

1. Guide utilisateur – Utilisation du fichier `index.html`

1.1. Objectif du système

Ce petit outil permet de :

- Suivre les **sorties de produits** données aux patients.
- Comparer ces sorties avec :
 - le **stock cible** et les **seuils d'alerte** (pour éviter les ruptures),
 - les **commandes déjà passées**.
- Chercher rapidement une **référence produit** pour savoir :
 - combien a été consommé,
 - combien a été commandé,
 - si une **alerte** est déclenchée (risque de rupture).

Tout se fait via **une page web locale** : `index.html` (pas besoin d'Internet).

1.2. Fichiers fournis à l'utilisateur

L'utilisateur doit avoir, dans un même dossier (par exemple sur le Bureau) :

1. **Le programme :**

- `index.html`
- `xlsx.full.min.js` (bibliothèque Excel nécessaire au fonctionnement)

2. **Les trois fichiers Excel de données** (noms à adapter selon votre organisation, mais typiquement) :

- `sorties.xlsx ou sorties.csv`

Liste des produits donnés aux patients.

- `stock.xlsx ou stock.csv`

Liste de toutes les références disponibles, avec stock cible et seuil d'alerte.

- `commandes.xlsx`

Liste des produits commandés, avec les quantités.

Important : l'utilisateur **ne doit pas changer la structure** de ces fichiers (ni l'ordre ni le nom des colonnes), sinon le programme risque de ne plus fonctionner.

1.3. Structure des 3 fichiers Excel (vue d'ensemble)

1.3.1. Fichier 1 – Sorties patients (`sorties.xlsx` ou `sorties.csv`)

Contient une ligne par sortie de produit (par patient).

Exemple de colonnes (à titre indicatif, ne pas modifier dans votre version réelle) :

- Colonne A : Date
- ...
- Colonne I : **Référence produit** (c'est ce qui est utilisé par le programme)

Principes pour l'utilisateur :

- Ne pas insérer de lignes de titre supplémentaires.
- Toujours mettre la **référence produit** dans la même colonne (celle prévue à l'origine, ici I).

1.3.2. Fichier 2 – Stock théorique (`stock.xlsx` ou `stock.csv`)

Contient **une ligne par référence**.

Exemple de colonnes :

- Référence
- Libellé du produit
- Stock cible (quantité souhaitée en stock)
- Seuil d'alerte / minimum urgence (en dessous de ce nombre, il faut recommander)

Principes pour l'utilisateur :

- Ne pas ajouter / supprimer de colonnes sans validation d'un informaticien.
- Pour ajouter un nouveau produit :
 - le rajouter en **nouvelle ligne**,
 - renseigner au minimum : Référence, Libellé, Stock cible, Seuil d'alerte.



1.3.3. Fichier 3 – Commandes ([commandes.xlsx](#))

Contient **une ligne par commande** (ou par ligne d'une commande).

Ce fichier est généré par le logiciel renommer en “commandes.xlsx”

Principes pour l'utilisateur :

- Ne pas modifier l'ordre ou le nom des colonnes existantes.
 - Contrôler juste la colonne I ou 9 avec la **référence**.
-

1.4. Lancer le programme

1. Ouvrir le dossier contenant :
 - [index.html](#)
 - [xlsx.full.min.js](#)
 2. **Double-cliquer** sur [index.html](#).
 3. La page s'ouvre dans le navigateur (Chrome, Edge, etc.).
 4. Aucune installation n'est nécessaire.
-

1.5. Charger les fichiers Excel dans l'application

Sur la page `index.html`, vous voyez trois champs de type “**Choisir un fichier**” (ou “Parcourir”).

1. Sélectionner le fichier des sorties

- Cliquez sur le premier bouton **Choisir un fichier** (ex. “Fichier sorties / patients”).
- Sélectionner `"sorties.csv"`.
- Valider.

2. Sélectionner le fichier de stock

- Cliquer sur le deuxième bouton **Choisir un fichier** (ex. “Fichier stock / références”).
- Sélectionner `"stock.csv"`.
- Valider.

3. Sélectionner le fichier des commandes

- Cliquer sur le troisième bouton **Choisir un fichier** (ex. “Fichier commandes”).
- Sélectionner `"commandes.xlsx"`.
- Valider.

4. Cliquez sur le bouton “**Charger / Mettre à jour les données**”.

5. Attendre quelques secondes :

- Si tout se passe bien, un message indiquera que les données sont chargées.
- En cas de problème, un message d'erreur apparaît (voir section 1.8).

À refaire **à chaque fois que les fichiers Excel sont modifiés** (nouvelles sorties, nouvelles commandes, etc.).

