

# **Cahier des Charges**

## **YOWL Community**

---

# **Plan**

1. Introduction.....	5
1.1 Contexte du projet.....	5
1.2 Objectifs du projet.....	5
1.3 Périmètre du projet.....	5
2. Description Fonctionnelle.....	5
2.1 Fonctionnalités principales.....	5
2.1.1 Gestion des utilisateurs.....	5
2.1.2 Gestion des contenus.....	6
2.2 Interfaces utilisateur.....	6
3. Contraintes techniques.....	6
3.1 Technologies utilisées.....	6
3.2 Performance.....	6
3.3 Sécurité.....	7
4. Gestion du projet.....	7
4.1 Équipe projet.....	7
4.2 Planning prévisionnel.....	8
4.3 Budget prévisionnel.....	8
5. Maintenance et Support.....	9
5.1 Maintenance.....	9
5.2 Support.....	9

6. Validation.....	9
6.1 Critères de validation.....	9
Tableau de bord KPI.....	10
1. Nombre d'utilisateurs actifs mensuels :.....	10
2. Nombre moyen de commentaires/avis par utilisateur :.....	10
3. Taux d'engagement (likes, partages, réponses):.....	10
4. Temps moyen passé par session :.....	10
5. Taux de rétention (D1, D7, D30):.....	10
User Testing & Feedback.....	11
Tests utilisateurs.....	11
Feedback & analyse.....	11
Priorisation des fonctionnalités.....	11
MVP & Versionning.....	12

# **1. Introduction**

## **1.1 Contexte du projet**

Le projet **YOWL COMMUNITY** consiste à développer une application web communautaire où les utilisateurs peuvent partager, commenter et réagir sur n'importe quel contenu trouvé sur internet. L'objectif est de créer un espace digital simple, moderne et interactif, où chaque membre de la communauté peut donner son avis, échanger avec les autres et suivre des tendances.

## **1.2 Objectifs du projet**

- Créer une plateforme intuitive permettant de publier, commenter et interagir autour de contenus web.
- Favoriser la création d'une communauté active et engagée.
- Mettre en place des outils de suivi et d'analyse (KPI, dashboard).
- Garantir la compatibilité sur tous les supports (desktop, mobile, tablette).

## **1.3 Périmètre du projet**

Le projet couvre le développement de **YOWL Community**, incluant :

- La gestion des utilisateurs (inscription, login, rôles, profils).
- La publication et l'interaction autour de contenus (commentaires, likes, partages).
- Un dashboard KPI pour suivre l'activité et l'engagement.
- Un système de notifications pour informer les utilisateurs.

# **2. Description Fonctionnelle**

## **2.1 Fonctionnalités principales**

### **2.1.1 Gestion des utilisateurs**

- Authentification par email/mot de passe.
- Gestion des rôles : utilisateur, modérateur, administrateur.
- Profils utilisateurs : photo, bio, préférences.
- Interface de gestion pour l'administrateur permettant de créer, modifier et supprimer des utilisateurs.
- Accessibilité avancée : prise en charge des lecteurs d'écran, contraste renforcé et navigation clavier.
- Gamification (version future) : système de badges ou points pour encourager l'engagement.

### **2.1.2 Gestion des contenus**

- Publication de liens et d'articles.
- Commentaires et réponses.
- Réactions (like, dislike).
- Notifications en temps réel.
- Rapports et analyses
- Dashboard avec indicateurs clés (utilisateurs actifs, posts publiés, engagement).
- Export des données (CSV/PDF).
- Suivi de l'évolution de la communauté (MAU, DAU, taux d'engagement).

### **2.2 Interfaces utilisateur**

- Interface claire et responsive (desktop, mobile, tablette).
- Page d'accueil avec flux de contenus.
- Pages Login / Register simples et intuitives.
- Tableau de bord avec graphiques (engagement, activité).

## **3. Contraintes techniques**

### **3.1 Technologies utilisées**

- Langages de programmation : Javascript, php.
- Base de données : MySQL.
- Frameworks : Vue.js pour le frontend, Laravel pour le backend et Tailwind pour le style.
- Serveur d'hébergement : Serveur cloud (AWS ou équivalent).

### **3.2 Performance**

- Le logiciel doit pouvoir gérer jusqu'à 5000 utilisateurs simultanés.
- Le temps de réponse pour chaque action de l'utilisateur ne doit pas excéder 3 milisecondes
- Le système doit être scalable pour supporter une montée en charge (jusqu'à 15 000 utilisateurs simultanés dans une version future).
- Optimisation des requêtes avec mise en cache (Redis ou équivalent) pour réduire le temps de réponse.
-

### 3.3 Sécurité

- Hashage des mots de passe (bcrypt).
- Conformité RGPD (gestion des données personnelles).
- Chiffrement des données sensibles (AES-256).
- Mise en place d'un système d'authentification à deux facteurs (2FA) dans une version ultérieure.
- Audit régulier du code et des accès serveur (logs, monitoring).

