

# web プログラミングレポート課題

25G1051 近藤巧望

2025 年 12 月 28 日

## 目次

1	はじめに	2
第 I 部 利用者用仕様書		2
1	システム概要	2
2	目的	2
3	主な機能	2
4	利用方法	3
4.1	星 5 キャラクター一覧表示	3
4.2	星 4・5 武器一覧表示	4
4.3	聖遺物一覧表示	5
第 II 部 管理者用仕様書		5
1	システムの概要と目的	6
2	システムの流れ	6
3	運用方法	7
4	エラー処理	7
第 III 部 開発者用仕様書		7
1	システム概要	8
2	目的と機能	8
3	HTTP メソッドとリソース名一覧	8
3.1	星 5 キャラクター所持数一覧	8
3.2	星 4・5 武器所持数一覧	8
3.3	スコア 40 以上の聖遺物一覧	9

4	データ構造	9
4.1	星5 キャラクターデータ . . . . .	9
4.2	星4・5 武器データ . . . . .	9
4.3	聖遺物データ . . . . .	10
5	ファイルごとの役割	11
5.1	星5 キャラクター一覧表示システム . . . . .	11
5.2	星4・5 武器一覧表示システム . . . . .	11
5.3	聖遺物一覧表示システム . . . . .	11
6	ページ遷移について	11
7	授業外から取り入れた技術	12

# 1 はじめに

本システムは、ゲーム「原神」における「星 5 キャラクター所持数一覧」「星 4・5 武器所持数一覧」「スコア 40 以上の聖遺物一覧」をそれぞれ管理するためのシステムである。各項目の新規登録、詳細表示、編集、および削除機能を提供する。特に、聖遺物一覧のシステムに関しては、聖遺物セットごとに部位を管理し、セット効果の発動状況を考慮した上で、聖遺物スコアが 40 以上のものを一覧表示する機能を備えている。そのため、本仕様書では、聖遺物セットの管理機能と所持聖遺物の管理機能を分けて記載する。github リポジトリは以下の URL にて公開している。

- [https://github.com/kondou33/webpro\\_06](https://github.com/kondou33/webpro_06)

## 第 I 部

### 利用者用仕様書

#### 1 システム概要

本システムは、ゲーム「原神」における以下の 3 つの要素を管理するためのものである。

- 星 5 キャラクター所持数一覧
- 星 4・5 武器所持数一覧
- スコア 40 以上の聖遺物一覧

各要素に対して、新規登録、詳細表示、編集、および削除の機能を提供する。聖遺物一覧に関しては、聖遺物セットごとに部位を管理し、セット効果の発動状況を考慮した上で、聖遺物スコアが 40 以上のものを一覧表示する機能を備えている。

#### 2 目的

本システムの目的は、ユーザーが自身の所持する星 5 キャラクター、星 4・5 武器、および聖遺物を効率的に管理できるようにすることである。星 5 キャラクターの管理は、キャラクターの育成状況や役割を把握するために役立ち、星 4・5 武器の管理は、装備の最適化や戦略の立案に寄与する。武器はレアリティが高いものほど強力ではあるが、キャラクターとの相性も重要であることや、星 4 武器でも優秀なものが存在することから、星 4・5 武器の両方を管理できるようにしている。聖遺物の管理に関しては、セット効果の発動状況を考慮することで、より効果的なビルト構築を支援する。これらの管理機能を通じて、ユーザーはゲーム内の戦力強化や戦略的なプレイを促進できることを目的としている。

#### 3 主な機能

本システムは、以下の主要な機能を提供する。

- 星 5 キャラクター所持数一覧の管理
  - 新規登録
  - 詳細表示
  - 編集

- 削除
- 星 4・5 武器所持数一覧の管理
  - 新規登録
  - 詳細表示
  - 編集
  - 削除
- スコア 40 以上の聖遺物一覧の管理
  - 聖遺物セットの管理
    - \* セット名称の新規登録
    - \* セット名称の削除
  - 所持聖遺物の管理
    - \* 新規登録
    - \* 詳細表示
    - \* 編集
    - \* 削除

## 4 利用方法

### 4.1 星 5 キャラクター一覧表示

まずは、次の図 1 のように、google Chrome などのウェブブラウザで `http://localhost:8080/chara` にアクセスすることで、星 5 キャラクター一覧表示画面が表示される。



