

HarpoTab

Cahier des Charges

Convertisseur de Partitions Musicales
vers Tablatures pour Harmonica

Auteur : Mathurin C.
Date : 11 décembre 2025
Version : 1.0
Statut : Spécifications Phase 1

GitHub : <https://github.com/mathurinc/harpotab>

Table des matières

1	Présentation du Projet	2
1.1	Objectif	2
1.2	Portée	2
2	Spécifications Fonctionnelles	2
2.1	Entrées Supportées	2
2.1.1	Phase 1 (Prioritaire)	2
2.1.2	Phase 2 (Développement ultérieur)	2
2.2	Types d'Harmonica Supportés	2
2.3	Sortie Générée	2
2.4	Traitement de la Mélodie	3
2.5	Transposition Automatique	3
3	Spécifications Techniques	4
3.1	Architecture	4
3.2	Technologies et Dépendances	4
3.2.1	Phase 1 : Partitions (PDF/JPEG)	4
3.2.2	Phase 2 : Audio (MP3/YouTube)	5
3.2.3	Déploiement avec Docker	5
3.3	Structure du Projet	6
4	Interface Utilisateur	8
4.1	Technologies	8
4.2	Pages Principales	8
4.2.1	Page d'Accueil	8
4.2.2	Page de Conversion	8
4.2.3	Page Résultat	8
4.3	Gestion des Erreurs	8
5	Workflow Utilisateur	9
5.1	Phase 1 : Partitions	9
5.2	Phase 2 : Audio/YouTube	9
6	Contraintes et Exigences	10
6.1	Performance	10
6.2	Qualité	10
6.3	Sécurité	10
6.4	Accessibilité	10
7	Limites et Restrictions	10
7.1	Limitations Connues	10
7.2	Cas Non Supportés	11
8	Planning de Développement	12
8.1	Phase 1 : Fonctionnalités de Base	12
8.2	Phase 2 : Extraction Audio	12

8.3 Phase 3 : Améliorations	12
9 Critères de Succès	13
9.1 Phase 1	13
9.2 Phase 2	13
10 Documentation Requise	14
11 Maintenance et Évolution	14
11.1 Support à Long Terme	14
11.2 Évolutions Possibles	14
12 Références et Ressources	14
12.1 Technologies Utilisées	14
12.2 Documentation	14

1 Présentation du Projet

1.1 Objectif

HarpoTab est un outil de conversion automatique de partitions musicales et fichiers audio en tablatures pour harmonica. Le système extrait la mélodie principale et génère une tablature adaptée au type d'harmonica sélectionné, avec transposition automatique si nécessaire.

1.2 Portée

Le projet est divisé en deux phases de développement :

- **Phase 1** : Conversion de partitions (PDF, JPEG) vers tablature harmonica
- **Phase 2** : Extraction de mélodie depuis fichiers audio (MP3) et vidéos YouTube

2 Spécifications Fonctionnelles

2.1 Entrées Supportées

2.1.1 Phase 1 (Prioritaire)

- **PDF** : Partitions musicales au format PDF
- **JPEG/PNG** : Images de partitions scannées ou photographiées

2.1.2 Phase 2 (Développement ultérieur)

- **MP3** : Fichiers audio (extraction de mélodie)
- **YouTube** : Liens vers vidéos (extraction audio puis mélodie)

2.2 Types d'Harmonica Supportés

L'utilisateur doit pouvoir sélectionner parmi les types suivants :

- **Harmonica diatonique** (Richter 10 trous)
 - Tonalités : C, D, E, F, G, A, B \flat , B
- **Harmonica chromatique** (12/16 trous)
- **Autres types** (selon évolution du projet)

2.3 Sortie Générée

Un **PDF** généré via Lilypond contenant :

1. **Portée musicale** : Mélodie en notation classique

2. **Tablature harmonica** : Positionnée sous la portée, indiquant :
 - Numéro de trou
 - Sens du souffle (aspiration/expiration)
 - Techniques (bend, overblow, etc.)
3. **Accords** : Grille d'accords au-dessus de la portée
4. **Transposition** : Information sur la transposition appliquée (si nécessaire)
5. **Métadonnées** :
 - Titre du morceau
 - Tonalité originale et finale
 - Type d'harmonica utilisé
 - Tempo