1.6. Chercher une référence produit

1. Repérer le champ texte “Référence” ou “Rechercher une référence produit”.
2. Saisir la **référence exacte** du produit (exemple : 1503353).
3. Cliquer sur le bouton “Rechercher”
ou appuyer sur la touche Entrée du clavier.

La page affiche alors, pour cette référence :

- Le libellé du produit (nom).
- Le stock cible.
- Le total des **sorties** (quantité totale donnée aux patients).
- Le total des **commandes** pour cette référence.
- Le **stock théorique actuel** calculé.
- Le **seuil d'alerte**.
- Un statut :
 - **Vert / Normal** : au-dessus du stock cible ou du seuil d'alerte.
 - **Rouge / URGENT** : stock théorique en dessous du seuil d'alerte.

Interprétation simple pour l'utilisateur :

- Rouge = **commander rapidement**.
- Vert = pas de problème immédiat.

1.7. Zone d'alertes globales

Selon la version de la page, une section peut lister **toutes les références en alerte rouge** (sous le seuil d'alerte).

- Utilisation pratique :
 - Permet de voir en un coup d'œil **tous les produits à recommander en urgence**.
 - Cette liste est mise à jour à chaque clic sur “**Charger / Mettre à jour les données**”.
-

1.8. Messages d'erreur fréquents (et solutions simples)

1.8.1. « La bibliothèque Excel (XLSX) n'est pas chargée »

- Cause : le fichier `xlsx.full.min.js` n'est pas trouvé.
- Vérifications :
 1. `xlsx.full.min.js` doit être **dans le même dossier** que `index.html`.
 2. Le nom doit être exactement `xlsx.full.min.js` (sans `(1)`, sans `.txt`).

Si besoin, demander à l'informaticien de vérifier.

1.8.2. Erreur indiquant un problème de lecture d'un fichier

Exemples :

- Fichier vide,
- Mauvais format (ex. `.csv` ou `.xls` mal formé),
- Colonnes supprimées ou renommées.

Solution pour l'utilisateur :

- **Ne pas modifier la structure des colonnes.**
- En cas de doute, appeler l'informaticien pour vérifier les colonnes et les noms.

1.8.3. Aucune donnée trouvée pour la référence

- Vérifier :
 - Que la référence saisie existe bien dans les fichiers.
 - Qu'il n'y a pas d'espace en trop (avant ou après).
 - Qu'il n'y a pas d'erreur de frappe (0 / O, 1 / I, etc.).
-

1.9. Bonnes pratiques pour l'utilisateur

- Toujours **enregistrer** les fichiers Excel après avoir saisi de nouvelles données (sorties ou commandes) avant de les recharger dans [index.html](#).
- Ne jamais :
 - supprimer de colonnes,
 - renommer les colonnes existantes,
 - déplacer des colonnes.
- En cas d'évolution souhaitée (nouvelles colonnes, nouveaux calculs), en parler à l'informaticien.

2. Guide infogérance / informaticien

2.1. Architecture générale

- Application **100 % front-end** (HTML + JavaScript).
 - Aucune base de données, aucune communication réseau.
 - Lecture des fichiers Excel via la bibliothèque **SheetJS** (`xlsx.full.min.js`).
 - Le navigateur lit les fichiers locaux choisis par l'utilisateur via `FileReader` et `XLSX.read`.
-

2.2. Fichiers techniques

- `index.html`

Page principale contenant :

- le HTML (interface),
- le JavaScript métier (logique de lecture et de calcul),
- le CSS (style de base).

- `xlsx.full.min.js`

Bibliothèque SheetJS, version complète, minifiée.

2.3. Chargement et parsing des fichiers Excel

Le flux standard (déjà en place) est généralement :

1. Récupération des fichiers depuis `<input type="file">`.
2. Utilisation de `FileReader` pour les lire en `ArrayBuffer` ou `binary string`.
3. Appel de `XLSX.read(...)` avec le type approprié.
4. Conversion de la première feuille en JSON via
`XLSX.utils.sheet_to_json(...)`.

Stockage dans des variables JavaScript du type :

```
let donneesSorties = []; // fichier 1
let donneesStock = []; // fichier 2
let donneesCommandes = []; // fichier 3
```

5.

Les traitements suivants s'appuient ensuite sur ces tableaux.

2.4. Hypothèses actuelles sur la structure des fichiers

Important : Les colonnes sont actuellement codées en dur dans le JS.

Toute modification des entêtes ou de la structure Excel nécessite d'ajuster le code.

Typiquement :

- Fichier des sorties :
 - clé : **Référence** (colonne I dans Excel, mais dans le JS, on utilise le **nom de l'entête** de colonne, ex. "**Référence**" ou "**REF**").
 - champ quantité : "**Quantité**" (ou équivalent).
- Fichier de stock :
 - clé : **Référence** (entête identique à celui des sorties et commandes).
 - **stockCible** : stock théorique souhaité.
 - **seuilAlerte** : minimum urgence.
- Fichier des commandes :
 - clé : **Référence**
 - champ quantité : "**Quantité**" (commandée).

