

詳細設計書

新医療機関情報システム

作成日：2025 年 9 月 10 日

作成者：医療 DS 学科 加藤 稜也

1. 使用する技術・環境

- 言語：PHP
- DB：MySQL
- 外部サービス：Google Maps API（住所検索予定）

2. モジュール外部仕様

2.1 認証・認可モジュール

- 役割：ログイン/ログアウト、権限判定（一般/事務/管理者/システム管理者）
- 入力：ユーザーID、パスワード
- 出力：認証結果、有効判定表示、セッション
- エラー：認証失敗、ロック中、アカウント停止中
- 権限：全権限、以降の各モジュールへ権限判定を提供

2.2 マスタ管理モジュール

対象：hospital_types、ward_types、medical_departments、medical_services、clinical_pathways、kawasaki_university_facilities、kawasaki_university_departments

- 役割：各種マスタの CRUD と有効/無効管理、表示順管理
- 入力：コード、名称、表示順、有効フラグ
- 出力：一覧、参照、登録/更新結果
- エラー：重複コード、参照整合性エラー（使用中は無効化不可、条件付き？）
- 権限：システム管理者

※ CRUD：

Create（新規登録）、Read（参照・検索）、Update（修正）、Delete（削除）の略。

2.3 医療機関管理モジュール (hospitals テーブル)

- **役 割**：医療機関の基本情報 CRUD、検索/一覧/詳細、閉院/開院管理
- **入 力**：hospital_id, hospital_name, hospital_type_id, bed, has_pt/ot/st
- **出 力**：一覧/詳細、登録・更新・削除結果、エクスポート (Excel)
- **エラー**：必須不足、重複 (hospital_id)、外部キー不整合 (hospital_type_id)
- **権 限**：閲覧＝一般以上、CRUD＝管理者以上

2.4 検索・出力モジュール (一覧/CSV/Excel)

- **役 割**：横断検索 (名称・区分・地区・病床・在籍フラグ)、CSV/Excel 出力
- **入 力**：検索条件 (地域、診療科、診療日、紹介・逆紹介有無・年度・施設)、医療機関名、病院区分、医師会、理事長・病院長出身校)
- **出 力**：ページング済み一覧、CSV/Excel ファイル
- **エラー**：検索条件不正、出力上限超過
- **権 限**：閲覧＝一般以上

2.5 CSV インポートモジュール (一括登録/更新)

- **役 割**：医療機関/住所/連絡先/アサインの一括取込、ドライラン検証
- **入 力**：CSV (UTF-8、ヘッダ固定)
- **出 力**：成功/失敗件数
- **検 証**：必須/型/長さ/マスタ参照/ユニーク/整合性
- **権 限**：管理者以上

2.6 問い合わせ管理モジュール

- **役 割**：問い合わせの登録/参照/履歴管理、回答ステータス管理
- **入 力**：緊急度、内容
- **出 力**：問い合わせ一覧、詳細、進捗
- **権 限**：閲覧＝一般以上、管理＝システム管理者のみ

2.7 監査/ログ参照モジュール

- **役 割**：認証/閲覧/更新/エクスポート/権限変更の記録・検索・出力
- **入 力**：期間、ユーザー、イベント種別、対象 ID

- 出 力：ログ一覧
- 権 限：監査参照：システム管理者のみ

2.8 システム管理（通知・メンテナンス・設定）

- 役 割：運用メッセージ配信、メンテモード切替、設定変更
- 入 力：通知内容、運用コマンド、設定値
- 出 力：実行結果、履歴
- 権 限：システム管理者のみ

3. データ構造

別途ファイル（ [DB 構造](#) ）へ記述

4. 処理フロー・アルゴリズム

4.1 認証・認可

- **前 提**：ユーザーは ID・パスワードを入力。
権限は「一般」「一般（事務）」「管理者」「システム管理」
- **正常フロー**：
 - ① 入力された ID・パスワードを受け取る
 - ② パスワードをハッシュ化して DB と照合
 - ③ 一致すればログイン成功、セッション開始
 - ④ 権限に応じて利用可能な画面を切り替え
- **エラー処理**：
 - 不一致 → 「認証失敗」を表示
 - 5 回以上失敗 → アカウントロック
 - 停止ユーザー → 「利用できません」を表示
- **結 果**：ログイン成功時はメニューへ遷移

4.2 マスタ管理（病院区分・部署など）

- **前 提**：システム管理者のみ操作可
- **正常フロー**：
 - ① マスター一覧を表示
 - ② 新規登録/修正/削除/無効化の操作を実行
 - ③ DB へ反映
- **エラー処理**：
 - 重複コード → エラーメッセージ表示
 - 参照中データを削除しようとした場合 → 「削除不可」
- **結 果**：更新成功時は「保存しました」と表示

4.3 医療機関管理（病院情報）

- **前 提**：管理者以上のみ登録・更新可能。一般・事務は閲覧のみ

- **正常フロー：**
 - ① 医療機関一覧（医療機関 CD で検索）又は新規登録画面を開く
 - ② 病院名・区分・病床数などを入力
 - ③ バリデーション通過後、DB に反映
- **エラー処理：**
 - 必須不足 → 「未入力があります」
 - 重複 ID → 「既に登録されています」
 - 外部キー不整合（区分未登録など） → 「区分が存在しません」
- **結果：**登録/更新完了後、一覧画面に反映