図 1 Chrome での検索例

アクセスすると、図 2 のような一覧表示の画面が表示される。画面下部の「追加」ボタンをクリックすると、新規登録画面へ遷移する。また、各キャラクター名のリンクをクリックすると、詳細表示画面へ遷移する。詳細画面では、編集ボタンをクリックすると編集画面へ遷移し、削除ボタンをクリックすると削除処理が実行される。

← → ⌛ ⓘ localhost:8080/chara

□

### キャラクターの詳細一覧

キャラクターID	キャラクター名	元素属性	武器種別	命の星座凸数
1	刻晴	雷	片手剣	1
2	七七	氷	片手剣	3
3	フリーナ	水	片手剣	0
4	ディルック	炎	両手剣	2
5	シグウィン	水	弓	6

[追加](#)

図 2 星 5 キャラクター一覧表示画面

## 4.2 星 4・5 武器一覧表示

星 4・5 武器一覧表示画面へのアクセス方法および操作方法は、星 5 キャラクター一覧表示画面と同様である。<http://localhost:8080/weapon> にアクセスすると、図 3 のような一覧表示の画面が表示される。

← → ⌛ ⓘ localhost:8080/weapon

□

### 武器の詳細

武器ID	武器名	武器タイプ	レアリティ
1	天空の刃	片手剣	5
2	アモスの弓	弓	5
3	白雨心弦	弓	5

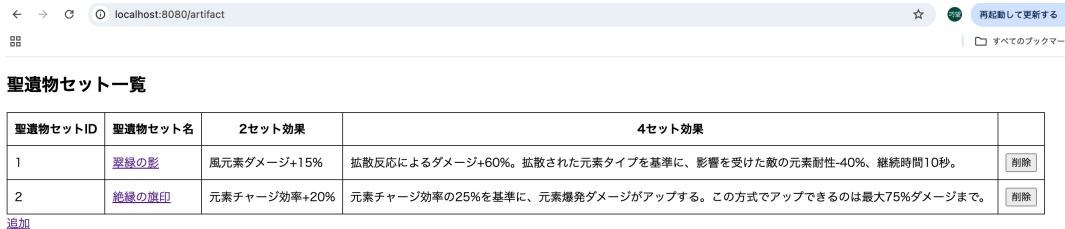
[追加](#)

図 3 星 4・5 武器一覧表示画面

使用方法も同様であり、画面下部の「追加」ボタンをクリックすると新規登録画面へ遷移し、各武器名のリンクをクリックすると詳細表示画面へ遷移する。詳細画面では、編集ボタンをクリックすると編集画面へ遷移し、削除ボタンをクリックすると削除処理が実行される。

#### 4.3 聖遺物一覧表示

聖遺物一覧表示画面へのアクセス方法も、星5キャラクター一覧表示画面と同様である。<http://localhost:8080/artifact>にアクセスすると、図4のような一覧表示の画面が表示される。

