

英検RAGアプリ MVP 設計書

0. 要件定義結果（MVP 要件定義書 - Draft v1.0）

No	項目	内容	優先度
1	ターゲットユーザー	小学生～高校生、社会人含む英検受験者全般	●
2	対応級	英検全級（5級～1級）を対象。ただし MVP では 5級～準2級を重点	●
3	データ構成	語彙（vocab）／長文（passages）／リスニング（listening）の3分類で構成	●
4	データ形式	CSVファイル（UTF-8 or Shift_JIS）＋メタデータ付き	●
5	Embedding モデル	OpenAI text-embedding-ada-002 を使用	●
6	ベクトルDB	Pinecone（Namespace：vocab, passages, listening）	●
7	LLM モデル	OpenAI gpt-3.5-turbo を使用	●
8	UI	Streamlit ベースの簡易 UI。質問欄と応答欄、級別選択（ドロップダウン）を提供	●
9	入力言語・出力言語	質問：日本語、英語混在可。応答：日本語出力が原則	●
10	Namespace フィルタリング	各CSV種別ごとに Pinecone namespace を分離、RAG構成時にフィルタ	●
11	検索件数	Pinecone top_k=3 程度を想定（将来的に調整可能）	●
12	応答形式	シンプルな要約 or 例文付き回答（出力形式指定なし）	●
13	メタ情報表示	コンテキスト元データ表示機能（タイトル・出典・元文など）	●
14	音声対応	音声読み上げ、TTS連携（gTTS など）	●
15	運用想定	ユーザー限定なし（公開利用可）、無料範囲での検証・評価を主目的	●
16	導入先・展開可能性	SAPIX・Z会・高校英語教材出版社等への展開も将来的に視野	●

