

Smart Pad 4i

洗練されたUIデザインを簡単に実現！ HTML作成テクニック

株式会社ミガロ、
RAD事業部技術支援課
國元 祐二



略歴

生年月日:1979年3月27日
最終学歴:2002年 追手門学院大学 文学部アジア文化学科卒業
ミガロ入社年月:2010年10月 株式会社ミガロ入社
社内経歴:2010年10月 RAD事業部配属

現在の仕事内容:

SmartPad4i(JC/400)、Business4Mobile、
Valenceの製品試験やサポート業務、
導入支援などを行っている。

- 1. はじめに
- 2. SmartPad4iでの開発
- 3. CSSフレームワーク
 - 3.1 HTMLとCSS
 - 3.2 CSSフレームワークとは?
 - 3.3 CSSフレームワークのメリット・デメリット
 - 3.4 CSSフレームワークの種類
 - 3.5 Bulma CSSフレームワークとは?
 - 3.6 Bulma CSSフレームワークの導入
 - 3.7 Bulma CSSフレームワークの使用
- 4. Webアイコンフォント
 - 4.1 Font Awesome
 - 4.2 Font Awesomeの導入
 - 4.3 Font Awesomeの使用法
- 5. HTML作成例
 - 5.1 メニュー画面の作成例
 - 5.2 入力欄の作成例
- 6. お勧めのUIカスタマイズ(CSSローディング表示)
- 7. さいごに

1. はじめに

システム開発に携わっていれば、頻りにUIという単語を聞くのではないだろうか？UIはユーザーインターフェース(User Interface)の略称で、利用者(ユーザー)と機械(アプリケーション)の間でやりとりするための接点(インターフェース)を指す。

UIは、アプリケーションの印象や使い勝手の良さをユーザーが判断する大きな要素になるため、非常に重要だ。UIについては様々な考えがあるが、直感的に操作できるUIが良いUIだと考える。

筆者は最近、スマートフォンを買い替えた。昔は分厚い取扱説明書が付いてきたものだが、そのスマートフォンには取扱説明書がついてなかった。しかし、取扱説明書がなくても操作や扱いに困ることはない。直感的に操作ができるようなUIなので、スマートフォンには取扱説明書を含めていないのだと思う。

機器だけでなく、一般的に利用される、アプリケーションも同様で、普段日常的に利用しているメール、SNSや動画配信サービス等のサービスも直感的に利用できるようにUIが設計されていることが多い。

弊社では、SmartPad4iというIBMiや5250アプリケーションのGUI化が行える開発ツールを販売している。

SmartPad4iで作成するアプリケーションのUIも直感的に操作できることが理想だと思う。

2. SmartPad4iでの開発

IBM iプログラム開発者がWebアプリケーションを作成するための開発ツールとして、SmartPad4iは非常に優れている。SmartPad4iの開発では、IBM iプログラム開発者は5250アプリケーション開発と同様に、IBM i側のファイルや業務ロジックをIBM iプログラム(RPG/COBOL/CLプログラム)で作成するが、5250アプリケーション開発でインターフェースとなる画面ファイル(DDS)の代わりにHTMLでインターフェースを作成する。

HTMLは簡単な言語のため、シンプルな画面を作成する場合、開発者は簡単に画面を作成できる。また、SmartPad4iで使用するHTMLはCSSやJavaScriptなどでカスタマイズできるため、自由度が高く様々な機能を追加することもできる。

最近の開発では、PCブラウザだけでなく、スマートデバイスの表示にも対応するため、デバイスのサイズに依存しない、レスポンシブデザインが必要になる場合も多い。レスポンシブデザインは一つのHTMLで、PCブラウザやスマートデバイス(スマートフォン、タブレット)のような画面サイズの異なる表示に対応するHTML作成技術だ。レスポンシブデザインに対応するには、HTMLとCSSの知識が必要となる。

一から全てCSSを作成するような事をしなくても、レスポンシブデザインで、使いやすい洗練されたUIのHTMLを簡単、効率的に開発する方法を本稿で紹介する。

3. CSSフレームワーク

3.1 HTMLとCSS

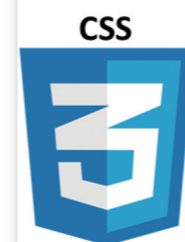
HTMLは「HyperText Markup Language」の略称である。「HyperText」とは、「Textを超える」という意味で、テキスト文章だけでなく、画像やリンクも含まれる情報が、文章構造「Markup」として表現された言語「Language」という意味だ。HTMLでも多少は文章の見た目や装飾を行うことができる。しかし、一般的には、視覚的・感覚的效果を定義する手段としてCSS(Cascading Style Sheets)を利用する。

CSSはHTMLやXMLで記述された文章の体裁、表示をカスタマイズするために利用されるスタイルシート言語である。1994年10月10日にホーコン・ウィウム・リー(Håkon Wium Lie)氏によって提案された。CSSはバージョンアップごとに機能が追加されて表現の幅が広がってきている。

【図1】

図1 CSSの歴史

CSS制定年数とバージョン	内容
1996年 CSS Level1	HTMLへ視覚的に装飾する事を目的に策定された
1998年 CSS Level2	CSS1を含む新しいプロパティが追加 CSS1からプロパティ数が約2倍になった
2011年 CSS Level2 Revision1	CSS level2の改訂や修正
2018年 CSS Level3	CSS2.1を中核に新しい機能の追加改良が実施された



CSS3では、ボーダーの線を角丸
アニメーション表示の拡張
ボックスのシャドー表示(影表示)
等様々な描画設定が追加された

CSSのプロパティ設定が増えていく分、HTML作成ではレスポンシブデザインの対応も加わりCSSの記述量が多くなってきた。少ないCSSの記述でHTMLを作成する手段として、CSSフレームワークを利用する方法がある。

3.2 CSSフレームワークとは？

CSSフレームワークはボタンやフォームの入力要素、レイアウトなどの部品が集まったCSSパーツ集のようなものだ。実現するために、たくさんのCSS記述が必要なデザインでも、