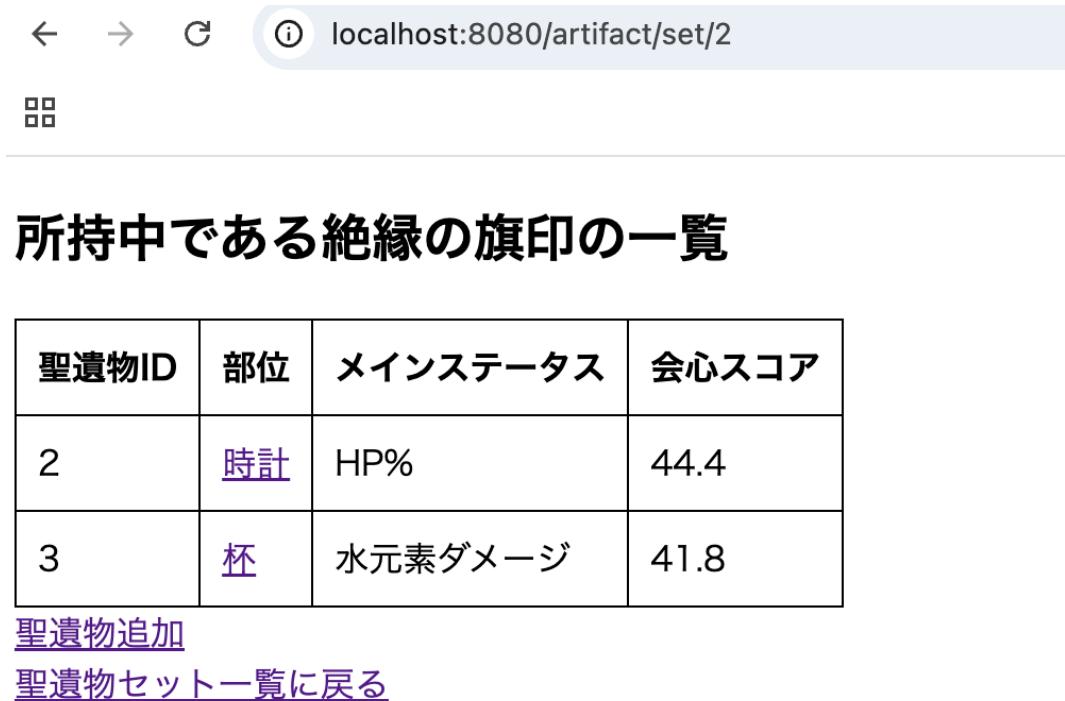


聖遺物セットID	聖遺物セット名	2セット効果	4セット効果
1	<a href="#">縁縁の影</a>	風元素ダメージ+15%	拡散反応によるダメージ+60%。拡散された元素タイプを基準に、影響を受けた敵の元素耐性-40%、継続時間10秒。
2	<a href="#">縁縁の旗印</a>	元素チャージ効率+20%	元素チャージ効率の25%を基準に、元素爆発ダメージがアップする。この方式でアップできるのは最大75%ダメージまで。

[追加](#)

図4 聖遺物一覧表示画面

図4の聖遺物一覧表示は、聖遺物セットごとに部位を管理しているため、星5キャラクター一覧表示画面や星4・5武器一覧表示画面とは構造が異なり、操作方法も異なる。セット一覧ページの画面下部にある「追加」ボタンをクリックすると、聖遺物セット名称の新規登録画面へ遷移する。各セット名のリンクをクリックすると、そのセットに所属する聖遺物部位一覧画面へ遷移する(図5)。



聖遺物ID	部位	メインステータス	会心スコア
2	<a href="#">時計</a>	HP%	44.4
3	<a href="#">杯</a>	水元素ダメージ	41.8

[聖遺物追加](#)

[聖遺物セット一覧に戻る](#)

図5 聖遺物部位一覧画面

部位一覧画面の画面下部にある「追加」ボタンをクリックすると、聖遺物部位の新規登録画面へ遷移する。各部位名のリンクをクリックすると、詳細表示画面へ遷移する。詳細画面では、編集ボタンをクリックすると編集画面へ遷移し、削除ボタンをクリックすると削除処理が実行される。

# 第 II 部

## 管理者用仕様書

### 1 システムの概要と目的

本システムは、Node.js および Express フレームワークを用いた、原神のキャラクター・武器・聖遺物データの管理ツールである。管理者がローカル環境またはサーバー上で本プログラムを実行することで、ユーザーに対し Web ブラウザ経由でのデータ登録・閲覧・編集・削除（CRUD 機能）を提供する。主な目的は、複雑な聖遺物のステータスおよび会心スコアを一元管理し、育成の効率化を図ることにある。

