GeoReferencer 機能仕様書

1. プロジェクト概要

1.1 アプリケーションの目的

GeoReferencerは、PNG画像（ハイキングマップなど）を国土地理院の地理院地図上に精密にジオリファレンス（地理的位置合わせ）することを専門とするWebアプリケーションです。最小二乗法による6パラメータアフィン変換技術を用いて、高精度な画像位置合わせを実現し、画像内座標をGPS座標に変換してGeoJSONなどの形式でエクスポートします。

1.2 主要機能

- **GPS座標データの読み込み・表示**（Excel形式）
- **PNG画像ファイルの読み込み・オーバーレイ表示**
- **JSONファイル連携による画像内座標データの読み込み**
- **精密アフィン変換によるジオリファレンス**（最小二乗法）
- **変換済みGPS座標データのGeoJSONエクスポート**
- **国土地理院APIによる標高データ取得**
- **自動ポイントマッチング機能**（IDベース）

1.3 技術的特徴

- **完全ES6モジュール構成**（モジュラーアーキテクチャ）
- **ローカル完結型**（サーバーレス、ファイルベース処理）
- **非同期初期化**（Promise-based確実な初期化）
- **精密座標変換**（Web Mercator ↔ WGS84）
- **標高データ取得**（国土地理院API連携）

2. システム構成

2.1 アーキテクチャ概要

GeoReferencerは完全なES6モジュール構成を採用し、機能別に分離されたコアモジュールからなるモジュラーアーキテクチャを採用しています。Firebaseなどの外部バックエンド依存を排除し、ローカルファイル操作を中心とした独立性の高い設計となっています。

2.2 モジュール構成と依存関係

```
GeoReferencerApp (app-main.js)
├── コア機能
│   ├── MapCore (map-core.js) [地図初期化・レイヤー管理]
│   ├── ImageOverlay (image-overlay.js) [画像オーバーレイ処理]
│   ├── GPSData (gps-data.js) [GPS/Excelデータ処理]
│   ├── Georeferencing (georeferencing.js) [精密アフィン変換処理]
│   │   └── AffineTransformation (affine-transformation.js) [アフィン変換計算]
│   ├── RouteSpotHandler (route-spot-handler.js) [ルート・スポットデータ管理]
```

```
│   ├── CoordinateDisplay (coordinate-display.js) [座標表示・マーカー管理]
│   ├── ElevationFetcher (elevation-fetcher.js) [標高データ取得]
│   └── DataImporter (data-importer.js) [データ読み込み統合]
└── ユーティリティ
    ├── UIHandlers (ui-handlers.js) [UI操作ハンドラー]
    ├── FileHandler (file-handler.js) [ファイル保存・管理]
    ├── Utils (utils.js) [ログ・エラーハンドリング]
    ├── MathUtils (math-utils.js) [数学計算・座標変換]
    └── Constants (constants.js) [定数定義]
```

2.3 外部依存関係

- **Leaflet.js v1.9.4**: 地図レンダリング（CDN経由）
- **SheetJS v0.18.5**: Excelファイル処理（CDN経由）
- **国土地理院タイル**: 地図データソース
- **国土地理院標高API**: 標高データソース

2.4 ファイル構成

```
GeoReferencer/
├── index.html                    # メインHTML
├── styles.css                    # 統合CSS（CSS変数使用）
├── README.md                     # プロジェクト概要
├── docs/                         # ドキュメント
│   ├── funcspec-202602.md        # 機能仕様書（本書）
│   └── UsersGuide-202602.md      # 利用者の手引
└── js/                           # JavaScriptモジュール
    ├── app-main.js               # メインアプリケーション
    ├── map-core.js               # 地図コア機能・レイヤー管理
    ├── image-overlay.js          # 画像オーバーレイ処理
    ├── gps-data.js               # GPS/Excelデータ処理
    ├── georeferencing.js         # 精密アフィン変換処理
    ├── affine-transformation.js  # アフィン変換計算専用
    ├── route-spot-handler.js     # ルート・スポットデータ管理
    ├── coordinate-display.js     # 座標表示・マーカー管理
    ├── elevation-fetcher.js      # 標高データ取得
    ├── data-importer.js          # データ読み込み機能
    ├── ui-handlers.js            # UI操作ハンドラー
    ├── file-handler.js           # ファイル処理統合
    ├── math-utils.js             # 数学・座標変換統合
    ├── utils.js                  # ユーティリティ・ログ機能
    └── constants.js              # 定数定義
```

3. 機能詳細

3.1 メインアプリケーション (GeoReferencerApp)

責任範囲: アプリケーション全体の初期化・統合・イベント管理

主要メソッド:

- `init()`: 非同期アプリケーション初期化
- `initializeModules()`: 各モジュールの依存関係を考慮した初期化
- `setupEventHandlers()`: UIイベント設定
- `handleMatchPoints()`: ジオリファレンス実行の統合処理
- `handleExportGeoJson()`: GPS変換済みデータのGeoJSON出力
- `handleFetchElevation()`: 標高データ取得処理

データフロー:

```
PNG読み込み → JSON読み込み（任意） → GPS Excel読み込み →
ジオリファレンス実行 → GPS変換 → 標高取得 → GeoJSON出力
```

3.2 データ読み込み機能 (DataImporter)

責任範囲: 各種ファイル（Excel, PNG, JSON）の読み込み処理

主要機能:

- **GPS Excel読み込み**: ポイント座標データの読み込み
- **PNG画像読み込み**: 地図画像の読み込み
- **JSON読み込み**: ルート・スポットなどの画像座標データの読み込み

