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

## **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）

## **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

## データ構造

別途ファイル（ [DB構造](https://github.com/w3422008/update_DB/blob/main/%E4%BB%95%E6%A7%98%E6%9B%B8%E7%AD%89/%E6%96%B0%E5%8C%BB%E7%99%82%E6%A9%9F%E9%96%A2%E3%82%B7%E3%82%B9%E3%83%86%E3%83%A0/%E4%BB%95%E6%A7%98%E6%9B%B8%E7%AD%89/DB/DB%E6%A7%8B%E9%80%A0%E6%A1%88.md) ）へ記述

## 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**：新規登録/修正/削除/無効化などが対象。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、時刻など）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

## 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

* フロントエンド
* バックエンド

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 区分 | パス |  | 区分 | パス |
| クラス | class/ |  | ページ | public/view/ |
| コントローラー | public/control/ |  | CSS | css/ |
| API | lib/ |  | JavaScript | Js/ |

### 5.6 ログ・監査

* 監査ログテーブル：unified\_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 |
| log\_type | 行った内容を保存（定型） |
| description | イベント詳細 |
| ip\_address | IPアドレス |
| old\_values | 変更前データ |
| new\_values | 変更後データ |
| additional\_data | 追加データ（任意） |
| created\_at | ログ記録日時 |