CSSフレームワークを利用すると、少ない記述で簡単、効率的にHTMLを作成することができる。

3.3 CSSフレームワークのメリット・デメリット

CSSフレームワークを利用するメリットには以下がある。

- ①デザイナーが作成するような見た目のUIを作成できる
- ②効率的にHTMLを作成できる
- ③HTMLの記述に統一感があり、チーム単位での開発に向いている

良い面ばかりのようであるが、CSSフレームワークにはもちろんデメリットもある。

- ①CSSフレームワークを利用するための学習コストが発生する
- ②CSSファイルに使われないものも含めて、フレームワーク全体のCSS設定が含まれるためファイルサイズが通常作成

するCSSファイルより大きなサイズになる

- ③軽微なデザインの調整が難しい場合がある
(レイアウトの微調整やフレームワークの規定デザインを一部変更する場合等)

メリット、デメリットもあるCSSフレームワークであるが、デメリットを考えると、レスポンシブデザインの対応や、洗練されたデザインのUIが適用されることのメリットは大きい。洗練されたUIのHTMLを効率的に作成する手段として、CSSフレームワークは非常に有用だ。

3.4 CSSフレームワークの種類

CSSフレームワークは数多く存在している。インターネット検索を行えば、数多くのフレームワークを見つける事ができるはずである。CSSフレームワークでは、Twitter社が開発したbootstrapやamazonやadobeなどのソフトウェア会社が採用しているFoundationなどが有名だ。数多くのCSSフレームワークがあり、それぞれ利用方法やメリット、メリッ

トも異なるため、どのCSSフレームワークを利用するか迷うことも多いと思う。

数多くあるCSSフレームワークの中から、本稿ではSmartPad4iに適したCSSフレームワークとしてBulma CSSフレームワーク (<https://bulma.io/>)を紹介したい。

3.5 Bulma CSSフレームワークとは？

Bulma CSSフレームワーク(Bulma)は2016年にMITライセンスの下で配布された新しいCSSフレームワークだ。Bulmaのメリットとしては、CSSの知識がなくてもデザインが可能であること、少しの学習コストで利用できる点がある。

数多くのCSSフレームワークの中でBulmaはSmartPad4iと相性が良い点が2つある。

ひとつは、CSSだけのシンプルなCSSフレームワークである点だ。通常、CSSフレームワークはフレームワークに付随するJavaScriptファイルとセットになっていることが多い。しかし、BulmaはCSSファイルだけのシンプルなCSSフレームワーク

のため、JavaScript部分は独自に記述することができる。

もうひとつは、通常のHTMLタグを使用している点だ。SmartPad4iではコンボボックス<select>タグやラジオボタン<input type="radio">をHTMLタグとして利用する必要がある。CSSフレームワークによってはコンボボックスなどを(li)タグとして記述する場合もあるが、Bulmaではselectタグをそのまま利用するため、JavaScriptでタグを書き換える必要もなく、<select>タグを<div>でラップするような、少しの追加記述だけで、SmartPad4iのコンボボックスやラジオボタンが使用できる。

3.6 Bulma CSSフレームワークの導入

Bulmaを利用するには、Bulma公式サイトからcssファイルをダウンロードする。【図2】



ダウンロードしたzipファイルを展開後/bulma/cssフォルダにあるbulma.min.cssをWebサイトのディレクトリにコピーして読み込むだけだ。

例えば、/smartpad4i/SP4IREP21/css/bulma.min.cssにCSSを配置した場合には、HTMLへ【ソース1】1-①のように記述する。

ソース 1

```
Bulma.cssの読み込み例【HTML】
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="shift_jis">
5 <title>メニュー画面例</title>
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <!-- css framework -->
8 <link rel="stylesheet" href="../../smartpad4i/html/SP4IREP21/css/bulma.min.css">
9 <!-- user css -->
10 <link rel="stylesheet" href="../../smartpad4i/html/SP4IREP21/css/style.css">
11 <meta name="format-detection" content="telephone=no">
12 </head>
13 <body>
~省略~
```

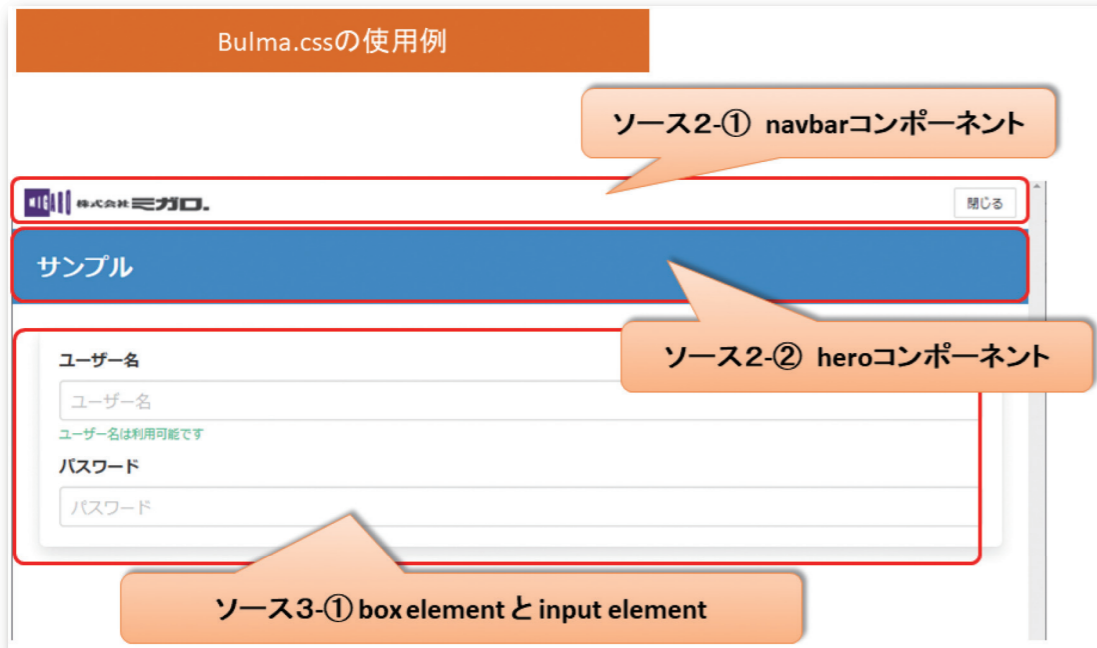

3.7 Bulma CSSフレームワークの使用

Bulmaはcssファイルを読みこんだ後、HTMLの記述で要素にclass属性を設定することで使用できる。また公式サイトに様々な記述例が記載されているため、公式サイトを確認することで大抵の画面は作成可能だ。