3.3 地図コア機能 (MapCore)

責任範囲: Leaflet地図の初期化・専用ペイン管理・スケールコントロール

主要機能:

- **非同期初期化**: Promise-baseの確実な地図初期化
- **専用ペイン管理**: z-index制御によるレイヤー構造
- **コントロール配置**: ズームコントロール、スケールコントロール

3.4 画像オーバーレイ処理 (ImageOverlay)

責任範囲: PNG画像の読み込み・表示・境界計算・アフィン変換対応

主要機能:

- **画像読み込み**: PNG専用のFileReader処理
- **境界計算**: Mercator投影補正を考慮した精密境界計算
- **アフィン変換対応**: 変換行列に基づく表示位置更新

3.5 GPS/Excelデータ処理 (GPSData)

責任範囲: Excelファイルの読み込み・変換・地図表示

対応フォーマット:

- **Excel**: 必須列（ポイントID、名称、緯度、経度）、オプション列（標高、備考）
- **検証**: 座標範囲チェック、数値形式検証

3.6 精密アフィン変換処理 (Georeferencing)

責任範囲: 最小二乗法による6パラメータアフィン変換・精度計算・座標同期

技術仕様:

- **変換方式**: 最小二乗法による6パラメータアフィン変換
- **最小制御点数**: 3点以上（推奨: 4点以上）
- **精度評価**: 残差計算による誤差評価

3.7 標高データ取得 (ElevationFetcher)

責任範囲: 国土地理院APIから標高データ取得・マーカーへの設定

主要機能:

- **標高取得**: fetchElevation(lng, lat) - 国土地理院API呼び出し
- **一括取得**: ポイント、ルート、スポットの標高を一括で取得
- **レート制限**: 0.5秒待機（API制限対応）

3.8 ルート・スポットデータ管理 (RouteSpotHandler)

責任範囲: 画像座標データ（JSON）の管理・表示

主要機能:

- **データ保持**: ルート、スポット、ポイントのメモリ管理
- **マーカー表示**: 各種マーカーの地図表示
- **座標変換**: ジオリファレンス結果に基づくGPS座標計算

3.9 ファイル処理統合 (FileHandler)

責任範囲: ファイル読み込み・保存機能

機能:

- **読み込み**: Text, Binary, DataURL形式での読み込み
- **保存**: Blob作成とダウンロードリンク生成によるローカル保存
- **検証**: 読み込み時のデータ検証とフォーマット判定

4. ユーザーインターフェース

4.1 UI構成

- **地図エリア**: 画面全体に表示される国土地理院地図
- **制御パネル**: 左上固定の操作パネル（読み込み、マッチング、標高、保存）
  - **読み込みセクション**: GPS、PNG、JSONの選択と読み込み
  - **マッチングセクション**: ジオリファレンス実行と結果表示
  - **標高セクション**: 標高取得対象の選択と実行

- **メッセージエリア**: 画面上部の一時メッセージ表示（成功・エラー・進捗）

4.2 ファイル読み込み機能

ポイントGPS読み込み

- **形式**: Excel (.xlsx)
- **機能**: 基準となるGPS座標点の読み込み
- **フィードバック**: 読み込み成功時に、既存データとのマージ確認および詳細件数（ポイント数）を表示

PNG画像読み込み

- **形式**: PNG
- **機能**: 背景となる地図画像の読み込み
- **フィードバック**: 画像ファイル名を表示

JSONファイル読み込み

- **形式**: JSON（独自スキーマ）
- **機能**: 画像上のルート・スポット・ポイント座標定義の読み込み
- **フィードバック**: 読み込み完了時に、ポイント・ルート・スポット・エリアの各件数を詳細に表示

4.3 ジオリファレンス操作

- **実行ボタン**: 画像とGPSデータのマッチングと変換実行
- **結果表示**: マッチング数、誤差情報の表示

4.4 データ出力

- **GeoJSON保存**: 変換後の全データをGeoJSON形式でダウンロード
- **ファイル名規則**: `[画像名略称]-GPS-[YYYYMMDD].json`形式で自動生成（例: `map-GPS-20260214.json`）

5. データ構造

5.1 JSON入力フォーマット (points/routes/spots)

画像上の座標(x, y)を持つJSONデータ構造。

```
{
  "points": [
    { "id": "A-01", "x": 100, "y": 200, "name": "Point A" }
  ],
  "routes": [
    {
      "name": "Route 1",
      "waypoints": [{ "x": 10, "y": 20 }, { "x": 50, "y": 60 }]
    }
  ],
  "spots": [
```

```
    { "name": "Spot 1", "x": 300, "y": 400, "description": "Viewpoint" }
  ]
}
```

5.2 GeoJSON出力フォーマット

標準的なFeatureCollection形式。

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [135.0, 35.0, 100.0]
      },
      "properties": {
        "name": "Point A",
        "type": "point",
        "id": "A-01"
      }
    }
  ]
}
```

6. 制限事項

- **ファイル形式**: 画像はPNGのみ対応
- **ブラウザ**: ES6モジュール対応ブラウザ必須
- **API制限**: 国土地理院標高APIのアクセス制限準拠
- **座標系**: WGS84のみ対応

7. 改訂履歴

- **v2.0** (2026-02-13): Firebase依存を完全に排除し、ローカルファイルベースのフローに刷新。GeoJSONエクスポート機能を追加。
- **v1.0** (2025-12-01): 初版リリース（Firebase版）