

# 母子手帳デジタルアプリ 簡易設計書

2024/06/28

新井 秀平

# 目次

- ①要件・開発手法
- ②各設計(ページごとの一連の流れ) →課題1
- ③追加機能(新規登録時の各ユースケースを想定した機能) →課題2
- ④追加機能(医師免許情報サイトへのスクレイピングによるリアルタイム承認) →課題2

# ①要件・開発手法

## 要件

ある地方自治体より、母子健康手帳のデジタル化を進めるべく、プロトタイプの開発を依頼されました。

## 開発範囲

- ・省令様式より「手帳の受け取り」から「3~4か月児健康診査」まで(P.1-P.25)
- ・以下の内容は除外する
- ・便色の確認の記録(P.19)

# ①要件・開発手法

想定環境(以下の2つより選択)

- ・Ubuntu 22.04 LTS (オンプレミス環境の想定か)
- ・Webホスティングサービス

例:AWSによるSaaSモデル

<https://aws.amazon.com/jp/application-hosting/>

ただし、AWSではスケーラビリティによる料金発生が行政の予算を上回る可能性や機密情報取り扱いによる責任範囲も想定し、断念。

Ubuntu 22.04 LTS での構築とした。

# ①要件・開発手法

## 開発手法

MERNスタックの運用 (MongoDB, Express, React, Node.js)

フロント・バックエンドがJavaScriptで作成でき、知識習得の時間が短縮できる。

原則、一人開発であるため、一貫性がある方が良い。

大人数による多数のアクセスも想定されるため、MongoDBによるNoSQLを想定。

## ②各設計(DB,フロントエンド,バックエンド)

### ・フロントエンド、DB

ここからはスライド1-2枚につき1ページ分の情報を載せることとする。流れは以下となる。

#### ①新規登録ページ、ユーザー情報

#### ②ログインページ

#### ③行政・支援者ログイン時の保護者選択ページ

#### ④母子手帳目次

#### ⑤母子手帳各ページ(同様の機能となるため、今回はP23,P24の2ページで紹介)

## ②-①新規登録ページ、ユーザー情報

登録ボタン押下時に

ユーザー情報テーブルに

右の情報を登録する。

その際、IDは自動採番を行う。

DB向けユーザー情報テーブル

マイナンバー(主キー) : String型

ID : number型

氏名 : String型

生年月日 : Date型

住所 : String型

電話番号 : number型

メールアドレス : String型

ロール : String型

### 初回登録ページ

氏名:

生年月日:  年 /  月 /  日

住所:

電話番号:

メールアドレス:

マイナンバー:

☒ 保護者

☐ 行政

☐ 支援者

## ②-②ログインページ

メールアドレスとマイナンバーの下4桁を入力。

ログインボタン押下時、マイナンバーをハッシュ化した後、

ユーザー情報と照合し、一致するか確認する。

一致を確認できた場合、ユーザー情報よりユーザーIDとロールをWindows APIのローカルストレージへ埋め込まれる。

その後、行政・支援者は③の保護者選択ページへ、保護者は④の母子手帳目次ページへ遷移する。

初回登録ページへボタン押下時、①の新規登録ページへ遷移する。

### ログインページ

メールアドレス:

マイナンバーの下4桁:



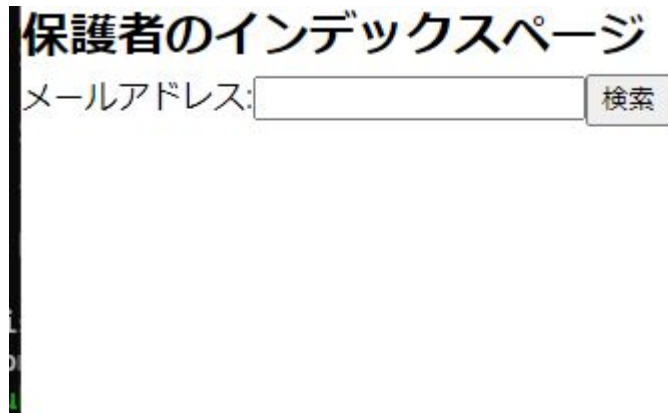
## ②-③行政・支援者ログイン時の保護者選択ページ

メールアドレスよりユーザー情報テーブルから  
保護者を検索し、取り出す機能。

ここから出力される保護者一覧には  
「メールアドレス、氏名、生年月日、電話番号」  
が表示される。

メールアドレスが同一の方が複数いた場合は  
表形式で複数表示される。

また氏名をクリックすることでローカルストレージに該当保護者のユーザーIDを  
埋め込み、④の母子手帳目次ページへ遷移する。



The screenshot shows a web interface titled '保護者のインデックスページ' (Guardian Index Page). Below the title is a search form with the label 'メールアドレス:' (Email Address:) followed by a text input field and a button labeled '検索' (Search). The page is partially obscured by a vertical black bar on the left side.