また、Bulmaでは、部品を組み合わせるように画面を作成し

ていくことができる。ナビゲーションメニューが作成できる navbarコンポーネント、タイトル表示に使用できるheroコンポーネント、簡単な入力欄の要素を設定した画面例とHTMLの例が【図3】【ソース2】【ソース3】だ。

図3 画面例



ソース2

navbarコンポーネントの例 2-①【HTML】

```
19 <!-- navbarコンポーネント -->
20 <nav class="navbar"> -----(1)
21   <div class="navbar-brand"> -----(2)
22     <a class="navbar-item"> -----(3)
23       
24     </a>
25   </div>
26   <div class="navbar-end"> -----(4)
27     <a href="#" id="BTNF3" class="navbar-item"> -----(3)
28       <button class="button is-small is-outlined"><span>閉じる</span></button>
29     </a>
30   </div>
31 </nav>
32 <!-- /navbarコンポーネント -->
```

heroコンポーネントの例 2-②【HTML】

```
33 <!-- heroコンポーネント -->
34 <section class="hero is-small is-info"> -----(1)
35   <div class="hero-body"> -----(2)
36     <h1 class="title is-4"> -----(3)
37       <span>サンプル</span>
38     </h1>
39   </div>
40 </section>
41 <!-- /heroコンポーネント -->
```

ソース3

box elementとinput elementの例 3-①【HTML】

```
42 <!-- container -->
43 <main class="container"> -----(1)
44   <div class="box mt-5"> -----(2)
45     <div class="field"> -----(3)
46       <label class="label">ユーザー名</label>
47       <div class="control"> -----(4)
48         <input class="input" type="text" placeholder="ユーザー名"> -----(5)
49       </div>
50       <p class="help is-success">ユーザー名は利用可能です</p> -----(6)
51     </div>
52     <div class="field">
53       <label class="label">パスワード</label>
54       <input class="input" type="password" placeholder="パスワード">
55     </div>
56   </div>
57 </main>
58 <!-- /container -->
```

class属性に設定しているclass名はBulmaに定義されているclass名になる。

では、【図3】について補足する前に、【図3】を構成するために必要となる、HTMLで多用するタグについて簡単に説明する。

まず、<div>タグはdivisionの略で分割を意味するタグに

なる。<div>タグ単体では特別な意味を持たず、CSSを適用することで意味のあるタグに変化する。

次に、<a>タグはanchorの意味で、anchorは錨を意味する単語である。<a>タグの要素をクリックすると、指定したURLに遷移することができる。

最後に、<label>タグと<input>タグについて説明する。<label>タグは要素の項目名と入力欄やラジオボタン、チェックボックスなどの要素と関連付けを行うためのタグである。<input>タグはフォームの入力欄に使用されるタグだ。type属性を設定することで、入力欄やラジオボタン、チェックボックス、ボタン等様々な形に変化するタグである。では、【図3】の説明に入る。【図3】の「navbarコンポーネント」は【ソース2-①】で構成されている。また、class属性に設定されているclass名は全て、Bulmaで定義されているclass名である。(1)では主要なナビゲーションを表すタグ<nav>タグにclass属性navbarを指定している。これにより、<nav>タグはBulmaのnavbarコンポーネントとして表示される。次の(2)では、navbarコンポーネント内の要素を設定する。<div>タグにnavbar-brand class属性を設定している。navbar-brandはブランドロゴの画像を定義する場合に指定するclass名である。設定するとロゴは左側で常に表示状態となる。22行目、27行目のnavbar内のタグには(3)のようにnavbar-itemを指定する必要がある。設定するとnavbar内の要素として認識される。(4)のclass名navbar-endはnavbar内の要素を右寄せで表示するために設定している。次に、図3の「heroコンポーネント」は【ソース2-②】で構成されている。heroコンポーネントはスクリーン全幅の印象的なバナーを表示することができるコンポーネントである。例えば、サイトタイトルなどを表示するのに使用できる。【ソース2-②】の(1)では、特定の<section>タグにclass属性heroとis-small,is-infoを指定している。class属性は半角スペース区切りでclass名を記述すると、複数のclass属性を設定することができる。<section>タグは意味や機能のひとまとまりを表す際に使用するタグである。heroと一緒に指定しているis-smallはheroコンポーネントの縦サイズとなる。例えば、is-largeと指定すれば、縦幅がより大きいバナーを表示することができる。is-infoは色の指定で、ブルーに表示するには、is-infoと指定する。例えば、is-successはグリーン、is-dangerはレッドのようにBulmaで指定された色を適用することができる。(2)のhero-bodyはheroコンポーネントの要素に指定する。(3)のtitleとis-4はタイトルの文字としてサイズ4で表示する意味になる。サイズはis-1~is-6まであり、大きい数値になると、小さい文字で表示できる。is-xを指定しない場合はデフォルトサイズのis-3が、文字サイズに適用される、少しフォントを小さくするためis-4を指定している。

