

国土交通省

Project LINKS × 地域交通DX推進プロジェクト「COMmmONS(コモンズ)」

2025年度 公共交通計画策定支援ツール開発プロジェクト

地域公共交通計画策定支援ツール「LINKS Mobilys」技術検証レポート付録



Project LINKS



COMmmONS
by MLIT

地域公共交通計画策定支援ツール「LINKS Mobilys」 システム設計書

国土交通省 総合政策局 情報政策課/モビリティサービス推進課

No.012-01



目次

1. 開発スコープ	- 1 -
1-1. 概要	- 1 -
1-2. システムを利用する業務全体像とシステム利用フロー	- 3 -
2. 開発するシステム：機能要件（FN/SL/AL/CO/HW/IF/UI）	- 6 -
2-1. システム機能（FN）	- 6 -
2-1-1. システムアーキテクチャ	- 6 -
2-1-2. システム機能一覧	- 7 -
2-1-3. システム機能の詳細	- 10 -
2-1-4. ソフトウェア・ライブラリ（SL）の詳細	- 55 -
2-1-5. 数理モデル・アルゴリズム（AL）の詳細	- 66 -
2-2. システムコンポーネント（CO）	- 96 -
2-2-1. システムコンポーネント図	- 96 -
2-2-2. システムコンポーネント一覧	- 97 -
2-3. ハードウェア（HW）	- 98 -
2-3-1. ハードウェアアーキテクチャ	- 98 -
2-3-2. ハードウェア一覧	- 99 -
2-3-3. ハードウェアの詳細	- 100 -
2-4. データインターフェース（IF）	- 101 -
2-4-1. データアーキテクチャ	- 101 -
2-4-2. データインターフェース一覧	- 103 -
2-4-3. データインターフェースの詳細	- 105 -
2-5. ユーザーインターフェース（UI）	- 120 -
2-5-1. 画面遷移図	- 120 -
2-5-2. ユーザーインターフェース一覧	- 121 -
2-5-3. ユーザーインターフェースの詳細	- 123 -
3. 開発するシステム：非機能要件（NF）	- 156 -
3-1. 非機能要件一覧	- 156 -
3-2. 非機能要件の詳細	- 158 -
4. 実証調査に利用するデータ（DT）	- 162 -
4-1. 実証調査に利用するデータ一覧	- 162 -
4-2. 実証調査に利用するデータの詳細	- 163 -
用語集	- 172 -

1. 開発スコープ

1-1. 概要

GTFS データ及びモビリティ・データ（乗降実績データ等）を活用した地域公共交通計画の立案支援を行うシステムである地域公共交通計画策定支援ツール「LINKS Mobilys」を開発する。

本システムに実装する主な機能は以下のとおりである。

- GTFS データから取得した路線、標柱等を地図上に可視化し、駅ポイントを中心としたバッファや道路ネットワーク等を用いた到達圏分析
 - GTFS データから取得した運行情報に対して、駅の追加削除や時刻表の編集等を行い、編集した GTFS データの保存、出力
 - GTFS データに基づいた運行記録（日報）帳票作成ツールの手入力シートと登録シートを作成、出力し、運行記録と GTFS データのルート ID や便名を適切にひも付け
 - IC カードシステムから入手した乗降実績データを投入して乗降実績や OD 輸送量を可視化
 - サービスレベルの変化に応じた公共交通分担率の変化のシミュレーション
- これらの機能を含めたシステム実装に先立ち、以下のアルゴリズムを構築する。
- 道路ネットワークを用いた到達圏分析アルゴリズム
 - 交通分担率の変化シミュレーションアルゴリズム
 - 乗降実績データと GTFS のひも付けアルゴリズム
 - インポートした GTFS の編集
 - 運行記録（日報）の作成
 - 路線と停留所によるグループ化

本システムは、2024 年度 Project LINKS で実施した「公共交通分野のオープンデータ化及び活用の促進に関する実証調査業務（<https://www.mlit.go.jp/links/use-cases/1896.html>）」の技術検証レポートを参考に作成した。

道路ネットワークを用いた到達圏分析は、GTFS から取得した駅ポイント及び道路ネットワーク分析により、駅ポイントから到達圏域を時間別に地図に表示する機能に使用する。

インプットデータにデジタル道路地図（DRM）又は OpenStreetMap（OSM）を活用したネットワーク分析を利用することで、徒歩→公共交通→徒歩などの複合的な交通手段を踏まえた解析を実施し、指定した時間内に到達可能な地域を可視化する。指定時間内に到達可能な地域と、人口データ等を重ね合わせ、到達圏の拡大面積や対象人口増加率を算出し、改善効果を定量的に提示することが可能となる。道路ネットワーク解析に当たっては、OpenTripPlanner などの OSS ライブラリを利用する。

交通分担率の変化シミュレーションは、GTFS データ及び乗降集計データをインプットデータとして、サービスレベルの変化に応じて公共交通分担率の変化のシミュレーションを行うものである。シミュレーションを行うためのパラメータは、運行経費、増便感度、減便感度、自家用車交通量等とし、断面交通量の変化及びこれに基づく便益をアウトプットとする。

エクセルや紙で自治体や事業者ごとの様式で整理された運行記録データや運行記録（日報）を、本プロジェクトで作成した運行記録（日報）帳票作成ツールから出力した様式に入力することで、乗降集計データを作成し、本システムへ入力することが可能となる。「乗降実績データ標準仕様書（鉄道・バス）」に準拠した乗降実績データは、本システムへ取り込み、システム内で GTFS とひも付け、乗降集計データや駅・バス停間 OD データを作成することが可能となる。これらのデータを本システム上で可視化・分析することができる。

1-2. システムを利用する業務全体像とシステム利用フロー

1. 業務フロー

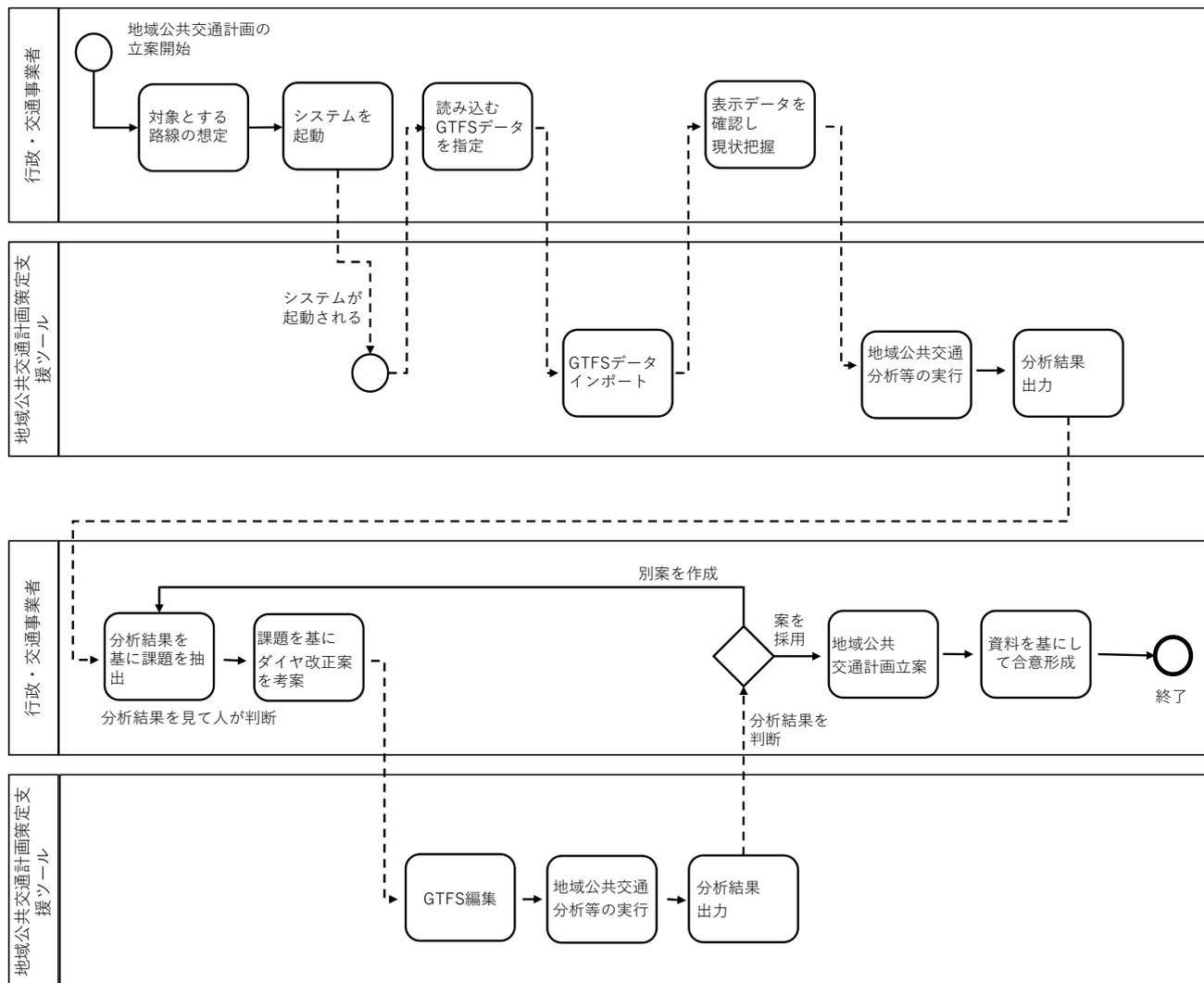


図 1-1 業務フローの全体像

2. システムシーケンス図

① 現状把握～課題抽出

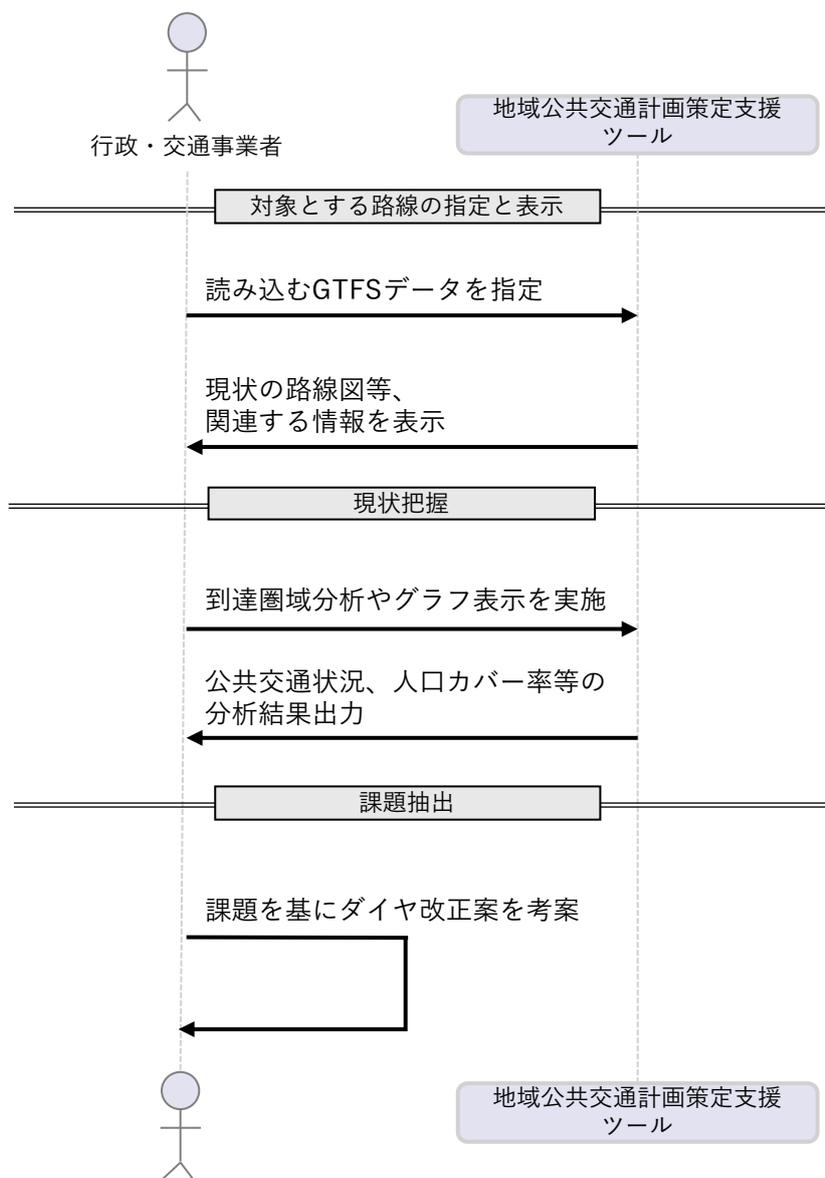


図 1-2 現状把握、分析、課題抽出

② 課題に基づいた計画立案

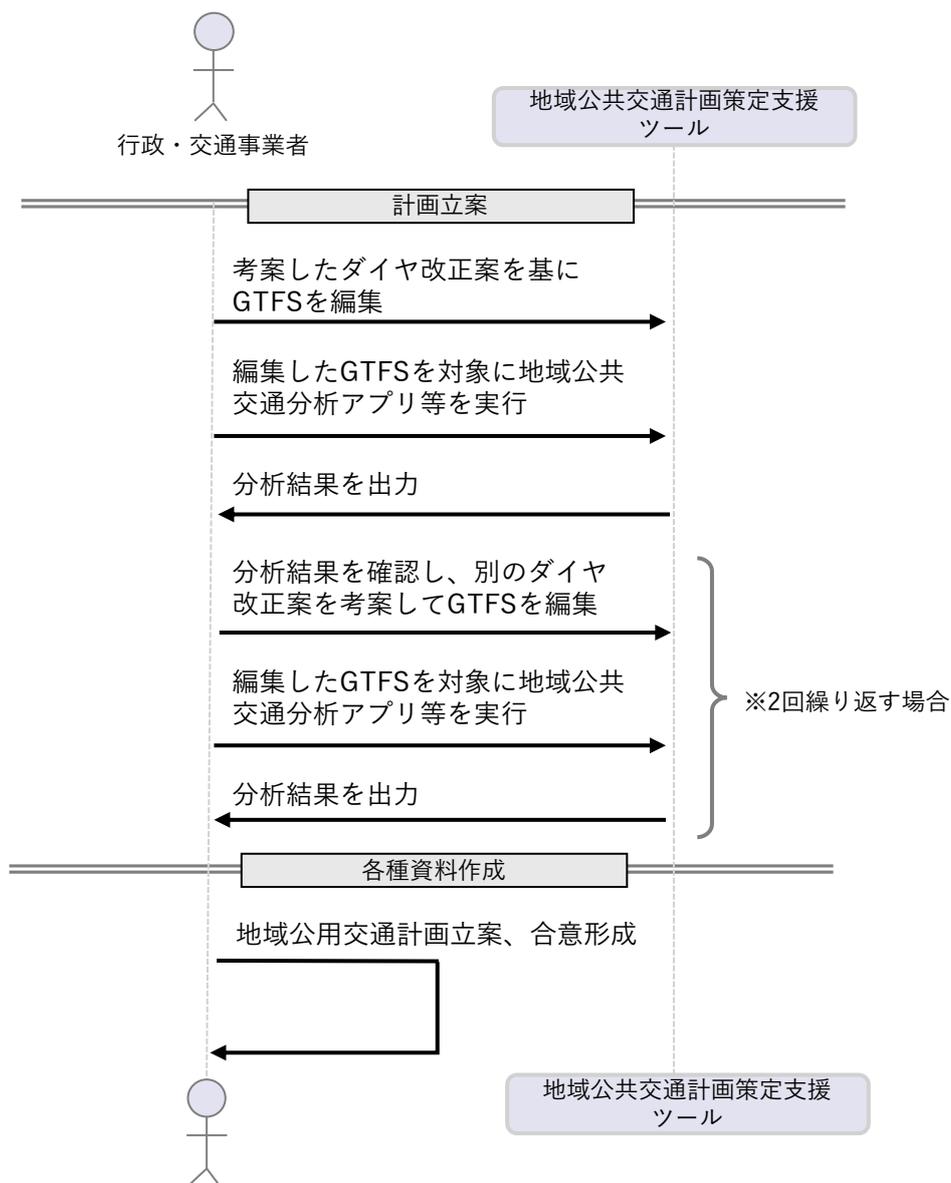


図 1-3 計画立案、資料作成、合意形成

2. 開発するシステム：機能要件（FN/SL/AL/CO/HW/IF/UI）

2-1. システム機能（FN）

2-1-1. システムアーキテクチャ

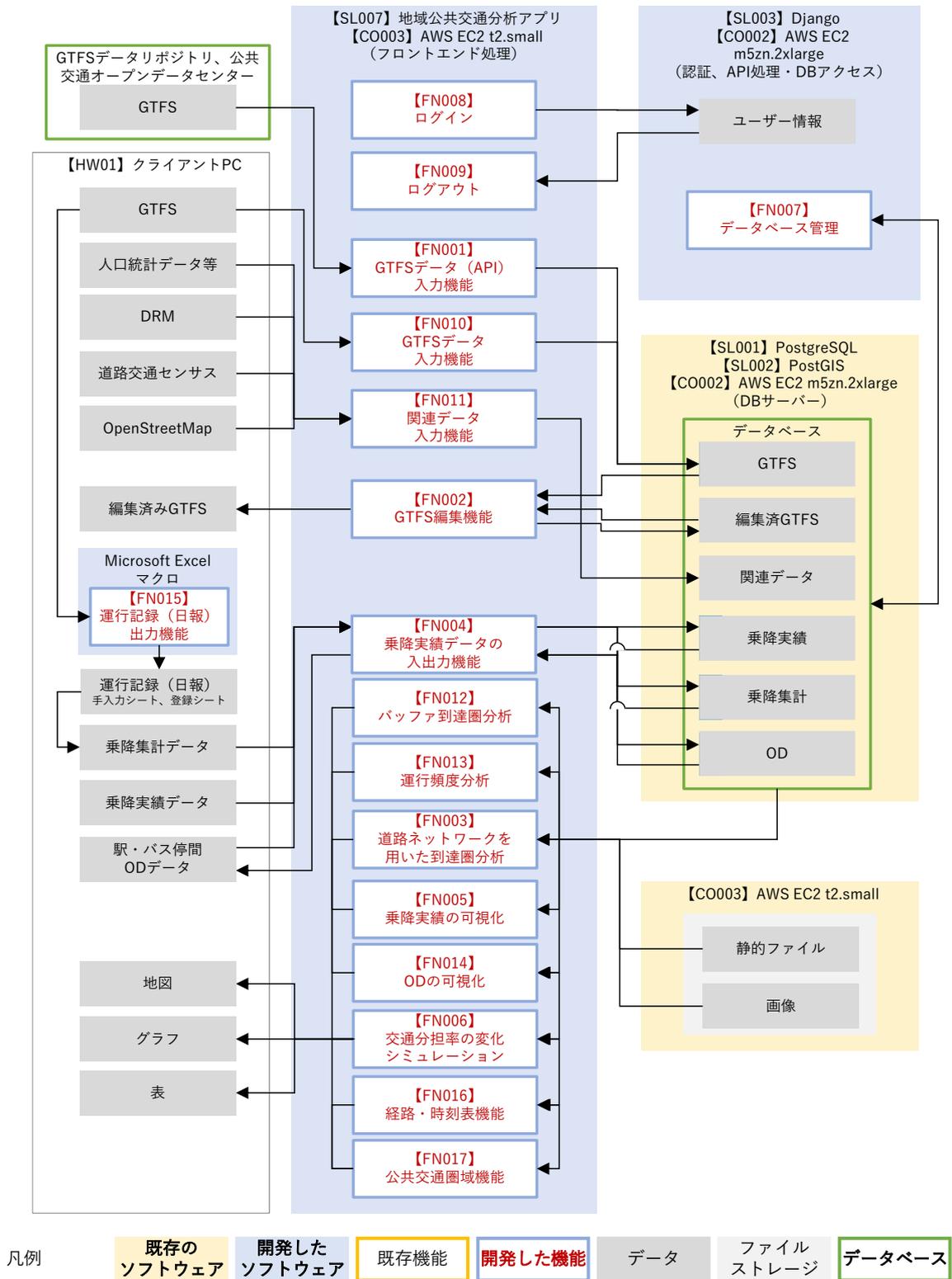


図 2-1 システムアーキテクチャ

2-1-2. システム機能一覧

表 2-1 システム機能一覧

※朱文字：新規開発

ID	機能名	機能説明
FN001	GTFS データ (API) 入力機能	<ul style="list-style-type: none"> ● GTFS データリポジトリや公共交通オープンデータセンターから API を利用して GTFS データをインポートする。 ● ローカル環境から GTFS データをインポートする。 ● インポートした GTFS データは一定期間プリセットとして保持する。
FN002	GTFS 編集機能	<ul style="list-style-type: none"> ● GTFS から取得した運行情報（駅、時刻表、路線等）を画面上で編集する。 ● 簡易的な操作手順で増減率または増減数の指定で便数のみを編集する。 ● 路線・便・停留所の追加・変更削除を含む GTFS の各項目を詳細に編集する。 ● 編集した運行情報を GTFS 形式で出力する。
FN003	道路ネットワークを用いた到達圏分析	<ul style="list-style-type: none"> ● DRM 又は OpenStreetMap を利用した道路ネットワーク分析による到達圏域を可視化する。 ● 任意の時間内に到達可能な到達圏域（道路ネットワーク分析）の可視化を行う。 ● 関連するオープンデータ（人口統計等）を地図上に可視化する。 ● アクセス可能な停留所、アクセス可能な施設、所要時間別到達圏域内の人口（カバー人口）を集計し、画面に表示する。 ● 地図やグラフを図（PNG 形式）、表をデータ（CSV 形式）でエクスポートする。
FN004	乗降実績データの入出力機能	<ul style="list-style-type: none"> ● ローカル環境から乗降集計データ、駅・バス停間 OD データ、乗降実績データをインポートする。 ● 乗降実績データを読み込み乗降集計データ、駅・バス停間 OD データを生成し、データベースやファイルに保存する。 ● 乗降実績データのインポートにあたっては、乗降実績データの情報を補完する。乗車・降車日時と停留所の情報からどの便に乗車したかを推定する。推定した便の情報から乗降集計データと駅・バス停間 OD データの作成に必要な GTFS の情報をひも付ける。 ● インポートした乗降実績データから、GTFS とひも付けた路線別の OD 輸送量を集計する。 ● インポートした乗降実績データから、集計した乗降集計データ

		<p>と駅・バス停間 OD データを作成する。</p> <ul style="list-style-type: none"> ● アカウント情報等とひも付けることでユーザー単位のセグメントを可能とする。
FN005	乗降実績の可視化	<ul style="list-style-type: none"> ● 乗降集計データに基づき、駅ごと又は複数系統を統合した系統ごとの乗降人数を集計して地図上に表示する。 ● 乗降集計データに基づき、停留所間の乗降人数の実績を集計して地図上にラインで表示する。 ● 地図上の路線の乗車人数、降車人数、通過人数をグラフで表示する。 ● 駅や停留所ごとの乗降人数をグラフで表示する。 ● 日単位だけでなく、路線単位を踏まえた分析を可能とする。 ● 複数系統を統合した系統ごとの集計も可能とする。 ● 地図やグラフを図 (PNG 形式)、表をデータ (CSV 形式) でエクスポートする。
FN006	交通分担率の変化シミュレーション	<ul style="list-style-type: none"> ● GTFS 及び乗降集計データを入力として、サービスの変化に応じた公共交通分担率の変化をシミュレーションする。
FN007	データベース管理	<ul style="list-style-type: none"> ● 格納されたデータを一元管理する。 ● データはアカウント情報とひも付けることで、当該アカウントに属するデータのみを参照・更新対象とする。 ● データベースのバックアップを取得する。
FN008	ログイン	<ul style="list-style-type: none"> ● 入力された ID/パスワードをサーバーに送信し、ログインを要求する。 ● 受け取った ID/パスワードを基に、データベース内のユーザー情報と照合し、認証可否を判定する。
FN009	ログアウト	<ul style="list-style-type: none"> ● アプリからログアウトする。
FN010	GTFS データ入力機能	<ul style="list-style-type: none"> ● ローカル環境に保存している GTFS データを入力する。 ● 選択した GTFS ファイルをドラッグ&ドロップ操作でデータをアップロード可能とする。 ● インポートした GTFS データは、アカウント管理機能等と連携することでプリセットとして保持する。
FN011	関連データ入力機能	<ul style="list-style-type: none"> ● 施設データをインポートする。 ● オープンデータ (人口統計) や DRM データ等をインポートする。
FN012	バッファ到達圏分析	<ul style="list-style-type: none"> ● 選択したポイントから任意の時間内に到達可能な、バッファ分析による到達圏域を可視化する。 ● GTFS データを地図上で可視化する。 ● 関連するオープンデータ (人口統計等) を地図上に可視化する。 ● アクセス可能な停留所、アクセス可能な施設、所要時間別到達圏域内の人口 (カバー人口) を集計し、画面に表示する。

		<ul style="list-style-type: none"> ● 地図やグラフを図（PNG 形式）、表をデータ（CSV 形式）でエクスポートする。
FN013	運行頻度分析	<ul style="list-style-type: none"> ● 選択した駅区間別の運行本数を集計して地図上に可視化する。 ● GTFS データを地図上で可視化する。 ● 関連するオープンデータ（人口統計等）を地図上に可視化する。 ● 時間帯別運行本数、路線別運行本数、停留所別運行本数を画面に表示する。 ● 地図やグラフを図（PNG 形式）、表をデータ（CSV 形式）でエクスポートする。
FN014	OD の可視化	<ul style="list-style-type: none"> ● 駅・バス停間 OD データを地図上にラインで表示する。 ● 停留所別の乗車人数、降車人数、利用者数（起点・終点）を表で表示する。 ● 指定した停留所を起点として終点別の降車人数を表で表示する。 ● 指定した停留所を終点として起点別の乗車人数を表で表示する。 ● 乗車停留所、降車停留所、輸送量の一覧を表で表示する。 ● 地図や表を図（PNG 形式）、データ（CSV 形式）でエクスポートする。
FN015	運行記録（日報）出力機能	<ul style="list-style-type: none"> ● ローカル環境に保存している GTFS データを運行記録（日報）帳票作成ツールに取り込む。 ● 運行記録（日報）手入力シートを出力する。 ● 運行記録（日報）登録シートを出力する。 ● 運行記録（日報）登録シートに記載した乗降数を乗降集計データとして出力する。
FN016	経路・時刻表機能	<ul style="list-style-type: none"> ● GTFS データの路線図を地図上に可視化する。 ● 停留所別の時刻表を画面上に表示する。 ● 地図と時刻表を図（PNG）形式でエクスポートする。
FN017	公共交通圏域機能	<ul style="list-style-type: none"> ● GTFS データの公共交通路線の全ての停留所から指定した距離（徒歩圏）内の範囲を「公共交通圏域」として地図上に可視化する。 ● 公共交通圏域内の施設タイプ別分布のグラフと表を画面に表示する。 ● 公共交通圏域内の年代別人口数（カバー人口）のグラフを画面上に表示する。 ● 地図やグラフを図（PNG 形式）、表をデータ（CSV 形式）でエクスポートする。

2-1-3. システム機能の詳細

開発機能の詳細要件を記す。なお、本業務において開発や改修を行う内容については、**朱文字**で示す。

【FN001】GTFS データ (API) 入力機能<新規開発>

- 本システム機能の概要
 - アカウント単位で、システム利用に必要なデータを入力可能とする
 - GTFS データリポジトリなどの外部 API を通じて GTFS データを取得・データベースへ格納する
- 本システム機能の入力・処理・出力のフローチャート

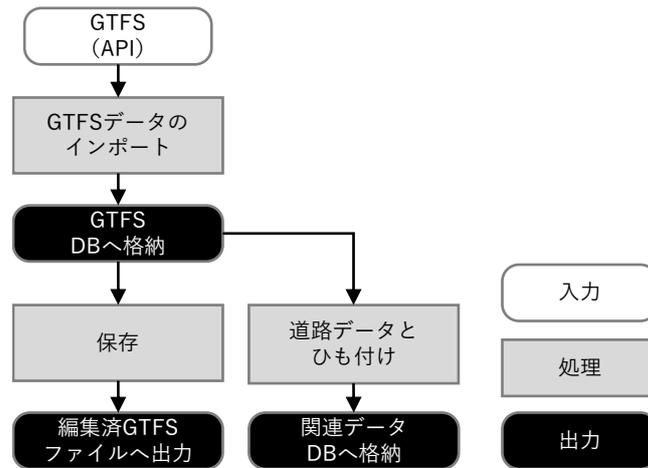


図 2-2 【FN001】のフローチャート

- 本システム機能の処理の詳細
 - GTFS データのインポート
 - ◇ 処理内容
 - データインポート画面で取得する GTFS データを指定し、外部サイトから API を通じて指定した条件に合致する GTFS データを取得する
 - インポートした GTFS データは一定期間プリセットとして保持する
 - 取得・指定した GTFS データの入力内容を確認し、不足データを補完する等の整備を行う
 - 整備した GTFS データのシナリオ名をつけてデータベースに保存する
 - インポートする際に路線と停留所によるグループ化を行う
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - 【AL106】路線と停留所によるグループ化
 - 道路データとひも付け
 - ◇ 処理内容
 - 【FN011】において読み込んだ道路データと、シナリオ名をつけて DB に格納した

GTFS データを用いて、OpenTripPlanner の道路ネットワーク解析を行うためのグラフオブジェクトを生成する。

- ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - OpenTripPlanner (ソフトウェア・ライブラリ【SL006】を参照)
- ◇ 利用するアルゴリズム
 - なし

