

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

## 1. はじめに

### 1.1. 本書の目的

本書は、AWS総合研修：WEB3層モデルの実装の課題として開発する「ナレッジ管理システム」の要件定義書です。

開発スコープ・機能要件・技術構成について、認識を合わせ 及び合意を得ることを目的とします

### 1.2. 設計書との関係

提示された設計書「ナレッジ管理システム\_アーキテクチャ設計書.md」は、Cloudflare + GCP構成を示しています。

本研修では、以下の方針でDocker版として実装します。

提示された設計書（Cloudflare + GCP構成）	本研修での実装（Docker版）
Cloudflare Workers	FastAPI（Pythonバックエンド）
Cloudflare D1（SQLite系RDB）	PostgreSQL
Cloudflare R2（オブジェクトストレージ）	ローカルファイルシステム
Cloud Run	不要（Dockerで代替）
LangGraph / Claude API	対象外（オプション実装）

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

## 2. システム概要

### 2.1. システム名

システム名：ナレッジ管理システム

### 2.2. システムの目的

社内ドキュメント（Markdown (.md) ファイル）をインポートし、一覧・検索・閲覧・関連確認ができるナレッジ管理システムを構築します。編集機能は持たず、インポート（インポートのみ）・閲覧・関係性探索に特化します。

### 2.3. ターゲットユーザー

- 想定は、社内のエンジニア等
- 研修課題としては、研修担当者、研修受講者

### 2.4. データソース

本システムに投入するデータは以下の通りです。

データ	データ形式	件数（想定）	備考
研修用・新規MDファイル	Markdown (.md)	50～100件程度	

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

### 3. システム構成（WEB3層モデル）

#### 3.1. WEB3層モデルの概要

本システムはWEB3層モデルで構成します。各層の役割は以下の通りです。

層	役割	使用技術
プレゼンテーション層	ユーザーインターフェース ブラウザからのアクセスを受け付ける	HTML、CSS、JavaScript Nginx（リバースプロキシ）
アプリケーション層	ビジネスロジック APIの処理・データの加工	FastAPI（Python）
データ層	データ・ファイル・メタデータ管理	メタデータ（PostgreSQL） 本文（ローカル）

#### 3.2. 技術スタック

コンポーネント	技術	役割
フロントエンド	HTML / CSS / JavaScript	UI画面
リバースプロキシ	Nginx	ルーティング・静的ファイル表示
バックエンドAPI	FastAPI（Python）	REST API
メタデータDB	PostgreSQL	ドキュメントのメタ情報管理
本文ストレージ	ローカルファイルシステム	Markdownファイル保存
コンテナ管理	Docker / Docker Compose	全サービスの統合起動

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

### 3.3. データストアの役割分離

操作	担当	理由
ドキュメント一覧・ページネーション	PostgreSQL	効率的なページネーションが得意
テキスト検索・フィルタ	PostgreSQL	条件に沿ったデータを適切に抽出
ドキュメント本文の表示	ローカルファイルシステム	Markdownファイルを直接返す
ブックマーク・閲覧履歴	PostgreSQL	ユーザー操作の記録

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

## 4. 機能要件（本研修のスコープ・MVP）

### 4.1. 本研修のスコープ・MVPの定義

本研修のスコープは、「Phase 1: 基本機能」の機能要件をMVP（実装ゴール）として定義します

No.	機能	概要	優先度
1	ファイルアップロード	Markdownファイルをシステムに登録する	必須
2	一覧表示	ドキュメントの一覧をページネーション付きで表示する	必須
3	検索	タイトル・タグ・カテゴリで絞り込み検索する	必須
4	閲覧	Markdownを整形してドキュメントの本文を表示する	必須
5	関連確認	タグ・Neo4jを利用して関連ドキュメントを表示する	必須
6	ブックマーク	ドキュメントをお気に入り登録する	要調整
7	閲覧履歴	最近閲覧したドキュメントを表示する	要調整