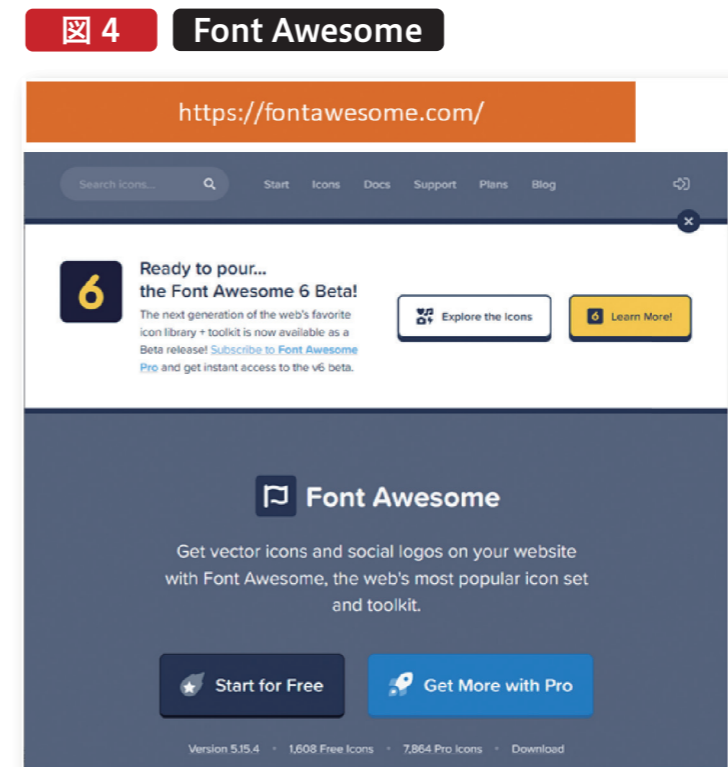
最後に、【図3】の「box elementとinput element」は【ソース3-①】で構成されている。この要素の前に定義されている(1)のmainタグに設定されている、containerについて説明する。containerはレイアウトに使用される。設定されると画面サイズに合わせて横幅が自動的に切り替わるようになっている。画面サイズがFullHD(1408px)以上であれば、containerの横幅は、1344pxに、画面サイズが1216px~1407pxの場合には、containerの横幅は1152pxになり、画面サイズが1024px~1215pxの場合には横幅が960pxとなる。画面サイズが1023px以下の場合には、containerの横幅は画面サイズに合わせて100%の横幅に変化する。レスポンシブデザインに対応できるシンプルな要素である。box elementは(2)でdivタグに指定している。boxを設定すると要素を配置できるシンプルなコンテナとなる。白い背景に影がついた見た目に変化する。フォームの入力要素を定義する場合には、(3)のようにclass属性にfieldを設定する。fieldはlabelやフォームの入力要素、help用のテキスト表示などを内包できるコンテナである。コンテナ内(4)のcontrolも単一のinput要素やselect要素、button要素、icon要素の表示を拡張するためのコンテナである。アイコンを設定する場合には、(4)のコンテナ内に追加する必要がある。(5)の<input>タグには、class属性inputを設定することで、入力欄が横幅全体に広がり、入力欄のエッジ部分が角丸に変化する。(6)はヘルプ用のテキストになる。<p>タグはParagraphの略で段落を意味するタグだ。class属性名helpを設定することで例えば、注意事項を記述、または、エラーメッセージを出力する際に利用できる。Bulmaではclass属性名を設定するだけで簡単に要素をカスタマイズできることが分かったと思う。また、Bulmaは他の機能やリソースを組み込み、利用できる点にも魅力がある。Bulmaはフォントアイコンのライブラリにも依存しないため、自分の好みのフォントアイコンも使用可能だ。より洗練されたUIを作成するために、Bulmaと併用して利用できるWebアイコンフォントを次節で紹介する。

4. Webアイコンフォント

WebアイコンフォントはHTMLのタグで簡単にアイコンを表示できるサービスだ。画像のアイコンと異なり、拡大しても画像のように荒くなることもなく、色やサイズも簡単に変更できる。スマートデバイスのWebアプリケーションで使用されることも多くなってきていることもあり、アイコンで直感的に機能が理解できるWebアイコンフォントは、洗練されたWebアプリケーションに必須の要素だと思う。Webアイコンフォントの中で最も有名なFont Awesomeについて説明する。

4.1 Font Awesome

WebアイコンフォントもCSSフレームワークと同様に様々あるが、世界で最も使用されているのはFont Awesomeだ。無料プランでも1500個を超えるアイコンが利用できる。【図4】



Smart Pad 4i

4.2 Font Awesomeの導入

ソース 4

```
Font Awesomeの読み込み【HTML】
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="shift_jis">
5 <title>メニュー画面例</title>
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link rel="apple-touch-icon" href="icon.png">
8 <!-- Font Awesome -->
9 <script src="https://use.fontawesome.com/releases/v5.3.1/js/all.js" defer ></script>
10 <!-- css framework -->
11 <link rel="stylesheet" href="../../smartpad4i/html/SP4IREP21/css/bulma.min.css">
12 <!-- user css -->
13 <link rel="stylesheet" href="../../smartpad4i/html/SP4IREP21/css/style.css">
14 <meta name="format-detection" content="telephone=no">
15 </head>
```

Font Awesomeを導入するのは簡単で、【ソース4】4-①のようにjsファイルを読み込むだけだ。

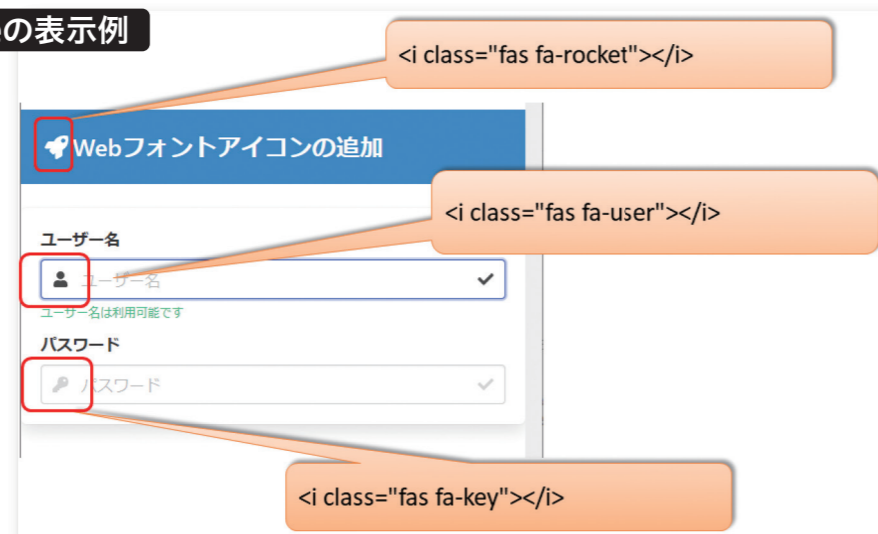
jsファイルはFont AwesomeのCDNからダウンロードできる。CDNは(コンテンツ・デリバリー・ネットワーク)の略称で、サーバのコンテンツを世界各地に分散する仕組みである。簡単に説明すると、世界中に分散されたサーバがあり、利