➤ 保存

- ◇ 処理内容
 - データベースのデータを保存する。
 - 保存時に GTFS のバリデーションチェックを実行する。
 - ZIP ファイル名をシナリオ名のとおり保存する。
- ◇ 利用するライブラリ
 - なし
- ◇ 利用するアルゴリズム
 - なし

● 本システム機能の入出力データの仕様

➤ 入力

- ◇ GTFS データ
 - 内容
 - GTFS Schedule (静的) データを外部サイト (GTFS データリポジトリ) から取得する。
 - 形式
 - TXT 形式
 - 利用するデータインターフェース
 - データインターフェース【IF002】GTFS 入力 IF (API) を参照

➤ 出力

- ◇ GTFS データ
 - 内容
 - 入力した GTFS データをデータベースに格納する。
 - 形式
 - データベースに格納
 - 利用するデータインターフェース
 - データインターフェース【IF017】GTFS データ入力 IF を参照
- ◇ 道路ネットワークのグラフオブジェクト
 - 内容
 - GTFS と OpenStreetMap あるいは DRM の道路データから生成した道路ネット

ワークのグラフオブジェクトをデータベースに格納する

- 形式
 - .pbf 形式
- 利用するデータインターフェース
 - データインターフェース【IF017】GTFS データ入力 IF を参照

【FN002】GTFS 編集機能<新規開発>

- 本システム機能の概要
 - 【FN001】【FN010】で読み込んだ GTFS データに含まれるルートや停留所・便数の要素を画面上で直接編集し、保存する
- 本システム機能の入力・処理・出力のフローチャート

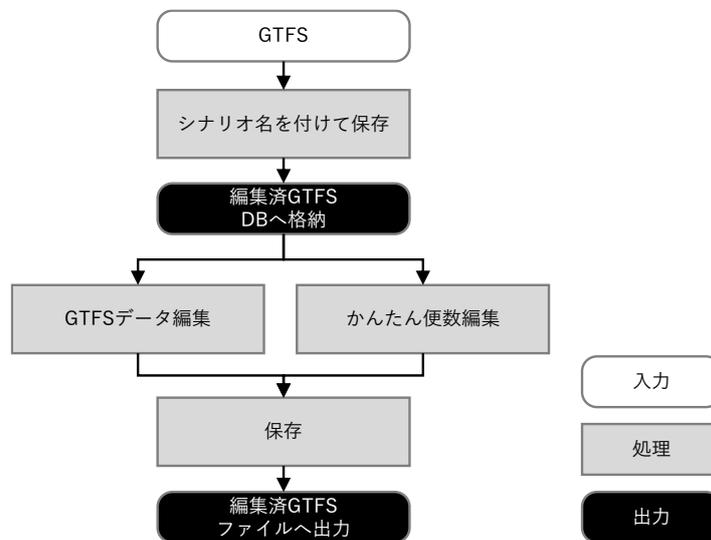


図 2-3 【FN002】のフローチャート

- 本システム機能の処理の詳細
 - シナリオ名をつけて保存
 - ◇ 処理内容
 - 編集した GTFS データにシナリオ名を付けてデータベースに保存する。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - なし
 - GTFS データ編集
 - ◇ 処理内容
 - 運行パターン編集
 - ルートパターン取得
 - ◇ 特定のシナリオにおける全ルートパターンを取得する。
 - ◇ stop_id、stop_name、service_id、agency_id、パターンメタデータ (運行回

- 数、間隔、行き先表示)を含む。
- ルートパターン作成
 - ◇ 新規ルートとその最初のパターンを作成する。
 - ◇ route_id 及びパターン特性に対する重複検証を行う。
 - ルートパターン削除
 - ◇ 1つ以上のルートパターンを削除する。
 - ◇ パターン削除後に該当ルートに運行が残っていない場合、自動的にルートを削除する。
 - ルートパターン更新
 - ◇ 既存ルートパターンの停留所順序を更新する。
 - ◇ パターン条件に一致する全運行を変更する。
 - 既存ルートへのパターン追加
 - ◇ 既存ルートに新規パターンを追加する。
 - ◇ ルートの存在確認とパターン重複チェックを行う。
 - フィード基本情報の編集と運行日の編集
 - シナリオ編集データの取得
 - ◇ フィード情報、カレンダー、カレンダー日付を含むシナリオ編集用データを取得する。
 - シナリオ更新
 - ◇ シナリオのメタデータ、フィード情報、カレンダーデータを更新する。
 - ◇ 便 ID で使用しているカレンダーエントリに対する検証を行う。
 - 停留所・標柱の編集
 - 新規停留所作成
 - ◇ 入力されたグループ ID と停留所名を基に、名前グループ、ID グループ、stop、グループひも付け、を作成する。
 - 停留所の一覧取得
 - ◇ 指定シナリオのグルーピング状態と、stop 名称グループ、stopID グループにひも付く stop 一覧を返す。
 - 標柱の複製
 - ◇ 指定シナリオに stop を新規作成する。
 - ◇ parent_stop_id が指定された場合は、親 stop のグループを継承する。それ以外の場合は、stop_name_group_id と stop_id_group_id を指定する。
 - ◇ translations があれば、データベースに新規作成または上書き更新する。
 - 標柱の編集
 - ◇ 指定した stop の緯度・経度を更新する。
 - 標柱の削除
 - ◇ 指定シナリオ上の stop を削除する。
 - ◇ 当該 stop が属していた名前グループ/ID グループの中心座標の緯度・経度を再計算し、グループに stop が残っていなければグループ自体を削除する。
 - ◇ シナリオの編集状態を更新する。

- 路線区間の短縮
 - ルートパターンの取得
 - ◇ 特定のシナリオにおける全ルートパターンを取得する。stop_id、stop_name、service_id、agency_id、パターンメタデータ（運行回数、間隔、時間）を含む。
 - ルートパターンの更新
 - ◇ 既存ルートパターンの標柱の順序を更新する。
 - ◇ パターン条件に合致する全運行を変更する。
- 便の編集
 - 便の一覧の取得
 - ◇ 指定した GTFS シナリオにおいて、路線ごとにまとめられたトリップの一覧を取得する。
 - 便の詳細取得
 - ◇ 特定のトリップの詳細を取得する。
 - ◇ トリップに関連する全ての stop_times を含む。
 - 便の編集
 - ◇ 指定されたシナリオと trip_id に対して、既存の便の情報を更新する。
 - ◇ service_id、direction_id、stop_times などの内容を変更する。
 - 新規の便作成
 - ◇ 指定されたシナリオの既存のルートの下に、新規の便（trip）を作成する。
 - 便の削除
 - ◇ 指定されたシナリオから、複数の便それに関連する stop_times をまとめて削除する。
 - 経路のプレビュー
 - ◇ 指定されたシナリオで、選択された停留所間の経路のプレビューを作成する。
 - カレンダー一覧の取得
 - ◇ シナリオに関連するカレンダー設定の一覧を取得する。
- ルート編集
 - ルート生成
 - ◇ GTFS 交通データ用のルートを作成する。
 - ◇ 「シナリオ内の全運行と路線のルート生成」「特定の運行のルート生成」「特定の路線のルート生成」の3つの動作モードで操作する。
 - ◇ ルート生成には OSRM ルートサービスを使用し、停留所間の正確な道路追従経路を作成する。
 - 停留所からのルート生成
 - ◇ 停留所のリストからルートポイントを生成する。
 - 座標からのルート生成
 - ◇ 停留所情報を使わずに座標の一覧からルートポイントを生成する。
 - ルート一括更新
 - ◇ ルートレコードの一括編集を行う。
 - ◇ 部分更新、完全更新、欠落しているルートポイントの作成を行う。

- ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
- ◇ 利用するアルゴリズム
 - なし
- かんたん便数 GTFS データ編集
 - ◇ 処理内容
 - データの取得
 - シナリオを指定してルートごとの運行パターンを取得する。
 - 各パターンでは、始発・終点の stop_name、trip_headsign、direction_id、service_ID、shape_id、運行本数 (interval) を集計して表示する。
 - 運行本数の変更
 - 同じ route_id、direction_id、service_id の便の運行本数 (interval) をまとめて追加または削除する。
 - 既存の路線ごとの運行本数に対し、増減率 (2.0 倍、0.5 倍等) 又は増減数 (+4 本、-10 本等) のいずれかを入力可能とする。
 - 入力した倍数または増減数に応じて、既存の運行本数に比例した便数を配分する。(数理モデル・アルゴリズム【AL104】インポートした GTFS の編集を参照)
 - データ詳細の取得
 - 指定したシナリオ、route_id、service_id に合う便の一覧 (trip) を返す。
 - 各便の始発停留所の出発時刻 (stop_sequence=1)、trip_id、route_id、service_id、trip_headsign、shape_id を返す。
 - 結果は service_id→departure_time の順でソートする。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - 【AL104】インポートした GTFS の編集
- 保存
 - ◇ 処理内容
 - データベースのデータを保存する。
 - 保存時に GTFS のバリデーションチェックを実行する。
 - ZIP ファイル名をシナリオ名のとおり保存する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし

- 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS データ
 - 内容
 - 【FN001】【FN010】で取得した GTFS データ
 - 形式
 - データベース形式
 - 利用するデータインターフェース
 - データインターフェース【IF018】GTFS DB 入出力 IF を参照
 - 出力
 - ◇ 編集済 GTFS データ
 - 内容
 - 編集後の GTFS データ
 - 形式
 - データベースに格納
 - TXT 形式
 - 利用するデータインターフェース
 - データインターフェース【IF018】GTFS DB 入出力 IF を参照
 - データインターフェース【IF009】編集済み GTFS 出力 IF を参照

【FN003】 道路ネットワークを用いた到達圏分析<新規開発>

- 本システム機能の概要
 - 【FN001】【FN010】で読み込んだ GTFS データ及び【FN002】で編集した GTFS データを地図上で可視化する
 - 任意の時間内に到達可能な到達圏域（道路ネットワーク分析）の可視化を行う
 - 人口統計のメッシュデータの重ね合わせの ON/OFF を可能とする。
- 本システム機能の入力・処理・出力のフローチャート

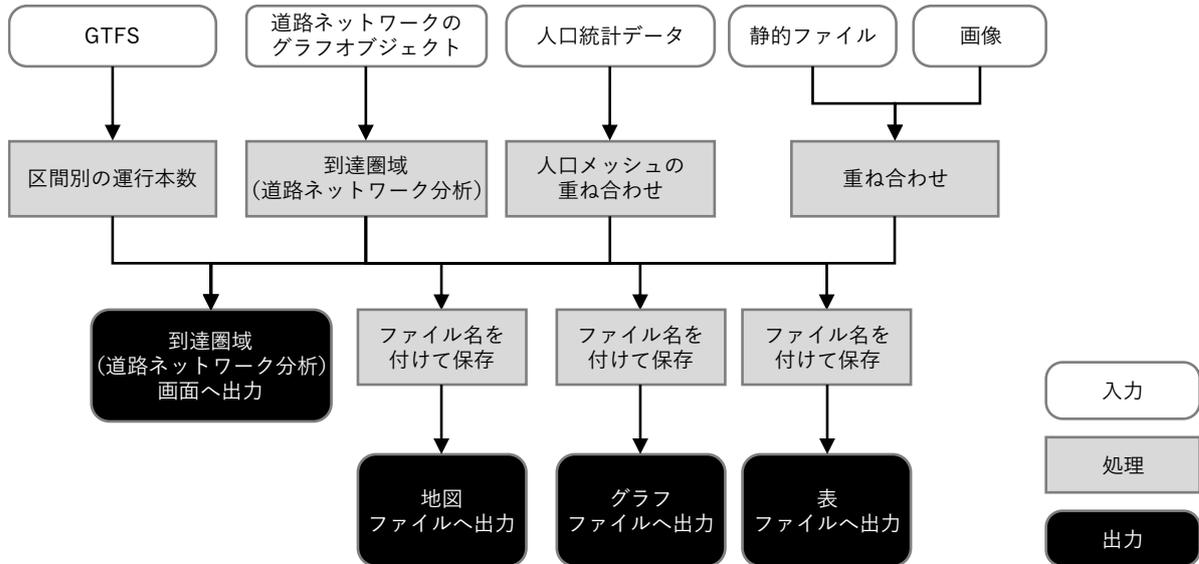


図 2-4 【FN003】 のフローチャート

- 本システム機能の処理の詳細
 - 到達圏域（道路ネットワーク分析）
 - ◇ 処理内容
 - 【FN001】【FN010】で GTFS データと道路データから生成した道路ネットワークのグラフを使用し、ユーザーが地図上で指定した地点（緯度・経度）から、徒歩と公共交通で到達可能なエリアをネットワーク分析で算出する
 - GTFS データが保有する停留所の緯度経度情報及び発着時刻等のデータを使用し、交通の乗換えを考慮した到達圏を算出する。
 - 任意の時間内に到達可能な到達圏域を地図上に表示する。
 - 任意の時間内に到達できる主要施設（病院、学校、商業施設など）を表示する。
 - 任意の時間内に到達できる到達圏域内の人口（カバー人口）を年齢層別に集計して表示する。
 - ◇ 利用するライブラリ
 - PostgreSQL（ソフトウェア・ライブラリ【SL001】を参照）
 - PostGIS（ソフトウェア・ライブラリ【SL002】を参照）
 - Django（ソフトウェア・ライブラリ【SL003】を参照）
 - Leaflet（ソフトウェア・ライブラリ【SL004】を参照）

- React.js (ソフトウェア・ライブラリ【SL005】を参照)
- OpenTripPlanner (ソフトウェア・ライブラリ【SL006】を参照)
- ◇ 利用するアルゴリズム
 - 【AL101】道路ネットワークを用いた到達圏分析アルゴリズムを参照
- 人口メッシュの重ね合わせ
 - ◇ 処理内容
 - 人口メッシュデータを地図上に表示する。
 - 到達圏域内の総人口を算出する。到達時間 10 分単位 (例: 10 分圏、20 分圏、30 分圏 …) の到達圏ポリゴンを作成し、各到達圏ポリゴンと 250mメッシュ人口データのメッシュポリゴンを重ね合わせた空間重ね合わせ処理を行い、到達圏ごとの人口集計グラフを表示する。
 - 各メッシュポリゴンについて、次の判定・処理を行う。
 - ◇ 到達圏ポリゴンに完全に含まれる場合、メッシュ人口をそのまま加算
 - ◇ 到達圏ポリゴンに一部のみ含まれる場合は、重なり面積比を算出し、メッシュ人口を面積比で案分して加算
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - Leaflet (ソフトウェア・ライブラリ【SL004】を参照)
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
 - ◇ 利用するアルゴリズム
 - なし
 - ファイル名を付けて保存
 - ◇ 処理内容
 - ファイルに名前を付けてデータを保存する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
 - 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS データ
 - データの内容
 - 【FN001】【FN010】で入力あるいは【FN002】で編集した GTFS 形式のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース

- データインターフェース【IF020】可視化用 DB 出力 IF を参照
- ◇ 道路ネットワークのグラフオブジェクト
 - 内容
 - 【FN011】で OpenStreetMap あるいは DRM の道路データから生成した道路ネットワーク
 - 形式
 - グラフオブジェクト
 - 利用するデータインターフェース
 - データインターフェース【IF020】可視化用 DB 出力 IF を参照
- ◇ 人口統計データ
 - 内容
 - 2020 年の国勢調査に基づく 250mメッシュごとに集計した人口
 - 形式
 - JSON 形式
 - 利用するデータインターフェース
 - データインターフェース【IF020】可視化用 DB 出力 IF を参照
- 出力
 - ◇ 到達圏域（道路ネットワーク分析）画面へ出力
 - 内容
 - 地図：地点からの到達可能エリア
 - グラフ：到達圏内の停留所数
 - 形式
 - 画面上に表示
 - 利用するデータインターフェース
 - なし
 - ◇ 地図画像の出力
 - 内容
 - 地図：運行頻度図、到達圏域図
 - 形式
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF012】地図出力 IF を参照
 - ◇ グラフの出力
 - 内容
 - グラフ：時間帯別の運行本数、到達圏内の停留所数
 - 形式
 - PNG 形式
 - 利用するデータインターフェース

- データインターフェース【IF013】グラフ出力 IF を参照
- ◇ 表の出力
 - 内容
 - 表：時間帯別の運行本数、到達圏内の停留所数
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF014】表のデータ出力 IF を参照

【FN004】乗降実績データの入出力機能<新規開発>

- 本システム機能の概要
 - ユーザーが保有する乗降集計データ、駅・バス停間 OD データ、乗降実績データをローカル環境から読み込み、乗降集計データ、駅・バス停間 OD データを生成し、データベースやファイルに保存する
- 本システム機能の入力・処理・出力のフローチャート

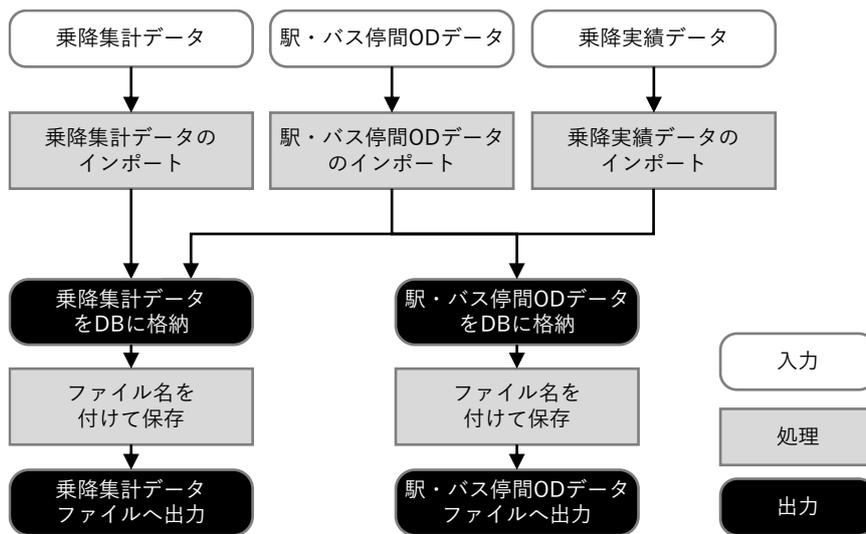


図 2-5 【FN004】のフローチャート

- 本システム機能の処理の詳細
 - 乗降集計データのインポート
 - ◇ 処理内容
 - 乗降集計データをデータベースに保存する
 - アカウント情報等とひも付けることでユーザー単位のセグメントを可能とする
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - ◇ 利用するアルゴリズム
 - なし

- 駅・バス停間 OD データのインポート
 - ◇ 処理内容
 - 駅・バス停間 OD データをデータベースに保存する
 - アカウント情報等とひも付けることでユーザー単位のセグメントを可能とする
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - ◇ 利用するアルゴリズム
 - なし
- 乗降実績データのインポート
 - ◇ 処理内容
 - システム上で可視化・分析するために、乗降実績データを GTFS データとひも付けて乗降集計データ及び駅・バス停間 OD データを CSV 形式で出力する。
 - 乗降実績データは route_id、stop_id 等を使用して GTFS データとひも付けを行う。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - 【AL103】乗降実績データと GTFS のひも付けアルゴリズム
- ファイル名をつけて保存
 - ◇ 処理内容
 - ファイルに名前を付けてデータを保存する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 乗降集計データ
 - データの内容
 - 各事業者が作成する運行記録データや運行記録(日報)から作成した乗降集計データ
 - どの停留所で何人の乗客が乗車、降車したかを示す
 - データの形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF004】乗降集計データ入力 IF を参照
 - ◇ 駅・バス停間 OD データ
 - データの内容
 - 乗降実績データ等から作成した駅・バス停間 OD データ

- ある停留所で乗ってある停留所で降りた乗客が何人いるかを示す
- データの形式
 - CSV 形式
- 利用するデータインターフェース
 - データインターフェース【IF004】乗降集計データ入力 IF を参照
- ◇ 乗降実績データ
 - データの内容
 - 乗降実績データ標準仕様により定義された様式のデータ
 - IC カードの利用実績を 1 件ごとに記録したデータ
 - 利用日時や便、乗降した停留所等の情報を含む
 - データの形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF005】乗降実績データ入力 IF を参照
- 出力
 - ◇ 乗降集計データ (DB 格納)
 - 内容
 - 各停留所での乗客の乗車数と降車数
 - 形式
 - データベースに格納
 - 利用するデータインターフェース
 - データインターフェース【IF019】乗降実績 DB 入出力 IF を参照
 - ◇ 乗降集計データ (CSV 出力)
 - 内容
 - 各停留所での乗客の乗車数と降車数
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF019】乗降実績 DB 入出力 IF を参照
 - ◇ 駅・バス停間 OD データ (DB 格納)
 - 内容
 - 入起点と終点の停留所の組合せでの利用者数
 - 形式
 - データベースに格納
 - 利用するデータインターフェース
 - データインターフェース【IF019】乗降実績 DB 入出力 IF を参照
 - ◇ 駅・バス停間 OD データ (CSV 出力)
 - 内容

- 起点と終点の停留所の組み合わせでの利用者数
- 形式
 - CSV 形式
- 利用するデータインターフェース
 - データインターフェース【IF011】 駅・バス停間 OD データ出力 IF を参照

【FN005】 乗降実績の可視化

- 本システム機能の概要
 - 乗降集計データに基づき、停留所ごとの乗降人数を集計し、地図上に表示する
- 本システム機能の入力・処理・出力のフローチャート

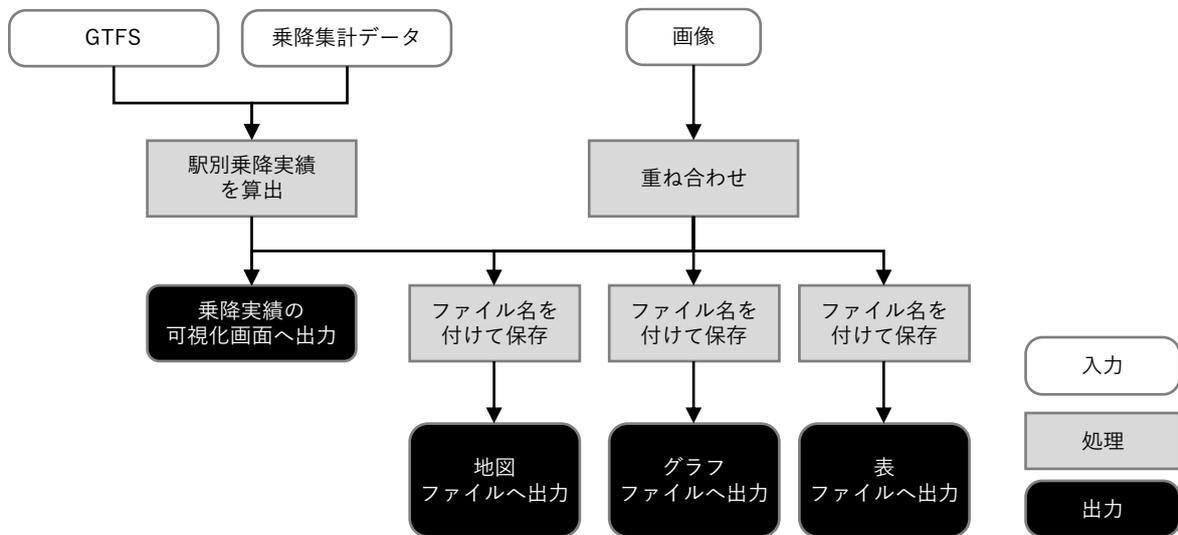


図 2-6 【FN005】 のフローチャート

- 本システム機能の処理の詳細
 - 駅別乗降実績を算出
 - ◇ 処理内容
 - ユーザーが画面上で入力した路線や運行日等の集計範囲に関する情報を取得する
 - 乗降集計データと GTFS データから停留所ごとの乗降数を算出し、集計する
 - 集計した結果を駅別乗降実績として地図上に表示する
 - 日付と時間帯、路線 (route_id) によるフィルタリングを可能とする
 - 複数系統を統合した系統ごとの集計も可能とする
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - なし
 - 重ね合わせ
 - ◇ 処理内容

- 画像を地図上に表示する。
- ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - Leaflet (ソフトウェア・ライブラリ【SL004】を参照)
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
- ◇ 利用するアルゴリズム
 - なし
- ファイル名をつけて保存
 - ◇ 処理内容
 - ファイルに名前を付けてデータを保存する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 乗降集計データ
 - データの内容
 - 各事業者が作成する運行記録データや運行記録(日報)を基に、「どの便で」「どの停留所で」「何人の人が乗降したか」を整理したデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - データインターフェースを参照
 - ◇ GTFS データ
 - データの内容
 - 【FN001】GTFS データ (API) 入力機能<新規開発> 【FN010】GTFS データ入力機能<新規開発>で入力あるいは【FN002】GTFS 編集機能<新規開発>で編集したGTFS形式のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - データインターフェース【IF018】GTFS DB 入出力 IF を参照
 - ◇ ユーザー入力情報
 - データの内容
 - ユーザーがシステム上で入力する駅別乗降実績の表示に必要な情報

- データの集計範囲を指定する情報
 - データの形式
 - なし
 - 利用するデータインターフェース
 - なし
- 出力
- ◇ 乗降実績の可視化画面へ出力
 - 内容
 - 地図：駅別乗降実績を画面に出力する
 - グラフ：駅別乗降実績
 - 形式
 - 画面上に表示
 - 利用するデータインターフェース
 - なし
 - ◇ 地図画像の出力
 - 内容
 - 路線の乗降実績、区間の時間帯別利用者数、停留所別利用者数
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF012】地図出力 IF を参照
 - ◇ グラフの出力
 - 内容
 - 乗降実績
 - 形式
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF013】グラフ出力 IF を参照
 - ◇ 表の出力
 - 内容
 - 乗降実績
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF014】表のデータ出力 IF を参照

【FN006】交通分担率の変化シミュレーション<新規開発>

- 本システム機能の概要
 - サービスレベル（運行本数）の変更に応じた公共交通分担率の変化をシミュレーションする
 - ※DRM は一般財団法人日本デジタル道路地図協会が販売する有償データであるため、本機能を使用する場合は、ユーザーが DRM を購入する必要がある

- 本システム機能の入力・処理・出力のフローチャート

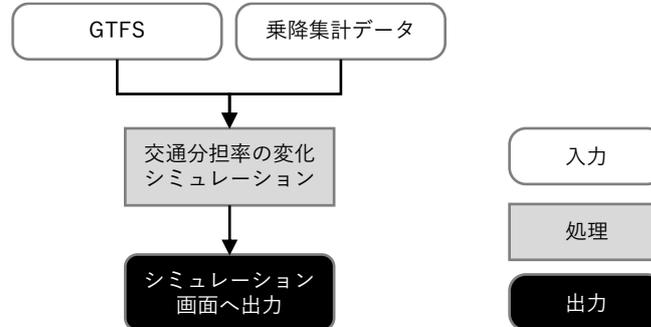


図 2-7 【FN006】のフローチャート

- 本システム機能の処理の詳細
 - 交通分担率の変化シミュレーション
 - ◇ 処理内容
 - GTFS・乗降集計データ・DRM・道路交通センサスを基に、特定の区間における交通量や混雑の程度を算出する
 - DRMの「交通センサス区間番号」と道路交通センサスの「交通調査基本区間番号」でデータをひも付ける
 - 変更内容に対して、増便感度、減便感度を乗じて、公共交通利用者の増減人数を算出する
 - ◇ 利用するライブラリ
 - PostgreSQL（ソフトウェア・ライブラリ【SL001】を参照）
 - PostGIS（ソフトウェア・ライブラリ【SL002】を参照）
 - ◇ 利用するアルゴリズム
 - 【AL102】交通分担率の変化シミュレーションアルゴリズムを参照
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS データ
 - データの内容
 - 【FN001】【FN010】で入力あるいは【FN002】で編集したGTFS形式のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - データインターフェース【IF018】GTFS DB 入出力 IF を参照
 - ◇ 乗降集計データ
 - データの内容

- 各事業者が作成する運行記録データや運行記録（日報）を基に、「どの便で」「どの停留所で」「何人の人が乗降したか」を整理したデータ
- 【FN004】で入力・出力した乗降集計データ
- データの形式
 - データベース形式
- 利用するデータインターフェース
 - データインターフェース【IF019】乗降実績 DB 入出力 IF を参照
- 出力
 - ◇ シミュレーション画面
 - 内容
 - サービスレベル（運行本数）の変更に応じた公共交通分担率の変化
 - 形式
 - 画面上に表示
 - 利用するデータインターフェース
 - データインターフェース【IF020】可視化用 DB 出力 IF を参照

【FN007】データベース管理<新規開発>

- 本システム機能の概要
 - 【FN008】でログインしたアカウント単位で、【FN001】等で入力・作成したデータ（GTFS/編集済み GTFS、乗降集計データ、乗降実績データ、駅・バス停間 OD データ、人口統計等）を一元管理する
 - データは account_id をキーとする論理的分離により保護され、当該アカウントに属するデータのみが参照・更新対象となる
 - データベースは PostgreSQL（PostGIS 拡張）上に構築する
 - データベースのバックアップは、本機能より定期保存する
- 本システム機能の入力・処理・出力のフローチャート

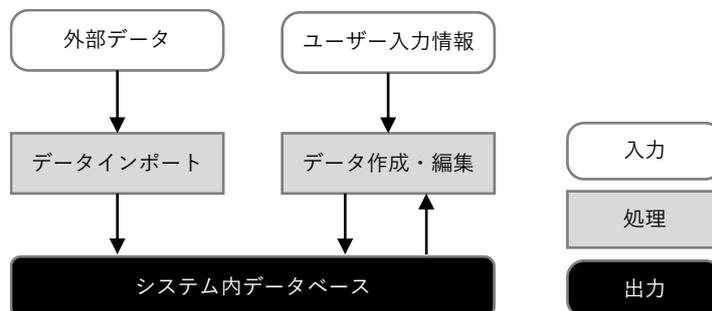


図 2-8 【FN007】のフローチャート

- 本システム機能の処理の詳細
 - データインポート
 - ◇ 処理内容
 - ログイン時のアカウント情報に基づき account_id をリクエストコンテキストに設定し、以降の