## 4. Gestion du projet

### 4.1 Équipe projet

- **Chef de projet** : Orlando DODAHO
- **Lead front-end** : Lauret CHACHA
- **Lead back-end** : Gilchrist KANTE
- **Lead QA/testeurs** : Nervely JEAN-CHARLES
- **QA/Testeurs** : chaque membre de l'équipe aura aussi un rôle de testeur transversal pendant la phase de validation, pour couvrir un maximum de scénarios.

## 4.2 Planning prévisionnel

Étape	Durée	Détail
Analyse des besoins	24 h	<ul style="list-style-type: none"> <li>- Prise en main du brief</li> <li>- Identification des attentes du client</li> <li>- Identification des entités impliquées</li> </ul>
Choix techniques & artistiques	24 h	<ul style="list-style-type: none"> <li>- Définition du stack technologique</li> <li>- Réalisation de la maquette</li> </ul>
Développement	72 h	<ul style="list-style-type: none"> <li>- Initialisation du projet</li> <li>- Intégration de la maquette</li> <li>- Développement des fonctionnalités</li> </ul>
Phase de test	48 h	<ul style="list-style-type: none"> <li>- Recrutement de bêta-testeurs</li> <li>- Prise en compte des feedbacks</li> </ul>
Livraison et mise en production	24 h	Préparation de la démo et restitution finale

## 4.3 Budget prévisionnel

Élément	Coût estimé
Développement logiciel	7 000 \$
Infrastructure (serveurs, etc.)	100 \$/mois
Maintenance et support	30 \$/mois
Hébergement pouvant évoluer vers un forfait supérieur selon la charge	+200 \$/mois

## **5. Maintenance et Support**

### **5.1 Maintenance**

- Correction des bugs : garantie de 2 mois après livraison.
- Mise à jour régulière du logiciel pour l'optimisation des performances et la sécurité.

### **5.2 Support**

- Support utilisateur disponible via mail.
- Temps de réponse moyen pour les tickets : 24h.

## **6. Validation**

### **6.1 Critères de validation**

- Toutes les fonctionnalités du MVP doivent être opérationnelles.
- Le logiciel doit passer par une phase de test utilisateur avant mise en production.
- Validation par le chef de projet et les parties prenantes à chaque étape clé du projet.

## Tableau de bord KPI

Cinq (05) indicateurs clés pour mesurer la performance et l'efficacité de l'application :

### **1. Nombre d'utilisateurs actifs mensuels :**

- **Description** : total des utilisateurs qui se connectent et interagissent avec la plateforme chaque mois
- **Objectif** : mesurer la croissance et l'adoption de l'application
- **Visualisation** : Courbe de progression mensuelle

### **2. Nombre moyen de commentaires/avis par utilisateur :**

- **Description** : ratio entre le nombre total de commentaires et le nombre d'utilisateurs actifs
- **Objectif** : évaluer l'engagement de la communauté
- **Visualisation** : histogramme ou jauge

### **3. Taux d'engagement (likes, partages, réponses):**

- **Description** : nombre total d'interactions (likes, partages, réponses) divisé par le nombre total de commentaires
- **Objectif** : mesurer la vitalité des discussions
- **Visualisation** : diagramme circulaire (likes vs partages vs réponses)

### **4. Temps moyen passé par session :**

- **Description** : durée moyenne des sessions utilisateurs
- **Objectif** : vérifier l'attractivité et l'ergonomie de l'application
- **Visualisation** : graphique en barres (comparaison semaine /mois)

### **5. Taux de rétention (D1, D7, D30):**

- **Description** : pourcentage d'utilisateurs qui reviennent après leur première utilisation
- **Objectif** : évaluer la fidélisation des utilisateurs
- **Visualisation** : courbes comparatives (D1, D7, D30)

# User Testing & Feedback

## *Tests utilisateurs*

- **Echantillon :** 5-10 utilisateurs (13-35 ans)
- **Scénarios :** inscription, commentaire, réaction, recherche
- **Méthode :** observation + questionnaire (clarté, facilité, utilité)

## *Feedback & analyse*

Regrouper les retours en trois (03) catégories :

- Fonctionnalités bien comprises et appréciées
- Points à améliorer (ergonomie, rapidité, accessibilité )
- Problèmes bloquants (bugs, incompréhensions majeures)

## **Indicateurs à mesurer:**

- Taux de réussite des tâches
- Temps moyen pour accomplir une action
- Niveau de satisfaction (score 1 à 5)

## *Priorisation des fonctionnalités*

- Haute valeur + Faible effort : **priorité immédiate**
- Haute valeur + Forte effort : version future
- Faible valeur + faible effort : bonus
- Faible valeur + fort effort : **à éviter**

## **MVP & Versionning**

### **MVP :**

- Inscription,
- Avis/commentaire,
- Lecture/réaction, recherche,

### **V1 :**

- Partage externe enrichi, modération basique,
- Dashboard personnel améliorer,
- Multilingue,
- Notifications en temps réels, profils,

### **V2 :**

- Système de recommandations personnalisées (contenus suggérés selon préférences),
- Partage multi-réseaux enrichi (X (ex Twitter), LinkedIn, WhatsApp, etc.),
- Gamification (badges, classements),
- Intégration d'API tierces (actualités, tendances sociales, etc.).