

2^e Année de Licence Informatique
Université Paris 8
Institut d'Enseignement à Distance

Réalisation de Programme

Projet : MoodRSS

Cahier des Charges & Spécifications

Le 03 avril 2024

Table des matières

1	Cahier des Charges	2
1.1	Présentation du Projet	2
1.1.1	Objectifs	2
1.2	Fonctionnalités	2
1.2.1	Recherche de Flux RSS	2
1.2.2	Analyse des Sentiments	2
1.2.3	Tableau de Bord	2
1.3	Exigences Générales	2
1.3.1	Accessibilité	2
1.3.2	Performance	3
1.4	Documentations	3
1.5	Phases du Projet	3
1.5.1	Les spécifications	3
1.5.2	Conception	3
1.5.3	Implémentation	3
1.5.4	Test et Validation	3
1.5.5	Déploiement	3
2	Spécifications	4
2.1	Architecture de l'Application	4
2.1.1	Composants Principaux	4
2.1.2	Technologies Utilisées	4
2.2	Fonctionnalités	4
2.2.1	Recherche de Flux RSS	4
2.2.2	Analyse des Sentiments	4
2.2.3	Tableau de Bord	5
2.3	Sécurité	5
2.4	Hébergement	5
2.4.1	Type de Serveur	5
2.4.2	Mise en Place de DDNS	5
2.5	Tests & distinction développement / production	6
2.5.1	Tests Unitaires	6
2.5.2	Tests Fonctionnels	6
2.5.3	Distinction développement / production	6
2.6	Versioning	6
2.7	Documentations	6

1 Cahier des Charges

MoodRSS est une application web destinée à analyser les flux RSS pour déterminer le sentiment des articles qu'ils contiennent et les afficher sur un tableau de bord interactif permettant de filtrer les articles par sentiment (positif, négatif ou neutre).

1.1 Présentation du Projet

L'application MoodRSS doit être accessible via un navigateur web et offrir une interface utilisateur intuitive pour rechercher, analyser et afficher les flux RSS. L'analyse des sentiments sera effectuée en temps réel pour tous les articles dès l'intégration du flux.

1.1.1 Objectifs

- Permettre aux utilisateurs d'intégrer des flux RSS.
- Analyser les sentiments des articles de manière automatisée.
- Afficher les articles avec un lien vers chacun d'eux.
- Offrir une fonctionnalités de filtrage des articles par sentiment.

1.2 Fonctionnalités

1.2.1 Recherche de Flux RSS

- **Entrée Utilisateur** : Champ de saisie pour entrer l'URL du flux RSS.
- **Validation** : Vérifier que l'URL saisie est valide et accessible.

1.2.2 Analyse des Sentiments

- **Analyse Automatisée** : Analyse de sentiment pour évaluer le sentiment des titres et résumés des articles
- **Catégorisation** : Classer les articles en sentiment positif, négatif ou neutre.

1.2.3 Tableau de Bord

- **Affichage des Articles** : Liste des articles avec titre, résumé, date de publication, auteur, et sentiment.
- **Filtrage** : Option pour filtrer les articles par sentiment (positif, négatif, neutre).

1.3 Exigences Générales

1.3.1 Accessibilité

- L'application doit être accessible via les principaux navigateurs web (Chrome, Firefox, Safari, Edge).
- L'interface utilisateur doit être responsive, c'est-à-dire adaptée à différents types d'appareils (ordinateurs & smartphones).

1.3.2 Performance

- L'application doit pouvoir gérer de nombreux utilisateurs simultanément sans dégradation des performances.
- Les analyses de sentiment doivent être effectuées rapidement pour ne pas ralentir l'expérience utilisateur.

1.4 Documentations

- Une documentation utilisateur doit être fournie pour aider les utilisateurs à utiliser l'application.
- Une documentation technique doit être disponible pour les développeurs.

1.5 Phases du Projet

1.5.1 Les spécifications

Définir les fonctionnalités et exigences techniques basées sur les besoins du client, et les documenter sous forme compréhensible pour l'équipe de développement.

1.5.2 Conception

Découper le logiciel en composants, définir l'architecture et répartir les tâches au sein de l'équipe.

1.5.3 Implémentation

Coder les fonctionnalités du logiciel en suivant les spécifications.

1.5.4 Test et Validation

Vérifier que chaque composant et l'ensemble du système fonctionnent correctement via des tests.