用者は最も適切なサーバからデータをダウンロードできる。もちろん、開発者が管理するWebサーバ上にFont Awesomeのファイルを配置することもできるが、CDNを利用するとサーバの負荷を減らすことができるため、理由がなければCDNからダウンロードすることをお勧めする。

4.3 Font Awesomeの使用方法

Font Awesomeは<i>タグを記述するだけで利用できる。例えば、ロケットのフォントアイコンを表示する場合には、<i class="fas fa-rocket"></i>を追加するだけだ。【図5】

図5 Font Awesomeの表示例



class属性に設定するfa-xxxxを変更することで様々なアイコンを表示することができる。利用可能なアイコンについ

ては公式サイトを検索欄に入力して探すことができる。【図6】

図6 Font Awesome アイコン検索



結果が表示される

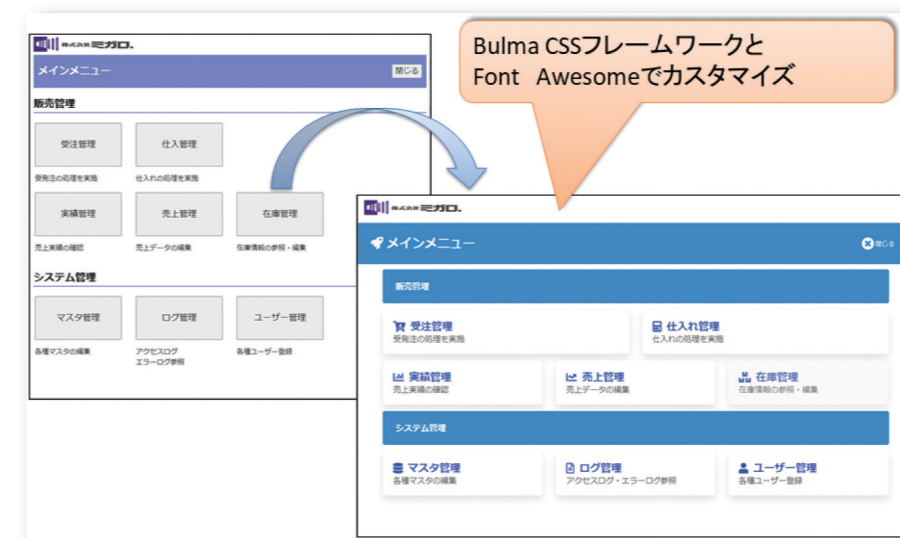
5.HTML作成例

BulmaとFont Awesomeを利用すると簡単に洗練されたUIのHTMLが作成可能だ。SmartPad4iの画面(HTML)を例にして説明する。

5.1メニュー画面の作成例

Bulmaを使用すると【図7】のように簡単に見栄えが良い画面が作成できる。また、フォントアイコン表示により、直感的にどのような機能であるかわかりやすい画面にもなる。

図7 メニュー画面例



さらに、Bulmaを利用すると、HTML作成時に少し意識するだけで、レスポンシブデザインのHTMLを作成することができる。【図8】

図8 レスポンシブデザイン



例で作成したメニュー画面は、「Tile element」で作成して、アイコンはFont Awesomeを利用した。【図9】【ソース5】

図9 Tile element



ソース5

```

Tile element を利用したメニュー【HTML】
50 <div class="tile is-ancestor mb-0">----- (1)
51 <div class="tile is-parent">----- (2)
52 <a href="#" class="box-link" onclick="startApp(1); return false;">----- (3)
53 <article class="tile is-child box">----- (4)
54 <p class="title has-text-link-dark">
55 <i class="fas fa-cart-arrow-down"></i>----- (5)
56 <span class="app-title">受注管理</span>
57 </p>
58 <p class="subtitle">受発注の処理を実施</p>
59 </article>
60 </div>
61 </div>
62 <div class="tile is-parent">
63 <a href="#" class="box-link" onclick="startApp(2); return false;">
64 <article class="tile is-child box">
65 <p class="title has-text-link-dark">
66 <i class="fas fa-calculator"></i>
67 <span class="app-title">仕入れ管理</span>
68 </p>
69 <p class="subtitle">仕入れの処理を実施</p>
70 </article>
71 </div>
72 </div>
73 </div>

```

```

box-link【CSS】
.box-link{
  display:flex;
  width:100%;
}

```

divタグにclass属性tileを設定するだけでTile Elementを利用できる。(1),(2),(4)のclass属性tileはネストして利用することになる。補足をすると、【ソース5】(1)のクラス名is-ancestorは「Tile element」の祖先要素となる。ネストして定義するclass属性tileの要素をラップする要素だ。また、mb-0は要素下部の余白を0にするmargin-bottom:0pxの意味だ。mb-0からmb-6まで設定できる。詳細は公式サイトを確認してほしい。

(4)の箇所のarticleタグに設定したclass属性is-child, boxは設定したタグの子要素をboxの見目で表示するために指定している。(5)の箇所はFont Awesomeのアイコンを表示するように記述した。