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

## 4.2. 機能詳細

### 4.2.1 ファイルアップロード

- Markdownファイル (.md) をフォームからアップロードできる
- アップロード時にタイトル・タグ・カテゴリを手動入力できる
- ファイル名を初期タイトルとして自動セットする
- アップロードした本文をローカルFSに保存する
- メタデータ（タイトル・タグ・カテゴリ・投稿日時）をPostgreSQLに登録する
- taggedリレーションのみを登録する

### 4.2.2 一覧表示

- 登録されたドキュメントの一覧を表示する
- 1ページあたり20件表示し、ページネーションで切り替えられる
- 更新日時の降順で並べる
- タイトル・カテゴリ・タグ・更新日時を一覧に表示する

### 4.2.3 検索

- タイトルのキーワード検索ができる（LIKE検索）
- カテゴリで絞り込みができる
- タグで絞り込みができる
- 複数条件を組み合わせて検索できる

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

#### 4.2.4 閲覧

- Markdownをレンダリングして表示する（見出し・コード・テーブルなど）
- タイトル・カテゴリ・タグ・更新日時をメタ情報として表示する
- 閲覧履歴を自動で記録する ※要調整（シングルユーザーのため、簡易実装。もしくは、未実装とする。）

#### 4.2.5 ブックマーク

- ドキュメントをお気に入り登録する ※要調整（シングルユーザーのため、簡易実装。もしくは、未実装とする。）

#### 4.2.6 閲覧履歴

- 最近閲覧したドキュメントを表示する ※要調整（シングルユーザーのため、簡易実装。もしくは、未実装とする。）

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

#### 4.4. 対象外機能

##### Phase 2: 概念グラフ

- LLMによる概念抽出 + Neo4j登録
- 関連ドキュメントのサイドバー表示
- グラフ探索ビュー

##### Phase 3: 外部連携

- DOCX / PDF インポート対応
- Google Drive連携（手動インポート）
- 外部リンク（スプレッドシート等）の登録

##### Phase 4: 自動化・AI

- Drive Webhookによる自動再インポート
- AIアシスタントとの統合（ラーニングシステムと共有）
- ラーニングシステムとのオントロジー共有

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

## 5. 画面一覧

No.	機能	概要	URL (予定)
1	ドキュメント一覧画面	登録済みドキュメントの一覧・検索	/ (ルート)
2	ドキュメント詳細画面	本文表示 (Markdown)	/documents/{id}
3	ドキュメント登録画面	ファイルアップロード (メタ情報入力)	/upload
4	検索結果画面	キーワード・タグ・カテゴリでの絞り込み結果	/search

## 6. 開発スケジュール

Week	期間	作業内容	成果物
1	2/17 ~ 2/20	要件定義・基本設計	要件定義書
2	2/24 ~ 2/27	基本設計・詳細設計	基本設計書、詳細設計書
3	3/2 ~ 3/5	構築 (インフラ) Docker環境・DB構築	docker-compose
4	3/9 ~ 3/13	構築 (アプリ) ・テスト	最終成果物 (システム)

要件定義書	システム名	サブシステム名	初版	版数	更新日	作成者	
	ナレッジ管理システム	Phase 1: 基本機能	2026/2/19	1	—	鶴原	

## 7. 非機能要件

### 動作環境：

- Docker / Docker Compose が動作するローカル環境

### ブラウザ：

- Firefox、Google Chrome（最新版）

### アクセス方法：

- http://localhost でアクセスできること

### データ量：

- 50～100件のMarkdownファイル

### レスポンス：

- 一覧取得：3秒以内 / 詳細表示：5秒以内（目安）

### ソース管理：

- GitHubにて公開（READMEあり）

### 認証：

- 不要（研修環境のためシングルユーザー想定）

### システムテスト：

- システムテストはシナリオベースとする（簡易テスト）。