DB 入出力は当該 account_id に限定する

外部システム及びユーザーのローカル環境よりデータを取得してデータベースに保存する
処理の詳細は以下を参照

- 【FN001】 GTFS データ (API) 入力機能
- 【FN002】 GTFS 編集機能
- 【FN003】 道路ネットワークを用いた到達圏分析
- 【FN004】 乗降実績データの入出力機能
- 【FN005】 乗降実績の可視化
- 【FN006】 交通分担率の変化シミュレーション
- 【FN008】 ログイン
- 【FN009】 ログアウト
- 【FN010】 GTFS データ入力機能
- 【FN011】 関連データ入力機能
- 【FN012】 バッファ到達圏分析
- 【FN013】 運行頻度分析
- 【FN014】 OD の可視化
- 【FN016】 経路・時刻表機能
- 【FN017】 公共交通圏域機能

◇ 利用するライブラリ

- PostgreSQL (ソフトウェア・ライブラリ 【SL001】 を参照)
- PostGIS (ソフトウェア・ライブラリ 【SL002】 を参照)

◇ 利用するアルゴリズム

- なし

▶ データ作成・編集

◇ 処理内容

- システム内データベース内のデータからデータを作成して追加する
処理の詳細は以下を参照
【FN004】 乗降実績データの入出力機能
- ユーザーからの入力を受け取り、入力内容に基づいてシステム内データベースのデータに
対して検索・追加・更新・削除等の処理を実行する
処理の詳細は以下を参照
【FN002】 GTFS 編集機能

◇ 利用するライブラリ

- PostgreSQL (ソフトウェア・ライブラリ 【SL001】 を参照)
- PostGIS (ソフトウェア・ライブラリ 【SL002】 を参照)

◇ 利用するアルゴリズム

- なし

- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 外部データ
 - データの内容
 - 外部システム及びユーザーのローカル環境より取得するデータ
 - データの詳細は以下機能のデータ仕様を参照
 - 【FN001】 GTFS データ (API) 入力機能
 - 【FN010】 GTFS データ入力機能
 - 【FN011】 関連データ入力機能
 - ◇ ユーザー入力情報
 - データの内容
 - ユーザーが画面上で入力したデータ
 - データの形式
 - なし
 - 利用するデータインターフェース
 - なし
 - 出力
 - ◇ システム内データベース
 - データの内容
 - システム内で保持している編集済み GTFS データや乗降実績等のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - なし
- **【FN008】 ログイン<新規開発>**
 - 本システム機能の概要
 - ユーザーが入力したログイン ID とパスワードを確認し、アカウント認証を行いログインの可否を制御する

- 本システム機能の入力・処理・出力のフローチャート

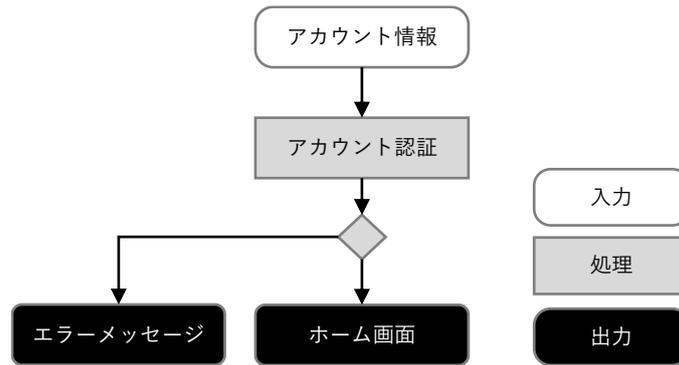


図 2-9 【FN008】 のフローチャート

- 本システム機能の処理の詳細

- アカウント認証

- ◇ 処理内容

- ユーザー情報がデータベースに保存されている情報と一致するかを検証し、一致すればホーム画面を、一致しなければエラーメッセージを表示する

- ◇ 利用するライブラリ

- React (ソフトウェア・ライブラリ【SL005】を参照)
- Django (ソフトウェア・ライブラリ【SL003】を参照)

- ◇ 利用するアルゴリズム

- なし

- 本システム機能の入出力データの仕様

- 入力

- ◇ ユーザー情報 (ログイン ID、パスワード)

- データの内容
 - ログイン画面でユーザーが入力したログイン ID 及びパスワード
- データの形式
 - TXT 形式
- 利用するデータインターフェース
 - データインターフェース【IF001】ログイン IF を参照

- 出力

- ◇ ホーム画面の表示

- データの内容
 - ログインが成功した場合はホーム画面
 - ログインに失敗した場合はエラーメッセージ
- データの形式
 - 画面上に表示
- 利用するデータインターフェース
 - なし

【FN009】 ログアウト<新規開発>

- 本システム機能の概要
 - 保持されたユーザー情報を破棄し、ユーザーの認証状態を終了する
- 本システム機能の入力・処理・出力のフローチャート

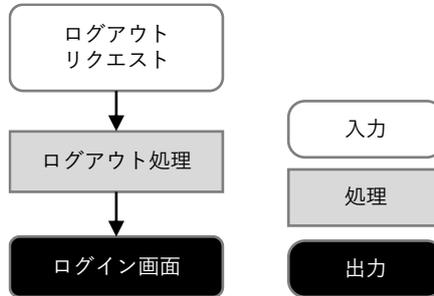


図 2-10 【FN009】のフローチャート

- 本システム機能の処理の詳細
 - ログアウト処理
 - ◇ 処理内容
 - 認証中ユーザーの認証情報を無効化し、ログイン画面を表示する
 - ◇ 利用するライブラリ
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ ログアウトリクエスト
 - 内容
 - ユーザーがログアウトボタンをクリックすることで、ログアウトを要求する
 - 形式
 - なし
 - 利用するデータインターフェース
 - なし
 - 出力
 - ◇ ログイン画面
 - 内容
 - ログイン画面
 - 形式
 - 画面上に表示
 - 利用するデータインターフェース
 - なし

【FN010】 GTFS データ入力機能<新規開発>

- 本システム機能の概要
 - アカウント単位で、システム利用に必要なデータを入力可能とする
 - ローカル環境からの GTFS データをインポートする
 - ドラッグ&ドロップ操作でファイルをアップロード可能とする
- 本システム機能の入力・処理・出力のフローチャート

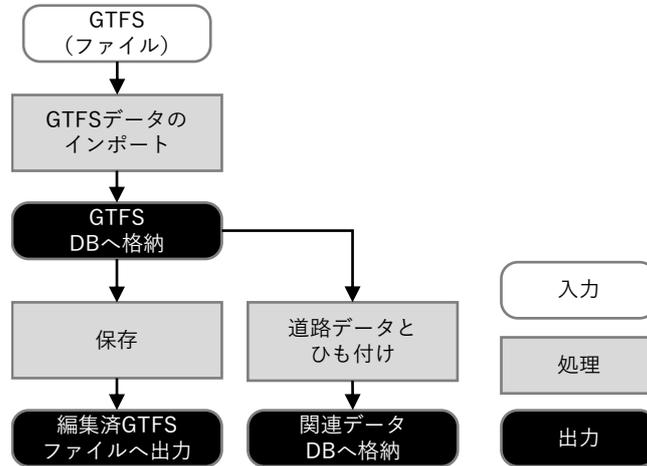


図 2-11 【FN010】のフローチャート

- 本システム機能の処理の詳細
 - GTFS データのインポート
 - ◇ 処理内容
 - GTFS ファイルを指定する
 - インポートした GTFS データは、アカウント管理機能等と連携することでプリセットとして保持する
 - 取得・指定した GTFS データの入力内容を確認し、不足データを補完する等の整備を行う
 - 整備した GTFS データのシナリオ名を付けてデータベースに保存する
 - インポートする際に路線と停留所によるグループ化を行う
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - 【AL106】路線と停留所によるグループ化
 - 道路データとひも付け
 - ◇ 処理内容
 - 【FN011】において読み込んだ道路データと、道路データとシナリオ名をつけて DB に格納した GTFS データを用いて、OpenTripPlanner の道路ネットワーク解析を行うためのグラフオブジェクトを生成する。
 - ◇ 利用するライブラリ

- PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
- PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
- OpenTripPlanner (ソフトウェア・ライブラリ【SL006】を参照)
- ◇ 利用するアルゴリズム
 - なし
- 保存
 - ◇ 処理内容
 - データベースのデータを保存する。
 - 保存時に GTFS のバリデーションチェックを実行する。
 - ZIP ファイル名をシナリオ名のとおり保存する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS データ
 - 内容
 - ローカル環境からドラッグ&ドロップでアップロードした GTFS のファイル
 - 形式
 - TXT 形式
 - 利用するデータインターフェース
 - データインターフェース【IF003】GTFS 入力 IF (ローカル) を参照
 - 出力
 - ◇ GTFS データ
 - 内容
 - 入力した GTFS データをデータベースに格納する。
 - 形式
 - データベースに格納
 - 利用するデータインターフェース
 - データインターフェース【IF017】GTFS データ入力 IF を参照
 - ◇ 道路ネットワークのグラフオブジェクト
 - 内容
 - GTFS と OpenStreetMap あるいは DRM の道路データから生成した道路ネットワークのグラフオブジェクトをデータベースに格納する
 - 形式
 - PBF 形式
 - 利用するデータインターフェース

➤ データインターフェース【IF017】GTFS データ入力 IF を参照

【FN011】関連データ入力機能<新規開発>

- 本システム機能の概要
 - アカウント単位で、システム利用に必要なデータを入力可能とする
 - 関連するオープンデータとして人口統計データや、OpenStreetMap あるいは DRM の道路データをインポートする
- 本システム機能の入力・処理・出力のフローチャート

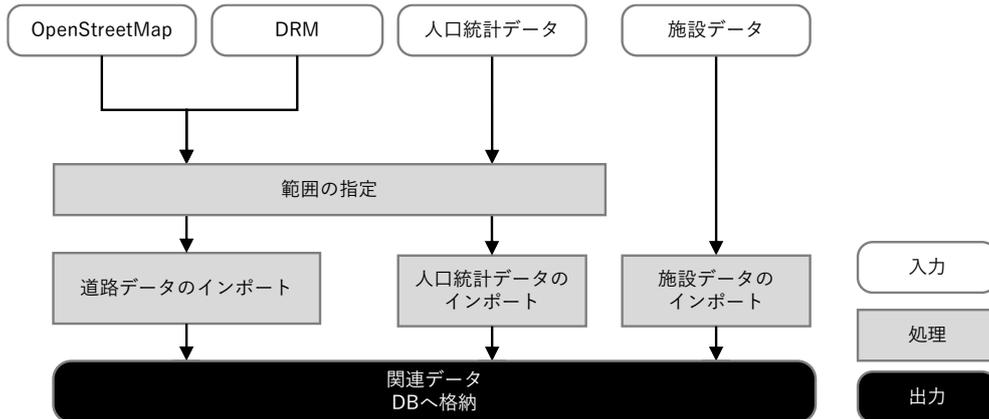


図 2-12 【FN011】のフローチャート

- 本システム機能の処理の詳細
 - 範囲の指定
 - ◇ 処理内容
 - OpenStreetMap、DRM の道路データ、人口統計データをアップロード可能とする。
 - 施設データをローカル環境からドラッグ&ドロップでアップロード可能とする。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
 - 道路データのインポート
 - ◇ 処理内容
 - アップロードした道路データとシナリオ名をつけて DB に格納した GTFS データを用いて、OpenTripPlanner の道路ネットワーク解析を行うためのグラフオブジェクトを生成する。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - OpenTripPlanner (ソフトウェア・ライブラリ【SL006】を参照)
 - ◇ 利用するアルゴリズム
 - なし

- 人口統計データのインポート
 - ◇ 処理内容
 - 国勢調査の 250mメッシュ人口データを、アップロード可能とする。
 - アップロードした人口統計データをデータベースに保存する。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - なし
- 施設データのインポート
 - ◇ 処理内容
 - 施設タイプ、名前、緯度、経度、備考の施設 CSV データを、ローカル環境からドラッグ&ドロップ操作でアップロード可能とする。
 - アップロードした施設データをデータベースに保存する。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ OpenStreetMap
 - 内容
 - 分析範囲の OpenStreetMap データ
 - 建物・水域など不要オブジェクトを除いた、道路系タグや規制・プラットフォームのオブジェクトとする。例：highway=* (motorway/trunk/primary...service)
 - 形式
 - .osm.pbf 形式
 - 利用するデータインターフェース
 - データインターフェース【IF008】 OpenStreetMap 入力 IF を参照
 - ◇ DRM (デジタル道路地図)
 - 内容
 - 国道・都道府県道以上の基本道路リンク (ノード間の道路) のデータ。
 - 一般財団法人日本デジタル道路地図協会が販売する有償データ。
 - 使用する場合は、ユーザーが DRM を購入する必要がある。
 - 形式
 - 固定長テキストレコード (Shift-JIS)
 - 日本測地系

- 利用するデータインターフェース
 - データインターフェース【IF007】DRM 入力 IF を参照
- ◇ 人口統計データ
 - 内容
 - 総務省統計局「地域メッシュ統計（国勢調査）」から 4 分の 1 地域メッシュ（約 250m×250m）単位で集計された 2020 年国勢調査（2020 年）の人口データ。
 - 総人口と年齢 3 区分（15 歳未満、15～64 歳、65 歳以上）を利用する。
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF006】人口統計入力 IF を参照
- ◇ 施設データ
 - 内容
 - 施設タイプ、名前、緯度、経度、備考の施設 CSV データ。
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF026】施設データ入力 IF を参照
- 出力
 - ◇ 道路ネットワークのグラフオブジェクト
 - 内容
 - GTFS と OpenStreetMap あるいは DRM の道路データから生成した道路ネットワークのグラフオブジェクトをデータベースに格納する
 - 形式
 - データベースに格納
 - 利用するデータインターフェース
 - データインターフェース【IF023】関連データ入力 IF を参照
 - ◇ 人口統計データ
 - 内容
 - 入力した人口統計データをデータベースに格納する。
 - 形式
 - データベースに格納
 - 利用するデータインターフェース
 - データインターフェース【IF023】関連データ入力 IF を参照

【FN012】 バッファ到達圏分析

- 本システム機能の概要
 - 【FN001】【FN010】で読み込んだ GTFS データ及び【FN002】で編集した GTFS データを地図上で可視化する
 - 任意の時間内に到達可能な到達圏域（バッファ分析）の可視化を行う
 - 人口統計のメッシュデータの重ね合わせの ON/OFF が可能とする
 - 施設データの重ね合わせの ON/OFF が可能とする
- 本システム機能の入力・処理・出力のフローチャート

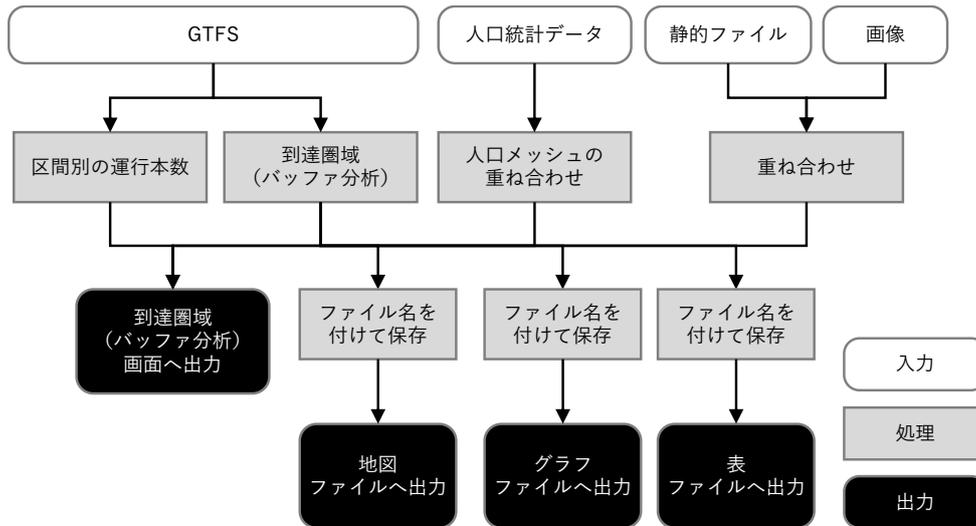


図 2-13 【FN012】 のフローチャート

- 本システム機能の処理の詳細
 - 区間別の運行本数
 - ◇ 処理内容
 - ユーザーが画面上で駅区間を設定する。
 - GTFS データを基に、ユーザーが指定した条件で駅区間別の運行本数を集計する。
 - 集計した結果を運行頻度図として地図上に表示する。
 - アクセス可能な施設、所要時間別到達圏域内の人口を集計し、グラフ表示する。
 - 時間内でアクセス可能な施設を表示する。
 - 到達圏域内の人口を年齢層別に集計する。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - Leaflet (ソフトウェア・ライブラリ【SL004】を参照)
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
 - ◇ 利用するアルゴリズム
 - なし

- 到達圏域（バッファ分析）
 - ◇ 処理内容
 - ユーザーが地図上で指定した地点（緯度・経度）から、徒歩及び公共交通を使用して到達可能なエリアを計算する
 - バスの乗換計算に当たっては GTFS データが保有する停留所の緯度経度情報及び発着時刻等のデータを使用する
 - 所要時間に応じた到達圏域を地図上に表示する
 - ◇ 利用するライブラリ
 - PostgreSQL（ソフトウェア・ライブラリ【SL001】を参照）
 - PostGIS（ソフトウェア・ライブラリ【SL002】を参照）
 - Django（ソフトウェア・ライブラリ【SL003】を参照）
 - Leaflet（ソフトウェア・ライブラリ【SL004】を参照）
 - React.js（ソフトウェア・ライブラリ【SL005】を参照）
 - ◇ 利用するアルゴリズム
 - なし
- 人口メッシュの重ね合わせ
 - ◇ 処理内容
 - 人口メッシュデータを地図上に表示する。
 - 到達圏域内の総人口を算出する。到達時間 10 分単位（例：10 分圏、20 分圏、30 分圏…）の到達圏ポリゴンを作成し、各到達圏ポリゴンと 250mメッシュ人口データのメッシュポリゴンを重ね合わせた空間重ね合わせ処理を行い、到達圏ごとの人口集計グラフを表示する。
 - 各メッシュポリゴンについて、次の判定・処理を行う。
 - ◇ 到達圏ポリゴンに完全に含まれる場合、メッシュ人口をそのまま加算
 - ◇ 到達圏ポリゴンに一部のみ含まれる場合は、重なり面積比を算出し、メッシュ人口を面積比で案分して加算。
 - ◇ 利用するライブラリ
 - PostgreSQL（ソフトウェア・ライブラリ【SL001】を参照）
 - PostGIS（ソフトウェア・ライブラリ【SL002】を参照）
 - Django（ソフトウェア・ライブラリ【SL003】を参照）
 - Leaflet（ソフトウェア・ライブラリ【SL004】を参照）
 - React.js（ソフトウェア・ライブラリ【SL005】を参照）
 - ◇ 利用するアルゴリズム
 - なし
- ファイル名をつけて保存
 - ◇ 処理内容
 - ファイルに名前を付けてデータを保存する。
 - ◇ 利用するライブラリ

- なし
- ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS データ
 - データの内容
 - 【FN001】【FN010】 で入力あるいは【FN002】 で編集した GTFS 形式のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - データインターフェース【IF020】可視化用 DB 出力 IF を参照
 - ◇ 人口統計データ
 - 内容
 - 2020 年の国勢調査に基づく 250mメッシュごとに集計した人口
 - 形式
 - JSON 形式
 - 利用するデータインターフェース
 - データインターフェース【IF020】可視化用 DB 出力 IF を参照
 - 出力
 - ◇ 到達圏域（バッファ分析）画面へ出力
 - 内容
 - 地図：地点からの到達可能エリア
 - グラフ：到達圏内の停留所数
 - 形式
 - 画面上に表示
 - 利用するデータインターフェース
 - なし
 - ◇ 地図画像の出力
 - 内容
 - 地図：運行頻度図、到達圏域図
 - 形式
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF012】地図出力 IF を参照
 - ◇ グラフの出力
 - 内容
 - グラフ：時間帯別の運行本数、到達圏内の停留所数

- 形式
 - PNG 形式
- 利用するデータインターフェース
 - データインターフェース【IF013】グラフ出力 IF を参照
- ◇ 表の出力
 - 内容
 - 表：時間帯別の運行本数、到達圏内の停留所数
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF014】表のデータ出力 IF を参照

【FN013】運行頻度分析

- 本システム機能の概要
 - 【FN001】【FN010】で読み込んだ GTFS データ及び【FN002】で編集した GTFS データを地図上で可視化する
 - 運行頻度図を可視化する
 - 人口統計のメッシュデータの重ね合わせの ON/OFF を可能とする
 - 路線別・時間帯別の運行状況を可視化する
- 本システム機能の入力・処理・出力のフローチャート

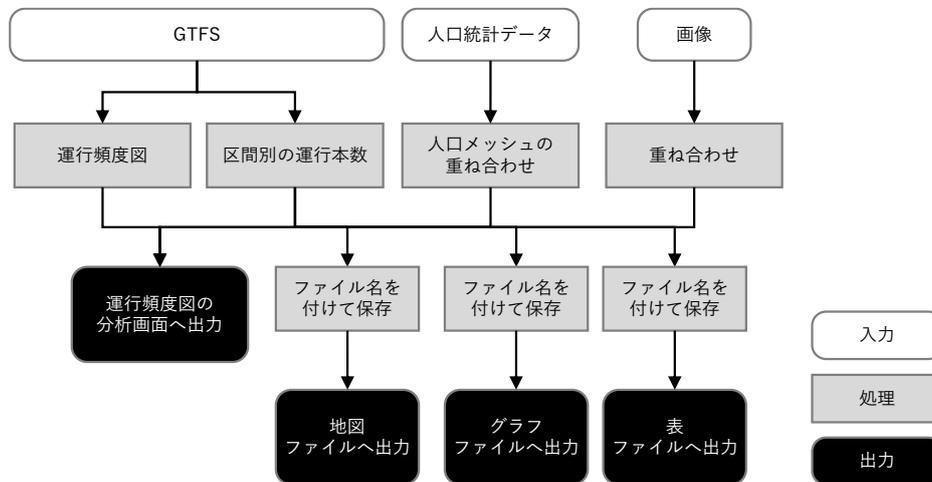


図 2-14 【FN013】のフローチャート

- 本システム機能の処理の詳細
 - 運行頻度図
 - ◇ 処理内容
 - ユーザーが画面上で路線や運行日等の条件を設定する
 - GTFS データを基に、ユーザーが指定した条件で停留所間の運行本数を集計する
 - 集計した結果を運行頻度図として地図上に表示する
 - 時間帯別運行本数、路線別運行本数をグラフ表示する

- 路線別運行本数、停留所別運行本数を表で表示する
- ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - Leaflet (ソフトウェア・ライブラリ【SL004】を参照)
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
- ◇ 利用するアルゴリズム
 - なし
- 区間別の運行本数
 - ◇ 処理内容
 - ユーザーが画面上で駅区間を設定する
 - GTFS データを基に、ユーザーが指定した条件で駅区間別の運行本数を集計する
 - 集計した結果を運行頻度図として地図上に表示する
 - 時間帯別運行本数、路線別運行本数をグラフ表示する
 - 路線別運行本数、停留所別運行本数を表で表示する
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - Leaflet (ソフトウェア・ライブラリ【SL004】を参照)
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
 - ◇ 利用するアルゴリズム
 - なし
- 人口メッシュの重ね合わせ
 - ◇ 処理内容
 - 人口メッシュデータを地図上に表示する。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - Leaflet (ソフトウェア・ライブラリ【SL004】を参照)
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
 - ◇ 利用するアルゴリズム
 - なし
- 重ね合わせ
 - ◇ 処理内容
 - 画像を地図上に表示する。

- ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - Leaflet (ソフトウェア・ライブラリ【SL004】を参照)
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
- ◇ 利用するアルゴリズム
 - なし
- ファイル名をつけて保存
- ◇ 処理内容
 - ファイルに名前を付けてデータを保存する。
- ◇ 利用するライブラリ
 - なし
- ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS データ
 - データの内容
 - 【FN001】【FN010】で入力あるいは【FN002】で編集した GTFS 形式のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - データインターフェース【IF020】可視化用 DB 出力 IF を参照
 - ◇ 人口統計データ
 - 内容
 - 2020 年の国勢調査に基づく 250mメッシュごとに集計した人口
 - 形式
 - JSON 形式
 - 利用するデータインターフェース
 - データインターフェース【IF020】可視化用 DB 出力 IF を参照
 - 出力
 - ◇ 運行頻度図の分析画面へ出力
 - 内容
 - 地図：停留所区間のラインの太さは、運行本数に応じて変化させて表示する
 - グラフ：時間帯別の運行本数
 - 形式
 - 画面上に表示

- 利用するデータインターフェース
 - なし
- ◇ 地図画像の出力
 - 内容
 - 地図：運行頻度図、到達圏域図
 - 形式
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF012】地図出力 IF を参照
- ◇ グラフの出力
 - 内容
 - グラフ：時間帯別の運行本数、到達圏内の停留所数
 - 形式
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF013】グラフ出力 IF を参照
- ◇ 表の出力
 - 内容
 - 表：時間帯別の運行本数、到達圏内の停留所数
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF014】表のデータ出力 IF を参照

【FN014】 OD の可視化

- 本システム機能の概要
 - 駅・バス停を地図上に表示する
- 本システム機能の入力・処理・出力のフローチャート

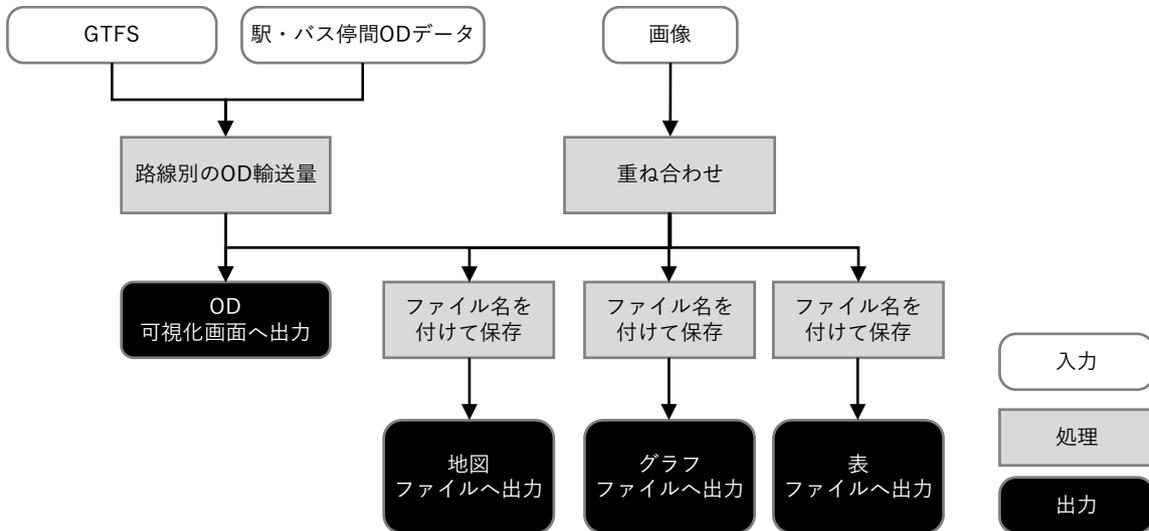


図 2-15 【FN014】 のフローチャート

- 本システム機能の処理の詳細
 - 路線別の OD 輸送量
 - ◇ 処理内容
 - ユーザーが画面上で入力した路線や運行日等の集計範囲に関する情報を取得する
 - 駅・バス停間 OD データと GTFS データから停留所別の OD 輸送量を算出し、集計する
 - 集計した結果を OD 輸送量として地図上に表示する
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ 【SL001】 を参照)
 - PostGIS (ソフトウェア・ライブラリ 【SL002】 を参照)
 - ◇ 利用するアルゴリズム
 - なし
 - 重ね合わせ
 - ◇ 処理内容
 - 画像を地図上に表示する。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ 【SL001】 を参照)
 - PostGIS (ソフトウェア・ライブラリ 【SL002】 を参照)
 - Django (ソフトウェア・ライブラリ 【SL003】 を参照)
 - Leaflet (ソフトウェア・ライブラリ 【SL004】 を参照)
 - React.js (ソフトウェア・ライブラリ 【SL005】 を参照)
 - ◇ 利用するアルゴリズム

- なし
- ファイル名をつけて保存
 - ◇ 処理内容
 - ファイルに名前を付けてデータを保存する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 駅・バス停間 OD データ
 - データの内容
 - ある停留所から乗ってある停留所で降りた乗客が何人いるかを示すデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - データインターフェース【IF004】乗降集計データ入力 IF を参照
 - ◇ GTFS データ
 - データの内容
 - 【FN001】【FN010】で入力あるいは【FN002】で編集した GTFS 形式のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - 路線、停留所、運行日等に関する情報
 - ◇ ユーザー入力情報
 - データの内容
 - ユーザーがシステム上で入力する駅別乗降実績の表示に必要な情報
 - データの集計範囲を指定する情報
 - データの形式
 - なし
 - 利用するデータインターフェース
 - なし
 - 出力
 - ◇ OD 可視化画面へ出力
 - 内容
 - 地図：停留所別利用者数、OD 流動図、OD 量を画面に出力する
 - グラフ：駅別 OD
 - 形式