### 2 システムの流れ

各システムのページ遷移は以下の通りである。

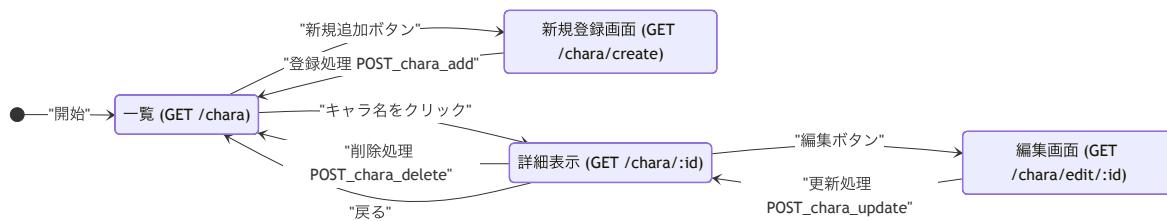


図 6 星 5 キャラクター一覧表示のページ遷移図

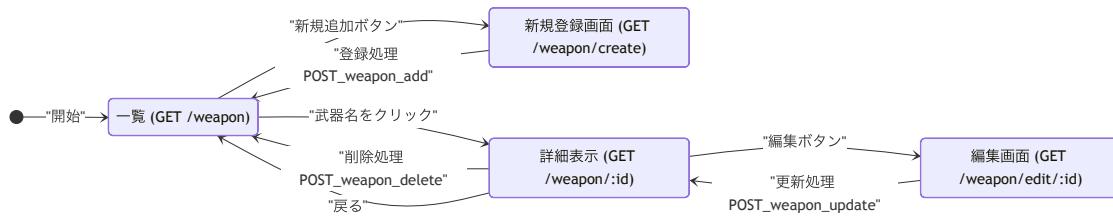


図 7 星 4・5 武器一覧表示のページ遷移図

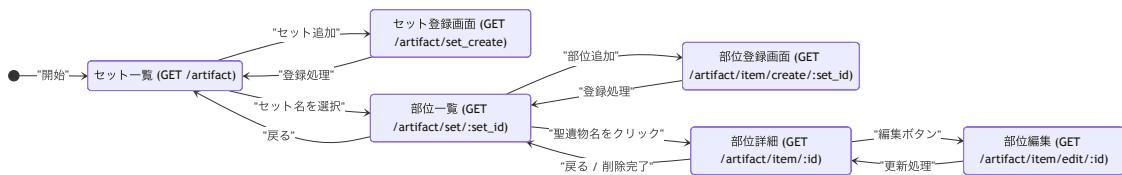


図 8 聖遺物一覧表示のページ遷移図

ページ遷移について、各システム共通の仕様として、最初の一覧表示画面から各詳細画面へ遷移し、詳細画面から編集画面へ遷移する形と、一覧表示から新規登録画面へ遷移する形を採用している。星5キャラクター一覧表示および星4・5武器一覧表示では、まず一覧表示画面から詳細画面へ遷移し、詳細画面から編集画面へ遷移する形を採用している。一方、聖遺物一覧表示では、聖遺物セットごとに部位を管理しているため、一覧表示画面からセットの部位一覧画面へ遷移し、部位一覧画面から詳細画面へ遷移する形を採用している。

### 3 運用方法

本システムを運用するには、Node.js および Express フレームワークがインストールされた環境が必要である。更に、冒頭で添付した GitHub リポジトリから `webpro_06` フォルダをクローンまたはダウンロードし、ターミナルで `webpro_06` フォルダに移動した後、以下のコマンドを実行して必要なパッケージをインストールする。

```
npm install
```

依存パッケージのインストールが完了したら、以下のコマンドを実行してアプリケーションを起動する。

```
node app5.js
```

アプリケーションが起動したら、Web ブラウザで `http://localhost:8080/chara` にアクセスすることで、星5キャラクター一覧表示画面が表示される。同様に、星4・5武器一覧表示画面は `http://localhost:8080/weapon`、聖遺物一覧表示画面は `http://localhost:8080/artifact` にアクセスすることで表示される。

### 4 エラー処理

システムが予期しない動作をした際や、不適切な操作が行われた場合は以下の挙動をとる。

- **存在しないデータの参照**: 削除済みのデータや存在しない ID にアクセスしようとした場合、システムは自動的に「一覧画面」へリダイレクトを行う。これにより、画面が真っ白になったりシステムが停止したりすることを防いでいる。
- **入力不備の防止**: 登録フォームでは、必須項目の入力漏れや数値以外の入力に対して制限を設けている。不適切な状態で送信ボタンを押した場合、ブラウザ側で警告が表示され、正しく入力されるまで登録処理は行われない。

# 第 III 部

## 開発者用仕様書

### 1 システム概要

本システムは、Node.js および Express フレームワークを用いた、原神のキャラクター・武器・聖遺物データの管理ツールである。管理者がローカル環境またはサーバー上で本プログラムを実行することで、ユーザーに対し Web ブラウザ経由でのデータ登録・閲覧・編集・削除（CRUD 機能）を提供する。主な目的は、複雑な聖遺物のステータスおよび会心スコアを一元管理し、育成の効率化を図ることにある。

