2025 年度後期 Web プログラミング期末課題仕様書

25G1064 佐土駿

2025 年 12 月 28 日

1　はじめに

本稿は，作成した 3 つの web アプリケーションに関連する仕様書を，「開発者向け仕様書」，「管理者向け仕様書」，「利用者向け仕様書」の 3 つに分けて記述する．ソースコードは GitHub 上に公開しており，以下の URL からアクセスできる．

`https://github.com/nazonomikan/webpro_06.git`

2　開発者向け仕様書

本節では，開発者向けに各 web アプリケーションのシステム構成，使用技術，ディレクトリ構成，主要なソースコードについて説明する．

2.1　app1:kimatu_saki

2.1.1　システム構成

本アプリケーションは，Node.js，Express を用いたサーバアプリケーションであり，テンプレートに EJS，静的資産に'public/' を使用する．サーバ実装は'app5.js' にまとめており，ビューは'views/'，静的ファイルは'public/' に配置している．サーバ起動は 'npm install'で依存をインストール後，'node app5.js'で行うことでできる．

2.1.2　目的

本アプリケーションの目的は，「プロジェクトセカイ カラフルステージ！ feat. 初音ミク」に登場するキャラクターのうちの一人である「天馬咲希」の所持カードを閲覧・管理し，さらに，実際のゲーム内での編成の参考にすることができるようにすることである．

2.1.3　機能詳細

本アプリケーションの機能の詳細について説明する．本アプリケーションでは，「プロジェクトセカイ カラフルステージ！ feat. 初音ミク」に登場するキャラクターのうちの一人である「天馬咲希」の所持カードの一覧表示，詳細表示，追加，編集，削除機能を使用できる．さらに，編成画面にて，所持しているカードの中から最大 5 枚を選択し，編成の総合力を計算することができる．

2.1.4　データ構造

app1 では，カードデータを’app5.js’ 内のメモリ上の配列 ‘saki‘で管理している．代表的なフィールドは以下の表 1 の通りである．

表 1: ’saki’ 配列の代表フィールド

項目名	詳細説明
id	カードの識別子，内部管理をするために使用している（数値）
name	カード名（文字列）
type	カードのタイプ（キュート/クール/ミステリアス/ハッピー/ピュア）
rarity	レアリティ（数値）（1-4）
skill_name	スキル名（文字列）
skill_level	スキルレベル（数値）（1-4）
rank	カードのマスターランク（数値）（0-5）
total_status	カードの初期総合力（数値）

2.1.5　ページ構造と HTTP メソッドとリソース名

本アプリケーションのページ構造，また，対応するメソッドについて図 1 を用いて説明する．

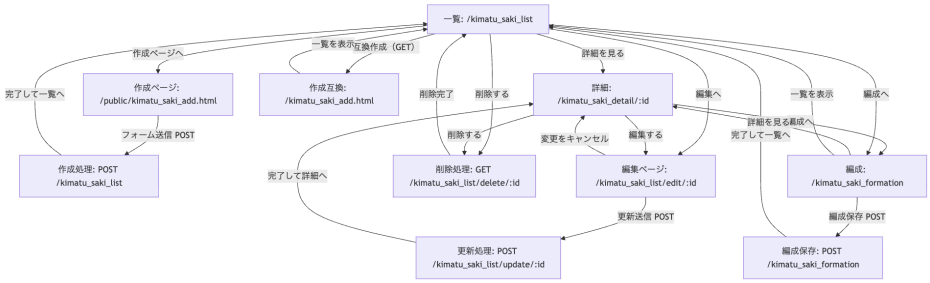

図 1: app1:kimatu_saki のページ構造

本アプリケーションの起点となるホームページは`/kimatu_saki_list` であり，ここから各機能へ遷移する．一覧表示からは，詳細表示，編集，削除，追加フォーム，編成画面へ遷移できる．詳細表示からは，編集，削除，編成画面へ遷移できる．編集ページからは，更新処理を経て，編集内容を確認するために詳細表示へ遷移する．削除，追加フォームからは，それぞれ削除処理，追加処

理を経て一覧表示へ遷移する．編成画面からは，選択したカードの編成を保存し，一覧表示または詳細表示へ遷移できる．また，使用しているファイル名と詳細について表 2 に示す．

表 2: app1:kimatu_saki の主要ファイル

ファイル名	説明
app5.js	サーバアプリケーション本体．saki 配列を管理し，ルーティングとロジックを実装している．
views/kimatu_saki_list.ejs	一覧表示ページのテンプレート．カード名，タイプが表示され，データ編集・削除ページ，編成ページへ行くことができる．
views/kimatu_saki_detail.ejs	詳細表示ページのテンプレート．一覧表示から選択されたカードのレアリティなどの詳細を表示する．また，データ編集，削除ページへ行くことができる．
views/kimatu_saki_edit.ejs	編集ページのテンプレート．id に対応するカードの編集フォームを表示する．
views/kimatu_saki_formation.ejs	編成画面のテンプレート．編成中のカードを選択し，総合力を計算する機能を提供する．
public/kimatu_saki_add.html	追加フォームの静的 HTML．カード追加用のフォームを提供する．
public/kimatu_saki.css	カスタム CSS ファイル．本システムにおける見た目の改善のために独自のスタイルを作成し，レスポンシブ対応の簡易実装.

また，主要なエンドポイントについて表 3 に示す．

表 3: app1:kimatu_saki の主要エンドポイント

HTTP メソッド	リソース名	説明
GET	/kimatu_saki_list	一覧表示
GET	/kimatu_saki_detail/:id	詳細表示
POST	/kimatu_saki_list	追加処理
GET	/kimatu_saki_list/edit/:id	編集ページ表示
POST	/kimatu_saki_list/update/:id	更新処理
GET	/kimatu_saki_list/delete/:id	削除処理
GET/POST	/kimatu_saki_formation	編成画面表示/保存

