DataShare – Spécifications fonctionnelles

Objectif

Permettre à des utilisateurs, anonymes ou enregistrés, de transférer un ou plusieurs fichiers via des liens de téléchargement temporaires, avec des options de protection et de gestion pour les utilisateurs connectés.

MVP (Minimum Viable Product)

US01 — Upload (avec compte)

Description :
Un utilisateur connecté peut déposer un fichier afin d'obtenir un lien de téléchargement unique, avec la possibilité de le retrouver dans son espace personnel.

Règles de gestion :

- Le fichier doit être stocké dans un système local ou cloud (selon l'implémentation choisie).
- Un identifiant unique (token) est généré pour le téléchargement.
- La date d'expiration par défaut est de 7 jours, configurable par l'utilisateur à l'envoi.
- Le fichier et ses métadonnées sont automatiquement supprimés à expiration.
- Optionnellement, l'utilisateur peut définir un mot de passe pour restreindre le téléchargement.
- Le fichier est lié à l'identifiant utilisateur.
- L'utilisateur peut retrouver ses fichiers dans un historique.

Contrôles de saisie :

- Taille maximale : 1 Go.
- Fichiers interdits : à définir selon la politique de sécurité (ex. .exe, .bat, etc.).
- Champ "mot de passe" : minimum 6 caractères, si renseigné.
- Date d'expiration : maximum 7 jours.
- Ajout possible d'un ou plusieurs tags.

Droits :

- Réservé aux utilisateurs authentifiés (voir US03 et US04).

US02 — Téléchargement via lien

Description :
Un destinataire peut télécharger un fichier en accédant à un lien unique généré après l'upload.

Règles de gestion :

- Le lien est associé à un identifiant unique non prédictible.
- Si un mot de passe a été défini, il est requis pour accéder au téléchargement.
- Un lien expiré ou invalide renvoie une erreur explicite.
- Les métadonnées du fichier (nom, type, taille, date d'expiration) sont visibles avant téléchargement.

Contrôles de saisie :

- Champ mot de passe : requis uniquement si le fichier à télécharger est protégé, à valider côté client et serveur.

Droits :

- Accessible à tout utilisateur disposant du lien, tant que celui-ci est valide.

US03 — Création de compte

Description :
Un utilisateur peut créer un compte pour déposer et gérer des fichiers, consulter l'historique, etc.

Règles de gestion :

- L'adresse email doit être unique.
- Le mot de passe est stocké de manière sécurisée (hashé, salé).
- Aucun rôle particulier n'est nécessaire (pas de profil administrateur dans le MVP).
- Aucun email de confirmation requis (sauf évolution).

Contrôles de saisie :

- Email : format valide, unique en base.
- Mot de passe : minimum 8 caractères.

Droits :

- Accessible à tous.
- Création d'un token JWT à la connexion pour authentifier les requêtes.

US04 — Connexion utilisateur

Description :
Un utilisateur peut se connecter avec ses identifiants pour accéder à son espace personnel.

Règles de gestion :

- L'authentification est basée sur email + mot de passe.
- Un token JWT est généré et transmis au client.

Contrôles de saisie :

- Email : format valide.
- Mot de passe : correspondance exacte avec le hash stocké.

Droits :

- Uniquement pour les utilisateurs déjà enregistrés.

US05 — Consultation de l'historique

Description :
Un utilisateur connecté peut consulter l'ensemble des fichiers qu'il a envoyés via l'interface de son espace personnel.

Règles de gestion :

- L'historique affiche le nom du fichier, sa taille, sa date d'envoi, sa date d'expiration, l'état du lien (valide ou expiré).
- Aucun tri ni filtrage n'est obligatoire dans le MVP.
- L'utilisateur peut supprimer manuellement un fichier avant son expiration.

Contrôles de saisie :

- Aucun.

Droits :

- Accessible uniquement à l'utilisateur connecté propriétaire des fichiers.

US06 — Suppression d’un fichier

Description :
Un utilisateur connecté peut supprimer un fichier qu’il a précédemment envoyé.

Règles de gestion :

- La suppression entraîne la suppression physique du fichier sur le système de stockage ainsi que de toutes ses métadonnées.
- La suppression est irréversible.
- Seuls les fichiers non expirés sont affichés par défaut dans l’historique.

Contrôles de saisie :

- Confirmation de l’action requise côté front-end.

Droits :

- L’utilisateur ne peut supprimer que ses propres fichiers.

Fonctionnalités avancées (optionnelles)

US07 — Upload anonyme

Description :
Un utilisateur non authentifié peut déposer un fichier, comme en mode connecté, afin d'obtenir un lien de téléchargement unique, valide pendant une durée limitée.