### 2 目的と機能

本システムの目的は、ユーザーが自身の所持する星 5 キャラクター、星 4・5 武器、および聖遺物を効率的に管理できるようにすることである。星 5 キャラクターの管理は、キャラクターの育成状況や役割を把握するために役立ち、星 4・5 武器の管理は、装備の最適化や戦略の立案に寄与する。武器はレアリティが高いものほど強力ではあるが、キャラクターとの相性も重要であることや、星 4 武器でも優秀なものが存在することから、星 4・5 武器の両方を管理できるようにしている。聖遺物の管理に関しては、セット効果の発動状況を考慮することで、より効果的なビルト構築を支援する。これらの管理機能を通じて、ユーザーはゲーム内での戦力強化や戦略的なプレイを促進できることを目的としている。

### 3 HTTP メソッドとリソース名一覧

#### 3.1 星 5 キャラクター所持数一覧

メソッド	リソース名	機能概要
GET	/chara	星 5 キャラクター一覧表示
GET	/chara/create	新規登録フォーム表示
POST	/chara_add	新規追加処理
GET	/chara/:id	詳細表示
GET	/chara/edit/:id	編集フォーム表示
POST	/chara_update/:id	編集処理
POST	/chara_delete/:id	削除処理

#### 3.2 星 4・5 武器所持数一覧

メソッド	リソース名	機能概要
GET	/weapon	星 4・5 武器一覧表示
GET	/weapon/create	新規登録フォーム表示
POST	/weapon_add	新規追加処理
GET	/weapon/:id	詳細表示
GET	/weapon/edit/:id	編集フォーム表示
POST	/weapon_update/:id	編集処理
POST	/weapon_delete/:id	削除処理

### 3.3 スコア 40 以上の聖遺物一覧

#### 3.3.1 聖遺物セットの管理

メソッド	リソース名	機能概要
GET	/artifact	聖遺物セット一覧表示
GET	/artifact/set_create	セット名称入力画面表示（静的 HTML への遷移）
POST	/artifact/set_add	セット追加処理
POST	/artifact/set_delete/:set_id	セット削除処理

#### 3.3.2 所持聖遺物の管理

メソッド	リソース名	機能概要
GET	/artifact/set/:set_id	指定したセットの部位一覧表示
GET	/artifact/item_create/:set_id	部位追加画面表示（ID のバケツリレー）
POST	/artifact/item_add/:set_id	部位追加処理
GET	/artifact/item/:id	部位詳細表示
GET	/artifact/item/edit/:id	部位編集画面表示
POST	/artifact/item_update/:id	部位更新処理
POST	/artifact/item_delete/:id	部位削除処理

## 4 データ構造

本システムは「管理番号」および「名称」をキーとして管理を行う。詳細画面では以下のデータ項目を保持する。

### 4.1 星 5 キャラクターデータ

- chara\_id (数値) : 管理番号
- chara\_name (文字列) : キャラ名
- element (文字列) : 元素
- limited\_chara (文字列) : 限定/恒常の区分
- levelbonus\_name (文字列) : 突破ボーナス項目名
- levelbonus\_value (数値) : 突破ボーナス値
- weapon\_type (文字列) : 武器種
- level (数値) : キャラレベル
- nomal\_talent\_level (数値) : 通常攻撃天賦レベル
- skill\_talent\_level (数値) : 元素スキル天賦レベル
- burst\_talent\_level (数値) : 元素爆発天賦レベル
- role (文字列) : 役割（アタッカー/サポート等）
- totu (数値) : 命の星座凸数

### 4.2 星 4・5 武器データ

- weapon\_id (数値) : 管理番号
- weapon\_name (文字列) : 武器名

- `weapon_type` (文字列)：武器種
- `limited` (文字列)：限定/恒常の区分
- `rank_level` (数値)：レアリティ (4 または 5)
- `level` (数値)：レベル
- `base_attack` (数値)：基礎攻撃力
- `sub_stat` (文字列)：サブステータス項目名
- `sub_stat_value` (数値)：サブステータス値
- `passive_name` (文字列)：武器効果説明文
- `refinement_level` (数値)：精錬ランク (1–5)
- `motif` (文字列)：モチーフ対象キャラ名

### 4.3 聖遺物データ

聖遺物については、聖遺物セットデータと所持聖遺物データの 2 つに分けて管理する。

#### 4.3.1 聖遺物セットデータ

- `set_id` (数値)：セット ID (紐付け用)
- `set_name` (文字列)：セット名 (例：黄金の劇団)
- `set2_effect` (文字列)：2 セット効果説明
- `set4_effect` (文字列)：4 セット効果説明

