

# プロジェクト仕様書

最終更新日: 2024/01/01

## 目次

- プロジェクト概要
- 対象ユーザー
- 機能要件
- 画面設計
- データベース設計
- API 設計
- 非機能要件
- 使用技術
- セキュリティ設計
- 更新履歴

## プロジェクト概要

### プロジェクト名

屋台配置図 管理デモアプリ（試作）

### 目的

30店舗の屋台を配置図（SVG）としてブラウザに表示し、各屋台をクリックすることで右側の詳細パネルから情報を編集できるWebアプリの試作版を作成する。

本アプリは **GitHub Pages** で公開する静的Webアプリ とし、バックエンドは使用しない（ローカル変数管理）。

### スケジュール

- 開発開始：2024/01/01

- リリース予定：未定

## 対象ユーザー

### 大学祭実行委員会（屋台配置・確認担当）

- いつ使うか：配置図の確認・調整時、大学祭準備期間中
- どういうときに使うか：屋台の状態や情報を素早く確認・修正したいとき

## 機能要件

### 1. 屋台配置図の表示

#### 概要：

屋台を SVG の矩形（rect）で表示し、配置図を可視化する。

#### 詳細仕様：

- 屋台数：合計30店舗
- <svg> 内に <rect> を用いて屋台を表現
- 各屋台は class="booth" を持つ
- 各屋台の id は B01 ~ B30
- 5列 × 6行のグリッド配置
- サイズ：width=80, height=60

#### 配置ルール：

Row1 (y=20): B01 B02 B03 B04 B05  
Row2 (y=100): B06 B07 B08 B09 B10  
Row3 (y=180): B11 B12 B13 B14 B15  
Row4 (y=260): B16 B17 B18 B19 B20  
Row5 (y=340): B21 B22 B23 B24 B25  
Row6 (y=420): B26 B27 B28 B29 B30

### 2. 屋台の状態と色分け

#### 状態定義：

- normal : 通常営業
- stop : 出店停止

色ルール :

状態	色
normal	#6fc46f (緑)
stop	#e25a5a (赤)

初期状態はすべて normal とする。

### 3. 右側編集パネル

表示項目 :

- 屋台ID (h2などで表示)
- 状態 (select)
  - 通常 (normal)
  - 停止 (stop)
- チェックリスト (checkbox × 3)
  - 火気確認
  - 衛生チェック
  - 価格掲示
- 価格 (number input)
- 責任者名 (text input)
- 保存ボタン

### 4. データ仕様 (script.js)

データ構造 :

```
const booths = {
  B01: {
    status: "normal",
    checklist: [false, false, false],
    price: "",
    person: ""
  },
  ...
};
```

永続化 :

- ・不要
- ・ページリロード時は初期化されてよい

## 5. 処理仕様

初期化処理：

- ・B01～B30 のデータを生成
- ・全屋台を normal 状態として色を反映

屋台クリック時：

- ・クリックされた <rect> の id を取得
- ・対応する屋台データを右パネルに反映
- ・保存ボタンの対象屋台を切り替える

保存処理：

- ・右パネルの入力内容をデータに反映
- ・屋台の色を更新
- ・保存完了を alert で通知

色更新関数：

```
function updateBoothColor(id) {  
  const elem = document.getElementById(id);  
  const status = booths[id].status;  
  
  if (status === "normal") {  
    elem.setAttribute("fill", "#6fc46f");  
  } else {  
    elem.setAttribute("fill", "#e25a5a");  
  }  
}
```

## 画面設計

### 1. メイン画面

画面名: BoothMapScreen

パス/識別子: /

## 表示項目:

- 左：屋台配置図（SVG）
- 右：屋台情報編集パネル

## 動作:

1. 屋台をクリック
2. 右パネルに屋台情報を表示
3. 情報を編集して保存
4. 状態に応じて色を更新

## データベース設計

使用しない（ローカル変数で管理）

## API 設計

使用しない（ローカルで完結）

## 非機能要件

- GitHub Pages で公開可能な静的構成であること
- ローカル環境・ブラウザのみで完結すること

## 使用技術

- HTML / CSS / JavaScript（バニラJS）
- フレームワーク不使用
- 単一ページ構成

# セキュリティ設計

- ・バックエンドを持たないため、外部通信なし
- ・個人情報を保存しない

## 更新履歴

日付	バージョン	更新内容	更新者
2024/01/01	1.0	初版作成	名前