

運用チーム向け手順書

(AI-OCR インタビューシステム — v2025-01-11)

1. 目的

AI-OCR 導入のためのヒアリングを **30 分** で実施し、
「業務の事実情報」と「導入に対する感情・期待」を同時に取得するための現場手順をまとめます。

2. 事前チェックリスト

2.1 システム要件

1. **OS**: macOS（この手順書はmacOS環境に特化）
2. **ブラウザ**: 最新版 Chrome
3. **Python**: Python 3.11以上がインストール済み
4. **回線**: 有線または安定したWi-Fi（動画通話可能レベル）
5. **ハードウェア**: マイク・スピーカーまたはヘッドセット

2.2 アカウント・認証情報

1. **Googleアカウント**: Firebase Console アクセス用
 2. **OpenAI アカウント**: API キー取得用
-

3. インストールと起動手順

3.1 Firebase事前設定（必須）

重要：システムが依存するFirebaseサービスを先に設定します。この手順を飛ばすと起動時にエラーが発生します。

1. Firebase プロジェクトの作成

- [Firebase Console](#) にアクセス
- Googleアカウントでログイン
- 「プロジェクトを追加」をクリック
- プロジェクト名を入力（例：joblab-test）
- Google Analyticsの設定（任意、推奨は有効）
- 利用規約に同意して「プロジェクトを作成」

2. ウェブアプリの追加

- プロジェクト概要ページで「ウェブ」アイコン（</> マーク）をクリック
- アプリのニックネームを入力（例：hearing-ai-app）
- 「このアプリのFirebase Hostingも設定する」はチェック不要
- 「アプリを登録」をクリック
- 表示される設定情報（API Key等）を後で.envファイルに使用するためメモ

3. Firestore API の有効化

- [Google Cloud Console - Firestore API](#)
- プロジェクトを選択
- 「有効にする」をクリック

4. Firestore データベースの作成

- Firebase Console の「Firestore Database」セクション
- 「データベースを作成」をクリック
- テストモードまたは本番モードを選択
- ロケーション：任意で設定

5. Authentication の設定

- （技術ドキュメント（システム）の4.4）を参照
- Firebase Console の「Authentication」セクション
- 「開始」をクリック
- 「Sign-in method」タブで「メール/パスワード」を有効化
- データベース構造は（技術ドキュメント（システム）の4.4）を参照すること

6. ユーザーアカウントの作成

- 「Users」タブで「ユーザーを追加」
- 各オペレーター用のメールアドレスとパスワードを設定

7. セキュリティルールの設定

 Firestore の「ルール」タブで以下を設定：

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // 認証済みユーザーのみアクセス可能
    match /{document=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

3.2 初回セットアップ（エンジニア向け - macOSローカル環境）

注意： この手順はmacOSのローカル環境でシステムを動作させる方法を説明しています。

1. プロジェクトディレクトリに移動

```
cd （当システムのあるディレクトリ）
```

2. 仮想環境の作成と有効化

```
python3.11 -m venv .venv
source .venv/bin/activate
pip install --upgrade pip
```

3. 依存関係のインストール

```
#依存関係のインストール
pip install -r requirements.txt

# エラーする場合はレガシーリゾルバを使用（推奨）
pip install --use-deprecated=legacy-resolver -r requirements.txt

# または個別インストール（エラー時）
pip install python-dotenv==1.0.1
pip install firebase-admin==6.9.0
pip install chainlit==2.5.5
pip install langchain-openai==0.3.6
```