2.1.6　リソース名ごとの機能の詳細

各エンドポイントの機能の詳細について説明する．主なエンドポイントの機能は，表 4 の通りである．

表 4: app1:kimatu_saki のリソース名ごとの機能詳細

リソース名	機能詳細
/kimatu_saki_list	saki 配列を EJS に渡して一覧を描画する．
/kimatu_saki_detail/:id	指定 id のオブジェクトを検索して詳細を表示.
/kimatu_saki_list (POST)	フォームから送られた値を受け取り，メモリ配列を更新する（DB は未使用）.
/kimatu_saki_list/edit/:id	指定 id のオブジェクトを検索して編集フォームを表示.
/kimatu_saki_list/update/:id	フォームから送られた値を受け取り，メモリ配列を更新する（DB は未使用）.
/kimatu_saki_list/delete/:id	指定 id のオブジェクトを検索してメモリ配列から削除する（DB は未使用）.
/kimatu_saki_formation	編成中のキャラの total_status に対してランク補正（rank * 600）とタイプボーナス，および定数の増幅（コード中に定義）を適用して計算し，合計値を表示する．

2.1.7　授業外使用技術について

本節では，授業で扱わなかったが本アプリケーションで採用した技術とその採用理由について説明する．

総合力計算方法　編成画面で選択された各カードの `total_status` に対して，カードの `rank` に基づくランク補正（コード中では実際の増加量を参考に `rank * 600` としている），タイプごとのボーナス，および定数による増幅を順次適用して合計値を算出する手法を採用した．理由は，実機に近い補正ルールを再現することでユーザーが現実的な編成評価を行えるようにするためである．計算は主にサーバ側で一貫して行い，表示はクライアント側で受け取ってレンダリングすることで，サーバの整合性とクライアントの応答性を両立させている．

編成画面　編成画面は最大 5 枚までのカードをチェックボックスで選択し，選択状態をクライアント側で一時保持してからサーバへ送信・保存する仕組みとしている．編成の保存，総合力計算のトリガーはフォーム送信／非同期通信いずれにも対応させており，ユーザーが一覧画面を離れずに操作を完了できるようにした．採用理由は，詳細画面に遷移して戻る従来のフローだとスクロール位置や選択状態が失われる可能性があるためであり，選択の即時反映と一覧の継続的な参照を可能にすることで操作性を向上させるためである．

CSS(見た目・UI)　本節では，実際にビューで使用されている ‘public/kimatu_saki.css‘ の主要ルールを行単位で示し，目的を表 5 に記述する．

表 5: app1: 使用している CSS（行ごとの詳細）

CSS（行）	説明
`body.blue-theme { font-family: -apple-system, "Helvetica Neue", Arial, sans-serif; }`	システムフォント優先の指定．レンダリング速度と OS に自然な見た目を確保するため．
`body.blue-theme { background: linear-gradient(180deg, #e6f0ff, #f8fbff); }`	ページ全体の背景グラデーション．視認性と統一感の向上．
`body.blue-theme { color: #012a4a; }`	主要なテキスト色の指定．背景とのコントラスト調整．
`body.blue-theme { padding: 24px; }`	ページ内余白．コンテナとブラウザ端の間隔を確保．
`.container { max-width: 960px; }`	コンテンツ幅の上限を設定し読みやすさを保つ．
`.container { margin: 0 auto; }`	横中央寄せ．
`.container { background: #ffffff; }`	コンテンツ領域の背景色．カード風の白地を作る．
`.container { border-radius: 10px; }`	角丸で柔らかい見た目に．
`.container { box-shadow: 0 6px 18px rgba(2, 6, 23, 0.06); }`	浮き上がりを演出し視覚的な階層を付与．
`.container { overflow: hidden; }`	子要素のはみ出しを抑制して角丸等と合わせる．
`.container { padding: 18px; }`	コンテンツ内余白．
`.page-header { display: flex; }`	ヘッダ内をフレックスで整列．横並びレイアウトに使用．
`.page-header { justify-content: space-between; }`	左右に要素を分散配置．
`.page-header { align-items: center; }`	垂直方向の中央揃え．
`.page-header { gap: 12px; }`	要素間の隙間を確保．
`.page-header { margin-bottom: 12px; }`	下部の余白．
`.page-header h2 { margin: 0; }`	見出しの余白リセット．
`.page-header h2 { color: #013a63; }`	見出しの色指定．

CSS（行）	説明
`.page-header h2 { font-size: 1.25rem; }`	見出しの文字サイズ.
`.saki-table { width: 100%; }`	テーブルを横幅に合わせる.
`.saki-table { border-collapse: collapse; }`	隣接するセルの境界を重ねて表示.
`.saki-table { margin-top: 6px; }`	テーブル上部の余白.
`.saki-table th, .saki-table td { padding: 12px 10px; }`	セル内余白．可読性向上のため広めに取る.
`.saki-table th, .saki-table td { text-align: left; }`	テキスト左揃え.
`.saki-table th, .saki-table td { border-bottom: 1px solid #e6eef8; }`	セル間の区切り線．行の区別を明確化.
`.saki-table th { background: linear-gradient(90deg, #d9eefb, #c6e6ff); }`	ヘッダ行の背景グラデーション．視認性向上.
`.saki-table th { color: #003a63; }`	ヘッダのテキスト色.
`.saki-table th { font-weight: 600; }`	ヘッダの太字指定.
`.saki-table tr:nth-child(even) td { background: #f4f9ff; }`	偶数行の背景．帯状で読みやすくするため.
`.saki-table tr:hover td { background: #e8f3ff; }`	行ホバー時の背景色変更で強調.
`.saki-table tr:hover td { transition: background .12s; }`	ホバー背景の遷移速度を指定．滑らかな変化.
`.saki-table a { color: #0b66b3; }`	テーブル内リンクの色指定.
`.saki-table a { text-decoration: none; }`	リンク下線を外す.
`.saki-table a:hover { color: #024b85; }`	リンクホバー時の色.
`.saki-table a:hover { text-decoration: underline; }`	ホバー時に下線を表示しリンク性を示す.
`.footer-actions { display: flex; }`	フッタのアクション群を横並びに.
`.footer-actions { gap: 10px; }`	ボタン間隔を確保.
`.footer-actions { margin-top: 14px; }`	フッタ上部の余白.