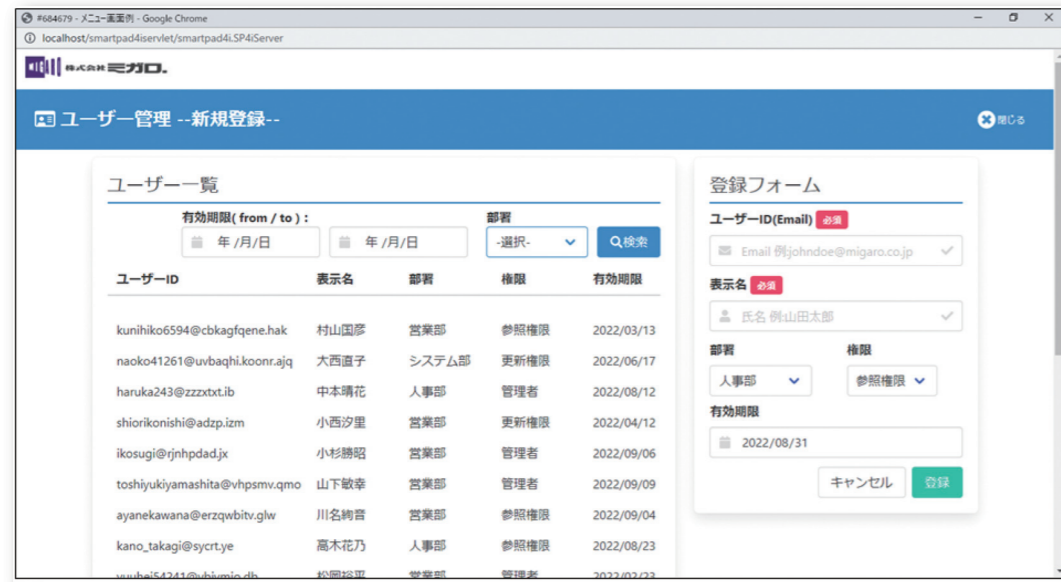
さらに、今回タイル要素をアプリケーションのリンク要素にするため、(3)の箇所にアンカータグ<a>を追加している。アンカータグのサイズ調整には、独自にbox-linkというクラスを定義した。独自に追加したCSSはこれだけで、後はBulmaの機能ですべて記述できた。Bulmaを利用することで、HTMLの見目は洗練され、レスポンシブデザインにも簡単に対応できる。さらに、CSSを記述する時間を大幅に短縮もできた。

また、Bulmaは入力欄のカスタマイズも容易で、洗練された入力画面を作成可能だ。次節では入力画面の作成例を紹介する。

5.2 入力欄の作成例

入力フォームはBulmaの公式サイト例が豊富なため作成に困ることはないだろう。入力欄の例として作成したのが【図10】だ。左側に一覧、右側に入力フォームがある。

図10 入力欄の例



入力欄にはclass属性名inputを設定するだけで、自動的に横幅が100%に設定され、エッジが角丸の入力欄に変化する。【図10】の入力欄はアイコン表示をカスタマイズしている

ため、記述の追加が必要となる。【図11】の赤枠の記述例が【ソース6】だ。

図11 入力欄の例



ソース6

図11の入力欄のHTML記述【HTML】

```

131 <div class="field">------(1)
132 <label class="label">ユーザーID(Email)------(2)
133 <span class="tag is-danger">必須</span>------(3)
134 </label>
135 <div class="control has-icons-left has-icons-right">------(4)
136 <input id="INP01" class="input" type="email"
137     placeholder="Email 例:johndoe@migaro.co.jp">
138 <span class="icon is-small is-left">------(5)
139 <i class="fas fa-envelope"></i>
140 </span>
141 <span class="icon is-small is-right">------(6)
142 <i class="fas fa-check"></i>
143 </span>
144 </div>
145 <p class="help is-danger error" id="ERR01"></p>------(7)
146 </div>

```

【ソース6】のclass属性名は全て、Bulmaで定義されている。まず、(1)のように入力欄をラップするように、<div>タグを記述してclass属性名にfieldを設定する。fieldは入力欄等、要素のシンプルなコンテナである。次に、(2)の箇所で、入力欄のラベルを表示するため<label>タグを追加している。<label>タグにはclass属性labelを設定する。(3)の行でラベルの横に、「必須」を表示しているが、これもclass属性名にtagを設定して、Bulmaの「tag element」の表示を使用している。一緒に設定している、is-dangerで色を赤色にできる。アイコンを表示するためには、<div>タグにて(4)のように

has-icons-left has-icons-rightを追加する必要がある。もちろん、左側のみにアイコンを表示したい場合はhas-icons-leftだけで構わない。入力欄の定義の後、(5)(6)のようにspanタグでiconを定義する。is-small is-leftはis-smallでアイコンのサイズ、is-leftでアイコンの位置を指定できる。また、(7)のid属性ERR01、<p>タグはRPG側からエラーを入力するために定義している。JavaScriptとの連携も必要となるが、未入力の場合には【図12】のように、わかりやすいエラーメッセージを表示することができる。

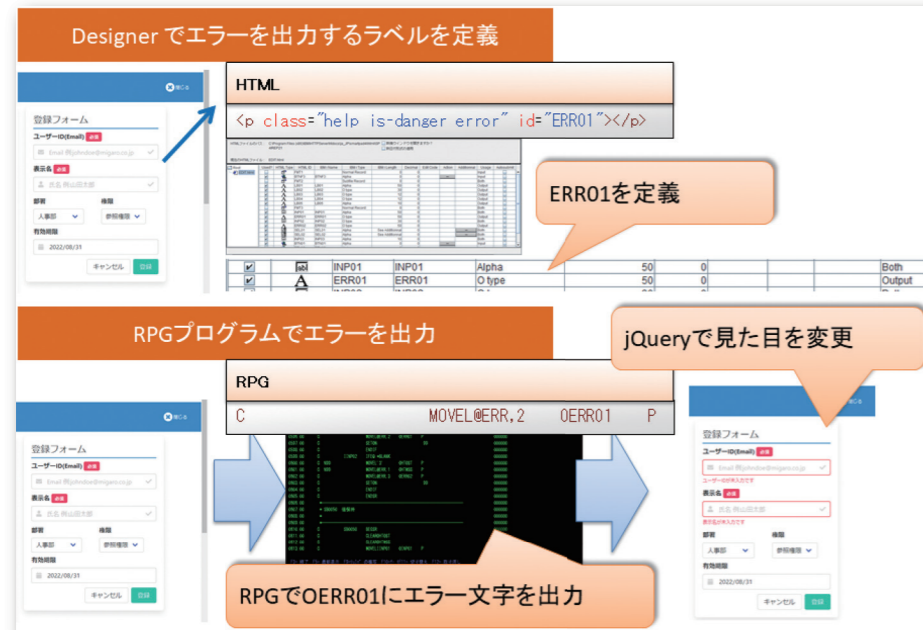
図12 エラー表示例



エラー表示の仕組みは、HTML側であらかじめエラーメッセージを表示するラベル(id属性ERR01)を定義しておき、Designerで属性を設定、RPGプログラムでエラーチェック