2.4 Traitement de la Mélodie

- **Extraction** : Isolation de la mélodie principale depuis :
 - Partitions piano (généralement main droite)
 - Partitions multi-instruments
 - Fichiers audio (Phase 2)
- **Simplification** : Réduction à une ligne mélodique monophonique
- **Adaptation** : Ajustement de la tessiture pour l'harmonica

2.5 Transposition Automatique

Le système doit :

1. Analyser la mélodie extraite
2. Vérifier la jouabilité sur l'harmonica sélectionné
3. Proposer une transposition optimale si nécessaire
4. Indiquer clairement la transposition appliquée sur le PDF final

3 Spécifications Techniques

3.1 Architecture

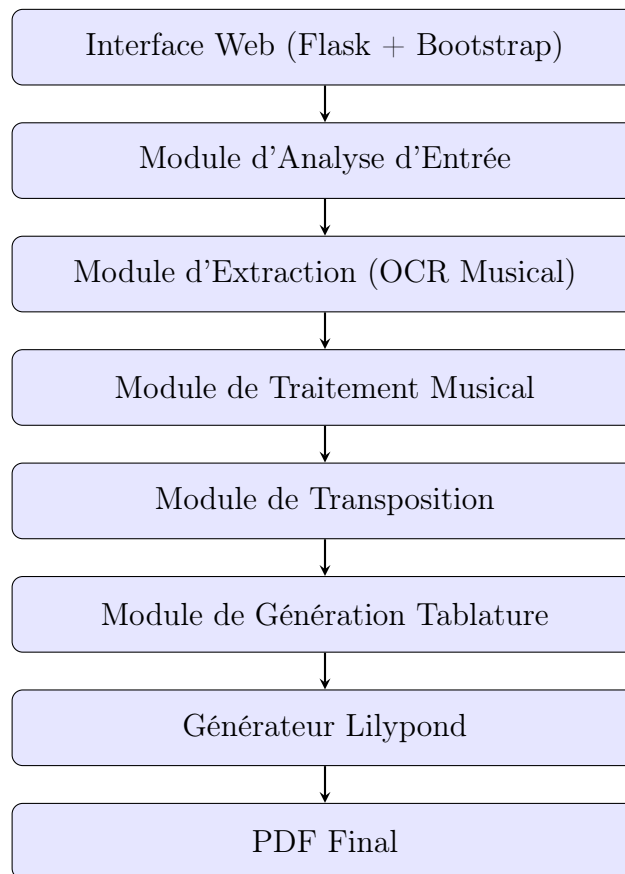


FIGURE 1 – Architecture du système HarpoTab

3.2 Technologies et Dépendances

3.2.1 Phase 1 : Partitions (PDF/JPEG)

Dépendances obligatoires :

- **Audiveris** : OCR musical pour lecture de partitions
- **Lilypond** : Génération de partitions et tablatures
- **Python 3.9+** : Langage principal
- **Flask** : Framework web
- **Bootstrap 5** : Interface utilisateur responsive

Bibliothèques Python :

```

1 flask>=3.0.0
2 pillow>=10.0.0
3 pdf2image>=1.16.0
4 opencv-python>=4.8.0
5 python-dotenv>=1.0.0
6 pytest>=7.4.0

```

Listing 1 – requirements.txt

Dépendances optionnelles :

- **OpenCV** : Prétraitement d'images
- **Tesseract** : OCR texte pour métadonnées

3.2.2 Phase 2 : Audio (MP3/YouTube)**Dépendances supplémentaires :**

- **yt-dlp** : Extraction vidéos YouTube
- **librosa** ou **essentia** : Analyse audio, extraction mélodie
- **aubio** : Détection de hauteur (pitch detection)

```

1 librosa>=0.10.0
2 yt-dlp>=2023.0.0
3 pydub>=0.25.0
4 numpy>=1.24.0
5 scipy>=1.10.0

```

Listing 2 – Dépendances Phase 2

3.2.3 Déploiement avec Docker

Pour faciliter le déploiement et garantir la reproductibilité de l'environnement, HarpoTab utilisera Docker pour l'environnement de production.