- 画面上に表示
- 利用するデータインターフェース
 - なし
- ◇ 地図画像の出力
 - 内容
 - 停留所別利用者数、OD 流動図、OD 量
 - 形式
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF012】地図出力 IF を参照
- ◇ 表の出力
 - 内容
 - 停留所別利用者数、起点・終点一覧、バス停間 OD 一覧
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF014】表のデータ出力 IF を参照

【FN015】 運行記録（日報）出力機能<新規開発>

- 本システム機能の概要
 - 現場の運転手が乗降数を記録するための手入力用の運行記録（日報）帳票作成ツールを作成する
 - 自治体職員等が現場の紙の運行記録（日報）を基に乗降実績を転記するための登録用の運行記録（日報）帳票作成ツールを作成する
 - 登録用の運行記録（日報）帳票作成ツールに転記された情報から乗降集計データを出力する
- 本システムの機能の入力・処理・出力のフローチャート

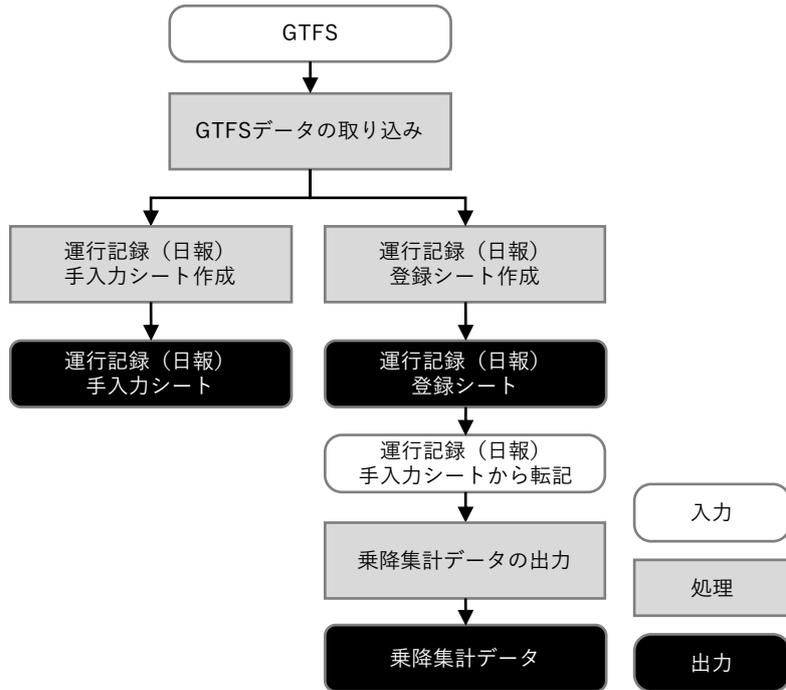


図 2-16 【FN015】のフローチャート

- 本システム機能の処理の詳細
 - GTFSデータの取り込み
 - ◇ 処理内容
 - GTFS ファイルを指定する
 - GTFS データを Microsoft Excel のシートに保存する
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
 - 運行記録（日報）手入力シート作成
 - ◇ 処理内容
 - Microsoft Excel のマクロにより GTFS とひも付いた運行記録（日報）手入力シートを作成する。
 - ◇ 利用するライブラリ
 - なし

- ◇ 利用するアルゴリズム
 - 【AL105】 運行記録（日報）の作成
- 運行記録（日報）登録シート作成
 - ◇ 処理内容
 - Microsoft Excel のマクロにより GTFS とひも付いた運行記録（日報）登録シートを作成する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - 【AL105】 運行記録（日報）の作成
- 乗降集計データの出力
 - ◇ 処理内容
 - 運行記録（日報）登録シートに登録されたデータを乗降集計データとして出力する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - 【AL105】 運行記録（日報）の作成
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS
 - 内容
 - GTFS Schedule（静的）データを外部サイト（GTFS データリポジトリ）から取得する。
 - 形式
 - TXT 形式
 - 利用するデータインターフェース
 - データインターフェース【IF024】 運行記録（日報）帳票作成ツールへの GTFS 入力 IF を参照
 - ◇ 運行記録（日報）手入力シートから転記
 - 内容
 - 自治体職員等が転記した乗降実績
 - 形式
 - Excel 形式
 - 利用するデータインターフェース
 - なし
 - 出力
 - ◇ 運行記録（日報）手入力シート
 - 内容

- 現場の運転手が乗降実績を記録するための運行記録（日報）フォーマット
- 形式
 - Excel 形式
- 利用するデータインターフェース
 - データインターフェース【IF010】運行記録（日報）帳票作成ツール出力 IF を参照
- ◇ 運行記録（日報）登録シート
 - 内容
 - 自治体職員等が乗降実績を転記するための運行記録（日報）フォーマット
 - 形式
 - Excel 形式
 - 利用するデータインターフェース
 - データインターフェース【IF010】運行記録（日報）帳票作成ツール出力 IF を参照
- ◇ 乗降集計データ
 - 内容
 - どの停留所で何人の乗客が乗車、降車したかを示す
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF025】乗降集計データ出力 IF を参照

【FN016】経路・時刻表機能

- 本システム機能の概要
 - GTFS データに基づき、経路・時刻表を地図上に表示する
- 本システムの機能の入力・処理・出力のフローチャート

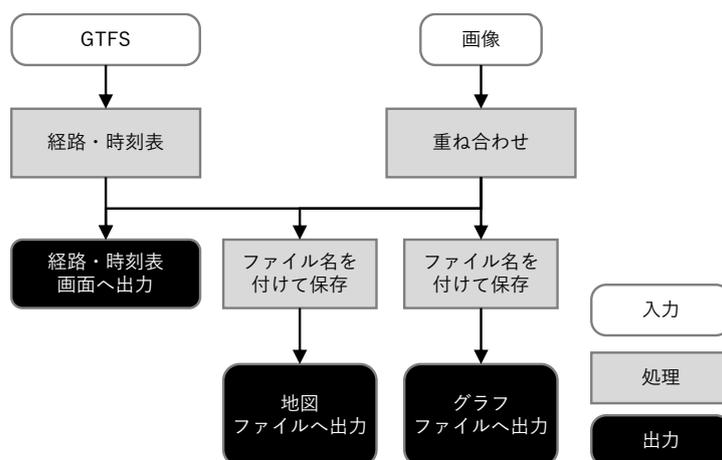


図 2-17 【FN016】のフローチャート

- 本システム機能の処理の詳細
 - 経路・時刻表
 - ◇ 処理内容
 - GTFS データから標柱／停留所、路線グループ、タイムレンジ、往復別、運行パターン別の経路・時刻表を地図上に表示する
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - ◇ 利用するアルゴリズム
 - なし
 - 重ね合わせ
 - ◇ 処理内容
 - 画像を地図上に表示する。
 - ◇ 利用するライブラリ
 - PostgreSQL (ソフトウェア・ライブラリ【SL001】を参照)
 - PostGIS (ソフトウェア・ライブラリ【SL002】を参照)
 - Django (ソフトウェア・ライブラリ【SL003】を参照)
 - Leaflet (ソフトウェア・ライブラリ【SL004】を参照)
 - React.js (ソフトウェア・ライブラリ【SL005】を参照)
 - ◇ 利用するアルゴリズム
 - なし
 - ファイル名をつけて保存
 - ◇ 処理内容
 - ファイルに名前を付けてデータを保存する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS データ
 - データの内容
 - 【FN001】【FN010】で入力あるいは【FN002】で編集した GTFS 形式のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - 路線、停留所、運行日等に関する情報
 - 出力

- ◇ 経路・時刻表画面へ出力
 - 内容
 - 地図：経路を画面に出力する
 - グラフ：駅別時刻表
 - 形式
 - 画面上に表示
 - 利用するデータインターフェース
 - なし
- ◇ 地図画像の出力
 - 内容
 - 経路
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF012】地図出力 IF を参照
- ◇ グラフの出力
 - 内容
 - 時刻表
 - 形式
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF013】グラフ出力 IF を参照
- ◇ 表の出力
 - 内容
 - 時刻表
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF014】表のデータ出力 IF を参照

【FN017】公共交通圏域機能

- 本システム機能の概要
 - 選択した公共交通路線の全ての停留所から指定した距離（徒歩圏）内の範囲を「公共交通圏域」として可視化し、その圏域内にある施設や人口を集計する
- 本システムの機能の入力・処理・出力のフローチャート

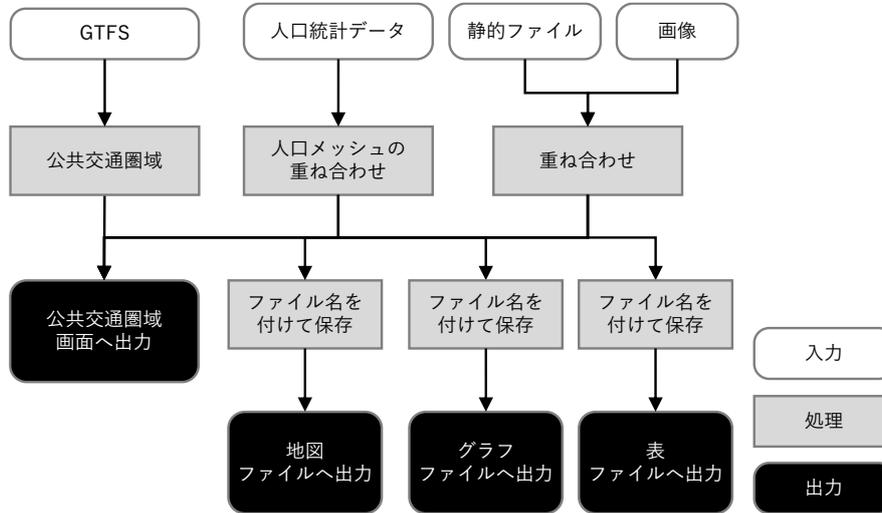


図 2-18 【FN017】のフローチャート

- 本システム機能の処理の詳細
 - 公共交通圏域
 - ◇ 処理内容
 - 選択した公共交通路線の全ての停留所から指定した距離（徒歩圏）内の範囲を「公共交通圏域」として地図上に表示する
 - ◇ 利用するライブラリ
 - PostgreSQL（ソフトウェア・ライブラリ【SL001】を参照）
 - PostGIS（ソフトウェア・ライブラリ【SL002】を参照）
 - ◇ 利用するアルゴリズム
 - なし
 - 人口メッシュの重ね合わせ
 - ◇ 処理内容
 - 人口メッシュデータを地図上に表示する。
 - ◇ 利用するライブラリ
 - PostgreSQL（ソフトウェア・ライブラリ【SL001】を参照）
 - PostGIS（ソフトウェア・ライブラリ【SL002】を参照）
 - Django（ソフトウェア・ライブラリ【SL003】を参照）
 - Leaflet（ソフトウェア・ライブラリ【SL004】を参照）
 - React.js（ソフトウェア・ライブラリ【SL005】を参照）
 - ◇ 利用するアルゴリズム
 - なし

- 重ね合わせ
 - ◇ 処理内容
 - 静的ファイル、画像を地図上に表示する。
 - ◇ 利用するライブラリ
 - PostgreSQL（ソフトウェア・ライブラリ【SL001】を参照）
 - PostGIS（ソフトウェア・ライブラリ【SL002】を参照）
 - Django（ソフトウェア・ライブラリ【SL003】を参照）
 - Leaflet（ソフトウェア・ライブラリ【SL004】を参照）
 - React.js（ソフトウェア・ライブラリ【SL005】を参照）
 - ◇ 利用するアルゴリズム
 - なし
- ファイル名をつけて保存
 - ◇ 処理内容
 - ファイルに名前を付けてデータを保存する。
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ GTFS データ
 - データの内容
 - 【FN001】【FN010】で入力あるいは【FN002】で編集した GTFS 形式のデータ
 - データの形式
 - データベース形式
 - 利用するデータインターフェース
 - データインターフェース【IF018】GTFS DB 入出力 IF を参照
 - ◇ 人口統計データ
 - 内容
 - 2020 年の国勢調査に基づく 250mメッシュごとに集計した人口
 - 形式
 - JSON 形式
 - 利用するデータインターフェース
 - データインターフェース【IF020】可視化用 DB 出力 IF を参照
 - 出力
 - ◇ 公共交通圏域画面へ出力
 - 内容
 - 地図：公共交通圏域を画面に出力する

- グラフ：施設タイプ別分布、年代別人口
- 形式
 - 画面上に表示
- 利用するデータインターフェース
 - なし
- ◇ 地図画像の出力
 - 内容
 - 経路
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF012】地図出力 IF を参照
- ◇ グラフの出力
 - 内容
 - 時刻表
 - 形式
 - PNG 形式
 - 利用するデータインターフェース
 - データインターフェース【IF013】グラフ出力 IF を参照
- ◇ 表の出力
 - 内容
 - 時刻表
 - 形式
 - CSV 形式
 - 利用するデータインターフェース
 - データインターフェース【IF014】表のデータ出力 IF を参照

2-1-4. ソフトウェア・ライブラリ (SL) の詳細

表 2-2 ソフトウェア・ライブラリー一覧

※朱文字：新規開発・既存改修

ID	名称	バージョン	内容
SL001	PostgreSQL	15	リレーショナルデータベース
SL002	PostGIS	3.3	PostgreSQL に地理空間データの管理機能を拡張した地理空間データベース
SL003	Django	5.2	Web フレームワーク、RESTful API 実装
SL004	Leaflet	1.9.4	地図描写のライブラリ
SL005	React.js	19.2.1	フロントエンドのフレームワーク
SL006	OpenTripPlanner	1.5.0	到達圏 (isochrone) 解析用のルーティングエンジン
SL007	MUI (Material UI)	7.1.2	React.js 向けの UI コンポーネントライブラリ
SL008	Docker	28.1.1	仮想環境で実行するためのプラットフォーム
SL009	OSRM (Open Source Routing Machine)	6.0.0	経路検索を行うエンジン
SL010	地域公共交通分析アプリ	0.1	地域の公共交通に関するデータを編集・分析・シミュレーション

【SL001】 PostgreSQL

- ベンダー
 - オープンソース
- 公式サイト
 - <https://www.postgresql.org/>
 - <https://www.postgresql.jp/>
- 本ソフトウェア・ライブラリの概要
 - オープンソースのリレーショナルデータベース管理システム (RDBMS)。標準 SQL に準拠しトランザクションや外部結合、ストアードプロシージャ、トリガー等豊富な機能を備えている。
- 開発するシステムにおいて利用する機能の詳細
 - GTFS データ等の格納・管理。路線 (shapes)、停留所 (stops)、経路 (trips) などをテーブルとして保存し、属性情報を効率的に検索・更新する。
- イメージ

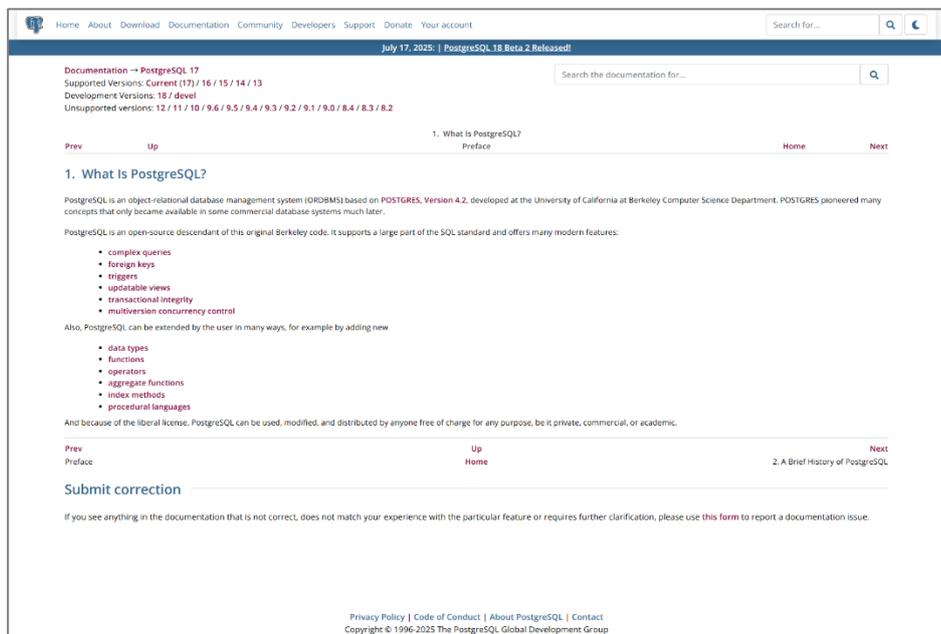


図 2-19 PostgreSQL ドキュメンテーション (<https://www.postgresql.org/docs/current/intro-what-is.html>)

【SL002】 PostGIS

- ベンダー
 - オープンソース
- 公式サイト
 - <https://postgis.net/>
- 本ソフトウェア・ライブラリの概要
 - PostgreSQL データベースに地理空間情報を格納・操作するためのオープンソース拡張モジュール。GIS（地理情報システム）機能を実現し、点・線・多角形などの空間データ型や距離・面積計算、空間検索などの豊富な関数を SQL から利用できる。
- 開発するシステムにおいて利用する機能の詳細
 - GTFS データ等の格納・管理。路線（shapes）、停留所（stops）、経路（trips）などを PostGIS 対応のテーブルとして保存し、空間データとして表示・計算・検索を行う。
- イメージ

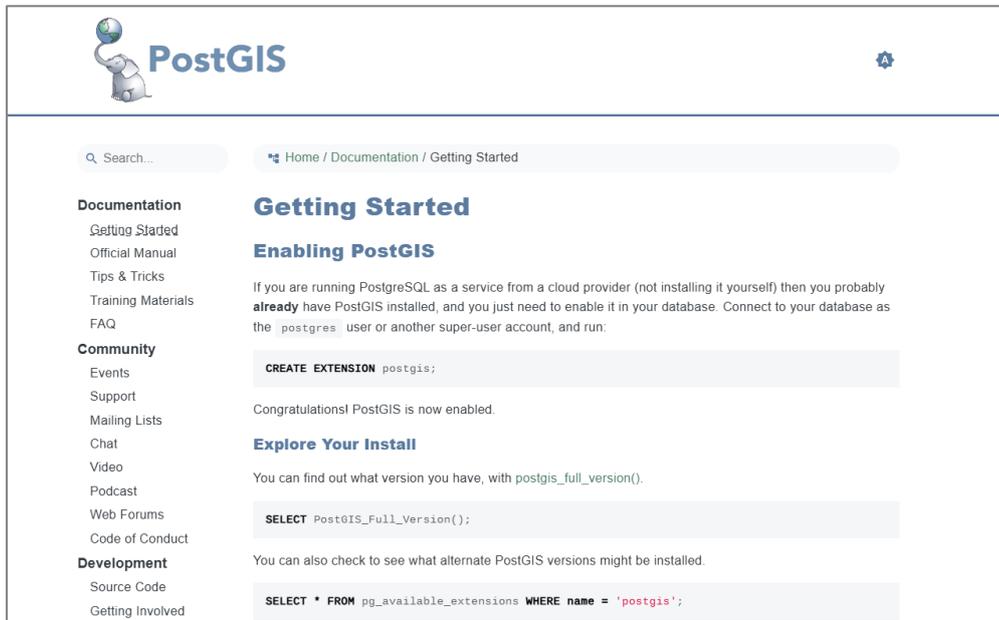


図 2-20 PostGIS ドキュメンテーション (https://postgis.net/documentation/getting_started/)

【SL003】 Django

- ベンダー
 - オープンソース
- 公式サイト
 - <https://www.djangoproject.com/>
- 本ソフトウェア・ライブラリの概要
 - Django は Python 製の Web アプリケーションフレームワーク。MTV (Model-Template-View) 構造を採用し、管理画面やセキュリティ機能が標準で備わっている。データベース連携や URL ルーティングも簡単に扱え、業務系から API 開発まで幅広く活用されている。
- 開発するシステムにおいて利用する機能の詳細
 - GTFS (routes、stops、trips)、駅・バス停間 OD データ、解析結果などをモデル定義し、PostgreSQL + PostGIS 上で検索・管理。
 - REST API サーバーとして、GTFS/乗降集計データの CRUD、地理解析呼び出し、シミュレーション実行。
- イメージ

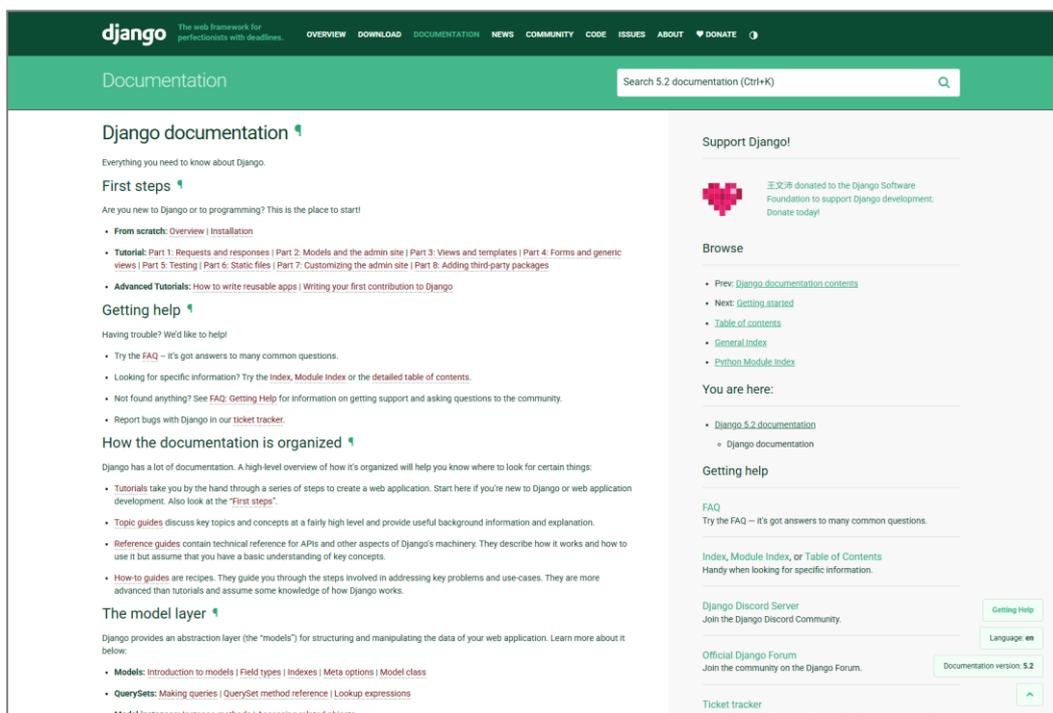


図 2-21 Django ドキュメンテーション (<https://docs.djangoproject.com/en/5.2/>)

【SL004】 Leaflet

- ベンダー
 - オープンソース
- 公式サイト
 - <https://leafletjs.com/>
- 本ソフトウェア・ライブラリの概要
 - オープンソースの地図ライブラリ。JavaScript で動作し、Web アプリや GIS システムに地図機能を組み込める。地理院タイルや OpenStreetMap などの地図タイルを表示し、マーカーやポップアップなどの機能も豊富に有する。
- 開発するシステムにおいて利用する機能の詳細
 - 停留所（ポイント）、路線（ポリライン）をレイヤーとして重ねて表示
 - タイルサーバーから地図タイルを読み込み、ズーム・パン操作を実現
 - OpenTripPlanner や PostGIS で生成したポリゴンを GeoJSON レイヤーとして読み込み、色分けや透過表示
- イメージ

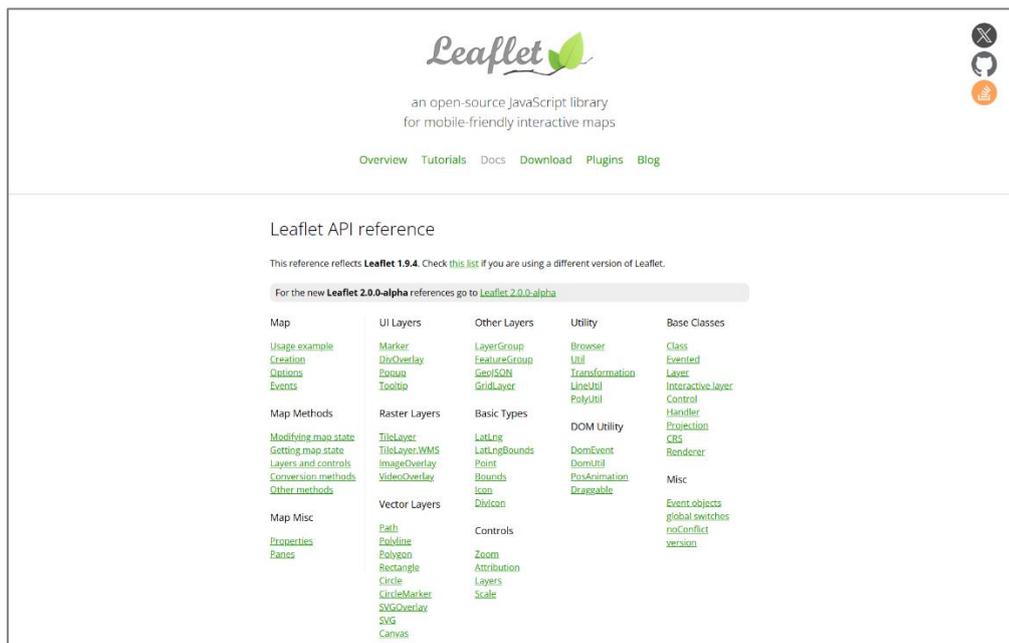


図 2-22 Leaflet ドキュメンテーション (<https://leafletjs.com/reference.html>)

【SL005】 React.js

- ベンダー
 - オープンソース
- 公式サイト
 - <https://ja.react.dev/>
- 本ソフトウェア・ライブラリの概要
 - Facebook が開発した JavaScript ライブラリで、動的でインタラクティブな Web ユーザーインターフェースを構築できる。コンポーネントベースの設計で、SPA（シングルページアプリ）やダッシュボード、フォームなどを効率よく開発できる。
- 開発するシステムにおいて利用する機能の詳細
 - UI 構築
- イメージ

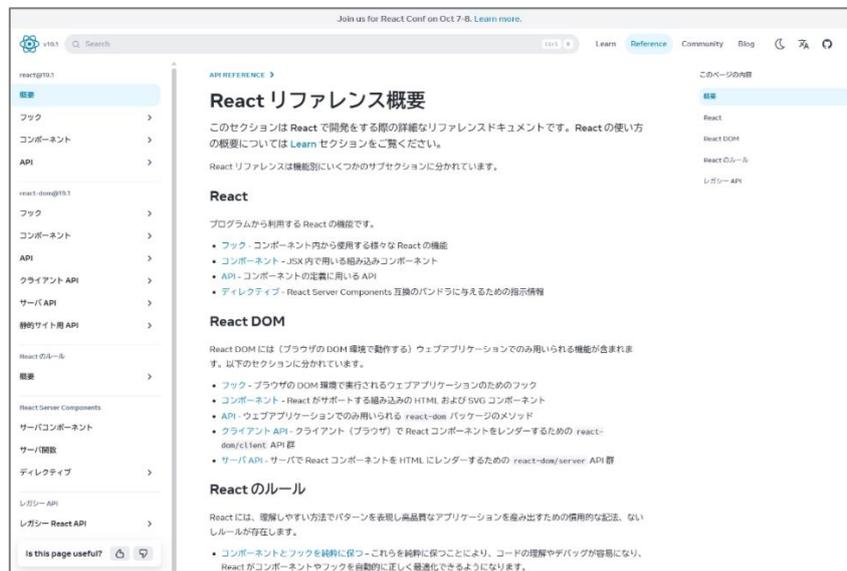


図 2-23 React リファレンス (<https://ja.react.dev/reference/react>)

【SL006】 OpenTripPlanner

- ベンダー
 - オープンソース
- 公式サイト
 - <https://www.opentripplanner.org/>
- 本ソフトウェア・ライブラリの概要
 - GTFS と OSM（道路網）を組み合わせた複合ルーティング・解析のためのオープンソースエンジン
- 開発するシステムにおいて利用する機能の詳細
 - 徒歩→公共交通→徒歩のシームレスな経路探索を実行し、最適ルートや所要時間を取得
 - 指定地点から 15 分/30 分/60 分といった複数の到達圏ポリゴンを一括生成
- イメージ

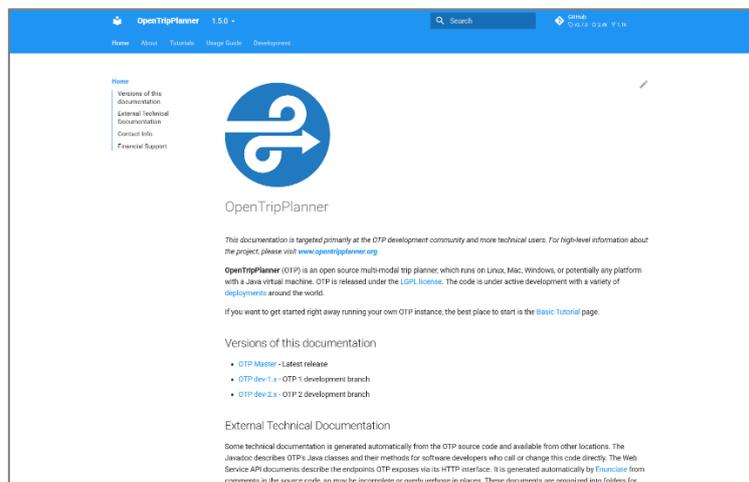


図 2-24 OpenTripPlanner ドキュメンテーション(<https://docs.opentripplanner.org/en/v1.5.0/>)

【SL007】 MUI (Material UI)