CSS（行）	説明
`.footer-actions { flex-wrap: wrap; }`	画面幅が狭い時に折り返す.
`.btn { display: inline-block; }`	ボタンをインラインブロック化し幅や余白を制御可能に.
`.btn { padding: 8px 12px; }`	ボタン内余白.
`.btn { border-radius: 6px; }`	ボタン角丸.
`.btn { text-decoration: none; }`	リンクボタンの下線を消す.
`.btn { color: #fff; }`	ボタン文字色.
`.btn { background: #0b66b3; }`	デフォルトボタン背景色.
`.btn { box-shadow: 0 2px 6px rgba(11, 102, 179, 0.18); }`	ボタンに軽い影を付け視覚的に浮かせる.
`.btn:hover { background: #085aab; }`	ホバー時に色を暗くして押下感を出す.
`.btn.add { background: #0b66b3; }`	追加ボタンの色（デフォルトと同じ）.
`.btn.formation { background: #0a5e9d; }`	編成ボタンの色差分.
`.btn.home { background: #05507a; }`	ホームへ戻るボタンの色.
`.link.edit { margin-right: 8px; }`	編集リンクの右余白.
`.link.edit { color: #045fb4; }`	編集リンクの色.
`.link.delete { color: #b02a2a; }`	削除リンクを赤系で強調.
`@media (max-width:720px) { .container { padding: 12px; } }`	小画面でのコンテナ内余白を縮める.
`@media (max-width:720px) { .saki-table th, .saki-table td { padding: 8px; } }`	小画面でセル内余白を縮小して表示領域を確保.
`@media (max-width:720px) { .page-header { flex-direction: column; align-items: flex-start; } }`	ヘッダを縦並びにして狭い画面に対応.
`@media (max-width:720px) { .footer-actions { flex-direction: column; align-items: stretch; } }`	フッタアクションを縦並びにして操作しやすくする.
`@media (max-width:720px) { .btn { width: 100%; text-align: center; } }`	モバイルでボタンをフル幅にして押しやすくする.

2.2 app2:kimatu_gakumasu

2.2.1 システム構成

本アプリケーションは，app1 と同様に Node.js，Express を用いたサーバアプリケーションであり，テンプレートに EJS，静的資産に'public/' を使用する．サーバ実装は'app5.js' にまとめており，ビューは'views/'，静的ファイルは'public/' に配置している．サーバ起動は 'npm install'で依存をインストール後，'node app5.js'で行うことでできる．

2.2.2 目的

本アプリケーションの目的は，学園アイドルマスターのプロデュース中に登場するカードの一部を確認し，管理することで，プロデュース中の参考にすることができるようにすることである．

2.2.3 機能詳細

本アプリケーションの機能の詳細について説明する．本アプリケーションでは，学園アイドルマスターのプロデュース中に登場するカードの一部の一覧表示，詳細表示，追加，編集，削除機能を使用できる．

2.2.4 データ構造

app2 では，学マスのカードデータを'app5.js' 内のメモリ上の配列'gakumasu' で管理している．代表的なフィールドは以下の表 6 の通りである．

表 6: 'gakumasu' 配列の代表フィールド

項目名	詳細説明
id	カードの識別子，内部管理をするために使用している（数値）
name	カード名（文字列）
cost	カードのコスト（数値）
cost_type	コストタイプ（文字列）（ノーマル，やる気，好印象）
rarity	レアリティ（文字列）（SR，SSR，レジェンド）
unlock_level	カードの解放レベル（数値）（1〜）
effect	カードの効果（文字列）

2.2.5 ページ構造と HTTP メソッドとリソース名

本アプリケーションのページ構造，また，対応するメソッドについて図 2 を用いて説明する．

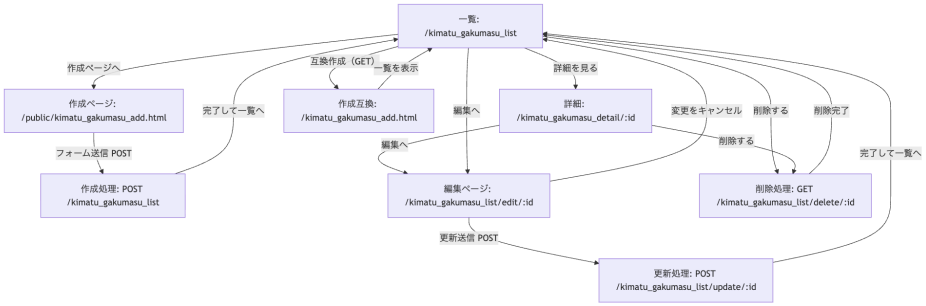

図 2: app2:kimatu_gakumasu のページ構造

本アプリケーションの起点となるホームページは`/kimatu_gakumasu_list`であり，ここから各機能へ遷移する．一覧表示からは，詳細表示，編集，削除，追加フォームへ遷移できる．詳細表示からは，編集，削除へ遷移できる．編集ページからは，更新処理を経て，編集内容を確認するために詳細表示へ遷移する．削除，追加フォームからは，それぞれ削除処理，追加処理を経て一覧表示へ遷移する．

2.2.6　リソース名ごとの機能の詳細

各エンドポイントの機能の詳細について説明する．主なエンドポイントの機能は，表 7 の通りである．

表 7: app2:kimatu_gakumasu のリソース名ごとの機能詳細

リソース名	機能詳細
/kimatu_gakumasu_list	gakumasu 配列を EJS に渡して一覧を描画する．
/kimatu_gakumasu_detail/:id	指定 id のオブジェクトを検索して詳細を表示.
/kimatu_gakumasu_list (POST)	フォームから送られた値を受け取り，メモリ配列を更新する（DB は未使用）.
/kimatu_gakumasu_list/edit/:id	指定 id のオブジェクトを検索して編集フォームを表示.
/kimatu_gakumasu_list/update/:id	フォームから送られた値を受け取り，メモリ配列を更新する（DB は未使用）.
/kimatu_gakumasu_list/delete/:id	指定 id のオブジェクトを検索してメモリ配列から削除する（DB は未使用）.