Avantages de Docker :

- **Isolation complète** : Toutes les dépendances (Python, Audiveris, Lilypond) dans un conteneur
- **Reproductibilité** : Même environnement en développement et production
- **Facilité de déploiement** : Une seule commande pour lancer l'application
- **Portabilité** : Fonctionne sur n'importe quel système supportant Docker

Architecture Docker :

```

1 HarpoTab/
2     Dockerfile                # Image principale
3     docker-compose.yml        # Orchestration
4     .dockerignore             # Fichiers ignorer

```

Listing 3 – Structure Docker

Utilisation :

```

1 # Build de l'image
2 docker build -t harpotab:latest .
3
4 # Lancement du conteneur
5 docker run -p 5000:5000 harpotab:latest
6
7 # Ou avec docker-compose
8 docker-compose up

```

Listing 4 – Commandes Docker

Note sur le développement

Pour le développement local, l'utilisation d'un environnement virtuel Python (venv) reste recommandée pour plus de simplicité et de rapidité. Docker sera utilisé principalement pour la production et la distribution.

3.3 Structure du Projet

```

1 HarpoTab/
2     app.py                    # Application Flask
3     principale
4     config.py                # Configuration
5     requirements.txt          # D pendances Python
6     setup.sh                 # Script d'installation
7     CAHIER_DES_CHARGES.md    # Ce document
8
9     modules/
10        ocr_reader.py         # Lecture partitions
11        melody_extractor.py   # Isolation m lodie
12        music_analyzer.py     # Analyse musicale
13        transposer.py         # Transposition
14        harmonica_mapper.py    # Mapping tablature
15        lilypond_generator.py  # G n ration PDF
16
17    data/
18        harmonica_maps/
19            diatonic_C.json    # Mapping C
20            diatonic_G.json    # Mapping G
21        templates/
22            lilypond_template.ly
23
24    static/
25        css/
26        js/
27        uploads/

```



```
28     templates/  
29         base.html  
30         index.html  
31         upload.html  
32         result.html  
33  
34     tests/  
35         test_ocr.py  
36         test_transposition.py  
37         test_lilypond_generation.py
```

Listing 5 – Arborescence du projet

4 Interface Utilisateur

4.1 Technologies

- **Framework** : Flask
- **CSS** : Bootstrap 5
- **Design** : Responsive, mobile-first

4.2 Pages Principales

4.2.1 Page d'Accueil

- Présentation du projet
- Fonctionnalités principales
- Bouton "Commencer"

4.2.2 Page de Conversion

1. **Section Upload** :
 - Glisser-déposer ou sélection fichier
 - Formats acceptés : PDF, JPEG, PNG
2. **Section Configuration** :
 - Sélecteur type d'harmonica
 - Sélecteur tonalité
 - Options avancées (facultatif)
3. **Bouton Conversion**

4.2.3 Page Résultat

- Aperçu du PDF généré
- Informations de transposition
- Bouton téléchargement PDF
- Bouton "Nouvelle conversion"

4.3 Gestion des Erreurs

Messages clairs pour :

- Format de fichier non supporté
- Partition illisible (mauvaise qualité OCR)
- Mélodie non extractible
- Morceau injouable sur l'harmonica sélectionné

5 Workflow Utilisateur

5.1 Phase 1 : Partitions

1. L'utilisateur accède à l'interface web
2. Upload d'un fichier PDF ou JPEG
3. Sélection du type d'harmonica (ex : Diatonique C)
4. Clic sur "Convertir"
5. Traitement backend :
 - (a) OCR musical (Audiveris)
 - (b) Extraction de la mélodie
 - (c) Détection des accords
 - (d) Analyse de jouabilité
 - (e) Transposition si nécessaire
 - (f) Génération tablature
 - (g) Compilation Lilypond → PDF
6. Affichage du résultat
7. Téléchargement du PDF

5.2 Phase 2 : Audio/YouTube

1. Upload MP3 ou lien YouTube
2. Extraction audio (si YouTube)
3. Analyse spectrale
4. Extraction mélodie principale
5. Transcription en notes
6. Suite identique à Phase 1 (étapes 5-7)