Règles de gestion :

- Les règles de gestion sont identiques à US01 (l’upload avec compte), mais le fichier n’est pas lié à l'identifiant utilisateur.
- Le lien de téléchargement est généré de la même manière que lors de US01.

Contrôles de saisie :

- Les contrôles de saisie sont identiques à l’upload avec compte.
- Champs facultatifs : mot de passe, date d’expiration.

Droits :

- Accessible uniquement aux utilisateurs non authentifiés.
- L’upload anonyme ne donne pas accès à un historique ou à une gestion du fichier.

US08 — Gestion des tags

Description :
Un utilisateur connecté peut ajouter des tags à ses fichiers afin de mieux les organiser.

Règles de gestion :

- Un fichier peut avoir de 0 à N tags.
- Les tags sont définis librement par l'utilisateur (pas de suggestions dans le MVP).
- Les tags permettent un filtrage simple dans l'historique (facultatif).

Contrôles de saisie :

- Tag : texte libre, longueur maximale 30 caractères.
- Aucun doublon autorisé pour un même fichier.

Droits :

- Réservé aux utilisateurs connectés.

US09 — Ajout d'un mot de passe à un fichier

Description :
L'utilisateur (anonyme ou connecté) peut protéger l'accès au fichier par mot de passe.

Règles de gestion :

- Le mot de passe est requis lors du téléchargement.
- Il est stocké sous forme hashée (non visible ni réversible).
- Aucun mécanisme de récupération du mot de passe n'est prévu.

Contrôles de saisie :

- Minimum 6 caractères.
- Alphanumérique conseillé (aucune contrainte forte dans le MVP).

Droits :

- Ouvert à tous les utilisateurs lors de l'upload.

US10 — Expiration automatique des fichiers

Description :
Les fichiers envoyés expirent automatiquement selon la durée définie à l'envoi.

Règles de gestion :

- Durée par défaut : 7 jours.
- L'utilisateur peut choisir une durée personnalisée (entre 1 et 7 jours).
- Une tâche planifiée (cron ou équivalent) purge les fichiers expirés chaque jour.
- À l'expiration : suppression du fichier et des données associées.

Contrôles de saisie :

- Durée maximale : 7 jours.
- Le champ d'expiration est optionnel mais validé côté serveur.

Droits :

- Applicable à tous les utilisateurs (anonymes et connectés).

Contraintes techniques

Tout le code dans un repository GitLab ou GitHub avec historique Git propre. Le respect de la norme "conventional commit" est un plus.

Stack technique (à choisir)

- **Back-end** : 4 options
  - Spring Boot (Java) OU
  - NET Core (C#) OU
  - NestJS (TypeScript) OU
  - PHP Symfony / Laravel
- **Front-end** : 3 options
  - Angular OU
  - React OU
  - VueJS
- **Base de données** : 2 options
  - PostgreSQL OU
  - MongoDB
- **Stockage** : 2 options
  - Système de fichiers local OU
  - Stockage AWS S3

Suivi de qualité et maintenance

Un **plan de suivi de qualité et maintenance** est attendu. Il doit couvrir les spécifications relatives aux éléments de testing, de sécurité et de performance, répartis en plusieurs fichiers dans le repository :

- TESTING.md
- SECURITY.md
- PERF.md
- MAINTENANCE.md

Le plan doit contenir, au minimum, les éléments suivants :

Plan de tests – TESTING.md

- Tests unitaires sur les fonctionnalités obligatoires du MVP
- Tests end-to-end (avec Cypress ou équivalent, au moins 2–3 scénarios critiques)
- Critères d'acceptation pour les tests
- Instructions d'exécution
- Seuil de couverture minimal (objectif 70 %) indicatif avec une capture d'écran du rapport de couverture de code.

Garanti de sécurité – SECURITY.md

- Scan de sécurité basique (ex. dépendances/vulnérabilités npm)
- Analyse succincte des résultats du scan
- Analyse succincte des décisions

Suivi de performance – PERF.md

- Test de performance rapide sur un endpoint critique
    - Utiliser k6 (ou autre outil de performance)
    - Résultats et interprétation de test
- Budget de performance côté front (bundle, performance du navigateur)
    - Note que côté back, il y a déjà le test de performance qui valide cette partie.
- Suivi des métriques, ex.
    - Temps de réponse
    - Taille de fichiers
- Captures de logs ou métriques de performance (navigateur, serveur)

Documentation de maintenance – MAINTENANCE.md

- Procédures de mise à jour des dépendances
    - Leur fréquence
    - Leurs risques

Meilleures pratiques à suivre

- Architecture REST API
- Authentification JWT
- Validation des données côté client et serveur
- Gestion d'erreurs appropriée
- Accessibilité prise en compte des utilisateurs PSH
- Scripts de déploiement : installation, configuration BDD