2.2.7　授業外使用技術について

本節では，授業で扱わなかったが本アプリケーションで採用した技術とその採用理由について説明する．

CSS(見た目・UI)　実際にビューで使用されている ‘public/kimatu_saki.css‘ の主要ルールを同様に行単位で示し，目的を表 8 に記述する．

表 8: app2: 使用している CSS（行ごとの詳細）

CSS（行）	説明
`body.blue-theme { font-family: -apple-system, "Helvetica Neue", Arial, sans-serif; }`	システムフォント優先の指定.
`body.blue-theme { background: linear-gradient(180deg, #e6f0ff, #f8fbff); }`	背景グラデーション.
`body.blue-theme { color: #012a4a; }`	主要テキスト色.
`body.blue-theme { padding: 24px; }`	ページ余白.
`.container { max-width: 960px; margin: 0 auto; background: #ffffff; border-radius: 10px; box-shadow: 0 6px 18px rgba(2,6,23,0.06); overflow: hidden; padding: 18px; }`	コンテンツ領域のレイアウトと装飾（幅・中央寄せ・背景・角丸・影・余白）.
`.page-header { display:flex; justify-content:space-between; align-items:center; gap:12px; margin-bottom:12px; }`	ヘッダのフレックス配置.
`.saki-table { width:100%; border-collapse:collapse; margin-top:6px; }`	テーブル表示の基本設定.
`.saki-table th, .saki-table td { padding:12px 10px; text-align:left; border-bottom:1px solid #e6eef8; }`	セルの余白・整列・区切り線.
`.saki-table th { background: linear-gradient(90deg,#d9eefb,#c6e6ff); color:#003a63; font-weight:600; }`	ヘッダの背景・色・太字.
`.saki-table tr:nth-child(even) td { background:#f4f9ff; }`	偶数行の背景.
`.saki-table tr:hover td { background:#e8f3ff; transition:background .12s; }`	行ホバーでの強調.
`.saki-table a { color:#0b66b3; text-decoration:none; }`	テーブル内リンクの見た目.

CSS（行）	説明
.footer-actions { display:flex; gap:10px; margin-top:14px; flex-wrap:wrap; }	フッタのボタン群の配置.
.btn { display:inline-block; padding:8px 12px; border-radius:6px; text-decoration:none; color:#fff; background:#0b66b3; box-shadow:0 2px 6px rgba(11,102,179,0.18); }	ボタンの基本スタイル.
.btn hover { background:#085aab; }	ボタンホバー時の色変化.
.btn.add { background:#0b66b3; }	追加ボタン.
.btn.formation { background:#0a5e9d; }	編成ボタン.
.btn.home { background:#05507a; }	ホーム戻るボタン.
.link.edit { margin-right:8px; color:#045fb4; }	編集リンクの余白と色.
.link.delete { color:#b02a2a; }	削除リンクの色.
@media (max-width:720px) { .container { padding:12px; } .saki-table th, .saki-table td { padding:8px; } .page-header { flex-direction:column; align-items:flex-start; } .footer-actions { flex-direction:column; align-items:stretch; } .btn { width:100%; text-align:center; } }	モバイル対応のレイアウト調整（余白・折返し・ボタン幅など）.

2.3 app3:kimatu_compass

2.3.1 システム構成

本アプリケーションは，app1，app2 と同様に Node.js，Express を用いたサーバアプリケーションであり，テンプレートに EJS，静的資産に’public/’ を使用する．サーバ実装は’app5.js’ にまとめており，ビューは’views/’，静的ファイルは’public/’ に配置している．サーバ起動は ‘npm install‘で依存をインストール後，‘node app5.js‘で行うことでできる．

2.3.2 目的

本アプリケーションは,「コンパス 戦闘摂理解析システム」におけるカード情報の一部を確認し，管理することで，ゲームプレイの参考にすることができるようにすることである．

2.3.3 機能詳細

本アプリケーションの機能の詳細について説明する．本アプリケーションでは,「コンパス 戦闘摂理解析システム」に登場するカードの一部の一覧表示，詳細表示，追加，編集，削除機能を使用できる．

2.3.4 データ構造

app3 では，コンパスのカードデータを'app5.js' 内のメモリ上の配列 'compass'で管理している．代表的なフィールドは以下の表 9 の通りである．

表 9: 'compass' 配列の代表フィールド

項目名	詳細説明
id	カードの識別子，内部管理をするために使用している（数値）
name	カード名（文字列）
type	カードの属性（文字列）（火属性，水属性，木属性，無属性）
attack	攻撃力（数値）
defence	防御力（数値）
hp	体力（数値）
cool_time	クールダウン（数値）
damage_type	ダメージタイプ（文字列）（近距離，遠距離，連撃，周囲）
damage	ダメージ量（%）（数値）
damage_count	ダメージ回数（数値）
effect	効果（文字列）

2.3.5 ページ構造と HTTP メソッドとリソース名

本アプリケーションのページ構造，また，対応するメソッドについて図 3 を用いて説明する．

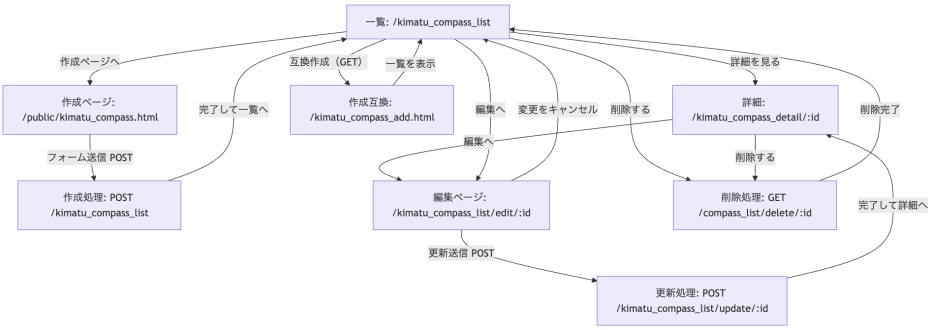

図 3: app3:kimatu_compass のページ構造

本アプリケーションの起点となるホームページは`/kimatu_compass_list`であり，ここから各機能へ遷移する．一覧表示からは，詳細表示，編集，削除，追加フォームへ遷移できる．詳細表示からは，編集，削除へ遷移できる．編集ページからは，更新処理を経て，編集内容を確認するために詳細表示へ遷移する．削除，追加フォームからは，それぞれ削除処理，追加処理を経て一覧表示へ遷移する．

2.3.6　リソース名ごとの機能の詳細

各エンドポイントの機能の詳細について説明する．主なエンドポイントの機能は，表 10 の通りである．

表 10: app3:kimatu_compass のリソース名ごとの機能詳細

リソース名	機能詳細
/kimatu_compass_list	compass 配列を EJS に渡して一覧を描画する．
/kimatu_compass_detail/:id	指定 id のオブジェクトを検索して詳細を表示.
/kimatu_compass_list (POST)	フォームから送られた値を受け取り，メモリ配列を更新する（DB は未使用）.
/kimatu_compass_list/edit/:id	指定 id のオブジェクトを検索して編集フォームを表示.
/kimatu_compass_list/update/:id	フォームから送られた値を受け取り，メモリ配列を更新する（DB は未使用）.
/kimatu_compass_list/delete/:id	指定 id のオブジェクトを検索してメモリ配列から削除する（DB は未使用）.

