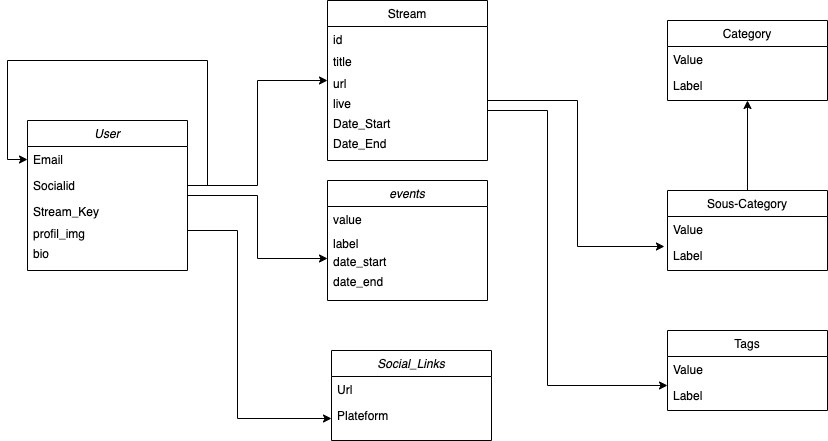
### Diagramme D’ACTIVITE GLOBAL



### Conception

Dans cette section, la phase de conception du sprint qui se basera sur le diagramme de classe et la décomposition en micro services

#### Diagramme de classes



**Figure 14: Diagramme de Classe**

### Réalisation

#### Back end

Dans un but permanent de gestion d’un nombre important d’utilisateurs et d’un flux énorme de données le choix s’est porté sur une architecture micro service. Ce qui permettra non seulement un travail en équipe harmonieux mais aussi une décentralisation des aspects fonctionnel de la plateforme, et partant, sera délivré un produit maintenable et extensible.

Ainsi, en plus du reverse proxy, du service d’enregistrement et le service de configurations l’équipe développera les micro services suivants :

* Micro-service d’authentification ;
* Micro service de gestion des utilisateurs et des abonnements ;
* Micro service de gestion des Streams ;
* Micro service de gestion des évènements ;
* Micro service de gestion des catégories.

#### Base de données

Chaque micro service aura sa propre base de données de type NoSql, dans une vision de pouvoir introduire facilement une architecture Big Data avec l’évolution du projet.

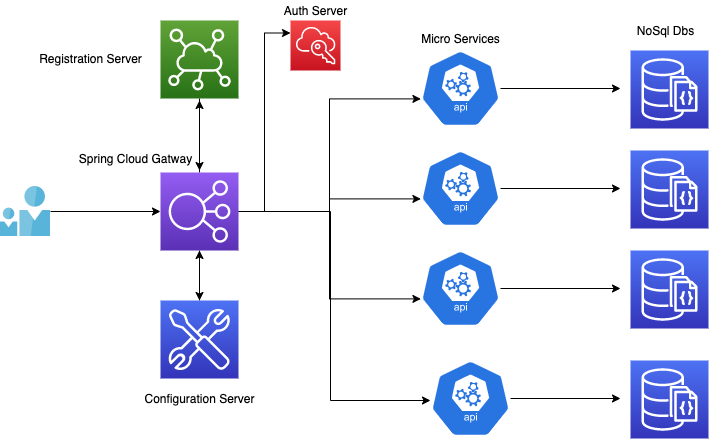
#### Front end

L’équipe sera appelée à développer deux applications web et mobile a base des technologie React et React Native qui communiquent envia des requêtes http avec les micro services en passe le reverse proxy.

### Scalabilité

En plus de la **Scalabilité** horizontale et la possibilité de gérer un grand flux de requêtes acquise par l’architecture micro services adoptés, l’accent sera mis sur les performances des serveurs utilises en vue d’assurer une **Scalabilité** verticale.

### Architecture Finale



## Sprint 2

Le deuxième sprint consiste à réaliser les parties maîtresses de la plateforme notamment le system de streaming et le system de chat.

### Sprint Backlog

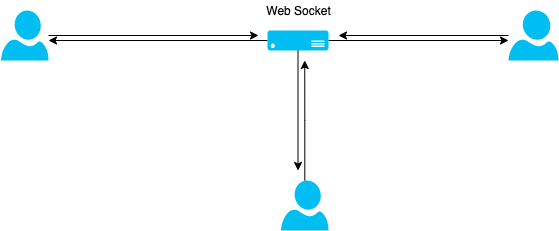
|  |  |
| --- | --- |
| **Exigence** | **Sous taches** |
| **2.1** | System Chat |
| **2.2** | System de streaming |
| **2.3** | Intégration |

**Tableau 4 : Sprint 1 Backlog**

### System de chat

Faisant parti de la catégorie des Real Time Application, la complexité d’un system de chat augmente en fonction des nombres d’utilisateurs. En plus sur une plateforme de streaming les besoins du chat sont ponctuels et dépendent constamment de la période du live, sans oublier que la sauvegarde et l’historique ne sont pas d’une telle importance comme le cas d’une application de messagerie. Le produit devra finalement offrir une meilleure latence et une meilleure **Scalabilité**.

#### Architecture Basique



#### Architecture Finale



### Système Vidéo

Le live streaming vidéo est un système complexe et lourd vu la quantité de données partagée sur internet, il faudra alors assurer et une qualité adapte à la bande passante et une latence optimale.

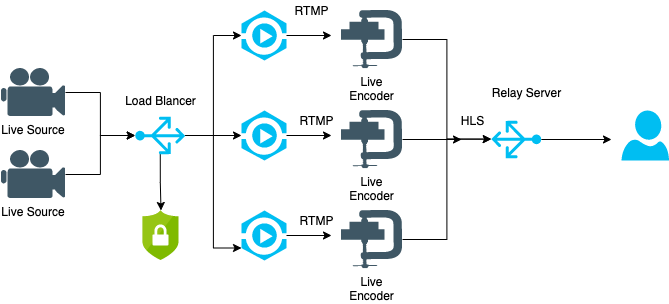
#### Architecture Basique



Media Server

#### Architecture Finale

Media Servers



### Intégration

Cette phase, consistera principalement à intégrer le système de chat et live streaming a la plateforme déjà mise en place.