1.5.5 Déploiement

Préparer l'infrastructure de production, déployer l'application, et la rendre accessible aux utilisateurs.

2 Spécifications

Ces spécifications techniques décrivent en détail les fonctionnalités et les exigences techniques nécessaires pour développer l'application MoodRSS, conformément au cahier des charges fourni. Elles comprennent des descriptions techniques, les entrées/sorties des fonctionnalités, les technologies utilisées et les critères de performance.

2.1 Architecture de l'Application

2.1.1 Composants Principaux

- **Frontend** : Interface utilisateur accessible via un navigateur web.
- **Backend** : Application Django pour gérer les interactions avec la base de données.
- **Base de Données** : Stockage des informations relatives aux flux RSS.
- **Analyse de Sentiment** : Module pour évaluer le sentiment des articles.

2.1.2 Technologies Utilisées

- **Frontend** : HTML5, CSS3, JavaScript et Bootstrap
- **Backend** : Python et son framework Django
- **Base de Données** : SQLite
- **Analyse de Sentiment** : VADER (VADER Sentiment Analysis)
- **Serveur Web** : Nginx
- **Conteneurisation** : Dockere et Docker Compose
- **Hébergement** : Serveur privé Synology

2.2 Fonctionnalités

2.2.1 Recherche de Flux RSS

Permet à l'utilisateur d'entrer l'URL d'un flux RSS pour en extraire et analyser les articles.

Entrées : URL du flux RSS (chaîne de caractères)

Sorties : Liste des articles extraits (HTML)

Méthodes :

- Formulaire HTML pour l'entrée de l'URL du flux RSS.
- Traitement de la requête POST pour récupérer et afficher les articles du flux RSS.

Détails Techniques :

- Utilisation d'une fonction pour lire les flux RSS.
- Validation de l'URL avec des expressions régulières.
- Gestion des erreurs pour les URL invalides ou les flux RSS inaccessibles.

2.2.2 Analyse des Sentiments

Analyse les sentiments des titres et résumés des articles en utilisant VADER.

Entrées :

- Titre de l'article (chaîne de caractères)
- Résumé de l'article (chaîne de caractères)

Sorties : Sentiment de l'article (positif, négatif, neutre) (chaîne de caractères)

Méthodes : Fonction pour analyser le sentiment d'un article donné.

Détails Techniques :

- Utilisation de la bibliothèque `nltk` et du module `SentimentIntensityAnalyzer` de `VADER`.
- Concatenation du titre et du résumé pour une analyse de sentiment plus précise.
- Implémentation d'une fonction pour utiliser le module importé et retourner le sentiment basé sur le score compound.

2.2.3 Tableau de Bord

Affiche les articles du flux RSS avec leurs sentiments et permet de filtrer par sentiment.

Entrées :

- URL du flux RSS (chaîne de caractères)
- Filtre de sentiment (positif, négatif, neutre, tout) (chaîne de caractères)

Sorties : Tableau de bord affichant les articles filtrés (HTML)

Méthodes :

- Vue Django pour afficher le tableau de bord.
- Formulaire HTML pour soumettre l'URL du flux RSS et le filtre de sentiment.

Détails Techniques :

- Utilisation de Django pour gérer les vues.
- Implémentation des vues Django pour traiter les requêtes GET et POST.
- Récupération et affichage des articles dans un template HTML avec des filtres appliqués.

2.3 Sécurité

- Implémentation de la protection CSRF pour les formulaires.
- Utilisation du protocole HTTPS pour chiffrer les communications.

2.4 Hébergement

2.4.1 Type de Serveur

Serveur Synology : Utilisation d'un serveur Synology et du logiciel Container Manager pour déployer et gérer les conteneurs Docker.

2.4.2 Mise en Place de DDNS

DDNS : Configuration d'un service de DNS dynamique pour fournir une adresse accessible publiquement.

2.5 Tests & distinction développement / production

2.5.1 Tests Unitaires

Tests unitaires pour toutes les fonctions.

2.5.2 Tests Fonctionnels

Fonctions permettant de tester l'application dans son ensemble.

2.5.3 Distinction développement / production

Développement du projet en distinguant l'environnement de développement et celui de production.

2.6 Versioning

Déploiement du code sur Github afin de travailler à plusieurs sur le même projet et d'assurer un versioning.

2.7 Documentations

- **Code** : Documentation de chaque élément du code.
- **Documentation technique** : Les outils employés, les paramètres
- **Documentation utilisateur** : Comment utiliser le programme