2.3.7　授業外使用技術について

本節では，授業で扱わなかったが本アプリケーションで採用した技術とその採用理由について説明する．

CSS(見た目・UI) 同様に，‘public/kimatu_saki.css‘ の主要ルールを行ごとに示し，目的を表 11 に記述する．

表 11: app3: 使用している CSS（行ごとの詳細）

CSS（行）	説明
`body.blue-theme { font-family:-apple-system, "Helvetica Neue", Arial, sans-serif; }`	システムフォント優先で見た目を OS ネイティブに合わせる．
`body.blue-theme { background:linear-gradient(180deg,#e6f0ff,#f8fbff); }`	ページ背景のグラデーション．
`body.blue-theme { color:#012a4a; padding:24px; }`	文字色とページ余白．
`.container { max-width:960px; margin:0 auto; background:#ffffff; border-radius:10px; box-shadow:0 6px 18px rgba(2,6,23,0.06); overflow:hidden; padding:18px; }`	コンテンツの幅・中央配置・背景・角丸・影・余白をまとめて指定．
`.page-header { display:flex; justify-content:space-between; align-items:center; gap:12px; margin-bottom:12px; }`	ヘッダ配置（フレックス）．
`.saki-table { width:100%; border-collapse:collapse; margin-top:6px; }`	テーブルの基本設定．
`.saki-table th, .saki-table td { padding:12px 10px; text-align:left; border-bottom:1px solid #e6eef8; }`	セル内余白・整列・区切り線．
`.saki-table th { background:linear-gradient(90deg,#d9eefb,#c6e6ff); color:#003a63; font-weight:600; }`	ヘッダの装飾．
`.saki-table tr:nth-child(even) td { background:#f4f9ff; }`	偶数行の背景で読みやすさ向上．
`.saki-table tr:hover td { background:#e8f3ff; transition:background .12s; }`	行ホバーで強調，アニメーションの指定．

CSS（行）	説明
.saki-table a { color:#0b66b3; text-decoration:none; }	テーブル内リンクの見た目.
.saki-table a:hover { color:#024b85; text-decoration:underline; }	リンクホバー時に視覚的な変化を付与.
.footer-actions { display:flex; gap:10px; margin-top:14px; flex-wrap:wrap; }	フッタボタンの配置と折返し対応.
.btn { display:inline-block; padding:8px 12px; border-radius:6px; text-decoration:none; color:#fff; background:#0b66b3; box-shadow:0 2px 6px rgba(11,102,179,0.18); }	ボタンの基本スタイル（視認性と操作性向上）.
.btn hover { background:#085aab; }	ホバー時のフィードバック.
.link.edit { margin-right:8px; color:#045fb4; }	編集リンクの余白と色.
.link.delete { color:#b02a2a; }	削除リンクを目立たせる色.
@media (max-width:720px) { .container { padding:12px; } .saki-table th, .saki-table td { padding:8px; } .page-header { flex-direction:column; align-items:flex-start; } .footer-actions { flex-direction:column; align-items:stretch; } .btn { width:100%; text-align:center; } }	モバイル画面でのレイアウト調整（余白・折返し・ボタン幅）.

3　管理者向け仕様書

本節では，管理者向けに各 web アプリケーションのインストールから起動までの手順，不具合情報について説明する．また，ここでは，代表である app1:kimatu_saki について説明する．app2,app3 についても同様の手順でインストール，起動が可能であるが，一部機能は実施していないため，注意が必要である．しかし，app2 と app3 は双方ともにシステム構成や使用技術は同じである．

3.1　目的

本アプリケーション群の管理者向けに，インストールから起動までの手順，不具合情報を提供することで，円滑な運用を支援することを目的とする．

3.2　本アプリケーションの流れ

本アプリケーションの遷移の流れについて図 4 を用いて説明する．

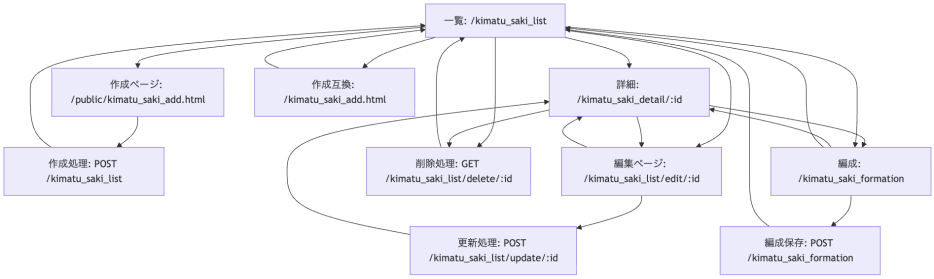

図 4: 管理者向け仕様書: アプリケーションの流れ

図 4 に示すように，本アプリケーションは，ホームページから各一覧ページへ遷移し，そこから詳細表示，編集，削除，追加フォームへ遷移することができる．図は app1:kimatu_saki を例に示しているが，app2,app3 も同様の流れである．しかし，app2,app3 については，編成画面の機能は実装していないため，編成，編成保存は存在しない．一覧ページから，各データの名前をクリックすることで，詳細表示へ遷移することができる．詳細表示ページからは，編集，削除へ遷移することができる．また，編集，削除は一覧ページからも遷移することができる．編集ページからは，更新処理を経て，編集内容を確認するために詳細表示へ遷移する．削除，追加フォームからは，それぞれ削除処理，追加処理を経て一覧表示へ遷移する．ここまでの流れは app1,app2,app3 で共通である．app1:kimatu_saki については，編成画面が存在し，編成画面からは，選択したカードの編成を保存し，一覧表示または詳細表示へ遷移できる．また，編成画面では，選択中のカードの総合力を計算し，表示する機能も存在する．