- ベンダー
 - オープンソース
- 公式サイト
 - <https://mui.com/material-ui>
- 本ソフトウェア・ライブラリの概要
 - React 向けの UI コンポーネントライブラリ。Google の Material Design ガイドラインに基づいてデザインされており、モダンな UI を簡単に構築できる。
- 開発するシステムにおいて利用する機能の詳細
 - WEB アプリの UI 構築
 - フォームや入力画面のデザイン
- イメージ

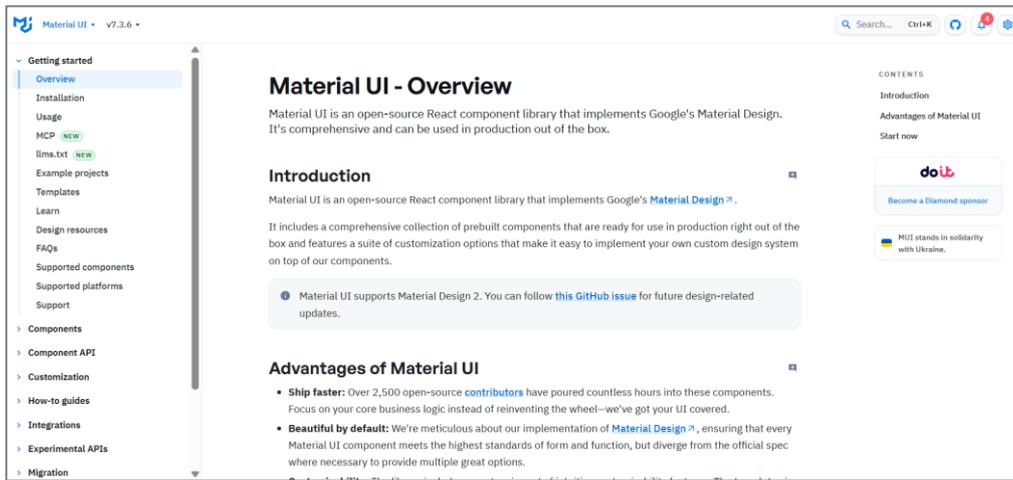


図 2-25 Material UI 概要 (<https://mui.com/material-ui/getting-started/>)

【SL008】 Docker

- ベンダー
 - オープンソース
- 公式サイト
 - <https://docs.docker.com/>
- 本ソフトウェア・ライブラリの概要
 - アプリケーションを軽量な仮想環境で動かすためのプラットフォーム
- 開発するシステムにおいて利用する機能の詳細
 - WEB アプリや API のデプロイ
 - マイクロサービスの構築
- イメージ

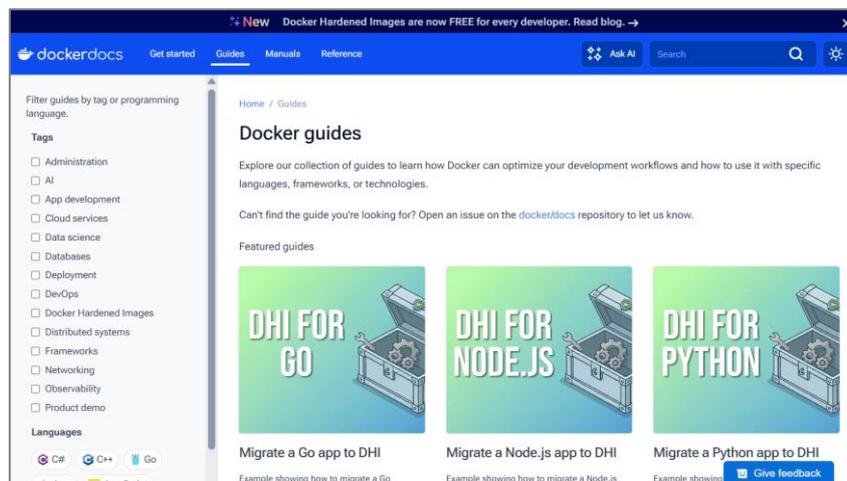


図 2-26 Docker ガイド (<https://docs.docker.com/guides/>)

【SL010】地域公共交通分析アプリ

- ベンダー
 - 新規開発
- 公式サイト
 - なし
- 本ソフトウェア・ライブラリの概要
 - 地域の公共交通に関するデータの編集・分析・シミュレーションを行う
- 開発するシステムにおいて利用する機能の詳細
 - GTFS 編集：運行本数の変更、路線の追加・変更・削除、便の追加・変更・削除、停留所の変更・追加・削除を行う
 - 可視化と分析：GTFS や停留所別乗降集計データ、駅・バス停間 OD データを地図と連動したグラフ・表で表示する
 - ◇ 路線や区間ごとの運行本数を示す「運行頻度図」
 - ◇ 道路ネットワークを使い到達範囲を算出する「到達圏分析」
 - ◇ 路線ごとの利用状況や断面輸送量分析
 - シミュレーション：公共交通の増減便数の変化を条件に、交通分担率の影響を試算する
 - 乗降実績データと GTFS のひも付け：乗降実績データを取り込み、GTFS データとひも付けをして、乗降集計データ、駅・バス停間 OD データを出力する
- イメージ

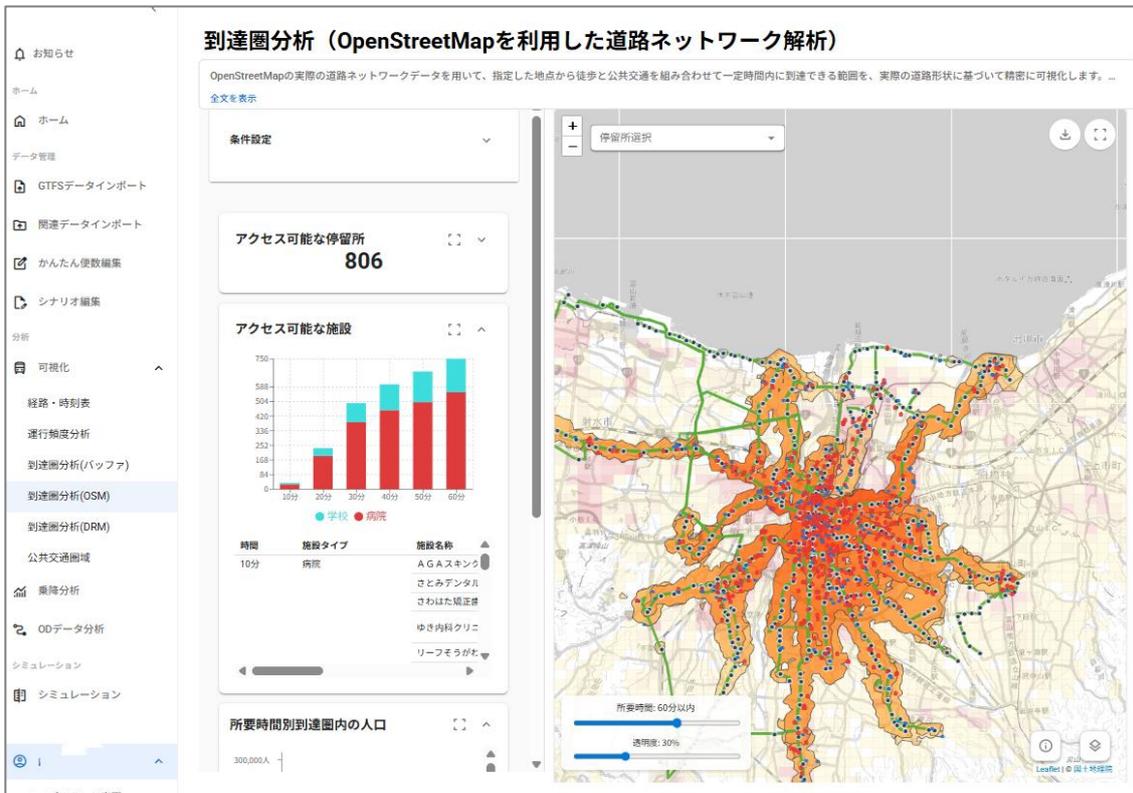


図 2-28 地域公共交通分析アプリ

2-1-5. 数理モデル・アルゴリズム（AL）の詳細

表 2-3 数理モデル・アルゴリズム一覧

※朱文字：新規開発・既存改修

ID	名称	説明	アルゴリズムを利用した機能
AL101	道路ネットワークを用いた到達圏分析 アルゴリズム	<ul style="list-style-type: none"> ● GTFS から取得した駅ポイント及び道路ネットワークのデータをインプットとし、道路ネットワーク分析による任意のポイントから到達圏域を時間別に算出する 	FN003
AL102	交通分担率の変化シミュレーション アルゴリズム	<ul style="list-style-type: none"> ● ダイヤ編成（増便など）前後の GTFS データをインプットとし、ダイヤ編成による整備効果を算出する 	FN006
AL103	乗降実績データと GTFS のひも付け アルゴリズム	<ul style="list-style-type: none"> ● 乗降実績データや運行記録データをインプットし、GTFS とひも付けられた乗降集計データ及び駅・バス停間 OD データを生成する 	FN004
AL104	インポートした GTFS の編集	<ul style="list-style-type: none"> ● インポートした GTFS の内容に含まれるルートや停留所・便数の要素を編集する 	FN002
AL105	運行記録（日報）の作成	<ul style="list-style-type: none"> ● 運行記録（日報）帳票作成ツールに GTFS をインポートして、設定した日付・路線の運行記録（日報）手入力シートを出力する ● 運行記録（日報）帳票作成ツールに GTFS をインポートして、設定した年月の運行記録（日報）登録シートを出力する ● 乗降集計データを CSV 形式で出力する 	FN015
AL106	路線と停留所によるグループ化	<ul style="list-style-type: none"> ● 同じ停留所名の停留所、路線を、それぞれ1つのグループにまとめる 	FN001 FN010

数理モデル・アルゴリズムの詳細を記す。なお、本業務において開発（新規・改修）を行う数理モデル・アルゴリズムを**朱文字**で示す。

【AL101】道路ネットワークを用いた到達圏分析アルゴリズム

- 本アルゴリズムの概要
 - 道路ネットワーク及び公共交通情報を活用して、任意の地点を基準とした時間別の到達圏域を計算し、地図上へ可視化するアルゴリズム。実際の移動経路と交通手段を考慮することで、現実的な圏域分析が可能となる。
 - **【SL006】OpenTripPlanner** を利用し、徒歩と公共交通を組み合わせることで指定時間内に到達可能な圏域を算出する。
- 本アルゴリズムを利用した機能
 - **【FN003】道路ネットワークを用いた到達圏分析**
- 本アルゴリズムの計算・処理の詳細
 - 従来のバッファ到達圏分析では、選択地点から一定距離内の範囲を単純に円形で表示する手法であり、道路構造を考慮していないため、現実の移動可能範囲との乖離（かいり）が生じる。
 - 本アルゴリズムでは、道路ネットワーク並びに公共交通機関の使用及び乗換えを考慮した到達圏域を解析する。徒歩→公共交通→徒歩といった複合的な移動手段を踏まえ、指定した時間内に到達可能な地域を算出する。
 - OpenTripPlanner の到達圏域算出のアルゴリズムは、以下のとおり。
 - ◇ 1. 出発地からの徒歩探索
 - OpenStreetMap から構築した道路ネットワークグラフを対象に、A*アルゴリズム (A-star search) で起点から徒歩のみで到達できる範囲を探索する。
 - 到達制約として徒歩距離上限を設定し、これを超える距離の探索は打ち切る。
 - 到達した停留所と到達時刻を初期停留所セットとして保存する。
 - 例：起点を自宅の緯度・経度、徒歩距離上限=1000mとする場合
200m先にバス停 A→到達 3分
500m先にバス停 B→到達 7分
900m先にバス停 C→到達 12分
{バス停 A：3分、バス停 B：7分、バス停 C：12分} を初期停留所セットとする。
 - ◇ 2. 公共交通探索 (RAPTOR)
 - GTFS の時刻表に基づいて、RAPTOR(Round-based Public Transit Optimized Router)で停留所の最短到着時刻を更新する。直通→1回乗換→2回乗換…と順ラウンドを繰り返して探索を広げ、更新がなくなった時点で終了。
 - ダイクストラ法のようにエッジを優先度付きキューで管理せず、停留所の到着時刻を配列で管理することにより処理を効率化。
 - 初期化：徒歩探索で到達した各停留所に「起点出発時刻+徒歩時間」を設定。それ以外の停留所は未到達∞とする。
 - 各ラウンド：前ラウンドで到達時刻が更新された停留所を起点に、それらを通る路線の時

刻表を順にスキャンし、「最も早く乗れる便」を探す。乗車可能な便が見付かったら、その便の停車順に従い、後続の停留所に到着できる最短時刻を更新。

- 更新があった停留所を、「次ラウンドで探索対象」に追加。更新がなくなったら探索終了。
- ◇ 3. 降車後の徒歩探索
 - 降車停留所から再び徒歩探索を行い、停留所から周辺の道路ネットワークに到達範囲を広げる。
- ◇ 4. 到達圏域の生成
 - ノード単位で得られた到達時間を地理空間上で連続的な面で表現できるよう変換し、指定カットオフごとに等値線抽出、ポリゴン化した到達圏域 (isochrone) を生成する。
- モードは徒歩と公共交通とし、考慮される経路パターンは以下のとおり。徒歩と公共交通を組み合わせた多段経路を含む。
 - ◇ 徒歩のみ
 - ◇ 徒歩→公共交通→徒歩
 - ◇ 徒歩→公共交通→公共交通→徒歩
- 到達圏分析の処理フローを以下に示す。

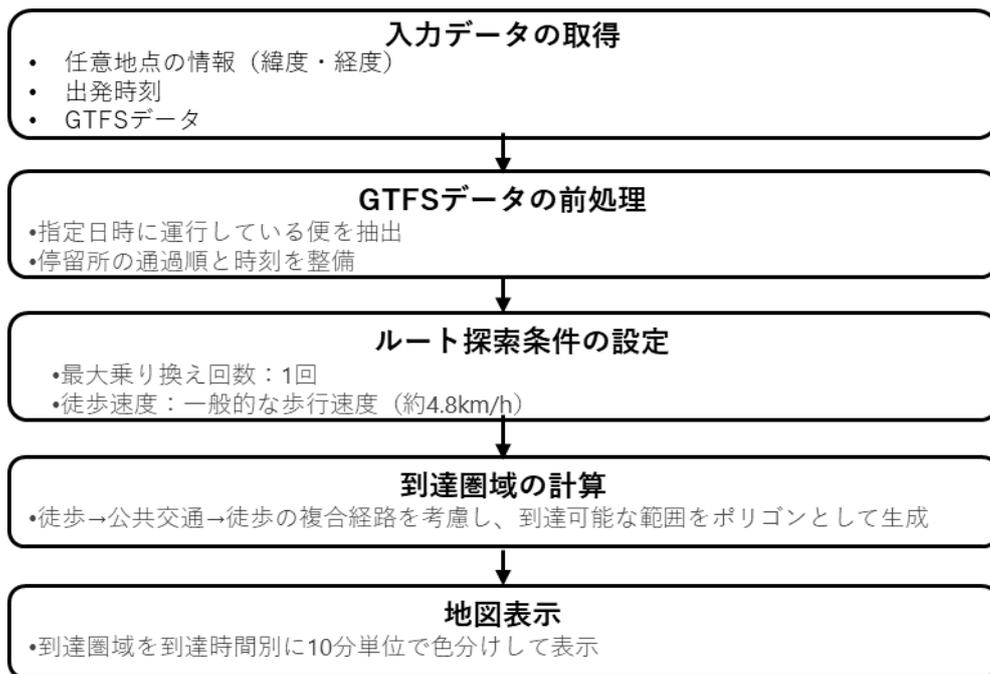


図 2-29 道路ネットワークを用いた到達圏分析のフローチャート

- 入力データの取得
 - ◇ ユーザーが指定した地点を地理座標として取得する。
 - ◇ 出発時刻又は到着時刻を指定し、GTFS の時刻情報と照合し、乗車可能な便を判定する。
 - ◇ 【FN001】【FN010】でインポートした GTFS のうち stops.txt、stop_times.txt、trips.txt、calendar.txt、calendar_dates.txt に記載されている情報を読み込む。
- GTFS データの前処理

- ◇ calendar.txt/calendar_dates.txt を参照し、指定日時に運行している service_id を抽出し、trips.txt と結合することで算出対象となる便 (trip_id) をフィルタリングする。
- ◇ stop_times.txt を trip_id ごとに stop_sequence 順に並び、停留所の通過順を時系列で整理する。
- ルート探索条件の設定
 - ◇ 停留所での到着・出発時刻差を用いて、最短乗換時間（例：5分）を条件とし、停留所間の距離から乗り換え可能な便を絞り込む。
 - ◇ 徒歩速度は約 4.8 km/h とする。
 - ◇ 徒歩移動の許容時間は 90 分までとする。
- 到達圏域の計算
 - ◇ 徒歩→バス→徒歩を考慮した経路を検索する。具体的には、出発地点から徒歩でアクセス可能な停留所を探索し、公共交通を用いて目的地候補となる停留所まで移動する。到着停留所から徒歩で移動可能な範囲を探索する。
 - ◇ これらの経路の合計移動時間が指定時間に収まるかを判定する。
 - ◇ 到達可能な地点群を空間的に結び、ポリゴンを表現する。
- 地図表示
 - ◇ 到達圏域を時間別に色分けして表示する。

【AL102】交通分担率の変化シミュレーションアルゴリズム

- 本アルゴリズムの概要
 - GTFS データ及び乗降集計データを入力として、運行本数の変更に応じて公共交通分担率変化のシミュレーションを行い、断面交通量の変化及びこれに基づく便益を算出する。

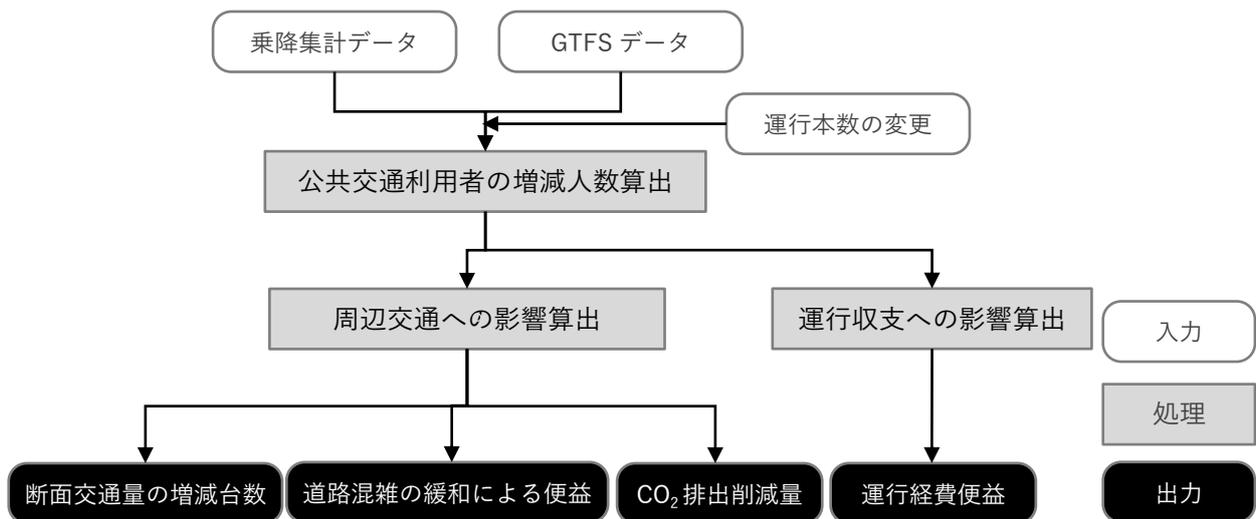


図 2-30 【AL102】交通分担率の変化シミュレーションアルゴリズムの概要

- 本アルゴリズムを利用した機能
 - 【FN006】交通分担率の変化シミュレーション
- 本アルゴリズムの計算・処理の詳細
 - 交通分担率の変化シミュレーションでは、運行本数の変更内容に対して、増便感度、減便感度を乗じ

て、公共交通利用者の増減人数を算出する。

- 公共交通の増減人数を基に、「費用便益分析マニュアル(令和7年8月 国土交通省道路局 都市局)」「鉄道プロジェクトの評価手法マニュアル(2012年改訂版、2025年3月一部変更) 国土交通省鉄道局)」等を参考に、便益、CO₂排出量等を算出する。
- 増減便による利用者数の算出及び便益の算出に用いている手法により、以下のような場合はシミュレーションができない。
 - ◇ 新規の路線を追加した時の便益
 - ◇ 路線を廃止(運行本数をゼロ)したときの便益
 - ◇ 路線や経由地、発着地点を変更したときの便益
 - ◇ 休日のみ運行の路線(便益算出時は平日の交通量を使用しているため)
 - ◇ エリア内を循環する路線やルートを往復する路線

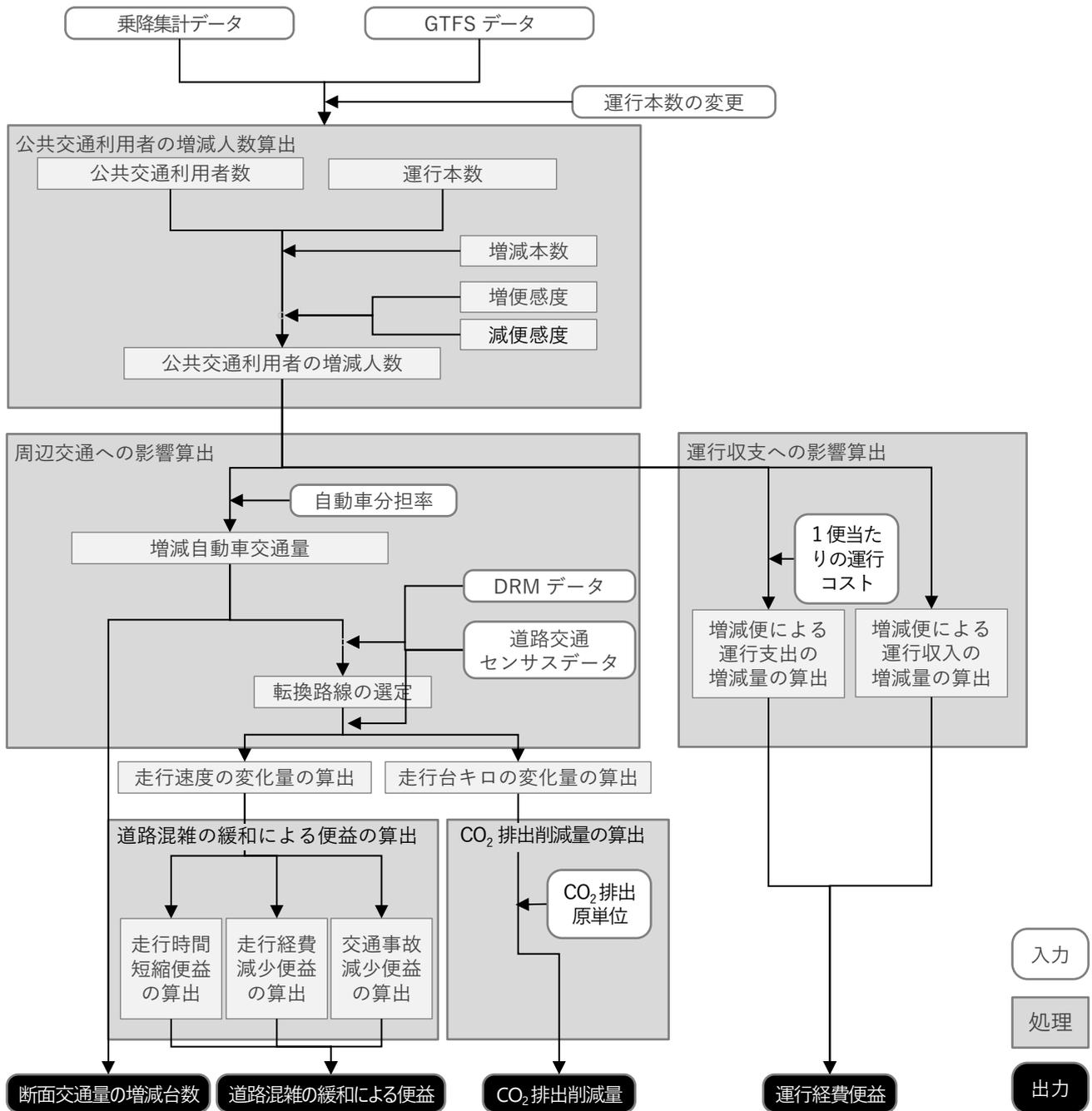


図 2-31 交通分担率の変化シミュレーションのフロー

- 乗降集計データ
 - ◇ IC カード等のデータを基に平日の路線ごとの1日の乗降者数を取得し使用する。
 - 乗降者数はICカードの利用者数からICカードの利用率を割り戻して算出する。
 - 平日1日以上 of データを使用することとし、複数日のデータがある場合には平日の平均値を使用する。
- GTFS データ
 - ◇ GTFS データの、route_id、agency_id、trips_id、shape_id、shape_pt_lat、shape_pt_lon、fare_id、price を用い、各路線の平日の運行本数、運行距離、運賃を取得または算出する。

- 一つの路線に複数の route_id がある場合には運行本数は合計、運行距離は平均を算出し使用する。
 - 運賃については、便益算出を簡素化するため、起点から終点まで移動していたと仮定し、起終点間の運賃を使用する。
- 運行本数の変更
- ◇ 現在の運行本数に対して、増便本数または減便本数（整数値）を入力する。
 - ◇ 減便本数は現状の運行本数未満とする。
- 公共交通利用者の増減人数算出
- ◇ 公共交通利用者の増減人数の算出式は以下のとおりとする。

増加人数（人／日）＝利用者数_{現状}×増便感度×（増便本数／運行本数_{現状}）

減少人数（人／日）＝利用者数_{現状}×増便感度×（減便本数／運行本数_{現状}）

増便感度、減便感度はともに 0.5 とし、ユーザーにより任意の値に変更可能とする。

- 周辺交通への影響算出
- ◇ 増減自動車交通量の算出
 - 増減自動車交通量の算出は以下のとおりとする。
- 増加自動車交通量（台／日）＝増加人数×自動車分担率－バス増便本数
- 減少自動車交通量（台／日）＝減少人数×自動車分担率－バス減便本数
- 自動車分担率は 0.465（令和 3 年度 全国都市交通特性調査）とし、ユーザーにより任意の値に変更可能とする。
- ◇ 転換路線の選定
 - 転換路線は道路交通センサスの対象路線を対象路線とし、路線の起終点間を結ぶ最短の経路をダイクストラ法で選定することとする。
 - route_id が複数ある場合には、各座標の重心を算出し、起終点とすることとする。
 - 道路交通センサスの GIS データについては DRM データを使用し、転換路線を選定する。
 - ◇ 走行速度の変化量の算出
 - 自動車交通量増減後の各路線の走行速度の変化量は、信号交差点密度別の混雑度と年平均旅行速度の関係の表（街路事業における費用便益分析マニュアル（案）（平成 12 年 6 月））を用いて算出する。
 - 算出方法は以下のとおりとする。

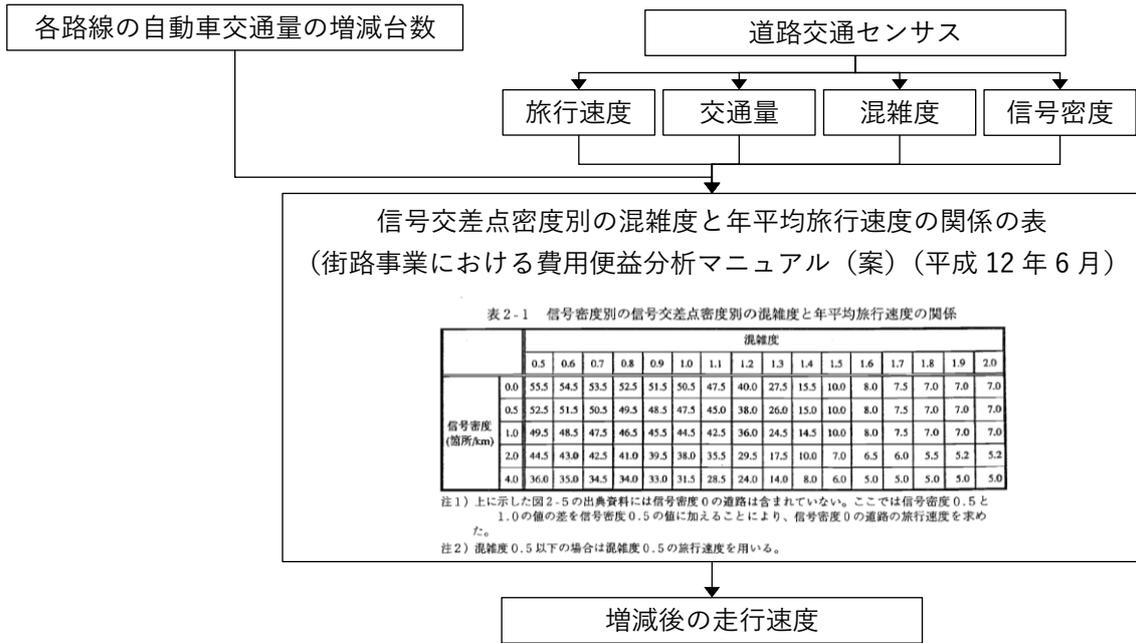


図 2-32 走行速度の算出方法

- 断面交通量の増減台数の算出
 - ◇ 断面交通量の増減台数は、増加自動車交通量（台／日）または減少自動車交通量（台／日）とする。
- 道路混雑の緩和による便益の算出
 - ◇ 走行時間短縮便益の算出
 - 走行時間短縮便益の算出方法は「費用便益分析マニュアル（令和 7 年 8 月）」を参考にし、以下のとおりとする。

$$\text{走行時間短縮便益} : B T = B T_0 - B T_w$$

$$\text{総走行時間費用} : B T_i = \sum \sum (Q_{i|j|l} \times T_{i|j} \times \alpha_j) \times 365$$

