## **Cahier des Charges : Application de Gestion d'Inventaire du Mydil**

### **1. Introduction et Contexte**

#### **1.1. Présentation du Projet**

Le Mydil de l'EPSI est un laboratoire d'innovation et de fabrication numérique (FabLab) mis à la disposition des étudiants. Il est équipé d'un parc de matériel informatique et électronique varié : ordinateurs, imprimantes 3D, casque de réalité virtuelle (VR), multiprises et outils pour l'électronique. La gestion actuelle de cet inventaire repose sur des méthodes manuelles, ce qui rend le suivi des prêts, la maintenance et la disponibilité du matériel complexes et chronophages.

Ce projet vise à développer une application web et mobile simple et centralisée pour optimiser la gestion de l'inventaire du Mydil. L'objectif est de fournir aux étudiants un moyen facile de consulter la disponibilité du matériel et de le réserver, tout en offrant aux administrateurs (personnel de l'EPSI, responsables du Mydil) un outil puissant pour suivre, gérer et maintenir l'équipement.

#### **1.2. Objectifs du Projet**

* **Centraliser l'information** : Créer une base de données unique et accessible pour tout le matériel du Mydil.
* **Simplifier la réservation** : Permettre aux étudiants de vérifier la disponibilité et de réserver du matériel en ligne.
* **Optimiser le suivi** : Assurer un suivi en temps réel des prêts, des retours et de l'état du matériel.
* **Faciliter la maintenance** : Planifier et tracer les opérations de maintenance préventive et corrective.
* **Responsabiliser les utilisateurs** : Mettre en place un historique clair des emprunts pour chaque étudiant.

#### **1.3. Cible Utilisateurs**

* **Étudiants de l'EPSI** : Utilisateurs principaux qui consultent et réservent le matériel.
* **Administrateurs du Mydil (enseignants, personnel technique)** : Utilisateurs en charge de la gestion de l'inventaire, de la validation des prêts et de la maintenance.

### **2. Périmètre Fonctionnel**

#### **2.1. Fonctionnalités pour les Étudiants**

##### **2.1.1. Authentification Unique (SSO)**

* **Connexion transparente** : Les étudiants se connecteront à l'application en utilisant leurs identifiants habituels de l'EPSI (même login et mot de passe que pour l'intranet, les emails, etc.). Il ne sera pas nécessaire de créer un nouveau compte.
* **Synchronisation automatique** : Les informations de base de l'étudiant (nom, prénom, classe) seront automatiquement récupérées depuis l'annuaire de l'école pour pré-remplir son profil utilisateur.
* **Fonctionnalité de Connexion ('Se connecter')** :
  + L'application affichera un bouton unique "**Se connecter avec les identifiants EPSI**".
  + Le clic sur ce bouton redirigera l'utilisateur vers la page de connexion SSO de l'école.
  + Après authentification réussie, l'utilisateur sera redirigé vers l'application en étant connecté à son compte personnel.

##### **2.1.2. Consultation de l'Inventaire**

* **Catalogue en ligne** : Affichage de tout le matériel disponible sous forme de fiches produits avec photo, description, caractéristiques techniques et statut (disponible, réservé, en maintenance).
* **Filtre et recherche** : Possibilité de rechercher un équipement par nom, catégorie (informatique, impression 3D, électronique) ou mot-clé.
* **Calendrier de disponibilité et vue tabulaire** :
  + Pour chaque équipement individuel, un **calendrier visuel** affichera les créneaux de réservation déjà pris et ceux disponibles.
  + Pour une **catégorie de matériel** (ex: "Casque VR"), une **vue synthétique (tableau ou planning)** affichera les **périodes de disponibilité** des différentes unités (ex: "Casque VR 1", "Casque VR 2") sur une période donnée (jour/semaine).