3.3　インストールから起動までの手順

3.3.1　インストール方法

ここでは，本アプリケーション群のインストール方法について説明する．運用環境は MacBookPro での利用を想定している．まず，本稿の冒頭で示した GitHub リンクより"webpro_06"のリポジトリをダウンロードする．続いて，ターミナルを開き，リポジトリのルートディレクトリに移動する．次に，依存パッケージのインストールを行うため，以下のソースコードを実行する．

```
npm install
```

次に，以下のソースコードを実行し，サーバーを起動する．

`npm start`

エラーが出力されなければ，サーバーの起動は成功である．これにより，ブラウザで `http://localhost:8080/kimatu_home` を開くことで，本アプリケーション群を利用できる．また，停止したい際は，実行中のターミナルで「Ctrl+C」を押すことで停止できる．本アプリケーション群についてとして，1 つ目一覧はプロセカ，2 つ目一覧は学マス，3 つ目一覧はコンパスのデータを管理することができる．

3.3.2 想定しうる不具合について

インストール時に想定しうる不具合として，以下のものがある．

- Node.js がインストールされていない場合，‘npm install‘コマンドが失敗する可能性がある．
- 依存パッケージのインストール中にネットワークエラーが発生する可能性がある．
- ポート 8080 が既に使用されている場合，サーバーの起動に失敗する可能性がある．

対処法として，Node.js がインストールされていない場合は，公式サイトからインストールを行う．依存パッケージのインストール中にネットワークエラーが発生した場合は，ネットワーク接続を確認し，再度 ‘npm install‘コマンドを実行する．ポート 8080 が既に使用されている場合は，別のポート番号を指定してサーバーを起動するか，既存のプロセスを停止する．また，ダウンロードしたリポジトリに不備 (起動できないなら package.json の不備など) がある場合は，再度 GitHub から最新のリポジトリをダウンロードする．さらに，データはサーバのメモリ内にあるためサーバ停止で消失してしまう．永続化が必要な場合はデータベース導入が必要となるため，注意が必要である．

4 利用者向け仕様書

本仕様書において説明するシステムは，プロジェクトセカイ カラフルステージ！（以下，プロセカ）において登場するキャラクターである「天馬咲希」のカードについてのデータ管理，編成シミュレーションを行うものである．本仕様書では，利用者向けにシステムの概要，目的，使用できる機能，操作方法について説明する．

4.1 目的

本アプリケーションでは，プロセカに登場するキャラクターである「天馬咲希」のカードデータの一覧や詳細について確認し，実際に編成をシミュレーションし，チャレンジライブの総合力の計算や，カード単体の総合力の確認を行い，より良い編成を考えることができるようにすることを目的としている．プロセカ内では，キャラクターごとに複数のカードが存在し，それぞれに異なるステータスや効果が設定され，さらに，マスターランクやタイプボーナスなどの要素が総合力に影響を与えるため，これらを考慮した編成シミュレーションは，プレイヤーの戦略的な意思決定を支援する重要な機能となる．

4.2　目的

学習目的のための CRUD 操作・画面設計練習．複数のデータ型（キャラクター／カード）を管理し，簡易的な計算（編成の総合力）を行うことを想定している．

4.3　使用できる機能

本アプリケーションでは，プロセカにおける「天馬咲希」のカードデータの一覧表示，詳細表示，追加，編集，削除機能を使用できる．また，編成画面では，複数のカードを選択して編成を保存し，選択したカードの総合力を計算して表示する機能も使用できる．