- ここで、
- $B T$: 走行時間短縮便益（円／年）
 - $B T_i$: 運行本数 i の場合の総走行時間費用（円／年）
 - $Q_{i|j}$: 運行本数 i の場合の区間 l における車種 j の交通量（台／日）
 - $T_{i|j}$: 運行本数 i の場合の区間 l における車種 j の走行時間（分）
 - α_j : 車種 j の時間価値原単位（円／分・台）
 - i : 運行本数変更後の場合 w 、変更前の場合 o
 - j : 車種
 - l : 区間

ここで、車種は乗用車類とし、時間価値原単位 α_j を 48.89 円／分・台とする。

◇ 走行経費減少便益の算出

- 走行時間短縮便益の算出方法は「費用便益分析マニュアル（令和7年8月）」を参考にし、以下のとおりとする。

$$\text{走行時間短縮便益} : BR = BR_0 - BR_W$$

$$\text{総走行時間費用} : BR_i = \sum_j \sum_l (Q_{ijl} \times L_l \times \beta_j) \times 365$$

ここで、	BR	: 走行経費減少便益（円／年）
	BR_i	: 運行本数 i の場合の総走行経費（円／年）
	Q_{ijl}	: 運行本数 i の場合の区間 l における車種 j の交通量（台／日）
	L_l	: 区間 l の延長（km）
	β_j	: 車種 j の時間価値原単位（円／分・km）
	i	: 運行本数変更後の場合 W 、変更前の場合 O
	j	: 車種
	l	: 区間

ここで、車種は乗用車類とし、「費用便益分析マニュアル（令和7年8月）」の一般道（市街地）の走行経費原単位を用いることとする。

◇ 交通事故減少便益の算出

- 交通事故減少便益の算出方法は「費用便益分析マニュアル（令和7年8月）」を参考にし、以下のとおりとする。
- なお、算定式は一般道路・D I D地区の算定式を用いることにする。

$$\text{年間総事故減少便益} \quad : \quad B A = B A_o - B A_w$$

$$\text{交通事故の社会的損失} \quad : \quad B A_i = \sum (A A_{i,l})$$

$$\text{2車線の場合} \quad A A_{i,l} = 1850 \times X_{1i,l} + 280 \times X_{2i,l}$$

$$\text{4車線以上の場合} \quad A A_{i,l} = 1110 \times X_{1i,l} + 370 \times X_{2i,l}$$

ここで、

$B A$: 年間総事故減少便益（千円／年）

$B A_i$: 運行本数 i の場合の交通事故の社会的損失（千円／年）

$A A_{i,l}$: 運行本数 i の場合の区間 l における交通事故の社会的損失（千円／年）

$X_{1i,l} = Q_{i,l} \times L_l$: 運行本数 i の場合の区間 l における走行台キロ（千台km／日）

$X_{2i,l} = Q_{i,l} \times Z_l$: 運行本数 i の場合の区間 l における走行台箇所（千台箇所／日）

$Q_{i,l}$: 運行本数 i の場合の区間 l における交通量（千台／日）

L_l : リンク l の延長（km）

Z_l : リンク l の信号交差点数（箇所）

i : 運行本数変更後の場合 w 、変更前の場合 o

l : 区間

ここで、区間が2車線の場合は、一般道路・D I D地区・2車線の交通事故損失額算定式を用い、4車線の場合は、中央帯の有無を考慮しない場合の一般道路・D I D地区・4車線の交通事故損失額算定式を用いることとする。

◇ 道路混雑の緩和による便益の算出

- 道路混雑の緩和による便益は以下のとおりとする。

$$\text{道路混雑の緩和による便益（円／年）} = B T + B R + B A$$

ここで、

$B T$: 走行時間短縮便益（円／年）

$B R$: 走行経費減少便益（円／年）

$B A$: 年間総事故減少便益（千円／年）

➤ CO₂排出削減量の算出

- ◇ CO₂排出削減量の算出方法は以下のとおりとする。

$$\text{CO}_2 \text{ 排出削減量} = \text{削減走行台キロ} \times \text{CO}_2 \text{ 排出原単位} \times 365$$

ここで、CO₂排出原単位は 123g-CO₂/台kmとする。

➤ 運行収支への影響算出

- ◇ 増減便による運行支出の増減量の算出

- 増減便による運行支出の増減量の算出方法は以下のとおりとする。

$$\text{運行支出の増減量} = \text{増減便本数} \times \text{路線延長} \times \text{台キロ当たりの運行コスト}$$

ここで台キロ当たりの運行コストは 520.9 円/台km（「2023/2024 年度版（令和 5・6 年度）日本のバス事業」（公益社団法人日本バス協会）の平均値）とし、ユーザーにより任意の値に変更可能とする。

- ◇ 増減便による運賃収入の増減量の算出

- 増減便による運行収入の増減量の算出方法は以下のとおりとする。

$$\text{運賃収入の増減量} = \text{増減利用者数} \times \text{起終点間の運賃}$$

- ◇ 運行経費便益の算出

- 運行経費便益の算出方法は以下のとおりとする。

$$\text{運行経費便益の算出（円/年）} = (\text{運賃収入の増減量} - \text{運行支出の増減量}) \times 365$$

【AL103】乗降実績データと GTFS のひも付けアルゴリズム

● 本アルゴリズムの概要

- 本アルゴリズムは、【DT005】乗降実績データを、GTFS と整合した【DT004】乗降集計データ及び【DT006】駅・バス停間 OD データへ変換するための処理手法を整理したものである。
- 一般に、IC カードシステム及び運賃箱システムから出力される乗降実績データには、乗車・降車日時や乗車・降車駅（停留所）名は含まれているが、どの便や路線に対応するかという項目は含まれず空欄となっているケースが多い。そのままでは GTFS と連携した分析に利用することができず、地図上への可視化や、路線単位・区間単位・停留所単位で分析ができないという課題がある。
- 乗降実績データは、国土交通省 COMmmmons の「モビリティ・データ標準化プロジェクト」にて策定した乗降実績データ標準仕様書（鉄道・バス）

[\(https://www.mlit.go.jp/commmmons/document/005/\)](https://www.mlit.go.jp/commmmons/document/005/) の様式に準拠したデータであり、乗降

実績データをインプットとして乗降集計データ、駅・バス停間ODデータを生成する。

- 本アルゴリズムを利用した機能
 - 【FN004】乗降実績データの入出力機能
- 本アルゴリズムの計算・処理の詳細
 - 乗降実績データを GTFS データと照合・統合し、乗降集計データ及び駅・バス停間 OD データを作成する。
 - 図 2-33 の「①stop_name と乗車・降車日時から trip_id と stop_sequence を推定」及び「②乗降実績データに必要な GTFS の項目をひも付け・補完」は、一定の条件を満たす場合に自動処理が可能な工程である。「③乗降実績データから乗降集計データへの変換処理」及び「④乗降実績データから駅・バス停間 OD データへの変換処理」は、システム上で自動的に実行される集計処理である。
 - 「①stop_name と乗車・降車日時から trip_id と stop_sequence を推定」及び「②乗降実績データに必要な GTFS の項目をひも付け・補完」で使用する乗降実績データの仕様は【DT007】乗降実績データに記載のとおりであり、表 2-4 の様式とする。表 2-4 「入力可否」の「①で推定」「②で補完」は処理の進行に伴うデータ状態の変化を明確にすることを目的とし、各処理段階で値が更新される項目を一覧にしたものである。
 - 「①stop_name と乗車・降車日時から trip_id と stop_sequence を推定」を実施する際には、表 2-4 で示す項目の CSV 形式の乗降実績データを準備する。その際、「乗車日時」「降車日時」と「乗車駅（停留所）名」「降車駅（停留所）名」（stop_name）の値は必須入力値となる。これらの値を使用して、「便コード」（乗車・降車した便の trip_id）及び「乗車停留所連番」「降車停留所連番」（その便の停留所順（stop_sequence））を推定する。
 - 「②乗降実績データに必要な GTFS の項目をひも付け・補完」は、①の処理でひも付けた「便コード」（乗車・降車した便の trip_id）、「乗車停留所連番」「降車停留所連番」（その便の停留所順（stop_sequence））を基に、「運行事業者コード」や「系統 ID」「経路 ID」「乗車駅（停留所）コード」などの GTFS の項目をひも付ける処理である。
 - 「③乗降実績データから乗降集計データへの変換処理」及び「④乗降実績データから駅・バス停間 OD データへの変換処理」は、②の処理で作成した乗降実績データから乗降集計データ及び駅・バス停間 OD データを生成する処理である。

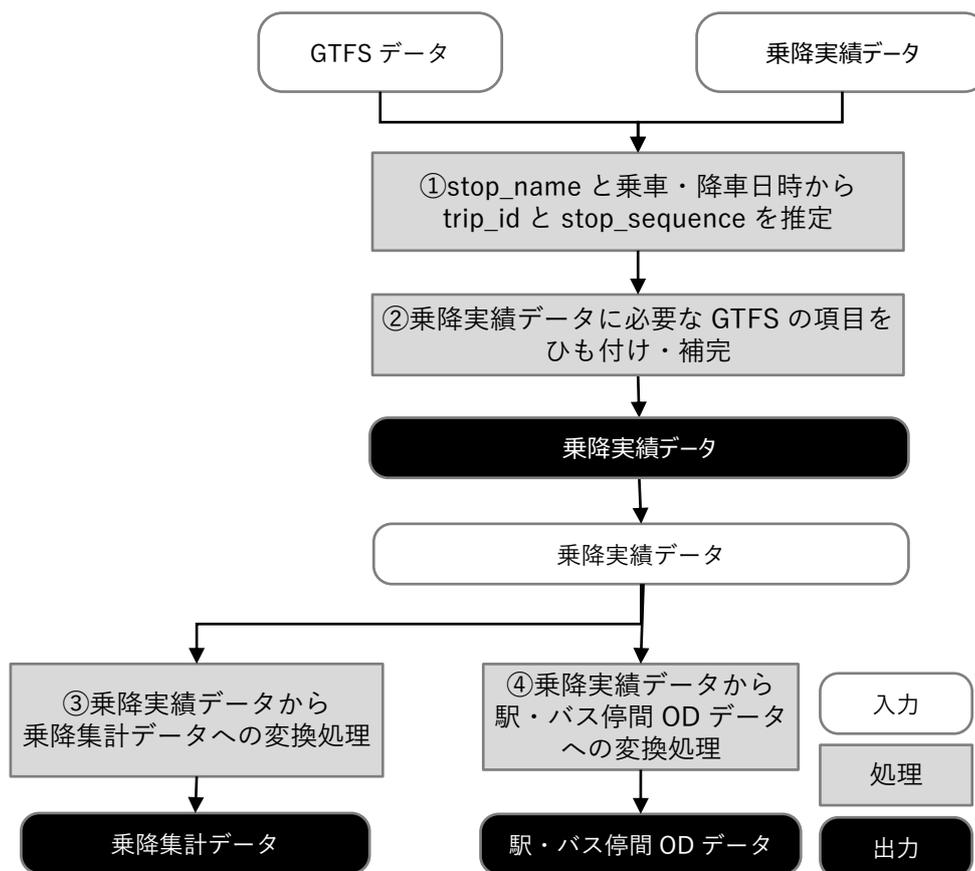


図 2-33 運行記録や乗降実績データから乗降集計データ及び駅・バス停間 OD データ生成のフローチャート

表 2-4 乗降実績データの項目及び各処理で値が更新される項目

No.	物理名	論理名	代替可能な GTFS の項目名	入力可否
1	ridership_record_id	乗降実績 ID		任意
2	ic_card_agency_identification_code	IC カード識別コード		任意
3	ic_card_issuer_code	IC カード発行事業者コード		任意
4	ic_card_issuer_name	IC カード発行事業者名		任意
5	ic_card_feature_type	IC カード機能区分		任意
6	ticket_type_area_code	券種エリアコード		任意
7	ticket_type	券種区分		任意
8	ticket_type_name	券種名		任意
9	ticket_valid_start_date	券有効開始日		任意
10	ticket_valid_end_date	券有効終了日		任意
11	transportation_mode_code	交通モードコード	routes.txt の「route_type」	②で補完
12	ic_card_usage_detail_id	IC カード利用明細 ID		任意
13	operating_agency_code	運行事業者コード	agency.txt の「agency_id」	②で補完

No.	物理名	論理名	代替可能な GTFS の項目名	入力要否
14	operating_agency_name	運行事業者名	agency.txt の 「agency_name」	②で補完
15	serviced_office_code	営業所コード	office_jp.txt の「office_id」	②で補完
16	serviced_office_name	営業所名	office_jp.txt の 「office_name」	②で補完
17	route_pattern_id	系統 ID	pattern_jp.txt の 「jp_pattern_id」	②で補完
18	route_pattern_number	系統番号	routes.txt の 「route_short_name」	②で補完
19	service_line_name	路線名	routes.txt の 「route_long_name」	②で補完
20	route_name	経路名	Routes.txt の 「route_long_name」 また は「route_short_name」の どちらか	②で補完
21	trip_code	便コード	trips.txt の「trip_id」	①で推定
22	timetable_number	ダイヤ番号		任意
23	vehicle_number	車両番号		任意
24	operation_type	処理区分		任意
25	operation_detail_type	処理詳細区分		任意
26	boarding_area_code	乗車エリアコード		任意
27	boarding_station_sequence	乗車停留所連番	stops.txt の 「stop_sequence」	①で推定
28	boarding_station_code	乗車駅（停留所）コード	stops.txt の「stop_id」	②で補完
29	boarding_station_name	乗車駅（停留所）名	stops.txt の「stop_name」	必須
30	boarding_at	乗車日時		必須
31	boarding_route_id	乗車経路 ID	routes.txt の「route_id」	②で補完
32	transfer_area_code_list	乗継エリアコードリスト	stops.txt の「stop_id」	②で補完
33	transfer_station_code_list	乗継駅（停留所）コードリスト	stops.txt の「stop_id」	②で補完
34	alighting_area_code	降車エリアコード		任意
35	alighting_station_sequence	降車停留所連番	stops.txt の 「stop_sequence」	①で推定
36	alighting_station_code	降車駅（停留所）コード	stops.txt の「stop_id」	②で補完
37	alighting_station_name	降車駅（停留所）名	stops.txt の「stop_name」	必須
38	alighting_at	降車日時		必須
39	payment_at	精算日時		任意

No.	物理名	論理名	代替可能な GTFS の項目名	入力要否
40	adult_challenged_passenger_count	大人障がい者利用者数		任意
41	adult_passenger_count	大人利用者数		任意
42	child_challenged_passenger_count	小児障がい者利用者数		任意
43	child_passenger_count	小児利用者数		任意
44	passenger_classification_type	利用者分類区分		任意
45	payment_type	支払い区分		任意

- ①stop_name と乗車・降車日時から trip_id と stop_sequence を推定
 - ◇ 一般に、IC カードシステム及び運賃箱システムから出力される乗降実績データには、乗車・降車日時や乗車・降車駅（停留所）名は含まれているが、利用者が乗車・降車した便（trip_id）が含まれていないケースが多い。そのため、どの便に乗車したかを推定する必要がある。また、循環路線では同一停留所に複数回停車する場合があります、停留所名だけでなく停留所順（stop_sequence）の推定も不可欠となる。
 - ◇ 本処理では、乗降実績データに記載されている乗車・降車日時と、乗車・降車停留所名（stop_name）を使用し、乗車・降車した便（trip_id）、その便の停留所順（stop_sequence）を推定する。
 - ◇ 推定で利用する乗降実績データは、出発時刻または到着時刻の両方が記録されており、一定の条件（遅延が 20 分以内で、乗車及び降車両方の情報が整備されているデータ）を満たす場合に限り使用可能とする。ただし、この推定値は遅延の度合いにも影響を受けるため、完全性及び正確性が常に保証されるものではない。

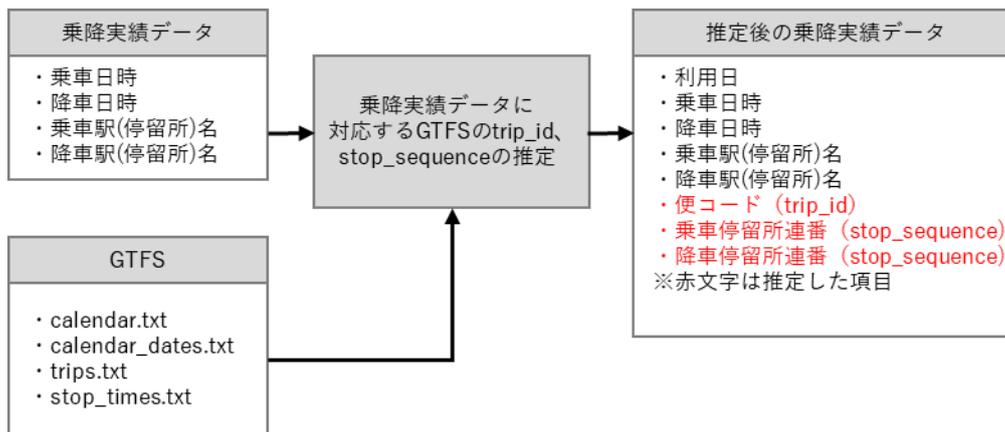


図 2-34 ①stop_name と乗車・降車時間から trip_id と stop_sequence を推定する処理の概要

- ◇ 本処理で推定に使用する項目は以下のとおりである。

表 2-5 ①stop_name と乗車・降車時間から trip_id と stop_sequence を推定する処理に使用する項目

No.	物理名	論理名	代替可能な GTFS の項目名
29	boarding_station_name	乗車駅（停留所）名	stops.txt の「stop_name」
30	boarding_at	乗車日時	
31	alighting_station_name	降車駅（停留所）名	stops.txt の「stop_name」
32	alighting_at	降車日時	

◇ GTFS と異なる停留所名の表記の整理

- 「乗車駅（停留所名）」「降車駅（停留所名）」は GTFS の stop_name と同じ表記とする必要がある。
- GTFS の stop_name と同じ表記になっていない場合は、「修正前の停留所名」と「stop_name」を一元的に管理できるように対照表を作成することで、停留所名と stop_name の置換が容易となる。
- ことでんグループ「IruCa」の対照表の作成例
 本実証内でことでんグループより受領した「IruCa」の乗降実績データを対象に、対照表を作成した例を示す。
 「IruCa」には一部の停留所名が stop_name と異なる表記で整理されていた。「IruCa」で整理されていた停留所名の特徴は以下のとおりである。これらの停留所については、stop_name との対照表を作成し、stop_name の表記に統一した。
 - 施設名が「・」で区切られている
 - 愛称が含まれている
 - アルファベットが全角・半角で混在している

表 2-6 「IruCa」の停留所名と stop_name の対照表の例

修正前の停留所名	GTFS の stop_name
県庁通り・中央公園前	県庁通り中央公園前
イオン高松東店	歯科医療センター・イオン高松東店
長神・キョーエイ空港店	長神
レインボーロード入口・ジャンプ前	レインボーロード入口
小坂総合運動公園前	小坂・総合運動公園前
J A 川東支店	JA 川東支店

- 富山地方鉄道「えこまいか」の対照表の作成例
 本実証内で富山地方鉄道より受領した「えこまいか」の乗降実績データを対象に、対照表を作成した例を示す。
 「えこまいか」の乗降実績データには、一部の停留所名が GTFS の stop_name と異なる表記で整理されていた。「えこまいか」で整理されていた停留所名の特徴は以下のとおりで

ある。これらの停留所については、stop_name との対照表を作成し、表記を統一した。

修正前の停留所名

- 1 漢数字で表記
- 2 括弧による表記無し
- 3 最後に「前」がついている

GTFS

- 1 数字で表記
- 2 括弧による表記あり
- 3 最後に「前」がついていない

乗降実績データ

乗車停留所・降車停留所	
0101 富山駅前	
...	
0301 総曲輪	
0303 星井町一丁目 1	
...	
0309 今泉 2	
0310 掛尾	
...	
0336 岩稲温泉楽今日館前 3	
0337 榆原口	
0338 榆原	
...	
1910 赤田	
1911 二俣	
...	

GTFS

Stop_name
富山駅前
...
総曲輪
星井町1丁目 1
...
今泉(笹津) 2
掛尾
...
岩稲温泉楽今日館 3
榆原口
榆原
...
赤田(辰尾団地)
二俣
...

図 2-35 「えこまいか」の停留所名と stop_name の対照表の例

- ◇ 乗降実績データ、乗車・降車日時、乗車・降車停留所名 (stop_name) を使用し、乗車・降車した便コード (trip_id)、その便の停留所順 (stop_sequence) を推定する。
- ◇ trip_id の推定手順は以下のとおりである。

表 2-7 乗降実績データから乗車した便 (trip_id) 及び stop_sequence の推定フロー

手順	処理内容
1	乗車日時及び降車日時から IC カードを機器にタッチした日を抽出し、その日に運行する trip_id のリストを抽出する。リスト抽出の手順は以下のとおりである。 <ul style="list-style-type: none"> ● calendar.txt から、乗車日時・降車日時に運行予定となっている service_id を抽出する。 ● calendar_dates.txt からその日だけ特別に運行される便や運休となる便があるかどうかをチェックし、運行情報を補正する。 ● 有効な service_id にひも付く trip_id を trips.txt から抽出する。
2	乗車駅 (停留所) 名が一致し、乗車日時が departure_time ± 5 分以上かつ departure_time ± 5 分以下の trip_id を取得する。 該当する trip_id が見つかった場合、stop_times.txt を参照し、乗車停留所に対応する stop_sequence を取得する。循環路線などで同一停留所が複数回登場し、複数の stop_sequence が見つかった場合は、乗車時間に最も近い通過時刻の stop_sequence を選定する。

手順	処理内容
	見つからない場合は、departure_time の±10分、±15分、±20分でマッチとループを回す。 本アルゴリズムが実装されている機能では遅延時間の許容値を0分・5分・10分・15分・20分で指定可能であり、指定した値がループの上限値となる。
3	降車も乗車と同様に実施する。 降車駅（停留所）名が一致し、降車日時が arrival_time - 5分前以上かつ arrival_time + 5分以下の trip_id を取得する。 該当する trip_id が見つかった場合、stop_times.txt を参照し、降車停留所に対応する stop_sequence を取得する。循環路線などで同一停留所が複数回登場し、複数の stop_sequence が見つかった場合は、降車時間に最も近い通過時刻の stop_sequence を選定する。 見つからない場合は、arrival_time の±10分、±15分、±20分でマッチとループを回す。
4	乗車と降車で推定した trip_id をリスト化する。推定した trip_id に一致するものがあれば、当該 trip_id を「乗車・降車した便」として推定する。 同じ route_id で複数の trip_id が候補にあった場合は、乗車時間と降車時間の差をそれぞれ比較し、より差が小さい trip_id を選定する。

乗降実績データ

乗車日時	2026年4月1日 13時5分
降車日時	2026年4月1日 13時15分
乗車(停留所)名	A停留所
降車(停留所)名	B停留所

4月1日に運行するTrip_id

- ・Aルート13時10分
- ・Aルート13時15分
- ・Bルート13時15分
- ・Cルート13時30分
- ・Dルート14時20分

乗車/降車日時が departure_time / arrival_time の ±5分以内の trip_id

- ・Aルート13時10分
- ・Aルート13時15分
- ・Bルート13時15分

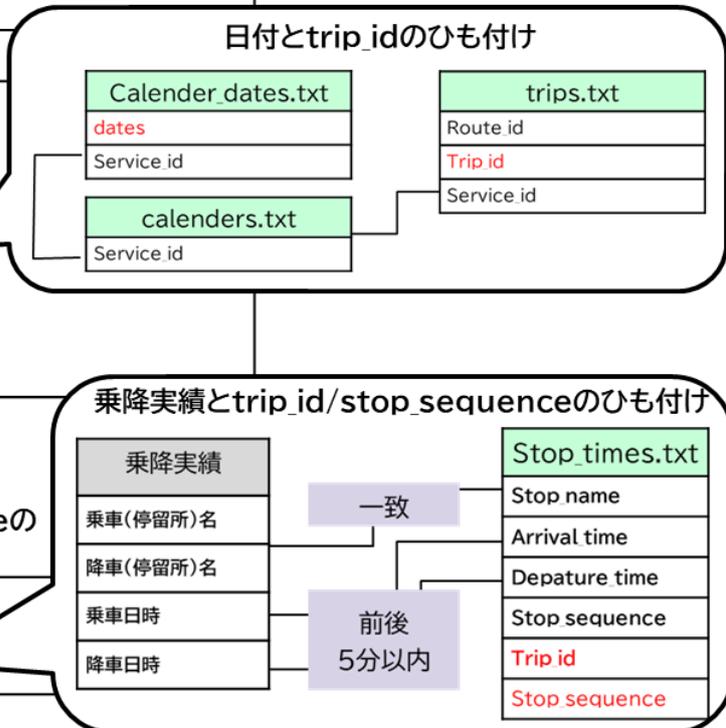


図 2-36 乗降実績データから乗車した便 (trip_id) 及び stop_sequence の推定のイメージ

- ◇ 本処理において trip_id 及び stop_sequence が推定されることにより、乗降実績データの以下の項目が更新され、値が補完される。

表 2-8 ①stop_name と乗車・降車時間から trip_id と stop_sequence で推定する項目

No.	物理名	論理名	代替可能な GTFS の項目名
21	trip_code	便コード	trips.txt の「trip_id」
27	boarding_station_sequence	乗車停留所連番	stops.txt の「stop_sequence」
35	alighting_station_sequence	降車停留所連番	stops.txt の「stop_sequence」

➤ ②乗降実績データに必要な GTFS の項目をひも付けと補完

- ◇ ここまでの処理で推定・補完した trip_id (便コード)、stop_name (乗車駅 (停留所) 名、降車駅 (停留所) 名) の情報を基に、【DT007】乗降実績データに必要なとなる agency_id、office_name、jp_parent_route_id、route_id、route_long_name、stop_id を GTFS からひも付ける。

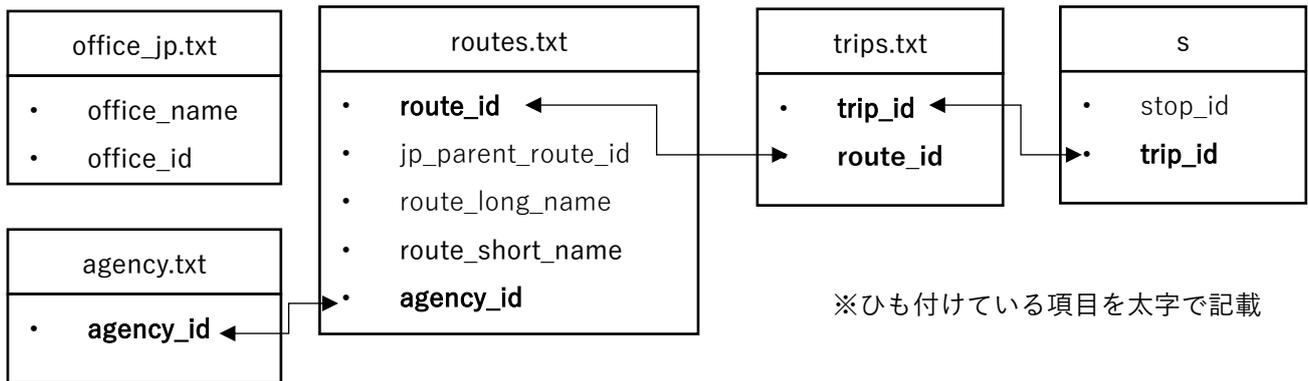


図 2-37 GTFS 各項目のひも付け

- ◇ ①の処理で確定した trip_id 及び stop_sequence を基に、routes.txt・stop_times.txt・stops.txt などの GTFS ファイルから、路線・便・停留所に関する情報を取得し、乗降実績データの空欄項目を GTFS の属性情報で補完する。補完される項目は以下のとおりである。
- ◇ これにより乗降実績データを完成させる。
- ◇ 完成した乗降実績データは③、④の処理で乗降集計データ及び駅・バス停間 OD データへ変換を行うために使用する。

表 2-9 ②乗降実績データに必要な GTFS の項目のひも付け・補完処理により補完される項目

No.	物理名	論理名	代替可能な GTFS の項目名
11	transportation_mode_code	交通モードコード	routes.txt の「route_type」
13	operating_agency_code	運行事業者コード	agency.txt の「agency_id」

No.	物理名	論理名	代替可能な GTFS の項目名
14	operating_agency_name	運行事業者名	agency.txt の 「agency_name」
15	serviced_office_code	営業所コード	office_jp.txt の「office_id」
16	serviced_office_name	営業所名	office_jp.txt の 「office_name」
17	route_pattern_id	系統 ID	pattern_jp.txt の 「jp_pattern_id」
18	route_pattern_number	系統番号	routes.txt の 「route_short_name」
19	service_line_name	路線名	routes.txt の 「route_long_name」
20	route_name	経路名	routes.txt の 「route_long_name」または 「route_short_name」のどち らか
26	boarding_station_code	乗車駅（停留所）コード	stops.txt の「stop_id」
31	boarding_route_id	乗車経路 ID	routes.txt の「route_id」
33	transfer_station_code_list	乗継駅（停留所）コードリスト	stops.txt の「stop_id」
36	alighting_station_code	降車駅（停留所）コード	stops.txt の「stop_id」