4.4 CSV インポート

- **前提：**管理者以上が実行可、CSV のヘッダは固定
- **正常フロー：**
 - ① CSV をアップロード
 - ② データ検証
 - ③ 配列へセッションへ挿入
 - ④ ユーザーへ最終確認を促す
 - ⑤ 問題なければ DB へ登録
- **エラー処理：**
 - ヘッダ不正 → 「データ形式が異なります」
 - 必須不足 → 「CSV データが登録されていません」
 - マスタ未登録 → 「医療機関 CD が登録されていません」
- **結果：**「登録が完了しました」と表示

4.5 Excel エクスポート

- **前提：**一般以上が利用可
- **正常フロー：**
 - ① 出力対象を検索条件で指定
 - ② DB からデータ抽出
 - ③ Excel ファイルに変換してダウンロード開始
- **エラー処理：**
 - 条件に一致するデータがない → 「該当データがありません」
- **結果：**Excel ファイルがダウンロードされる

4.6 問い合わせ管理

- **前 提**：ログインユーザーのみ利用可能
- **正常フロー**：
 - ① 問い合わせフォームに緊急度・内容を入力
 - ② DB に保存し、ステータスを「未登録」で登録
 - ③ 一覧画面で表示
- **エラー処理**：
 - 必須不足 → 「内容を入力してください」
- **結 果**：問い合わせ一覧に反映

4.7 監査ログ

- **前 提**：全操作が対象。参照はシステム管理者のみ
- **正常フロー**：
 - ① 操作発生時にログ（ユーザーID、操作内容、対象 ID、時刻）を保存
 - ② システム管理者は検索・閲覧が可能
- **エラー処理**：
 - 権限不足 → メニュー画面へ遷移
- **結 果**：監査ログが一覧で出力される

4.8 システム管理

- **前 提**：システム管理者のみ操作可
- **正常フロー**：
 - メンテナンス
 - ① メンテナンス設定を選択
 - ② コマンドを実行
 - ③ 結果を画面へ表示
 - 通知作成
 - ① 必要項目を入力
 - ② DB へ保存
 - ③ 一覧画面で表示
- **エラー処理**：
 - 権限不足 → メニュー画面へ遷移
 - 実行失敗 → 処理に失敗しました

- **結 果**：履歴に保存、必要に応じて通知

5. 命名規約

5.1 データベース

✧ テーブル

- 複数形の英語名を使用する
例：hospitals、addresses、contact_details
- スネークケース(単語間を「 _ 」でつなぐ)
例：hospital_types、hospital_departments

✧ カラム

- スネークケース(単語間を「 _ 」でつなぐ)
例：hospital_name、postal_code
- 論理削除/有効フラグ → is_active (真偽値)
- 日時 → created_at、updated_at (「DATETIME」型で統一)

✧ 主キー・外部キー

- 主キー：<テーブル名>_id
例：hospital_id、address_id
- 外部キー：参照先の主キー名を原則そのまま利用
例：hospital_id (addresses → hospitals 参照)

5.2 PHP

✧ クラス

- パスカルケース (単語毎の先頭を大文字にする)
例：HospitalService、ContactDetailController

✧ メソッド・関数

- キャメルケース (最初の単語のみ小文字にし、それ以外は大文字)
例：getHospitalList()、updateContactDetail()

✧ 変数

- スネークケース(単語間を「 _ 」でつなぐ)
例：\$hospital_list、\$contact_details

✧ 定数

- 大文字＋スネークケース
例：MAX_RETRY_COUNT、DEFAULT_TIMEOUT

✧ ファイル

- クラス名と一致させる

例：HospitalService.php、ContactDetailController.php

5.3 JavaScript

✧ 変数・関数

- キャメルケース

例：hospitalList、fetchHospitals

✧ 定数

- 大文字＋スネークケース

例：API_BASE_URL、DEFAULT_PAGE_SIZE

5.4 画面 ID

✧ 「SCR-xxx」の形式

✧ 区分毎に番号をまとめる

- 100 番台 → 医療機関
- 200 番台 → 医療機関管理/出力
- 300 番台 → ユーザー管理
- 400 番台 → 問い合わせ
- 500 番台 → システム管理
- 600 番台 → インポート

例：

- SCR-101 医療機関検索
- SCR-301 ユーザー管理
- SCR-503 メンテナンス

5.5 ファイル・ディレクトリ

✧ バックエンド

区分	パス
クラス	class/
コントローラー	public/control/
API	lib/

✧ フロントエンド

区分	パス
ページ	public/view/
CSS	css/
JavaScript	Js/

5.6 ログ・監査

✧ 監査ログテーブル：security_logs

✧ カラム：

カラム名	詳細
log_id	A_I（自動採番）
user_id	ユーザーID
event_type	行った内容を保存（定型）
severity	重要度（low,medium,high,critical）
description	イベント詳細
ip_address	IP アドレス
user_agent	ユーザーエージェント
session_id	セッション ID
target_resource	対象リソース
failure_reason	失敗理由
additional_data	追加データ
created_at	イベント発生日時