## ②-④母子手帳目次

項目をクリックすることで各ページへ遷移する。

### 母子手帳目次

- [表紙](#)
- [戸籍情報](#)
- [妊婦の健康状態等](#)
- [妊婦の職業と環境](#)
- [妊婦自身の記録\(1\)](#)
- [妊婦自身の記録\(2\)](#)
- [妊婦自身の記録\(3\)](#)
- [妊婦自身の記録\(4\)](#)
- [妊娠中の経過\(1\)](#)
- [妊娠中の経過\(2\)](#)
- [検査の記録](#)
- [予備欄\(医療機関が記入\)](#)
- [母親学級受講記録](#)
- [妊娠中と産後の歯の状態](#)
- [出産の状態](#)
- [出産後の母体の経過 母親自身の記録](#)
- [早期新生児期\(生後1週間以内\)の経過 後期新生児期\(生後1～4週\)の経過](#)
- [検査の記録 予備欄](#)
- [保護者の記録\(1か月頃\)](#)
- [1か月児健康診査](#)
- [保護者の記録\(3～4か月頃\)](#)
- [3～4か月児健康診査](#)

## ②-⑤母子手帳各ページ(P23 画面機能)

記録日:数値入力

首が座った日:数値入力

各設問:ラジオボタン

気持ち:文字入力

更新ボタン:押下時にDBのP23対応

テーブルへ更新を行う。

(テーブルにはユーザーID(主キー),記録日,首が座った日,各質問回答,気持ちが入る)

### 保護者の記録【3～4か月頃】

記録日: 2024 年 6 月 20 日

首が座った日: 6 月 18 日頃

あやすとよく笑いますか: ☐ はい ☒ いいえ

目つきや目の動きがおかしいのではないかと 気になりますか: ☐ はい ☒ いいえ

見えない方向から声をかけてみると、そちらの方を見ようとしていますか: ☐ はい ☒ いいえ

外気浴をしていますか: ☐ はい ☒ いいえ

子育てについて気軽に相談できる人はいますか: ☒ はい ☐ いいえ

子育てについて不安や困難を感じることはありますか: ☐ はい ☐ いいえ ☒ 何ともいえない

成長の様子、育児の心配、かかった病気、感想などを自由に記入しましょう。:

痔に困っている

更新

## ②-⑤母子手帳各ページ(P23 各処理、権限)

ここでは保護者の記録【3~4か月頃】の記録を行う。

よって編集権限は保護者と行政(管理者)のみとし、支援者は閲覧のみ可とする。

支援者は各項目の入力が不可となる。

処理1:ページ遷移時

ローカルストレージにあるユーザーIDをキーにP23対応テーブルへ検索を行う。

レコードが1件見つかった場合:レコードを基に自動記入を行う。

見つからなかった場合:DBへユーザーIDのみ記入したレコードを1件作成する。

処理2:更新ボタン押下時

レコードの更新を行う。

## ②-⑤母子手帳各ページ(P24 画面機能)

画面構成はP23に近く、  
ここでは割愛する。

### 3～4か月児健康診査

(年月日実施・か月日)

体重:g  
身長:cm  
胸囲:cm  
頭囲:cm

栄養状態:☐良☐要指導  
栄養法:☐母乳☐混合☐人工乳  
股関節開排制限:☐なし☐あり

健康状態:  
特記事項:  
施設名または担当者名:  
測定実施年月日:  
月齢:  
体重:  
身長:  
特記事項:  
施設名または担当者名:

## ②-⑤母子手帳各ページ(P24 各処理、権限)

ここでは3~4か月児健康診査の記録を行う。

よって編集権限は支援者と行政(管理者)のみとし、保護者は閲覧のみ可とする。

保護者は各項目の入力が不可となる。

処理1:ページ遷移時

ローカルストレージにあるユーザーIDを元にP24対応テーブルへ検索を行う。

レコードが1件見つかった場合:レコードを基に自動記入を行う。

見つからなかった場合:DBへユーザーIDのみ記入したレコードを1件作成する。

処理2:更新ボタン押下時

レコードの更新を行う。

## ②各設計(DB,フロントエンド,バックエンド)

### ・バックエンド

基本的にはMVCモデルに準拠し、Routesを経由することでフロント層とサーバー(Node.js)の疎通を図る。

<https://qiita.com/ksh-fthr/items/2daaaf3a15c4c11956e9>

ここではAxios.postを用いることでリクエストを送ることができる。

DBへはexpressフレームワークのRouterとMongoDBのライブラリmongooseを使用し、DBを操作する。また、DB操作を分けるためサービス層(DAO)も作る。

<https://qiita.com/takehilo/items/4b89f8ee432b0a06c691>

## ②各設計(DB,フロントエンド,バックエンド)

・バックエンドとフロントエンドの疎通例 (p23の更新ボタン押下時) 非同期処理

①フロントエンドのページよりバックエンドサーバーのapi/recordApi/updrecordP23へAxios.Postを行い、リクエストを送る。

②バックエンドのServer.jsにて名前解決を行い、/routes/recordRoutes/updrecordP23の命令を実行する。

③recordRoutesの/updrecordP23ではrouter.postを行い、コントローラー層のrecordcontroller.jsを経由してサービス層のrecordService.jsへ中継する。