を行い、RPG側からラベルのフィールドのアウトプットフィールド(OERR01)へ文字列を設定している。【図13】

図13 エラー表示の仕組み



エラーの文字がRPG側で出力された場合に、JavaScriptを利用して画面の見た目を変更した。

実装例のJavaScriptでは、jQueryを利用している。jQueryについては2019年度のテクニカルレポート、「SmartPad4iのインターフェース機能拡張」に取り上げているため、ここでは割愛する。

【図12】のように表示するために、【ソース6】(7)の<p>タグに

class属性errorを追加している。エラーメッセージがRPG側で設定された場合に、JavaScriptでエラーメッセージを表示状態に変化させ、入力欄の枠を赤に変更する。JavaScriptの例は【ソース7】になる。

ソース7

```

エラー表示を実行するJavaScript例【JavaScript】
275 <script>
276 //HTMLが読み込まれた後に実行
277 $(function(){-----(1)
278 //class属性 error(エラーメッセージ)の要素でループ
279 $('<strong>.error</strong>').each(function(){-----(2)
280 //エラーメッセージが出力されている場合
281 if($(this).text()){-----(3)
282 //エラーメッセージをフェードイン表示後、入力欄に赤枠を設定
283 $(this).fadeIn().prev('div').find('select, input')
284     .css('border-color', '#FF0000'); -----(4)
285 }
286 });
287 </script>

```

jQueryでは、HTMLが読み込まれた後に処理を実行するために、(1)のような記述を行う。(2)はclass属性名errorの要素をjQueryのセレクト、\$()で取得後、eachメソッドで取得し

た要素をループしている。\$()はjQueryで要素を取得するための関数でセクタと呼ばれる。

\$('.error')と記述すると、class属性にerrorが設定された要素を全て取得してくれる。

(3)や(4)で使用されている、thisはeachメソッドでループしている対象の要素だ。つまり、error属性が設定された要素にエラー内容の文字列がRPGプログラムから出力されている場合には、fadeInメソッドでフェードイン表示させ、一つ前の<div>タグ要素内にある<select>タグ、<input>

タグのcss属性border-colorプロパティを赤色(#FF0000)に設定している。

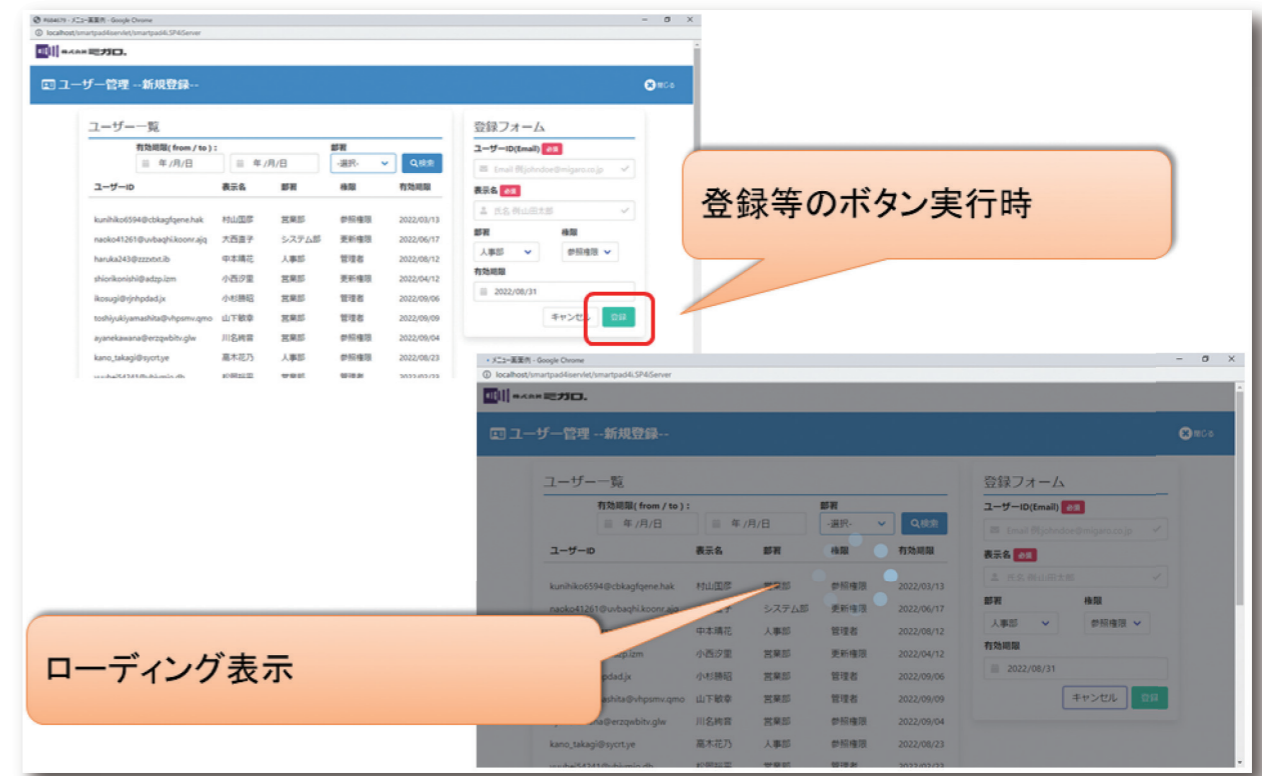
SmartPad4iでのHTML画面作成にBulmaを利用すると、短時間で洗練されたUIが作成できることが分かったと思う。その他、SmartPad4iアプリケーションに簡単な実装で取り入れられる機能をご紹介します。

6.お勧めのUIカスタマイズ(CSSローディング表示)

SmartPad4iをはじめとするWebアプリケーションは、「Webサーバへの要求(リクエスト)」→「Webサーバからの応答(レスポンス)」の流れで処理され、要求後に、応答を待つ必要がある。応答が返却される間の画面を工夫する