●＝確定 / ●＝今回あなたが単独で決定 / ●＝後回し／将来拡張

0.1 対象ユーザー

- ・小学生～高校生、および社会人学習者（英検全級対象）

0.2 対象機能（MVP）

- ・語彙検索：英単語の意味、品詞、例文を即時返却
- ・長文QA：過去問長文の要点要約・質問応答
- ・リスニングQA：スクリプト要約・質問応答

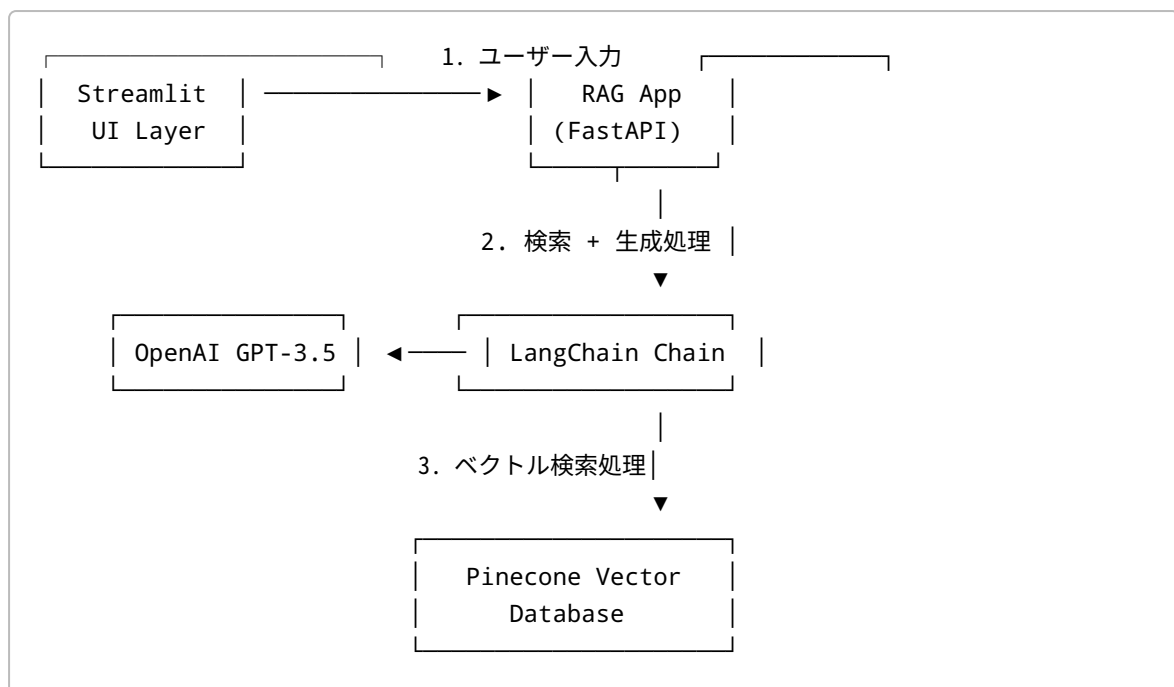
0.3 非機能要件

- ・レイテンシ：検索＋生成応答は3秒以内
- ・可用性：Streamlit Community Cloud で稼働可能
- ・コスト：約10USD/月以内（無料枠＋低量利用想定）

0.4 制約事項

- ・データは公開済CSVの要約版を使用
- ・API呼び出し数は1,000/月程度を想定

1. 全体アーキテクチャ



- ・ **Streamlit UI** : ユーザーインターフェース（質問入力、回答表示）
- ・ **RAG App 層** : 質問を受けて RetrievalQA チェーンを実行するアプリケーション層
- ・ **LangChain** : 複数のチャーン（Embedding → Vector Retrieval → LLM）を統合
- ・ **OpenAI GPT** : 実際の応答生成を担う大型言語モデル
- ・ **Pinecone** : 埋め込みベクトルの検索・管理を行うベクトルデータベース

2. 機能ブレイクダウン

項番	機能	詳細
1	データインジェスト	CSV（語彙・長文・リスニング）を読み込み、OpenAI 埋め込みを取得して Pinecone に upsert
2	ベクトル検索	質問文の埋め込みを生成 → Pinecone から関連ドキュメントを取得
3	応答生成	取得ドキュメントとユーザー質問を LLM（GPT-3.5）に渡し、回答を生成
4	Streamlit UI	質問入力フォーム、回答表示、履歴保存、級フィルタリング機能
5	環境設定・構成管理	.env 管理、依存ライブラリ管理（requirements.txt）
6	デプロイ準備	Dockerfile、CI/CD 設計、クラウドデプロイ

3. タスク & スケジュール詳細

期間	タスク	担当	備考
Day-1 (完了)	環境構築・API Key 発行	開発者	Python, venv, .env 作成
Day-2 (完了)	データ準備 & ingest.py 作成	開発者	CSV フォーマット設計
Day-3 (完了)	Pinecone アカウント作成 & データ登録	開発者	upsert CSV → Pinecone
Day-4 (完了)	RAG アプリ基本実装 (rag_app.py)	開発者	Streamlit + RetrievalQA 統合
Day-5	UI 改善	開発者	サイドバー、回答履歴、級選択
Day-6	級別フィルタ & モード切替	開発者	ドロップダウンで級レベル選択
Day-7	テスト & CI セットアップ	開発者	pytest、GitHub Actions
Day-8	デプロイ準備	開発者	Dockerfile、Heroku/AWS/Streamlit Cloud用
Day-9	デプロイ & SRE	開発者	本番検証、負荷テスト
Day-10	運用 & モニタリング	開発者	ログ監視、エラーアラート設定

今後の進め方

1. **Day-5:** Streamlit UI の画面調整・回答履歴保存を実装
2. **Day-6:** 質問に対して「5級〜1級」フィルタを追加
3. **Day-7:** ユニットテスト&CI/CD パイプライン構築
4. **Day-8〜9:** コンテナ化&クラウドデプロイ
5. **Day-10:** 本番運用体制の整備（監視・アラート）

以上を次スレッドの議題として取り上げ、各タスクの具体的なコーディング&レビューを進めていきましょう！