➤ ③乗降実績データから乗降集計データへの変換処理

- ◇ 「②乗降実績データに必要な GTFS の項目をひも付け・補完」で作成した乗降実績データを基に、【DT004】乗降集計データを作成する。ここでは、乗降実績データに乗車と降車の記録が両方ある場合のみを対象とする。
- ◇ データの準備
 - 以下のデータを準備する
【DT007】乗降実績データ
- ◇ 対象レコードの抽出
 - 乗降実績データのうち、乗車記録と降車記録が両方存在するレコードのみを抽出する。
- ◇ 乗車記録の集計
 - 乗降実績データの各レコードは「1 人の乗車あるいは降車」を表す。そのため、乗車停留所の 1 レコードを「当該便・停留所・日付に 1 人乗車した」とカウントできる。
 - 各レコードの乗車停留所・trip_id・route_id・agency_id・stop_id・stop_sequence・日付が記載されているレコード数を集計・抽出する。
- ◇ 降車記録の集計
 - 乗車記録の集計と同様に、各レコードの降車停留所・trip_id・route_id・agency_id・stop_id・stop_sequence・日付が記載されているレコード数を集計・抽出する。

◇ 集計結果の統合

- 乗車・降車それぞれの集計結果を、同じ trip_id・stop_id・日付ごとにまとめる。
- 乗降人数は、同じ trip_id・stop_id・日付ごとに集計する。
- 集計したデータを【DT004】乗降集計データのとおり成型する。

表 2-10 【DT004】乗降集計データのデータ定義

物理名	論理名	GTFS データとひも付けられる項目	属性の型
date	日付		文字 (YYYYMMDD,8 桁)
agency_id	事業者 ID	●	文字
route_id	路線 ID	●	文字
trip_id	便 ID	●	文字
stop_id	停留所 ID	●	文字
stop_sequence	通過順位	●	数値
count_geton	乗車人数		数値
count_getoff	降車人数		数値

➤ ④乗降実績データから駅・バス停間 OD データへの変換処理

- ◇ ②乗降実績データに必要な GTFS の項目をひも付けで作成した乗降実績データを基に、【DT006】駅・バス停間 OD データを作成する。
- ◇ 【DT006】駅・バス停間 OD データとは同一利用者が同一便において行った乗車と降車の組を指す。
- ◇ 同じカード ID で、同じ時間帯・同じ路線に乗った乗車・降車のペアを駅・バス停間 OD データとして抽出する。
- ◇ データの準備
 - 以下のデータを準備する
【DT007】乗降実績データ
- ◇ 乗車・降車ペアの作成
 - 乗降実績データをカード ID ごとにグループ化する。
- ◇ 同じ日付・同じ route_id・同じ trip_id で乗車・降車のペアを抽出する。
 - グループ化したカード ID の、同じ日付・route_id・trip_id のレコードを抽出し、乗車記録と降車記録のペアを作成する。
 - 乗車・降車のペアを記録時間でソートし、記録された時間が早いレコードの stop_id を stopid_geton、遅い時間の stop_id を stopid_getoff とする。
- ◇ OD ペアごとに集計
 - 作成した乗車・降車の OD ペアを、date、agency_id、route_id、stopid_geton、stopid_getoff ごとに集計する。
 - 同じ乗車・降車のペアが複数あれば、count として人数を合計する。

- ◇ 駅・バス停間 OD データの成型
 - 【DT006】 駅・バス停間 OD データのとおり成型する。

表 2-11 【DT006】 駅・バス停間 OD データのデータ定義

属性名	説明	GTFS データとひも付けられる項目	属性の型
date	日付		文字 (YYYYMMDD,8 桁)
agency_id	事業者 ID	●	文字
route_id	路線 ID	●	文字
stopid_geton	乗車停留所 ID	●	文字
stopid_getoff	降車停留所 ID	●	文字
count	利用者数		数値

【AL104】 インポートした GTFS の編集

- 本アルゴリズムの概要
 - インポートした既存の GTFS データを基に、運行本数や路線の変更等を複数設定したシミュレーション用データを GUI 上で編集する。
 - インポートで取り込んだ既存 GTFS のフィードをコピーして編集することとする。
- 本アルゴリズムを利用した機能
 - 【FN002】 GTFS 編集機能
- 本アルゴリズムの計算・処理の詳細
 - 運行本数の増減変更
 - ◇ 対象単位は、route_id × direction_id × service_id を集計単位として処理する。
 - ◇ 増減便数の入力は、以下のとおり。
 - 倍率指定： $N = \text{round}(\text{base} \times \text{rate}) - \text{base}$ (base=現行総便数、rate=倍率)
 - 本数指定： $N = +k$ (k=追加本数)
 - ◇ 増便の配分方針は、連続便の出発時刻差が大きい区間から順に、区間の中央へ便を挿入する。以下のとおり。
 - 第 1 便（始発）と最終便は固定とする。
 - 隣接便ごとの出発時刻差を算出し、最大時刻差の区間を選択し当該区間の中央時刻を新規便の基準時刻とする。
 - 最大時刻差の区間が複数ある場合は、前時刻側（早い時間帯）を優先とする。
 - 第 1 便をテンプレート便として選定し、その stop_times を平行移動して新規便の時刻表を生成する。
 - (例) 既存便：07:00、07:30、08:10、08:30、09:10 に 1 便追加
 - ・ 出発時刻差=[30、40、20、40]
 - ・ 最大時刻差は、07:30-08:10 (40 分) (同値の場合は早い時間帯を優先)
 - ・ 中央時刻 07:50 に 1 本挿入。07:30 便をテンプレート便とし、各停留所の到着・出発時刻を+20 分平行移動して stop_times を生成。

- ・結果：07:00、07:30、07:50、08:10、08:30、09:10（出発時刻差=[30、20、20、20、40]）
- ◇ 減便の配分方針は、前後の便と“より詰まっている便”を優先して間引く。以下のとおり。
 - 第1便（始発）と最終便は固定とする。
 - 隣接便ごとの出発時刻差を算出し、近接度が高い便を優先し1本ずつ間引く。近接度が同等の場合は、早い時刻の候補を優先する。
 - 始発・最終便固定により、路線の便数が2本以下のときは減便できない。制約により減便不可能な場合は警告する。
 - （例）既存便：07:00、07:20、07:32、08:10 から1便減らす
 - ・出発時刻差=[20、12、32]
 - ・候補A：07:20（前後の出発時刻差=20・12→合計32）
 - ・候補B：07:32（前後の出発時刻差=12・38→合計50）
 - ・近接度：A=1/32、B=1/50→Aを間引く
- ◇ GTFS反映は、以下のとおり。
 - 挿入：trips.txtに新規trip_idを採番し、stop_times.txtに平行移動後の停車時刻列を追加する。service_id等は既存を継承する。
 - 削除：対象trip_idをtrips.txt、stop_times.txtから削除する。
- 路線の変更（route）
 - ◇ 路線の新規追加、変更、削除する。
 - ◇ 「路線」とは、routes.txtのroute_idを単位とする。agency_id、route_short_name、route_long_name等の属性と、当該路線にひも付くtrips、stop_times、shapesを扱う。
 - ◇ 1路線に複数のdirection_id、service_idがある場合、route_id×direction_id×service_idを運行パターンとして扱う。
 - ◇ 路線の新規追加
 - route_idを一意に採番。agency_id、route_short_name、route_long_nameを入力する。
 - ◇ 路線の変更
 - route_short_name/route_long_name等の属性はroutes.txtの更新で完結する（tripsへは波及しない）。
 - 経路形状が変わる場合は、shape_idを新規採番して再作成する。新規区間は、OpenStreetMapを使って2地点間のルート検索を行い経路を推測し、shape.txtへ反映する。
 - ◇ 路線の削除
 - 削除対象のroute_idにひも付くtripsが存在しないことを確認の上、削除。存在する場合は警告を表示する。
 - ◇ GTFS反映は、以下のとおり。
 - routes.txt：追加/更新/削除する。
 - trips.txt：新設/更新/削除する（route_id変更時の一括更新を含む）。
 - stop_times.txt：停留所系列や所要時間変更に伴う再計算結果を反映する。
 - shapes.txt：経路変更時は新規shape_idを採番し差分管理する。
- 便の変更（trip）
 - ◇ 「便」はtrips.txtのtrip_idを単位とする。対応するstop_times.txtの時刻列を扱う。

- ◇ 便の新規追加
 - trip_id を一意に採番。route_id、service_id、direction_id を入力する。
 - stop_times を追加し、stop_id、stop_sequence で経路を定義し、各停留所の arrival_time / departure_time を入力する。
- ◇ 便の変更
 - 時刻変更は、始発便 arrival の入力値 (出発基準時刻との差分) を基準に、各停留所の arrival / departure に差分値を加算する。
 - 経路を編集 (経由する停留所を追加) する場合は、stop_times に、追加する停留所の stop_id、stop_sequence を定義し、各停留所の arrival_time / departure_time を個別に入力する。
- ◇ 便の削除
 - 削除対象の trip_id を trips.txt、stop_times.txt から削除する。
- ◇ GTFS 反映は、以下のとおり。
 - trips.txt : 追加 / 更新 / 削除する。
 - stop_times.txt : 便ごとの停車時刻列を追加 / 更新 / 削除する。
- 標柱の変更 (stop)
 - ◇ 「標柱」は stops.txt の stop_id を単位とし、stop_name、stop_lat / stop_lon などの属性、及び各便の stop_times 参照を扱う。
 - ◇ 標柱の新規追加
 - stop_id を一意に採番する。
 - stop_lat / stop_lon、stop_name を入力する。
 - ◇ 標柱の変更
 - 名称・属性変更は、stop_name、stop_lat / stop_lon を更新する (stop_times へは波及しない)。
 - ◇ 標柱の削除
 - 削除対象の stop_id を参照する stop_times が存在しないことを確認の上、削除。存在する場合は警告し、先に便側から当該停留所を外す。
 - ◇ GTFS 反映は、以下のとおり。
 - stops.txt : 追加 / 更新 / 削除する。
 - stop_times.txt : 標柱の新規追加・削除に伴って追加 / 削除する。
 - shapes.txt : 必要に応じ頂点列の補正 / 新規 shape_id 採番する。
 - 路線全体の経路を変更する場合は「路線の変更」、便個別は「便の変更」を行う。
- GTFS データの出力
 - ◇ 上記の編集結果を、GTFS 形式で出力する。

【AL105】運行記録 (日報) の作成

- 本アルゴリズムの概要
 - 運行記録 (日報) 帳票作成ツールである「unko-kiroku-nippou.xlsm」は、GTFS に基づく運行データを取り込み、乗降集計データの運行記録 (日報) 手入力シート及び運行記録 (日報) 登録シートを作成するとともに、登録された実績から乗降集計データの CSV 出力を行うことが可能な Excel マクロツールである
 - GTFS を入力して帳票を生成するため、最初に GTFS を展開し、必要ファイルを Excel シートへ取

り込む共通処理を有する

- 現場の運転手が乗降実績を記録するための運行記録（日報）手入力シートを作成する
 - ◇ ユーザーが指定する日付の特定路線（または全路線）の停留所時刻表を抽出し、日々の運行記録として印刷可能な帳票を生成する
 - ◇ ユーザーが指定する日付を基準に、運行する便（service_id）を確定し、路線選択に応じて stop_times を抽出、帳票として整形・印刷設定まで実施する
- 自治体職員等が現場の紙の運行記録（日報）を基に乗降実績を転記するための運行記録（日報）登録シートを作成する
 - ◇ ユーザーが指定する年月について、運行日×service_id×stop_times の組み合わせを展開した路線別のシートを生成し、入力欄（旅客種別ごとの乗降実績）と自動集計式を配置する
- 運行記録（日報）登録シートに転記された情報から乗降集計データを CSV 形式で出力する

- 本アルゴリズムを利用した機能

- 【FN015】 運行記録（日報）出力機能

- 本アルゴリズムの計算・処理の詳細

- 共通処理

- ◇ ZIP ファイル選択と展開

- GTFS ファイルを選択するダイアログを表示し、利用者に GTFS ファイルを選択させる
- VBA 内で解凍ロジックを持たず、OS 標準機能を使うことで依存を減らし、運用環境差を抑えるため、PowerShell により ZIP を展開する
- ZIP 内の階層構造が一定でない（1 階層深い）ケースへの耐性を確保するため、展開先フォルダー配下を再帰探索し、agency.txt 等の GTFS ファイルが存在するフォルダーを特定する

- ◇ 既存 GTFS シートの初期化

- 前回データの残存による混在や誤判定を防止するため、新規取り込み前に既存の GTFS 関連シートをクリアする

- ◇ テキスト読み込みと貼り付け

- 各 GTFS ファイルを読み込む
- 事業者や生成元により文字コードが揺れるケースでの取り込み失敗を減らすため、文字コードは UTF-8 を第一候補とし、失敗時には Shift-JIS で再試行する
- stop_times 等の大規模データでも処理時間を抑えるため、読み込んだ内容を CSV として分割し、必要に応じてダブルクォーテーションを除去する

- ◇ Excel への転記は、セル逐次書込みではなく配列一括貼り付けで行う

- 運行記録（日報）手入力シートの作成

- ◇ 運行日チェックと初期化

- ユーザーが「データ取り込みと作成」シートの日付欄に入力した値を日付として検証し、不正ならエラー表示し終了する
- 運行記録（日報）手入力シートの出力領域をクリアし、前回結果の残存を防ぐ

- ◇ stop_times 前処理（参照列付与とソート）

- stop_times に補助列を付与する

- J列に処理・並べ替え後の追跡用に連番を設定する
- K列に運行日判定・抽出で使用するために service_id を設定する
- K列 (service_id) は、trips から trip_id → service_id を Dictionary で引き、stop_times へ転記する
- stop_times を「service 内で trip がまとまり、trip 内で停留所順が整う」順にソートする
(例：K列降順→A列昇順→E列昇順)
- ◇ 運行タイプ (有効 service_id) の確定
 - ユーザーが「データ取り込みと作成」シートの日付欄に入力した日付から曜日と yyyyymmdd 形式を生成する
 - calendar から「該当曜日=1 かつ 期間内」の service_id を抽出する
 - calendar_dates で例外処理を適用する (exception_type=2:除外、exception_type=1:追加である)
 - 最終的に有効な service_id 集合を G12 へカンマ区切りで出力する
- ◇ 路線のプルダウンの生成
 - 有効 service_id から対象 route_id を抽出する (trips を参照し、Dictionary で重複排除する)
 - routes を参照し、表示用の路線名 (例：route_id+路線名) を生成する
 - 作業用シートに路線候補を配置し、路線部分にデータ検証 (プルダウン) として設定する
- ◇ データ抽出 (路線条件を反映)
 - 路線のプルダウンが空欄なら全路線対象、入力があれば該当 route の trip_id のみ対象とする
 - trips から対象 trip_id を取得する
 - stop_times から対象 trip の行を抽出し、「日報 (手入力用)」シートへ出力する
- ◇ 表示情報の付加
 - stops から stop_id→stop_name を Dictionary 化し、D列へ付加する
 - 便の識別に必要な属性として、trip_id+route_id→service_id 等を E列へ付加する
- ◇ 書式の整形とグルーピング
 - 連番を付与し、時刻列 (例：B列) を h:mm 形式に設定する
 - 便のまとまりが崩れないように並べ替え (例：F列昇順→A列昇順) を行う
 - 便単位のページ化のため、グループ番号 (trip_id 単位)、グループ内連番を付与する。グループ番号=1 であることが「便の先頭行」を示すトリガーとなる
- ◇ ヘッダー挿入 (帳票の基本枠)
 - 先頭に 12 行挿入し、タイトル/事業者/運行日/路線情報等のヘッダーを配置する
 - 旅客種別カテゴリーを 12~13 行目に配置し、セル色・罫線を設定する
- ◇ グループ別ヘッダー (便ごとに独立ページ化)
 - K列=1 の行の直前に 13 行挿入し、先頭ヘッダー (1~13 行目) をコピーして貼り付ける
 - これにより trip_id (便) 単位で独立したページを生成する
- ◇ 上下区分 (direction_id) の反映と後処理
 - trip_id/service_id をヘッダー行へ転記する (便の識別情報として残す)
 - trips の direction_id を取得し、上下区分 (0=復路、1=往路) としてヘッダーに表示する
 - 帳票に不要な作業列 (例：E~K) をクリアする

◇ 印刷設定

- 印刷範囲を設定し、帳票の区切り（例：「運行記録（日報）」行前）に改ページを挿入する
- A4 縦、横幅 1 ページ、余白調整等を行い、印刷結果の再現性を確保する

➤ 運行記録（日報）登録シートの作成

◇ 年月プルダウンの生成

- calendar の start_date～end_date から運行期間の最小・最大日付を取得する
- 期間内の全年月（yyyy 年 mm 月）を Dictionary で収集し重複排除する
- 「月次作成」シートの月次入力シート作成年月にプルダウンとして設定する

◇ 月次 GTFS のリスト作成（中間リスト生成）

- 選択年月の全日付（1 日～末日）を生成する
- 祝日リストを Dictionary に読み込み、曜日と合わせて平休区分を判定できるようにする
- 日付ごとに、有効 service_id 集合を確定する
- Calendar で曜日条件と期間で判定する
- calendar_dates で例外処理を適用する（exception_type=2：除外、exception_type=1：追加である）
- stop_times 全行を配列で読み込み、該当 service_id×日付のレコードを展開する
- 各レコードへ、帳票に必要な属性である日付・曜日・平休区分・service_id・trip_id・route_id
- stop_id・stop_sequence・departure_time・stop_name を付与する
- route_id→日付→trip_id→stop_sequence の順にソートし、A20 行目以降に出力する

◇ 旅客種別名の取得と配置

- ユーザーが「データ取り込みと作成」シートの種別 1～3 のセルに入力した値から旅客種別カテゴリを取得する（最大 3 種）
- 種別が全て空欄の場合も列構造を保つため、空欄 1 つとして扱う

◇ route_id 別シート作成（テンプレート複製）

- 中間リストの route_id 列の一意値を Dictionary で収集し昇順ソートする
- route_id ごとにテンプレート Monthly_template をコピーし、「月次_{route_id}」シートを作成する
- 中間リストから該当 route_id の行を抽出し、入力に適した列配置へ再構成して出力する（A 列：stop_sequence、B 列：departure_time (h:mm)、C 列：stop_id、D 列：stop_name、E 列～：種別入力欄、M 列以降：日付/曜日/平休/route_id/service_id/trip_id)
- テンプレートは誤編集防止のため保護し、xlSheetVeryHidden で非表示にする

◇ ヘッダーの作成

- 「月次_{route_id}」シートの 13 行目にヘッダーを作成する
- ユーザーが「データ取り込みと作成」シートの種別 1～3 のセルに入力した種別名を 2 セット横方向に配置する（乗車×種別数+合計、降車×種別数+合計）
- 12 行目に「通過」列を追加する

◇ ヘッダー情報取得（事業者・路線名・日付リスト）

- Agency シートから事業者名を取得し、「月次_{route_id}」シートの D4 セル事業者名へ転記する
- 各シートの route_id を取得し D5 セル路線 ID へ転記する
- routes から route_long_name を検索し D6 セル路線名へ転記する

- 絞り込み用のリストとして、「日付」列の一意値を収集・昇順ソートし、trip_id 列の右隣へ縦方向に出力する
- ◇ 合計計算・通過計算の設定
 - ユーザーが「データ取り込みと作成」シートの種別 1~3 のセルに入力した種別数を取得し typeCount を算出する
 - typeCount を用いて、乗車合計列／降車合計列／通過列の位置を動的に決定する
 - 14 行目に数式を設定し、対象範囲へ一括適用する
 - 通過列は行ごとの累計とし、A 列=1 で累計をリセットする
- ◇ direction_id 反映を取得して上下区分を更新
 - trips から Dictionary (キー: route_id/service_id/trip_id) を作成する
 - direction_id を表示用に変換 (0→「復路」、1→「往路」等) し、上下区分を更新する
 - 月次作成シートの作業領域 (例: D1~H1) をクリアし、必要に応じて 20 行目以降の中間データを削除する
- ◇ 月次乗車数サマリーシートの作成
 - 新規シート「月次乗車数サマリー」を作成する
 - 「月次_{route_id}」シートの所定範囲を集計する数式を設定する
 - 月末日から「1 日当たり利用者数」を算出する
 - 各シートから RouteName と集計値をリンク貼り付けして、一覧表を作成する
- ◇ 日付・便絞り込み機能
 - 日付選択フォーム (UserForm1) では、「日付」列のユニーク値を ListBox に表示し、選択日でオートフィルタを適用する
 - Trip_id 選択フォーム (UserForm2) でフィルタ後の表示行から trip_id を ListBox に表示し、選択 trip_id で追加フィルタを適用する
- 乗降集計データの CSV 出力
 - ◇ 抽出と正規化
 - ExportRideDataToSheet で、「月次_{route_id}」シートから、抽出条件を満たすデータの必要項目を所定の列順に並べ替えて「乗降 CSV」シートへ出力する
 - 抽出条件は、A 列が数値であり「乗車合計 \geq 1 または降車合計 \geq 1)」であるものとする
 - 出力項目は、date、agency_id、route_id、trip_id、stop_id、stop_sequence、count_geton、count_getoff とする
 - ◇ CSV ファイルの出力
 - ExportCSVFile により、ADODB.Stream を用いて、UTF-8 (BOM なし) で LF 改行の CSV 形式ファイルを生成する
 - ファイル名は「事業者名+"_boarding_alighting_records.csv"」とする

【AL106】路線と停留所によるグループ化

- 本アルゴリズムの概要
 - 同じ停留所名の停留所、路線を、それぞれ 1 つのグループにまとめる。停留所をまとめることにより、「東京駅」という名前を持つ停留所レコードが 10 個あっても、それらを全て「東京駅グループ」として統一的に管理できる。路線も同様である。

- 本アルゴリズムを利用した機能
 - 【FN001】 GFTS データ (API) 入力機能、【FN010】 GTFS データ入力機能
- 本アルゴリズムの計算・処理の詳細
 - 停留所名によるグループ化
 - ◇ 停留所 ID (stop_id) が異なっても同じ停留所名 (stop_name) を持つ全ての停留所を、1つのグループとしてまとめる。
 - ◇ 停留所名によるグループ化の処理フローを以下に示す。



図 2-38 停留所名によるグループ化

- 停留所 ID によるグループ化
 - ◇ 停留所 ID (stop_id) の命名パターンを分析し、物理的に近い停留所をグループ化する。「1_01」「1_02」のようにアンダースコアの前が同じ ID の場合は、同じ場所や近接した場所にあることを示す。
 - ◇ 停留所 ID によるグループ化の処理フローを以下に示す

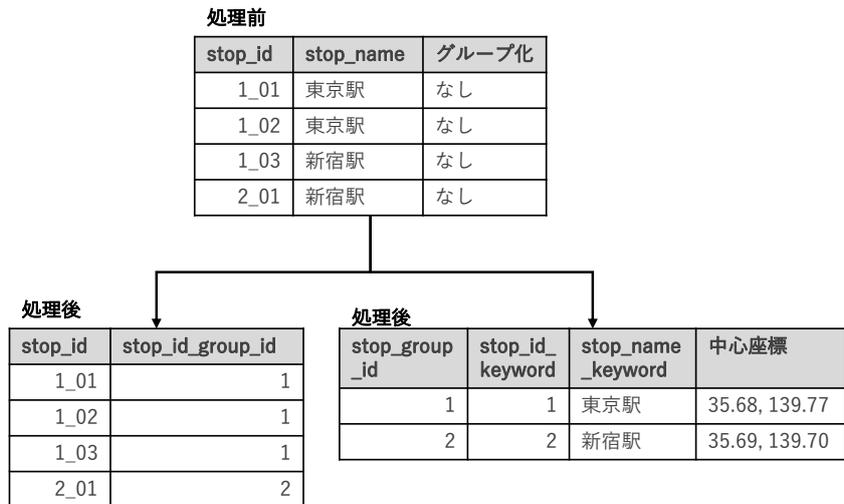


図 2-39 停留所 ID によるグループ化

- ルートキーワード抽出と分類
 - ◇ 路線名から最も重要なキーワードを抽出し、路線を分類する。例えば、「○△線(400_3_1)」と「○△線(400_3_2)」は、route_id は異なるが同じ路線名を持つため、「○△線」というキーワードでグループ化する。
 - ◇ route_id が異なる「○△線 400_3_1」と「○△線 400_3_2」でも、同じ路線名「○△線」を持つため、同じキーワードグループにまとめる。地図上では同じ色 (FF6B6B) で表示する。
 - ◇ ルートキーワード抽出と分類の処理フローを以下に示す。

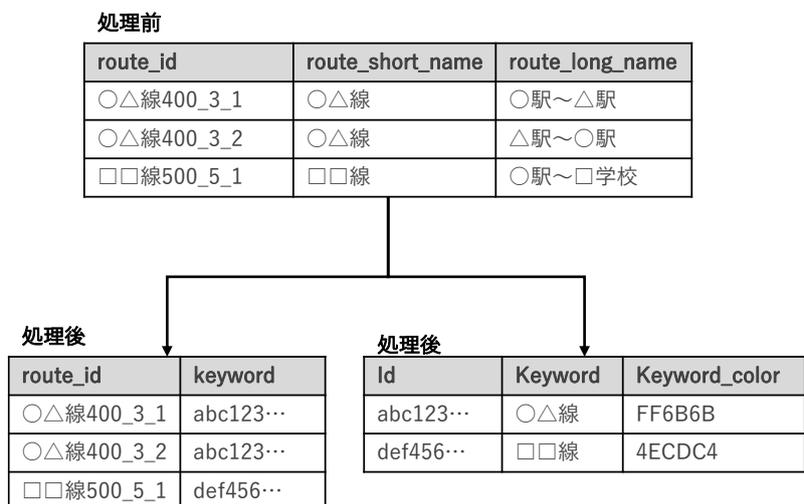


図 2-40 ルートキーワード抽出と分類

2-2. システムコンポーネント (CO)

2-2-1. システムコンポーネント図

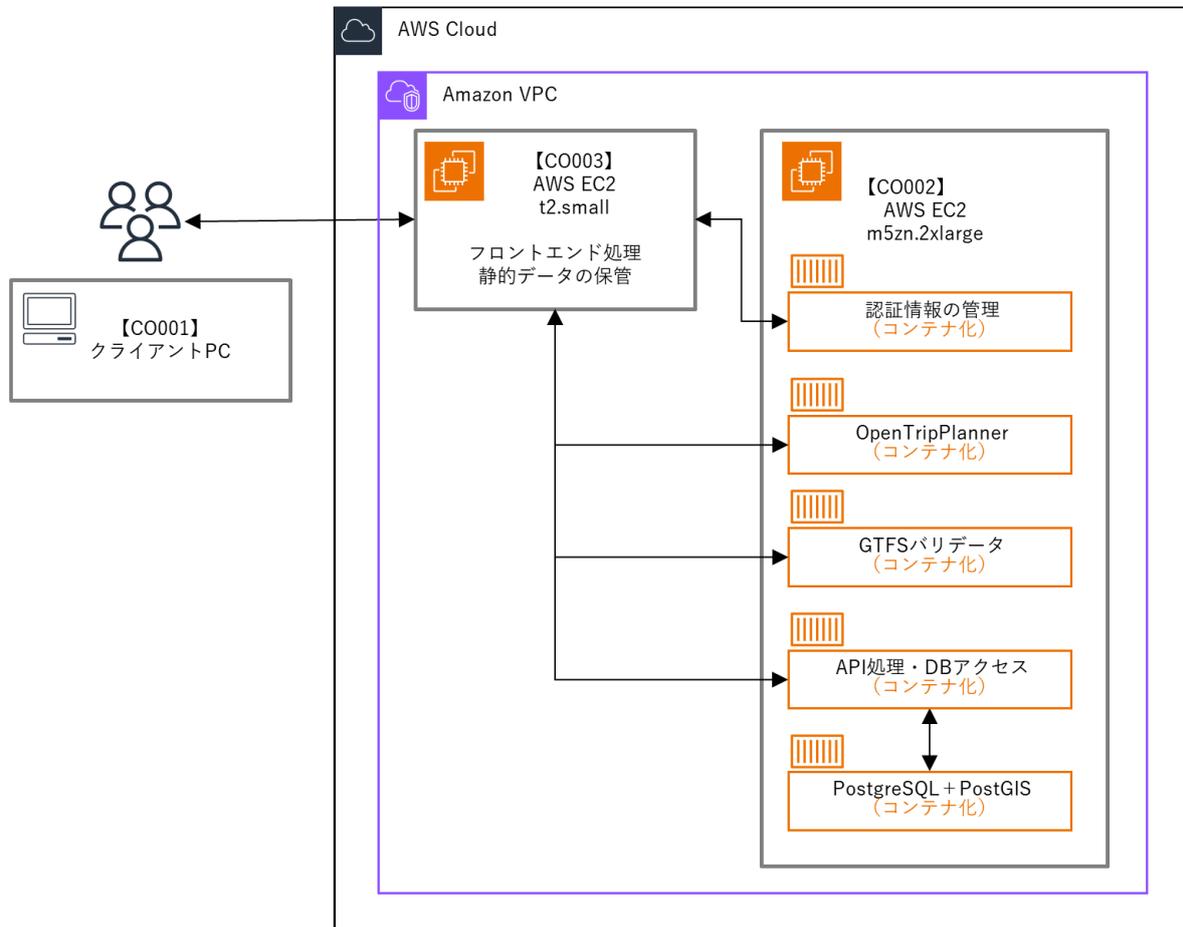


図 2-41 システムコンポーネント図

2-2-2. システムコンポーネント一覧

表 2-12 システムコンポーネント一覧

ID	種別	コンポーネント名	用途
CO001	クライアント PC	クライアント PC	<ul style="list-style-type: none"> ● アプリを操作するためのクライアント PC
CO002	認証情報の管理 API 処理・DB アクセス PostgreSQL PostGIS 用サーバー	AWS EC2 m5zn.2xlarge	<ul style="list-style-type: none"> ● データベースのユーザー名やパスワード等を扱うために使用する ● GTFS の API やバックエンドの DB へのアクセスを行うためのサーバー ● データベースである PostgreSQL とその拡張モジュールである PostGIS をインストールするためのサーバー
CO003	フロントエンド	AWS EC2 t2.small	<ul style="list-style-type: none"> ● Web ブラウザ上で動作するフロントエンドアプリケーション。処理の実行や結果の表示を行うアプリケーション