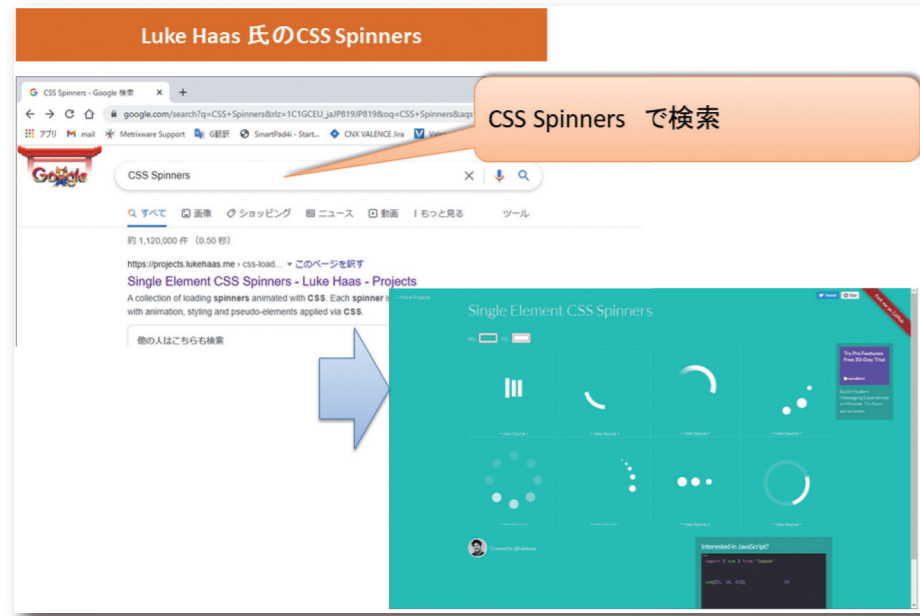
ことで、ユーザーの体感的な待ち時間軽減を図ることができる。例えば、多用するのが、ローディング画面の表示である。【図14】次に、SmartPad4iにおいてローディング画面を表示する方法を紹介する。

図14 ローディング表示



【図14】のローディング画面のアニメーションはCSSで実現している。CSSのアニメーションはLuke Haas氏のCSS Spinnersを使用した。【図15】

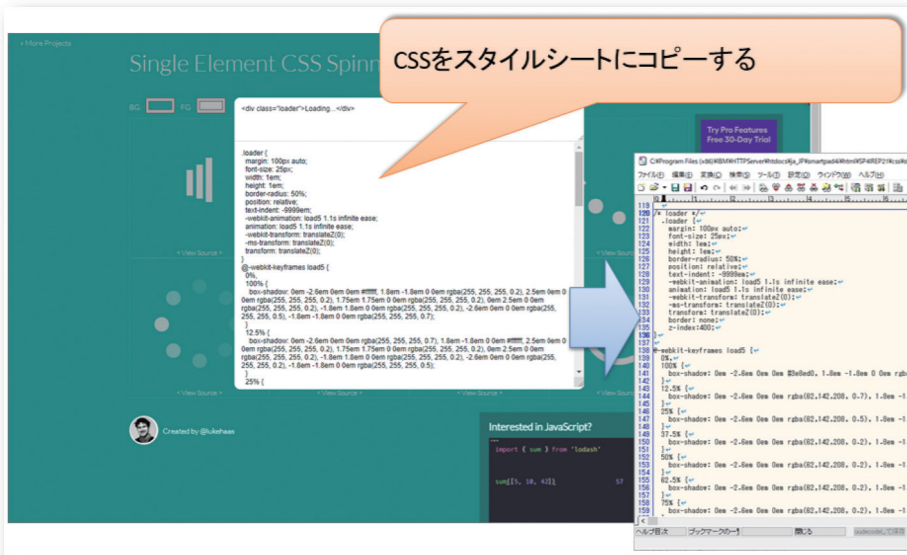
図 15 ローディングCSS Spinners



Luke Haas 氏のサイトでは動的にローディング画面のアニメーションが作成可能で、好きなデザインのローディングアニメーションを取得できる。ローディングアニメーションは

CSSのコードとして所得できるため、スタイルシートに張り付けて、HTML内にタグ<div class="loader">Loading...</div>を追加するだけで。【図16】

図 16 ローディングCSS Spinners



しかし、アニメーションはHTMLのタグを追加した箇所に表示されるため、アニメーションをラップする要素を作成して、画面中央に表示した。【ソース8】

ソース 8

HTML追加例【HTML】

```

16 <body>
17 <div class="loader-wrap" >
18 <div class="loader">Loading...</div>
19 </div>

```

divタグにclass名、loader-wrapを設定してclass名 loaderのタグを囲う

CSS追加例【CSS】

```

105 /* loaderを中央に表示*/
106 .loader-wrap {
107   position: fixed;
108   display: flex;
109   align-items: center;
110   justify-content: center;
111   width: 100%;
112   height: 100%;
113   top: 0px;
114   background-color:#000000;
115   opacity:0.5;
116   z-index:100;
117 }

```

wrapした要素のCSSを設定

【ソース8】の設定を行うことで、ローディング画面が画面中央に表示されるようになる。最後に、ローディング画面の表示、非表示をJavaScriptで制御する。【ソース9】

ソース 9

ローディング画面を表示するJavaScript例【JavaScript】

```

275 <script>
276 //HTMLが読み込まれた後に実行
277 $(function(){
278   //エラーメッセージ
279   ~【ソース8の箇所省略】~
280   //ローディングを非表示
281   $('<div class="loader-wrap">').fadeOut();
282 });
283
284 //IBMI送信時
285 function cansubmit(){
286   $('<div class="loader-wrap">').fadeIn();
287 }
288 </script>

```

HTML読み込み後、wrapした要素をフェードアウトで非表示

WEBサーバ送信前に実行される、SmartPad4iのcansubmit関数を利用してフェードインでローディング画面を表示

以上で、ローディング画面の実装が完了する。ローディング画面は簡単なCSS設定とJavaScriptで記述できるため、是非実装してほしい。

7.さいごに

本稿では、Bulma CSSフレームワークによるUI作成とレスポンシブ対応、WEBアイコンフォント、メニュー画面や入力画面の例、ローディング表示の画面カスタマイズを紹介した。数あるCSSフレームワークの中で、Bulmaは

SmartPad4i開発との相性がよいため、是非利用して洗練された、直感的に使用できるUIのアプリを作成してもらいたい。