6 Contraintes et Exigences

6.1 Performance

- Temps de traitement < 30 secondes pour une partition simple (2-3 pages)
- Temps de traitement < 2 minutes pour extraction audio (Phase 2)

6.2 Qualité

- Précision OCR $> 90\%$ pour partitions de qualité standard
- Extraction mélodie correcte pour 80% des cas simples

6.3 Sécurité

- Validation des uploads (type MIME, taille)
- Taille max : 10 Mo pour PDF/images, 50 Mo pour audio
- Nettoyage automatique des fichiers temporaires
- Pas de stockage permanent des uploads utilisateur

6.4 Accessibilité

- Interface accessible (WCAG 2.1 niveau AA)
- Support navigateurs modernes (Chrome, Firefox, Safari, Edge)

7 Limites et Restrictions

7.1 Limitations Connues

Musique supportée

- Mélodie monophonique uniquement
- Pas de polyphonie complexe
- Tempo et rythmes simples à modérés

Partitions

- Qualité d'image suffisante pour OCR
- Partitions standards (clé de sol/fa)
- Pas de notations manuscrites (Phase 1)

Audio (Phase 2)

- Mélodie claire et audible
- Pas de polyphonie excessive
- Qualité audio correcte (> 128 kbps)

7.2 Cas Non Supportés

Non supporté actuellement

- Partitions manuscrites
- Tablatures guitare en entrée
- Musique atonale ou très complexe
- Harmonisation automatique

8 Planning de Développement

8.1 Phase 1 : Fonctionnalités de Base

Prioritaire

Configuration environnement (Audiveris, Lilypond)

Interface web Flask + Bootstrap

Upload et validation fichiers

OCR musical (Audiveris)

Extraction mélodie depuis partition

Mapping notes → tablature harmonica diatonique

Génération PDF avec Lilypond

Système de transposition basique

Durée estimée : 4-6 semaines

8.2 Phase 2 : Extraction Audio

Développement ultérieur

Intégration yt-dlp (YouTube)

Extraction mélodie audio (librosa/essentia)

Transcription audio → notes

Tests et optimisation

Durée estimée : 3-4 semaines

8.3 Phase 3 : Améliorations

Évolutions

Containerisation Docker complète

Support harmonica chromatique

Détection automatique du tempo

Export MIDI

Mode batch (conversion multiple)

API REST

9 Critères de Succès

9.1 Phase 1

Objectifs Phase 1

L'application peut convertir une partition piano PDF simple en tablature harmonica

La transposition automatique fonctionne

Le PDF généré est lisible et bien formaté

L'interface web est intuitive et responsive

9.2 Phase 2

Objectifs Phase 2

Extraction mélodie réussie sur 70%+ des morceaux tests

Intégration YouTube fonctionnelle

10 Documentation Requise

- **README.md** : Installation et utilisation
- **INSTALLATION.md** : Guide détaillé d'installation des dépendances
- **API.md** : Documentation API (si développée)
- **CONTRIBUTING.md** : Guide de contribution
- **CHANGELOG.md** : Historique des versions

11 Maintenance et Évolution

11.1 Support à Long Terme

- Mises à jour des dépendances
- Corrections de bugs
- Améliorations basées sur retours utilisateurs

11.2 Évolutions Possibles

- Application mobile
- Mode hors-ligne
- Bibliothèque de morceaux pré-convertis
- Communauté de partage

12 Références et Ressources

12.1 Technologies Utilisées

- **Audiveris** : <https://github.com/Audiveris/audiveris>
- **Lilypond** : <https://lilypond.org/>
- **Flask** : <https://flask.palletsprojects.com/>
- **Bootstrap** : <https://getbootstrap.com/>

12.2 Documentation

- Documentation Audiveris : <https://audiveris.github.io/audiveris/>
- Documentation Lilypond : <https://lilypond.org/doc/>
- MusicXML Standard : <https://www.musicxml.com/>

Historique des Versions

Version	Date	Modifications
1.0	11/12/2025	Version initiale du cahier des charges

Signatures

Chef de Projet
Mathurin C.

Date
11 décembre 2025

Signature

HarpoTab - Convertisseur de Partitions pour Harmonica
Version 1.0 - Phase 1

GitHub : <https://github.com/mathurinc/harpotab>