4.4　操作方法

まず，Chrome などの検索エンジンより図 5 のように `http://localhost:8080/kimatu_home` にアクセスする．

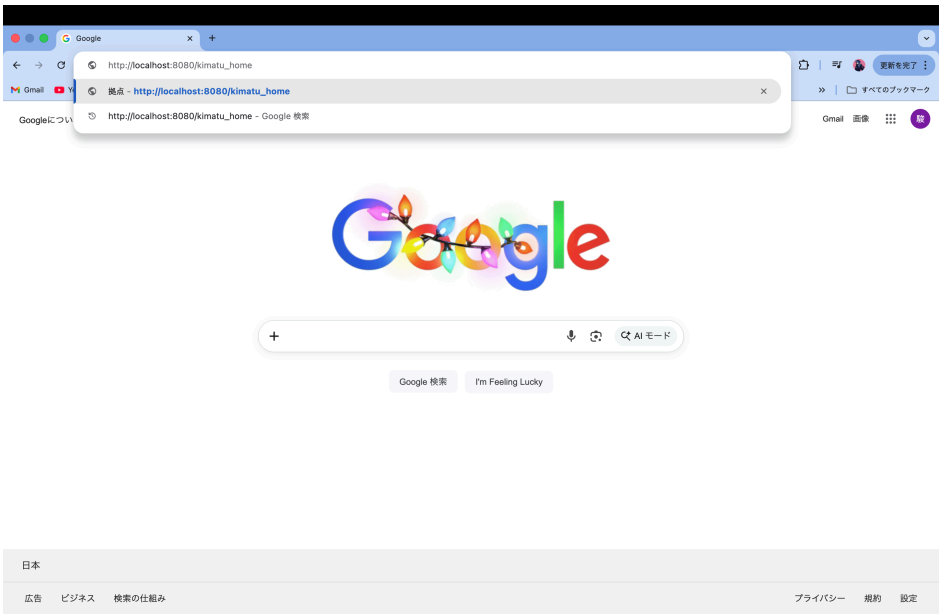

図 5: ブラウザからの検索例

次に，図 6 のようにホームページが表示されるため，画面中央上部の「１つ目閲覧」をクリックする．

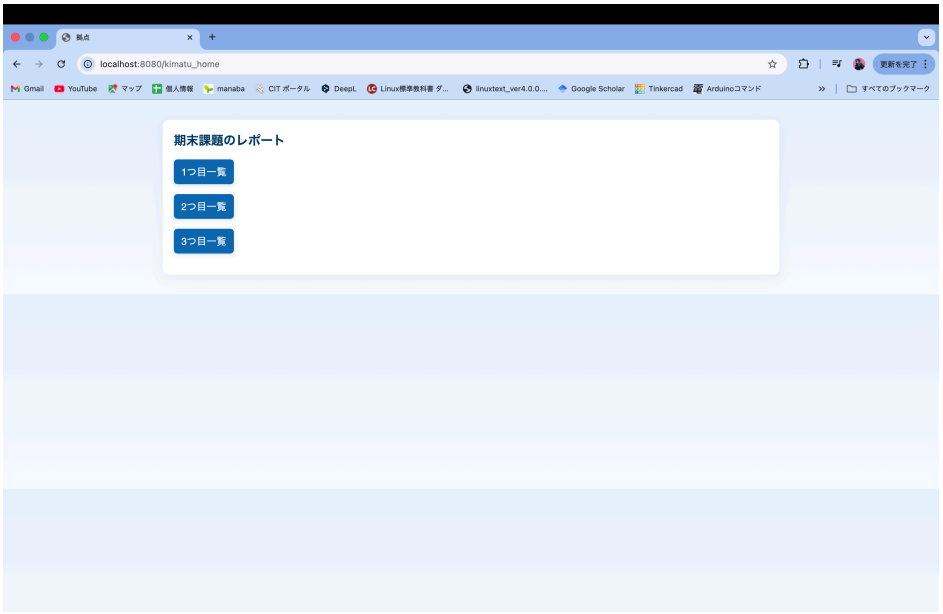

図 6: ホームページ

すると，図 7 のように「天馬咲希」のカード一覧ページが表示される．

図 7: カード一覧ページ

ここでは，カード名とタイプが表示されており，カード名をクリックすることで，図 8 のように詳細表示ページへ遷移することができる．

図 8: カード詳細表示ページ

詳細表示ページでは，カードのレアリティ，スキル名，スキルレベル，マスターランク，初期総合力などの詳細情報が表示される．また，詳細表示ページからは，編集，削除，編成画面へ遷移することができる．編集ページでは，図 9 のように，カードのステータスを編集することができる．

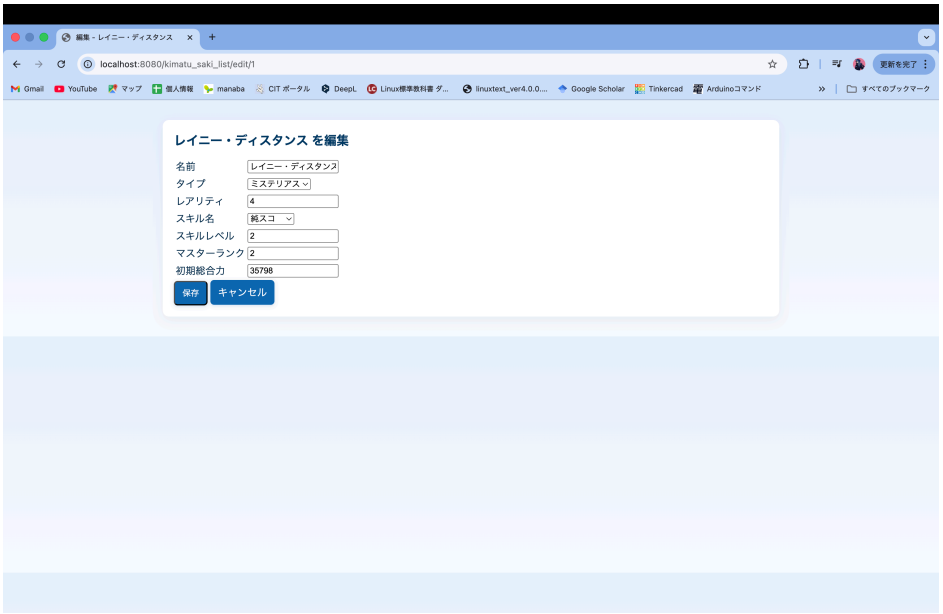

図 9: カード編集ページ

編集ページで変更を加えた後，「更新」ボタンをクリックすることで，編集内容が保存され，詳細表示ページへ移動する．削除ボタンを押すことで，図 10 のように，カードの削除を確認するこ

とができる．

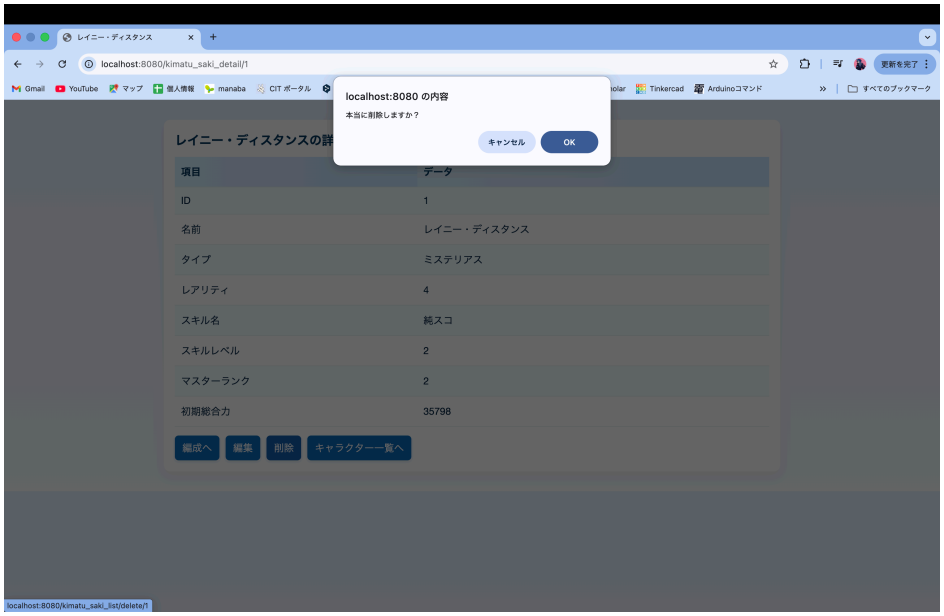

図 10: カード削除ページ

ここで「OK」をクリックすることで，カードが削除され，一覧表示ページへ移動しする．また，一覧表示ページの「追加フォームへ」ボタンをクリックすることで，図 11 のように，カードの追加フォームページへ遷移することができる．