#### 4.3.2 所持聖遺物データ

- `artifact_id` (数値)：管理番号
- `set_id` (数値)：所属するセットの ID
- `slot` (文字列)：部位 (花・羽・砂時計・杯・冠)
- `main_stat` (文字列)：メインステータス項目
- `main_stat_value` (数値)：メインステータス値
- `sub_stat1` (文字列)：サブステータス 1 項目
- `sub_stat1_value` (数値)：サブステータス 1 値
- `sub_stat2` (文字列)：サブステータス 2 項目
- `sub_stat2_value` (数値)：サブステータス 2 値
- `sub_stat3` (文字列)：サブステータス 3 項目
- `sub_stat3_value` (数値)：サブステータス 3 値
- `sub_stat4` (文字列)：サブステータス 4 項目
- `sub_stat4_value` (数値)：サブステータス 4 値
- `score` (数値)：聖遺物スコア。計算式は式 (1) 参照。

聖遺物スコアとは、キャラクターの火力向上に寄与するステータスを数値化したものであり、以下の式で算出される。聖遺物を使用するキャラクターのビルドに応じて、重視するステータスの重み付けが異なるため、攻撃力、元素チャージ効率、HP、防御力などのステータスは考慮せず、本システムでは会心率と会心ダメージに着目し、会心スコアとして定義している。

$$\text{Score} = (\text{会心率} \times 2) + (\text{会心ダメージ} \times 1) \quad (1)$$

## 5 ファイルごとの役割

本システムは、ロジックを担う `app5.js` と、各リソースの描画を担う EJS テンプレートファイル群および HTML ファイル群で構成されている。後述の表にて、各ファイルの役割を示す。

### 5.1 星 5 キャラクター一覧表示システム

ファイル名	役割
<code>app5.js</code>	メインロジック（ルーティング、CRUD 処理）
<code>chara.ejs</code>	星 5 キャラクター一覧表示画面テンプレート
<code>chara_new.html</code>	星 5 キャラクター新規登録画面テンプレート
<code>chara_detail.ejs</code>	星 5 キャラクター詳細表示画面テンプレート
<code>chara_edit.ejs</code>	星 5 キャラクター編集画面テンプレート

表 1 星 5 キャラクター一覧表示システムのファイル構成

### 5.2 星 4・5 武器一覧表示システム

ファイル名	役割
<code>app5.js</code>	メインロジック（ルーティング、CRUD 処理）
<code>weapon.ejs</code>	星 4・5 武器一覧表示画面テンプレート
<code>weapon_new.html</code>	星 4・5 武器新規登録画面テンプレート
<code>weapon_detail.ejs</code>	星 4・5 武器詳細表示画面テンプレート
<code>weapon_edit.ejs</code>	星 4・5 武器編集画面テンプレート

表 2 星 4・5 武器一覧表示システムのファイル構成

### 5.3 聖遺物一覧表示システム

ファイル名	役割
<code>app5.js</code>	メインロジック（ルーティング、CRUD 処理）
<code>artifact.ejs</code>	聖遺物セット一覧表示画面テンプレート
<code>artifact_set_new.html</code>	聖遺物セット名称新規登録画面テンプレート
<code>artifact_set.ejs</code>	聖遺物セット部位一覧画面テンプレート
<code>artifact_item_new.ejs</code>	聖遺物部位新規登録画面テンプレート
<code>artifact_item_detail.ejs</code>	聖遺物部位詳細表示画面テンプレート
<code>artifact_item_edit.ejs</code>	聖遺物部位編集画面テンプレート