##### **2.1.3. Gestion des Réservations**

* **Système de réservation simple** : L'étudiant peut sélectionner un équipement et choisir un créneau horaire (date et heure de début et de fin).
* **Panier de réservation** : Possibilité de réserver plusieurs articles en une seule fois.
* **Notifications** : Envoi d'emails ou de notifications push pour la confirmation, le rappel de début et le rappel de fin de prêt.
* **Historique des emprunts** : L'étudiant peut consulter son historique personnel d'emprunts.
* **Processus de Demande d'Emprunt** :
  1. **Sélection et Créneau** : L'étudiant sélectionne l'équipement et définit la période d'emprunt souhaitée.
  2. **Soumission de la Demande** : Un clic sur "**Réserver**" soumet la demande à l'administrateur. La demande passe au statut "**En attente de validation**".
  3. **Validation par l'Administrateur** : L'emprunt ne devient effectif qu'après validation par un administrateur du Mydil.
  4. **Confirmation/Refus** : L'étudiant reçoit une notification (email/push) l'informant de la validation ou du refus de sa demande, avec un motif clair en cas de refus.
  5. **Statut de la Réservation** : L'historique personnel de l'étudiant reflète le statut actuel (En attente, Validée, Refusée, En cours, Terminée).

#### **2.2. Fonctionnalités pour les Administrateurs**

##### **2.2.1. Gestion Complète de l'Inventaire**

* **Tableau de bord (Dashboard)** : Vue d'ensemble de l'état du parc.
* **Ajout/Modification/Suppression d'équipement** : Interface CRUD complète.
* **Gestion par QR Code** : Pour faciliter les processus de check-in/check-out.

##### **2.2.2. Gestion des Prêts et Retours**

* **Validation des prêts (Check-out) et des retours (Check-in)** : Interface dédiée.
* **Gestion du Flux de Demandes** :
  + Tableau de bord spécifique affichant toutes les **demandes d'emprunt en attente** de validation.
  + Fonctionnalités "**Valider**" et "**Refuser**" pour chaque demande, avec possibilité d'ajouter un commentaire.
* **Suivi des retards** avec alertes automatiques.

##### **2.2.3. Suivi de la Maintenance**

* Déclaration d'incident et planification/historique des interventions.

##### **2.2.4. Gestion des Utilisateurs**

* Synchronisation avec l'annuaire de l'école (via LDAP, SAML ou autre protocole SSO).
* Attribution des rôles (étudiant, administrateur).

### **3. Spécifications Techniques**

#### **3.1. Architecture**

* **Application Web** : Framework moderne (ex: React/Vue.js pour le front-end, Node.js/Django/Symfony pour le back-end).
* **Base de Données** : Base de données relationnelle (PostgreSQL ou MySQL).
* **API RESTful**.
* **Responsive Design**.

#### **3.2. Sécurité et Authentification**

* **Authentification SSO (Single Sign-On)** : La connexion se fera via le système d'authentification centralisé de l'EPSI. Il faudra se coordonner avec les services informatiques de l'école pour intégrer l'application avec leur fournisseur d'identité (par exemple, via un protocole comme SAML ou OpenID Connect, ou une connexion à un annuaire LDAP / Active Directory).
* **Gestion des droits** : Accès aux fonctionnalités limité en fonction du rôle.
* **Protection des données** : Respect des normes du RGPD.

### **4. Hébergement de l'Application**

#### **Option 1 : Hébergement Local (On-Premise) 🖥️**

* **Description** : Hébergement sur un serveur de l'EPSI.
* **Avantages** : Contrôle total, sécurité des données, coût à long terme.
* **Inconvénients** : Coût initial, maintenance interne requise.
* **Recommandation technique** : Utilisation de Docker pour la conteneurisation des services.

#### **Option 2 : Hébergement Cloud ☁️**

* **Description** : Location de ressources chez un fournisseur cloud (Scaleway, OVHcloud, Azure, AWS).
* **Avantages** : Flexibilité, maintenance réduite, haute disponibilité.
* **Inconvénients** : Coûts récurrents, dépendance à un tiers.

#### **Recommandation**

Pour un projet scolaire, l'hébergement local via Docker est une excellente option si l'école dispose des ressources. Sinon, un VPS chez Scaleway ou OVHcloud est la solution la plus simple.

### **5. Livrables Attendus**

* Dossier de conception technique.
* Code source complet et commenté.
* Application fonctionnelle déployée et accessible.
* Documentation utilisateur et technique.