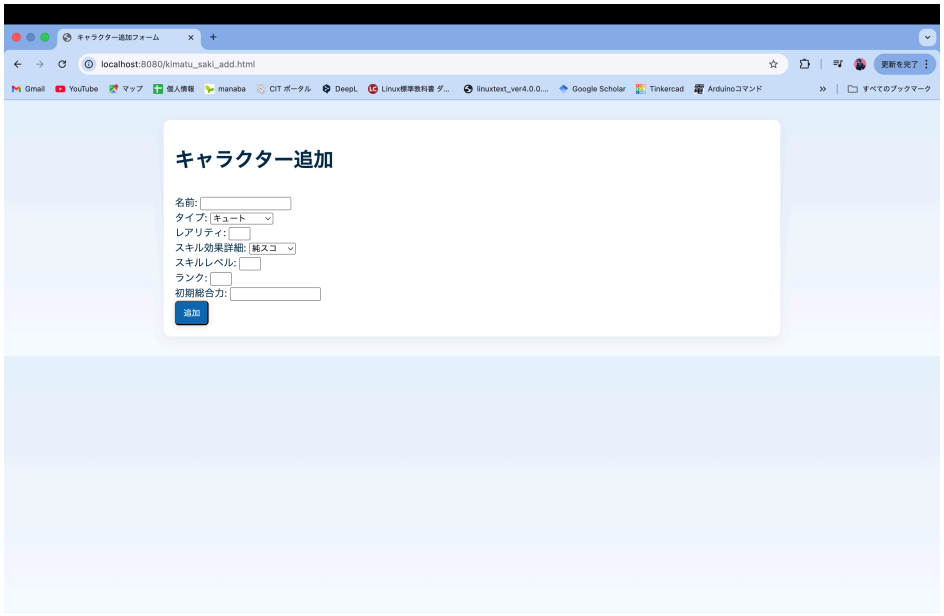

図 11: カード追加フォームページ

追加フォームページでは，新しいカードの情報を入力し，「追加」ボタンをクリックすることで，カードが追加され，一覧表示ページへ移動する．さらに，一覧表示ページの「編成画面へ」ボタンをクリックすることで，図 12 のように，編成画面へ遷移することができる．

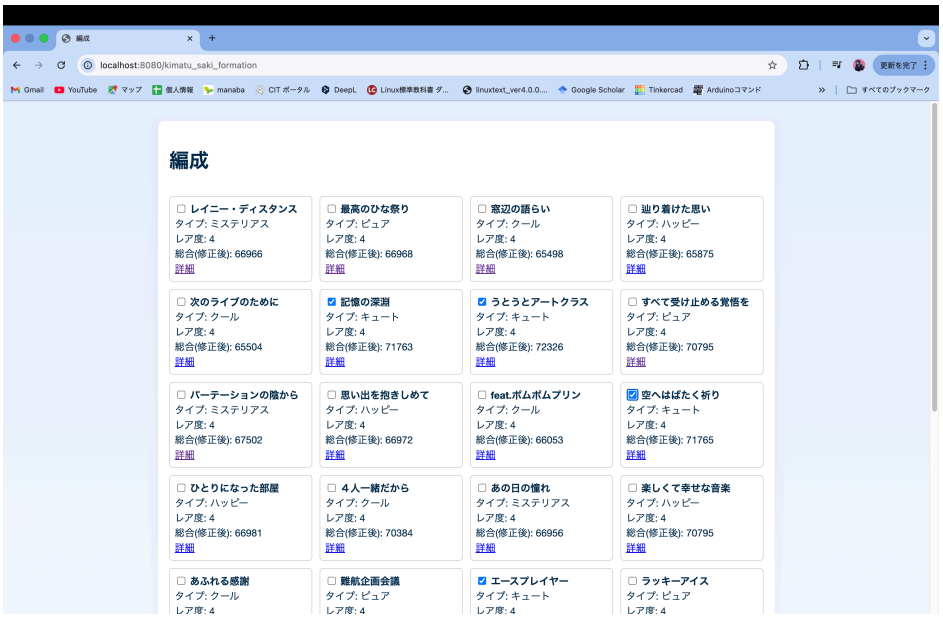

図 12: 編成画面

編成画面では，最大 5 枚までのカードを選択し，「編成保存」ボタンをクリックすることで，図 13 のように選択したカードの総合力が計算され，表示される．

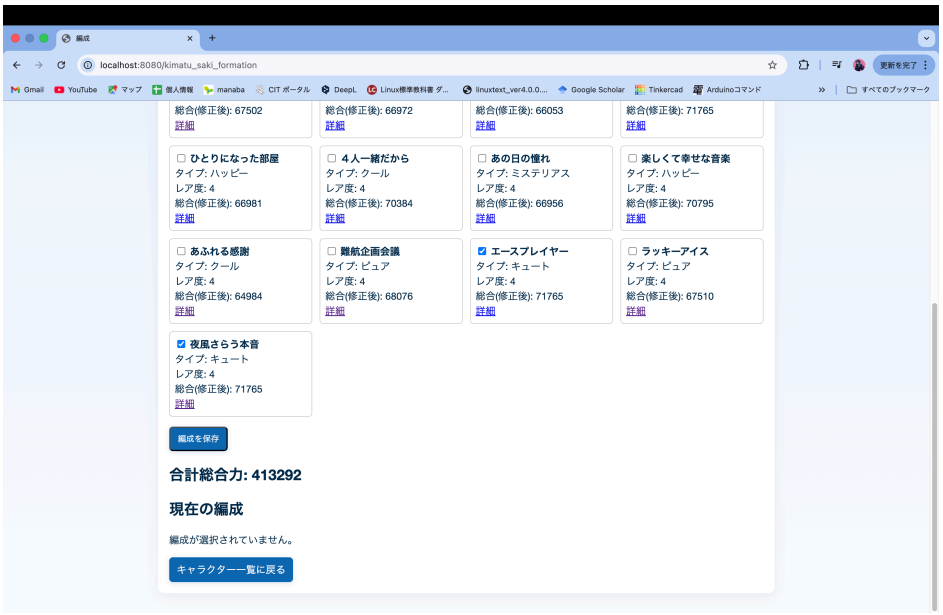

図 13: 編成画面（総合力計算後）

以上が，本アプリケーションの主な操作方法である．これらの機能を活用することで，プロセカにおける「天馬咲希」のカードデータの管理と編成シミュレーションを効果的に行うことができる．また，削除機能や編集機能を活用することで，カードデータの最新情報を維持したり，「天馬咲希」以外のキャラクターの情報も入れることができるため，より良い編成を考えるための参考にすることができる．