4. 環境変数ファイル（.env）の作成

プロジェクトルートの **.env** ファイルに以下の内容を取得した値に変更して設定

設定値の取得方法：

- **OPENAI_API_KEY**: [OpenAI Platform](#) でAPIキーを作成
- **Firestore設定値**: Firebase Console > プロジェクト設定 > 全般タブ > ウェブアプリの設定を表示
- **serviceAccountKey.json**: Firebase Console > プロジェクト設定 > サービスアカウントタブ > 「新しい秘密鍵の生成」をクリックしてダウンロード

```
# OpenAI API設定 (https://platform.openai.com/api-keys で取得)
OPENAI_API_KEY="sk-proj-xxx..."

# Chainlit認証設定（任意の文字列を設定）
CHAINLIT_AUTH_SECRET="your-secret-key"

# Firebase設定 (Firebase Console > プロジェクト設定 > 全般 > ウェブアプリ設定
から取得、各値を変更してください)
FIREBASE_API_KEY="AIzaSyB-xxx..."
FIREBASE_AUTH_DOMAIN="joblab-test.firebaseio.com"
FIREBASE_PROJECT_ID="joblab-test"
FIREBASE_STORAGE_BUCKET="joblab-test.firebaseio.com"
FIREBASE_MESSAGING_SENDER_ID="258749433945"
FIREBASE_APP_ID="1:258749433945:web:xxx..."
MEASUREMENT_ID="G-NW7NC6TYNY"

# Firebase Realtime Database URL (Firebase Console > Realtime Database
> データベースURLから取得)
FIREBASE_DATABASE_URL="https://joblab-test-default-
```

```
rtdb.firebaseio.com/"
```

```
# アプリケーション設定
USER_AGENT="my-app-bot"
```

```
# Firebase Admin SDK認証ファイル (Firebase Console > プロジェクト設定 > サービスアカウント > 新しい秘密鍵の生成でダウンロード)
GOOGLE_APPLICATION_CREDENTIALS="./serviceAccountKey.json"
```

5. 初回起動テスト

```
chainlit run main.py
```

- ブラウザで <http://localhost:8000> にアクセス
- ログイン画面が表示されることを確認
- Firebase認証が正常に動作することを確認

3.3 日常運用での起動手順

1. プロジェクトディレクトリに移動

```
cd /path/to/hearing-ai
```

2. 仮想環境の有効化

```
source .venv/bin/activate
```

3. システムの起動

```
chainlit run main.py
```

4. 起動確認

- ブラウザで <http://localhost:8000> にアクセス
- ログイン画面が表示されることを確認

3.4 インストール時トラブルシューティング（起動エラー対応）

よくあるエラーと対処法：（随時追記）

エラーメッセージ	原因	対処法
403 Cloud Firestore API has not been used...	Firestore API未有効	3.1の手順2を実行

エラーメッセージ	原因	対処法
404 The database (default) does not exist...	データベース未作成	3.1の手順3を実行
No module named 'chainlit'	依存関係未インストール	3.2の手順3を実行
Error loading .env file	環境変数ファイル未設定	3.2の手順4を実行

4. インタビュー実施フロー

手順	オペレーター操作	システム挙動
0	URL にアクセスし ログイン	Firebase 認証
1	「準備ができたなら '開始'」表示を確認	-
2	開始 と入力	ルール説明を受信
3	はい と入力して同意	Firestore にセッション作成
4	ネイチャリング ：AI から説明 → 回答	了承フラグが立つまで続く
5	インタビューフェーズ ：質問に回答	終了 で即終了可能
6	「ブラウザを閉じてください」表示	タブを閉じて完了

注意: いずれかの回答が 30 分を超えるとセッションがリセットされます。（セッション時間は変更可能）

5. 良い回答のコツ（仮）

- 数量・頻度・時間など **具体的な数字** を含める
- 不明点はその場で **確認質問** する
- 長い項目名はコピー＆ペーストで誤字防止
- データが無い場合は「後ほど確認します」と伝える

6. ブラウザ上でのトラブルシューティング (仮)

症状	原因	対処
ログイン失敗	PW 誤り	管理者にリセット依頼
ログイン後に白画面	キャッシュ	Ctrl+F5 でリロード
「入力時間超過」表示	タイムアウト	インタビュー再開始
Firestore に保存なし	通信断	エンジニアへ報告

7. エスカレーション手順（仮）

- Slack **#interview-ops** へ **[Critical]** を付けて投稿
- 記載内容：時刻 / メールアドレス / 症状概要

3. オンコールエンジニアが 15 分以内に応答

8. 用語集（随時追記）

- **ネイチャリングフェーズ**：目的説明と同意取得
- **業務内容フェーズ**：業務の事実情報ヒアリング
- **感情フェーズ**：期待・懸念など感情面を深掘り
- **開始**：インタビュー開始コマンド
- **終了**：即時終了コマンド