表 3 聖遺物一覧表示システムのファイル構成

## 6 ページ遷移について

各システムのページ遷移は以下の通りである。

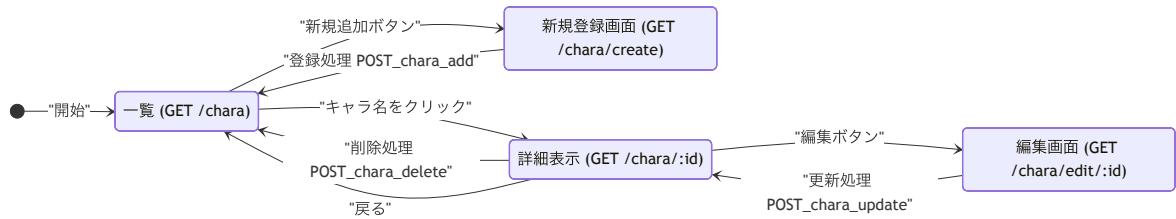


図 9 星 5 キャラクター一覧表示のページ遷移図

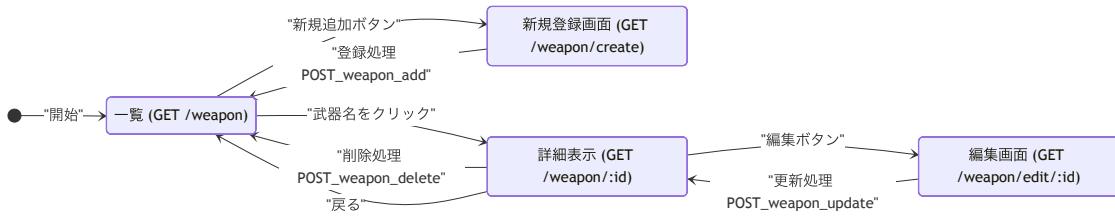


図 10 星 4・5 武器一覧表示のページ遷移図

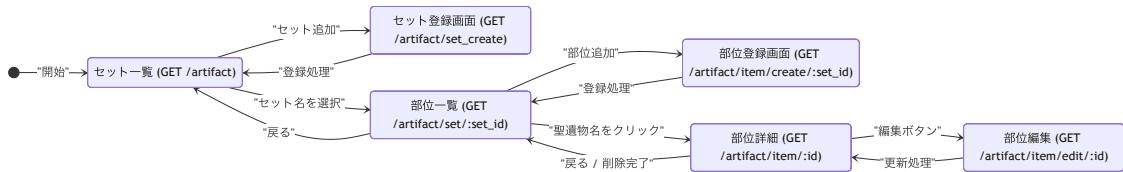


図 11 聖遺物一覧表示のページ遷移図

ページ遷移について、各システム共通の仕様として、最初の一覧表示画面から各詳細画面へ遷移し、詳細画面から編集画面へ遷移する形と、一覧表示から新規登録画面へ遷移する形を採用している。星 5 キャラクター一覧表示および星 4・5 武器一覧表示では、まず一覧表示画面から詳細画面へ遷移し、詳細画面から編集画面へ遷移する形を採用している。一方、聖遺物一覧表示では、聖遺物セットごとに部位を管理しているため、一覧表示画面からセットの部位一覧画面へ遷移し、部位一覧画面から詳細画面へ遷移する形を採用している。

## 7 授業外から取り入れた技術

新規データの登録時に、既存データと重複しない一意の `chara_id` を発行するため、次のアルゴリズムを実装した。

```
const id = character.length > 0 ? Math.max(...character.map(c => c.chara_id)) + 1 : 1;
```

このコードは、まず既存のキャラクターデータ配列 `character` の中から、各要素の `chara_id` を抽出し、最大値を求める。その後、最大値に 1 を加えることで、新しいキャラクターの ID を生成する。もし、既存データが存在しない場合は、初期値として 1 を割り当てる。この方法により、常に一意の ID が生成され、データの整合性が保たれる。

また、URL パラメータ等で指定された管理番号に基づき、配列内から单一のオブジェクトを取得する処理には `find` メソッドを用いている。例えば、星5キャラクターの詳細表示画面において、指定された ID に対応するキャラクターデータを取得するコードは以下の通りである。

```
const detail = character.find(chara => chara.chara_id == id);
```

このコードは、配列 `character` の中から、条件に一致する最初の要素を返す。ここでは、各キャラクターオブジェクトの `chara_id` が指定された ID と等しい場合に一致とみなしている。この手法は、配列のインデックス番号に依存せず、論理的な ID に基づいてリソースを確実に特定できる利点がある。

削除処理については、指定された管理番号に基づき、配列内から該当オブジェクトを除外するために `filter` メソッドを使用している。これは、元の配列を直接変更するのではなく、新しい配列を生成することで、指定された ID に一致しない要素のみを保持する方法である。

```
character = character.filter(chara => chara.chara_id != id);
```

このコードは、配列 `character` の中から、条件に一致しない要素を抽出し、新しい配列として再代入している。ここでは、各キャラクターオブジェクトの `chara_id` が指定された ID と等しくない場合にのみ、新しい配列に含める。この方法により、指定された ID のキャラクターが配列から削除される。