④recordService.jsにてmongooseを通じてDB操作を行い、レスポンスを返す。

⑤recordcontroller.jsからフロントエンド層へステータスコードを送る。



### ③追加機能(新規登録時の各ユースケース)

今回はアプリケーションの実装となるため、保護者と支援者、行政がどうやり取りし、インシデントを防げるか想定する必要がある。

考えられるケースは以下とした。

- ①保護者が行政とやり取りし新規登録する場合
- ②医療従事者が新規登録する場合
- ③支援者と保護者が母子手帳を共有する場合
- ④急患時に地域ではない医療従事者が新規登録する場合

※今回の実習では追加できなかったが、50時間プラスであれば実装できた機能である。

### ③追加機能(新規登録時の各ユースケース)

保護者のインデックスページ

メールアドレス:

①保護者が行政とやり取りし新規登録する場合

目的:**なりすましの防止**、正常な運用を促すため

①:保護者が役所訪問時にアプリの導入指示と同時に書面にてマイナンバーと電話番号、メールアドレスを記入してもらう。

②保護者が新規登録したあと、ユーザー情報を承認待ち状態にする。

③行政が書面をすり合わせて、問題なければ、ユーザー情報を承認する。

④保護者が母子手帳が利用できるようになる。

新規実装するもの:**承認ボタン**(②-③行政・支援者ログイン時の保護者選択ページに行政がログインした時に保護者一覧に保護者ごとに表示する)

### ③追加機能(新規登録時の各ユースケース)

#### ②医療従事者が新規登録する場合

目的:**なりすましの防止**、正常な運用を促すため

①:行政より地域医療従事者へ周知し、登録を行ってもらう。

②保護者が新規登録したあと、ユーザー情報を承認待ち状態にする。

③行政が書面をすり合わせてユーザー情報を承認する。

④保護者が母子手帳が利用できるようになる。

新規実装するもの:**職業選択欄**,**医師(看護師)免許ID記入欄**(ユーザー新規登録ページにロールの項目を削除し、代わりに職業選択欄と各免許IDの項目を追加する)

#### 初回登録ページ

氏名:

生年月日:  年 /  月 /  日

住所:

電話番号:

メールアドレス:

マイナンバー:

☒ 保護者  
☐ 行政  
☐ 支援者

### ③追加機能(新規登録時の各ユースケース)

#### ③支援者(登録済みの医療機関)と保護者が母子手帳を共有する場合

必須要件にもつながってくるが保護者がメールアドレスを通知し、支援者が検索を行うこととなる。

基本的にはこれでいいが、**支援者ログイン時には一度受診した保護者は優先表示したい。**

従って、ユーザー情報に受診した医師、助産師等のユーザーIDを配列型で保存し、支援者ログイン時に一覧を出せるようにしたい。

新規実装するもの:DBのユーザー情報へ受診した支援者のユーザーIDを配列型で保存、

フロントエンドでは支援者のログイン時にユーザーIDを基に各ページのテーブルへ副呼び出し検索を行い、遷移後、受診した保護者氏名を優先表示する。

### ③追加機能(新規登録時の各ユースケース)

#### ④急患時に地域ではない医療従事者が新規登録する場合

前提として一番早い手順は**保護者のIDでログインすること**となる。

運用方法:ヘルプマークにメールアドレスとマイナンバーの下4桁を記載する。

緊急時の情報閲覧はこの運用で可能となる。

ただ、情報編集ができないため、夜間の支援者新規登録のために**医師免許情報サイトへのスクレイピングによるリアルタイム承認機能**を実装する必要がある。



#### ④追加機能(医師免許情報サイトへのスクレイピングによるリアルタイム承認)

目的:夜間運用も考慮したリアルタイムの支援者ユーザー新規登録

支援者の新規登録時のみに適用する。

急患時は保護者としてログインすることで情報閲覧が可能だが、編集ができなくなる。

そこで、**医師免許情報サイトへのスクレイピング**を行い、医師情報が一致すれば登録できるように機能を追加する。

例:JavaScriptによるスクレイピング参考資料

<https://qiita.com/centerfield77/items/425780a673568d3008e6>

#### ④追加機能(医師免許情報サイトへのスクレイピングによるリアルタイム承認)

新規実装機能:サーバー側に**夜間モード**を実装する。

運用方法:支援者(医師,看護師)が新規登録を行った時、サーバーが夜間モードならば、医師免許サイトへ**医師(看護師)免許IDを基にスクレイピング検索**を行う。

医師免許サイト(厚労省):[https://licenseif.mhlw.go.jp/search\\_isei/](https://licenseif.mhlw.go.jp/search_isei/)

実装課題

なりすましによる多数新規登録に弱い。→ブルートフォースアタック対策のように複数回のログインへはIPまたはユーザー情報を基にスリープまたはロックを掛ける。