Pour valider exactement les noms utilisés, ouvrir **index.html** et rechercher les accès du type :

```
ligne["Référence"]
ligne["Quantité"]
ligne["Stock cible"]
ligne["Seuil alerte"]
```

2.5. Logique métier principale

2.5.1. Calcul des totaux

Pour une référence donnée `ref` :

- `totalSorties` = somme des quantités dans `donneesSorties` pour lesquelles `ligne["Référence"] === ref`.
- `totalCommandes` = somme des quantités dans `donneesCommandes` pour lesquelles `ligne["Référence"] === ref`.
- `stockCible` et `seuilAlerte` sont lus directement dans `donneesStock` pour la même référence.
- `stockActuelTheorique = stockCible - totalSorties + totalCommandes` (ou une variante selon ce qui a été codé).

2.5.2. Statut / alerte

En général, la logique est du type :

- si `stockActuelTheorique < seuilAlerte` → **Urgent / Rouge**
- sinon si `stockActuelTheorique` proche du seuil → **Alerte / Orange**
- sinon → **OK / Vert**

Cette logique peut être ajustée dans la fonction qui génère les valeurs de statut (chercher les mots-clés comme "ROUGE", "ALERTE", "OK" dans le JS).

2.6. Points d'extension fréquents

2.6.1. Ajouter un nouveau champ affiché

Exemple : afficher aussi la **consommation moyenne mensuelle**.

1. Ajouter une zone dans le HTML (par exemple un `<div>` dans la section résultats).
2. Dans la fonction qui traite la recherche par référence, calculer la nouvelle donnée (moyenne sur période, etc.).
3. Injecter la valeur calculée dans le DOM.

2.6.2. Modifier la structure des fichiers Excel

Si un changement de structure est demandé (ex. renommer une colonne, ajouter une colonne de “famille de produit”) :

1. Mettre à jour les fichiers Excel.

Adapter les accès dans le code JavaScript :

```
// Ancien :  
ligne["Référence"]  
  
// Nouveau (si la colonne se renomme) :  
ligne["Ref. produit"]
```

- 2.
3. Tester sur plusieurs références avec des données connues.

Recommandation : garder une copie **versionnée** de `index.html` (Git, ou au minimum des archives datées).

2.7. Mise à jour de la bibliothèque [xlsx.full.min.js](#)

Pour mettre à jour SheetJS :

1. Télécharger la nouvelle version de `xlsx.full.min.js` depuis SheetJS / CDN.
2. Remplacer le fichier dans le dossier.

Vérifier :

```
<script src="xlsx.full.min.js"></script>
```

- 3.
4. Tester :

- Chargement des 3 fichiers.
- Recherche d'une référence.
- Vérification des totaux.

En cas de doute, possibilité d'utiliser directement un CDN (si l'ordinateur a accès Internet) :

```
<script src="https://cdn.sheetjs.com/xlsx-latest/package/dist/xlsx.full.min.js"></script>
```

2.8. Débogage

- Ouvrir les **outils de développement** du navigateur (F12).
- Onglet **Console** :
 - Voir les erreurs JavaScript qui peuvent pointer sur :
 - un problème de lecture de fichier,
 - un champ inexistant (mauvais nom de colonne),
 - une faute de frappe dans le code.
- En cas de modifications :
 - Ajouter des `console.log(...)` pour :
 - inspecter les données lues,
 - vérifier le contenu des objets pour une référence donnée.

2.9. Sécurité et bonnes pratiques

- L'application fonctionne entièrement en **local** :
 - pas de données envoyées sur Internet,
 - pas de stockage côté serveur.
 - Recommandations :
 - Sauvegarder régulièrement :
 - les fichiers Excel,
 - le fichier `index.html` et `xlsx.full.min.js`.
 - Documenter les noms exacts des colonnes utilisés dans le code (une petite annexe technique).
-

2.10. Transmission du projet à un nouveau technicien

À transmettre au nouvel informaticien :

1. Le **dossier complet** contenant :
 - `index.html`
 - `xlsx.full.min.js`
 - un exemple anonymisé de `sorties.xlsx`
 - un exemple anonymisé de `stock.xlsx`
 - un exemple anonymisé de `commandes.xlsx`
2. Cette **documentation PDF**.
3. Un court mémo listant :
 - Les colonnes réellement utilisées et leurs noms exacts.
 - Les règles de calcul de stock théorique.
 - Les seuils utilisés pour passer d'un statut vert → orange → rouge (si spécifique).