2-3. ハードウェア (HW)

2-3-1. ハードウェアアーキテクチャ

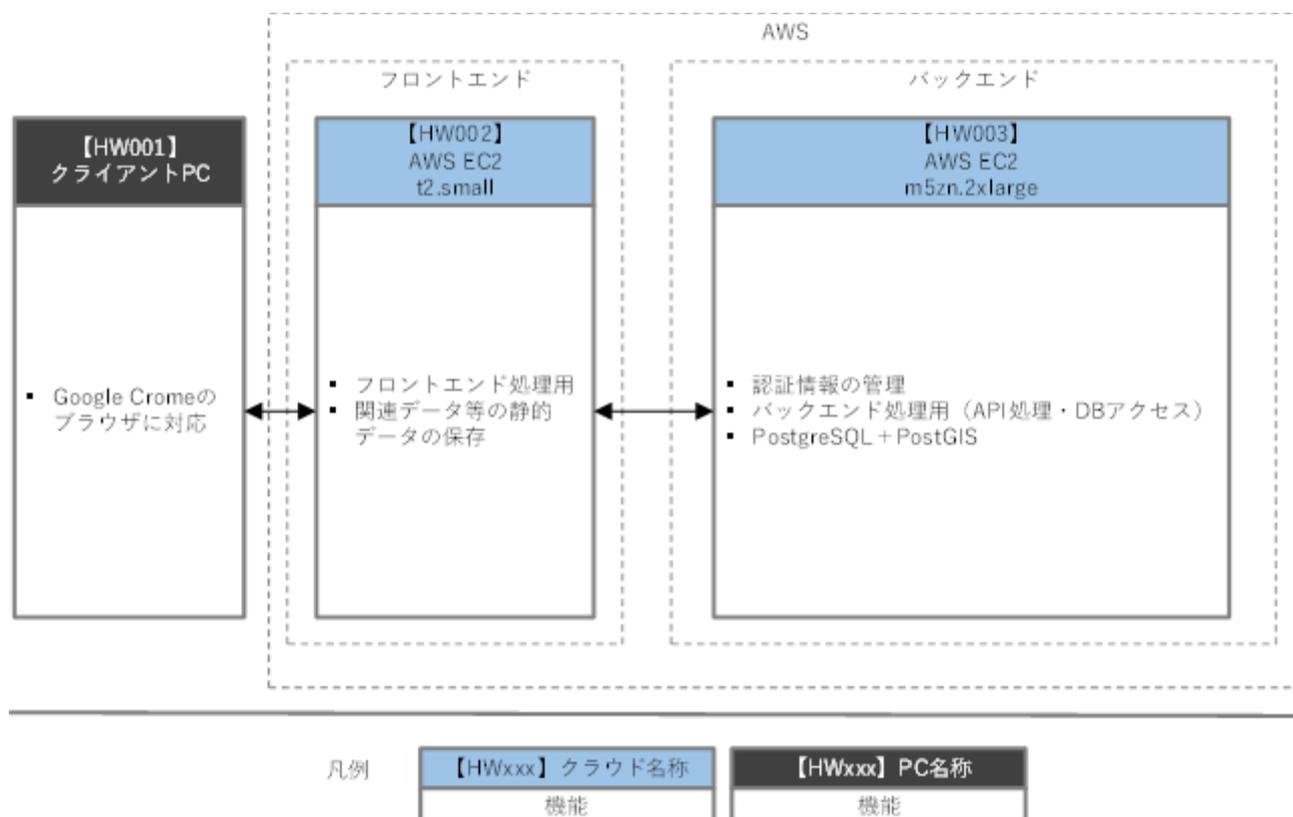


図 2-42 ハードウェアアーキテクチャ

2-3-2. ハードウェア一覧

表 2-13 ハードウェア一覧

※朱文字：新規開発・既存改修

ID	種別	ベンダー	品番	用途
HW001	PC	Lenovo	21KDS3SE00	● Google Chrome のブラウザに対応

2-3-3. ハードウェアの詳細

【HW001】 PC : Lenovo ThinkPad X1 Carbon Gen 12

- 本ハードウェアの概要
 - Google Chrome のブラウザに対応
- ベンダー
 - Lenovo
- 仕様・スペック
 - CPU : インテル® Core™ Ultra 7 vPro® Enterprise プロセッサー 165U
 - メモリ : 16GB
 - OS : Windows 11
- イメージ

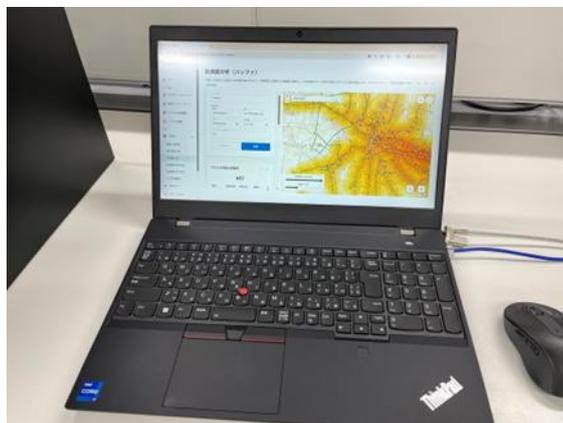


図 2-43 ThinkPad

2-4. データインターフェース (IF)

2-4-1. データアーキテクチャ

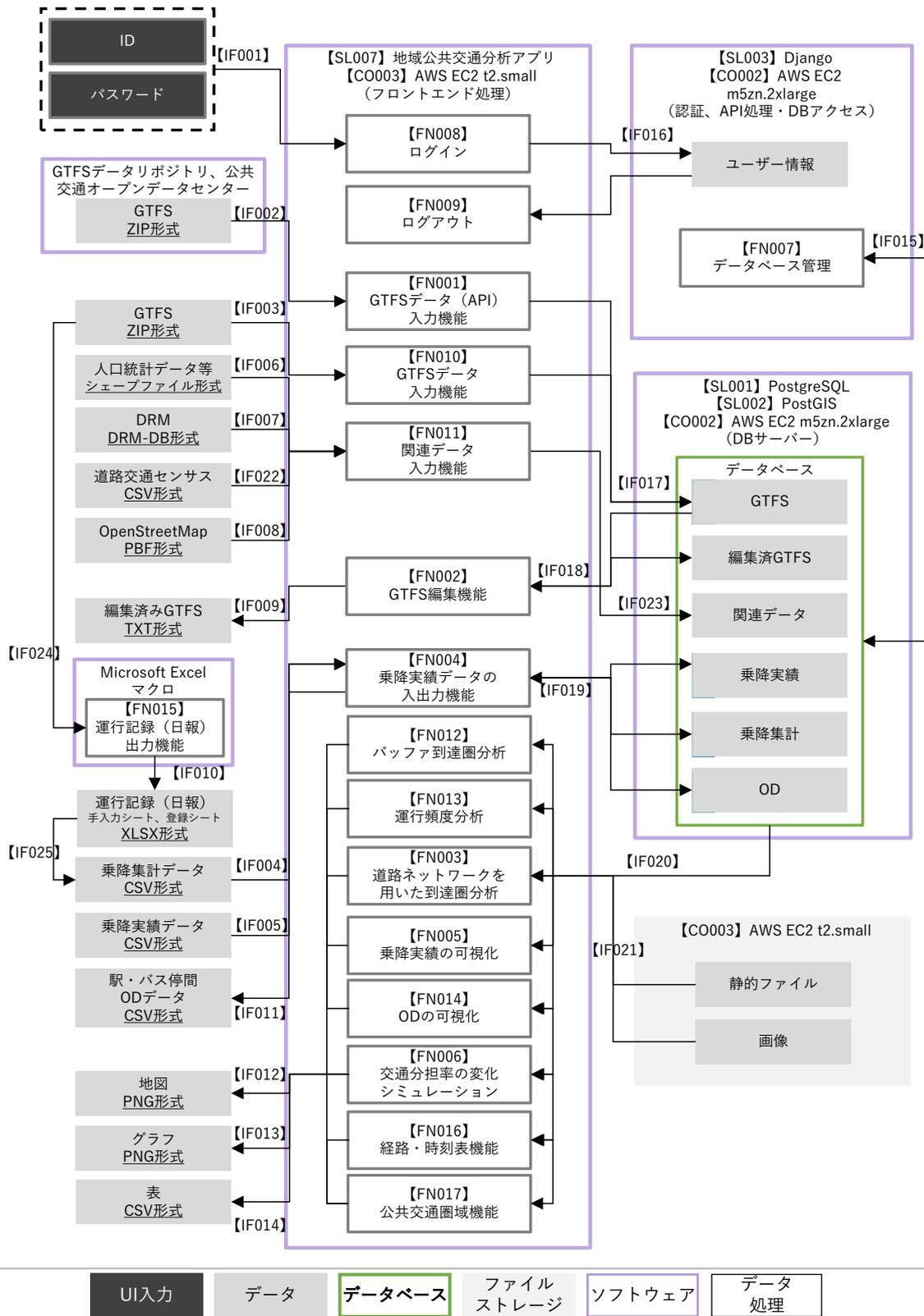


図 2-44 データアーキテクチャ

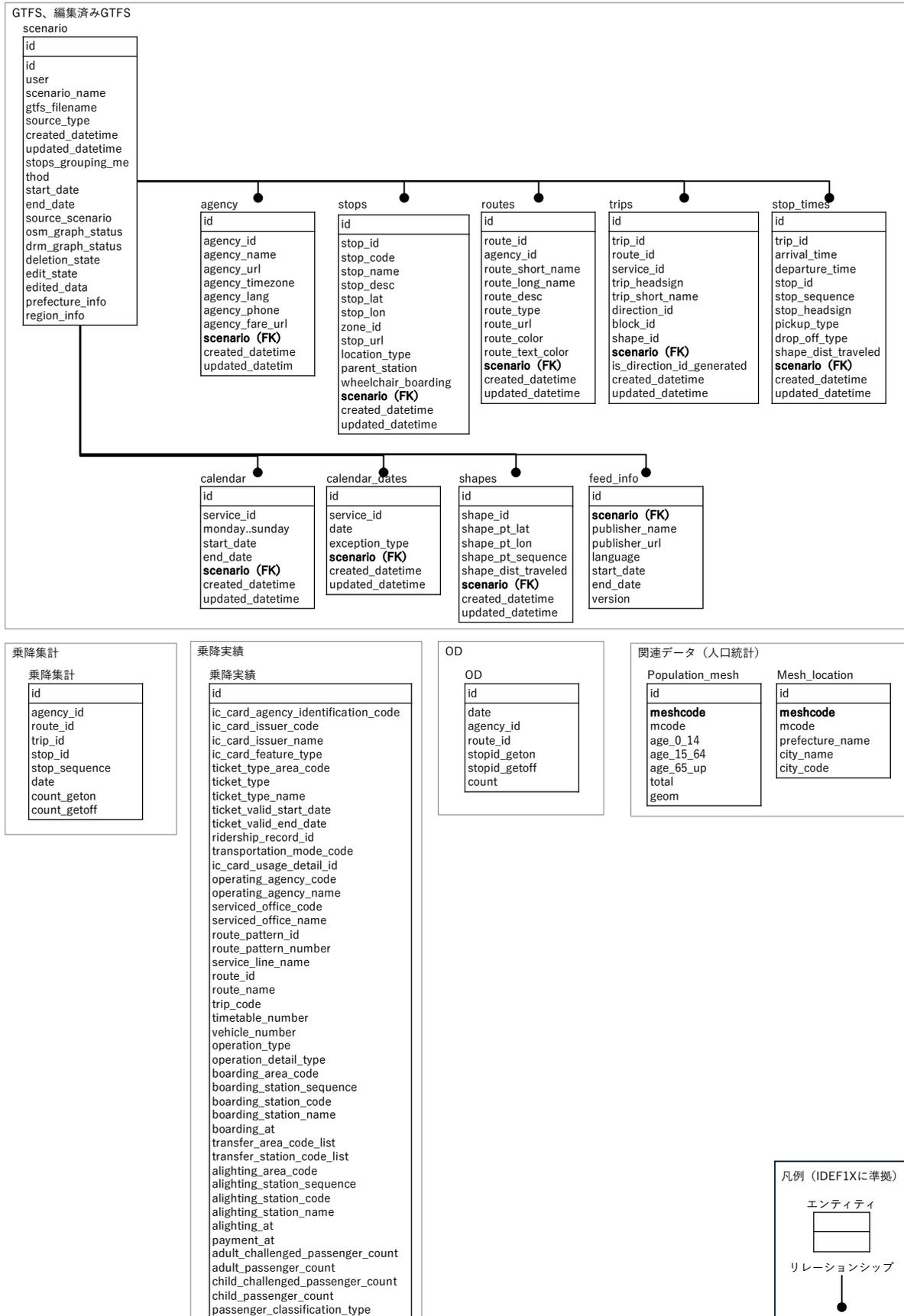


図 2-45 ER 図

2-4-2. データインターフェース一覧

表 2-14 データインターフェース一覧

※朱文字：新規開発・既存改修

ID	名称	出力側 ID	入力側 ID
IF001	ログイン IF	FN008	—
IF002	GTFS 入力 IF (API)	FN001	—
IF003	GTFS 入力 IF (ローカル)	FN010	—
IF004	乗降集計データ入力 IF	FN004	—
IF005	乗降実績データ入力 IF	FN004	—
IF006	人口統計入力 IF	FN011	—
IF007	DRM 入力 IF	FN011	—
IF008	OpenStreetMap 入力 IF	FN011	—
IF009	編集済み GTFS 出力 IF	—	FN002
IF010	運行記録 (日報) 帳票作成ツール出力 IF	—	FN015
IF011	駅・バス停間 OD データ出力 IF	—	FN004
IF012	地図出力 IF	—	FN003、FN005、 FN006、FN012、 FN013、FN014、 FN016、FN017
IF013	グラフ出力 IF	—	FN003、FN005、 FN006、FN012、 FN013、FN014、 FN016、FN017
IF014	表のデータ出力 IF	—	FN003、FN005、 FN006、FN012、 FN013、FN014、 N016、FN017
IF015	データベース入出力 IF	FN007	SL001、SL002
IF016	ユーザー情報 IF	FN008	CO006
IF017	GTFS データ入力 IF	SL001、SL002	FN001、FN010
IF018	GTFS DB 入出力 IF	SL001、SL002、FN002、FN004	
IF019	乗降実績 DB 入出力 IF	SL001、SL002、FN004	
IF020	可視化用 DB 出力 IF	FN003、FN005、 FN006、FN012、 FN013、FN014、 N016、FN017	SL001、SL002

IF021	静的データ IF	FN003、FN005、 FN006、FN012、 FN013、FN014、 N016、FN017	SL001、SL002
IF022	道路交通センサ入力 IF	FN001	—
IF023	関連データ入力 IF	SL001、SL002	NF011
IF024	運行記録（日報）帳票作成ツールへの GTFS 入力 IF	—	FN015
IF025	乗降集計データ出力 IF	—	—
IF026	施設データ入力 IF	FN001	—

2-4-3. データインターフェースの詳細

データインターフェースの詳細を記す。なお、本業務において開発（新規・改修）を行うデータインターフェースを**朱文字**で示す。

【IF001】 ログイン IF

- 本データインターフェースの概要
 - ログインに必要な ID・パスワードを受け取るためのデータインターフェース
- 本インターフェースを利用する機能
 - **【FN008】 ログイン**
- データ詳細
 - ID は、システム側で設定
 - パスワードは、ユーザー側で設定した値。英大文字小文字、数字、記号を利用可能とする。ハッシュ化して取り扱う。

【IF002】 GTFS 入力 IF (API)

- 本データインターフェースの概要
 - 外部サイト（GTFS データリポジトリ）が公開している GTFS データを入力するためのデータインターフェース
- 本インターフェースを利用する機能
 - **【FN001】 GTFS データ (API) 入力機能**
- 共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ POST
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - 共通リクエストパラメータ
 - ◇ なし
 - 共通レスポンスパラメータ

名称	説明
code	コード
message	メッセージ
body	実データ

➤ レスポンスデータ

```
{
  "code": 200,
  "message": "ok",
  "body": {
    // 各 API の返答結果
  }
}
```

● GTFS フィード一覧を取得

➤ 概要

- ◇ フィードの都道府県コードで検索し、GTFS フィード一覧を取得

➤ URL

- ◇ <https://api.gtfs-data.jp/v2/feeds>
- ◇ <https://api.gtfs-data.jp/v2/files>

➤ 使用例

- ◇ <https://api.gtfs-data.jp/v2/feeds?pref=16>

➤ リクエストパラメータ

名称	説明	値	必須
pref	都道府県コード	任意の文字列	-

➤ レスポンスパラメータ

名称	説明
organization_id	事業者 ID
organization_name	事業者名
organization_pref_id	事業者所在地都道府県コード
organization_web_url	事業者 Web ページ
organization_email	事業者 Email
feed_id	フィード ID
feed_name	フィード名
feed_pref_id	都道府県
feed_license	ライセンス
feed_license_url	ライセンス URL
feed_memo	フィード情報備考
feed_src_gtfs_current_url	静的情報：ダウンロード URL（現行）
feed_src_gtfs_next_url	静的情報：ダウンロード URL（次か現行（Google 用））
feed_src_gtfs_web_url	静的情報：ダウンロード URL（現行->過去->次）
real_time	リアルタイム情報

trip_update_url	リアルタイム情報：TripUpdate URL
vehicle_position_url	リアルタイム情報：VehiclePosition URL
alert_url	リアルタイム情報：Alert URL
update_interval	リアルタイム情報：更新間隔
last_updated_at	最終アップデート日
last_published_at	最終公開日
latest_feed_start_date	最新 GTFS 開始日
latest_feed_end_date	最新 GTFS 終了日
feed_is_discontinued	フィード廃止
feed_discontinued_date	フィード廃止日

➤ レスポンスサンプル

```
{
  "code": 200,
  "message": "ok",
  "body": [
    {
      "organization_id": "test-org1",
      "organization_name": "組織 1",
      "organization_pref_id": 1,
      "organization_web_url": "https://gtfs-data.jp",
      "organization_email": "naeae.myouji@aigid.jp",
      "feed_id": "rosenbus",
      "feed_name": "路線バス",
      "feed_pref_id": 23,
      "feed_license": "CC0 1.0",
      "feed_license_url": "https://creativecommons.org/publicdomain/zero/1.0/deed.ja",
      "feed_memo": "市内循環バスで夏のみ運行",
      "feed_src_gtfs_current_url": "https://...",
      "feed_src_gtfs_next_url": "https://...",
      "feed_src_gtfs_web_url": "https://...",
      "real_time": {
        "trip_update_url": "https://...",
        "vehicle_position_url": "https://...",
        "alert_url": "https://...",
        "update_interval": 10
      }
    },
    "last_updated_at": "2023-02-01",
  ]
}
```

```

        "last_published_at": "2023-02-01",
        "latest_feed_start_date": "2023-02-01",
        "latest_feed_end_date": "2023-12-31",
        "feed_is_discontinued": false,
        "feed_discontinued_date": ""
    }
]
}

```

【IF003】 GTFS 入力 IF (ローカル)

- 本データインターフェースの概要
 - ユーザーがローカル環境に保有している GTFS データを入力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN010】 GTFS データ入力機能
- データ詳細
 - GTFS 形式に準拠したデータを、ローカル環境からアップロードし入力する。
 - 利用するデータは、【IF002】 GTFS 入力 IF (API) と同じ。
 - 以下の技術資料を参照
 - ◇ https://www.mlit.go.jp/sogoseisaku/transport/sosei_transport_tk_000112.html

【IF004】 乗降集計データ入力 IF

- 本データインターフェースの概要
 - 「どの便で」「どの停留所で」「何人の人が乗降したか」が記録されているデータを入力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN004】 乗降実績データの入出力機能
- データ詳細

表 2-15 乗降集計データ属性情報

No.	属性名	説明	データ型
1	date	日付	YYYYMMDD 形式
2	agency_id	事業者 ID	文字
3	route_id	路線 ID	文字
4	trip_id	便 ID	文字
5	stop_id	停留所 ID	文字
6	stop_sequence	通過順位	数値
7	count_geton	乗車人数	数値
8	count_getoff	降車人数	数値

【IF005】乗降実績データ入力 IF

- 本データインターフェースの概要
 - ICカードの利用実績を一件ごとに記録したデータを入力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN004】乗降実績データの入出力機能
- データ詳細
 - 乗降実績データの乗降実績データ標準仕様書（鉄道・バス）標準仕様に準拠した形式

表 2-16 乗降実績データ属性情報

No.	属性名	説明	データ型
1	ic_card_agency_identification_code	ICカード識別コード	文字
2	ic_card_issuer_code	ICカード発行事業者コード	文字
3	ic_card_issuer_name	ICカード発行事業者名	文字
4	ic_card_feature_type	ICカード機能区分	Enum
5	ticket_type_area_code	券種エリアコード	文字
6	ticket_type	券種区分	Enum
7	ticket_type_name	券種名	文字
8	ticket_valid_start_date	券有効開始日	日付
9	ticket_valid_end_date	券有効終了日	日付
10	ridership_record_id	乗降実績 ID	数値
11	transportation_mode_code	交通モードコード	文字
12	ic_card_usage_detail_id	ICカード利用明細 ID	値
13	operating_agency_code	運行事業者コード	文字
14	operating_agency_name	運行事業者名	文字
15	serviced_office_code	営業所コード	文字
16	serviced_office_name	営業所名	文字
17	route_pattern_id	系統 ID	文字
18	route_pattern_number	系統番号	文字
19	service_line_name	路線名	文字
20	route_id	経路 ID	文字
21	route_name	経路名	文字
22	trip_code	便コード	文字
23	timetable_number	ダイヤ番号	文字
24	vehicle_number	車両番号	文字
25	operation_type	処理区分	Enum
26	operation_detail_type	処理詳細区分	Enum
27	boarding_area_code	乗車エリアコード	文字
28	boarding_station_sequence	乗車停留所連番	数値

29	boarding_station_code	乗車駅（停留所）コード	文字
30	boarding_station_name	乗車駅（停留所）名	文字
31	boarding_at	乗車日時	日付・時刻
32	transfer_area_code_list	乗継エリアコードリスト	文字（配列）
33	transfer_station_code_list	乗継駅（停留所）コードリスト	文字（配列）
34	alighting_area_code	降車エリアコード	文字
35	alighting_station_sequence	降車停留所連番	数値
36	alighting_station_code	降車駅（停留所）コード	文字
37	alighting_station_name	降車駅（停留所）名	文字
38	alighting_at	降車日時	日付・時刻
39	payment_at	精算日時	日付・時刻
40	adult_challenged_passenger_count	大人障がい者利用者数	数値
41	adult_passenger_count	大人利用者数	数値
42	child_challenged_passenger_count	小児障がい者利用者数	数値
43	child_passenger_count	小児利用者数	数値
44	passenger_classification_type	利用者分類区分	Enum

【IF006】人口統計入力 IF

- 本データインターフェースの概要
 - e-Stat の「小地域年齢（5 階級）別男女別人口」データを入力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN011】関連データ入力機能
- データ詳細

表 2-17 人口統計属性情報

No.	属性名	説明	データ型/長
1	id	連番	bigint
2	year	統計年度	integer
3	key_code	図形と集計データのリンクコード	character varying(20)
4	hyosyo	地域階層レベル	integer
5	cityname	市区町村名	character varying(50)
6	name	町丁・字等名	character varying(50)
7	t001082001	総数、年齢「不詳」含む	integer
8	t001082017	総数 15 歳未満	integer
9	t001082018	総数 15～64 歳	integer
10	t001082019	総数 65 歳以上	integer

No.	属性名	説明	データ型/長
11	t001082020	総数 75 歳以上	integer

【IF007】 DRM 入力 IF

- 本データインターフェースの概要
 - デジタル道路地図の基本道路網に含まれる路線・交通センサス番号等のデータを入力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN011】 関連データ入力機能
- データ詳細
 - DRM-DB の基本道路リンクデータを使用。道路交通センサスとひも付けを行う番号交通センサス区間番号を使用。
DRM-DB は A 版（現標準）日本測地系、文字コード「SJIS」を使用。

表 2-18 DRM データ使用する属性情報

No.	属性名	説明	データ型
1	交通センサス区間番号	道路交通センサスの「交通調査基本区間番号」とひも付けるための番号	数値

【IF008】 OpenStreetMap 入力 IF

- 本データインターフェースの概要
 - OpenTripPlanner で使用する OpenStreetMap の道路データを入力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN011】 関連データ入力機能
- データ詳細
 - 道路ネットワークを用いた到達圏分析に利用するため、建物・水域など不要オブジェクトを削除した、道路系タグや規制・プラットフォームのオブジェクト。例：highway=* (motorway/trunk/primary…service)
 - OSM PBF (Protocolbuffer Binary Format) 形式
 - 以下のフォーマット説明を参照
 - ◇ https://wiki.openstreetmap.org/wiki/PBF_Format

【IF009】 編集済み GTFS 出力 IF

- 本データインターフェースの概要
 - 路線や停留所情報、便数変更などの編集後の GTFS Schedule (静的) を ZIP で出力するためのデータインターフェース

- 本インターフェースを利用する機能
 - 【FN011】 関連データ入力機能
- データ詳細
 - GTFS Schedule に準拠し、agency.txt、routes.txt、trips.txt、stops.txt、stop_times.txt、calendar.txt または calendar_dates.txt、shapes.txt、feed_info.txt を出力する。
 - 文字コード：UTF-8 (CRLF)

【IF010】 運行記録（日報）帳票作成ツール出力 IF

- 本データインターフェースの概要
 - 便ごと・停留所ごとの乗降者数を記録するための運行記録（日報）手入力シート、運行記録（日報）登録シートを出力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN015】 運行記録（日報）出力機能
- データ詳細
 - 便ごと・停留所ごとの乗降人数を記録できる様式の出力
 - trip_id、stop_id は、GTFS と 1 対 1 でひも付いた運行記録様式とする

表 2-19 出力項目

No.	属性名	説明	データ型
1	事業者	GTFS の agency_name に準拠	文字列
2	運行日	ユーザーが出力時に指定	YYYYMMDD
3	平休	GTFS の service_id に準拠	文字列
4	路線 ID	GTFS の route_id に準拠	数値
5	路線名	GTFS の route_name に準拠	文字列
6	上下区分	GTFS の direction_id に準拠	文字列
7	便名	GTFS の trip_id に準拠	文字列
8	停留所	GTFS の stop_id に準拠	文字列
9	通過順位	GTFS の stop_sequence に準拠	数値
10	出発時刻	GTFS の departure_time に準拠	時刻

【IF011】 駅・バス停間 OD データ出力 IF

- 本データインターフェースの概要
 - 出発地から目的地まで (Origin-Destination (起点-終点)) の輸送量のデータを CSV で出力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN004】 乗降実績データの入出力機能

● データ詳細

表 2-20 駅・バス停間 OD データ属性情報

No.	属性名	説明	データ型
1	date	日付	YYYYMMDD 形式
2	agency_id	事業者 ID	文字
3	route_id	路線 ID	文字
4	stopid_geton	乗車停留所 ID	文字
5	stopid_getoff	降車停留所 ID	文字
6	count	利用者数	数値

【IF012】地図出力 IF

- 本データインターフェースの概要
 - GTFS 可視化・乗降実績を可視化した地図を PNG 形式で画像出力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN003】道路ネットワークを用いた到達圏分析
 - 【FN005】乗降実績の可視化
 - 【FN006】交通分担率の変化シミュレーション
 - 【FN012】バッファ到達圏分析
 - 【FN013】運行頻度分析
 - 【FN014】OD の可視化
 - 【FN016】経路・時刻表機能
 - 【FN017】公共交通圏域機能
- データ詳細
 - GTFS 可視化・乗降実績を可視化した地図の画像

【IF013】グラフ出力 IF

- 本データインターフェースの概要
 - グラフを PNG 形式の画像出力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN003】道路ネットワークを用いた到達圏分析
 - 【FN005】乗降実績の可視化
 - 【FN006】交通分担率の変化シミュレーション
 - 【FN012】バッファ到達圏分析
 - 【FN013】運行頻度分析
 - 【FN014】OD の可視化
 - 【FN016】経路・時刻表機能
 - 【FN017】公共交通圏域機能

- データ詳細
 - 乗降実績のグラフの画像

【IF014】 表のデータ出力 IF

- 本データインターフェースの概要
 - 表のデータを CSV 形式で出力する
- 本インターフェースを利用する機能
 - 【FN003】 道路ネットワークを用いた到達圏分析
 - 【FN005】 乗降実績の可視化
 - 【FN006】 交通分担率の変化シミュレーション
 - 【FN012】 バッファ到達圏分析
 - 【FN013】 運行頻度分析
 - 【FN014】 OD の可視化
 - 【FN016】 経路・時刻表機能
 - 【FN017】 公共交通圏域機能
- データ詳細
 - 表形式で表示する路線別運行本数、停留所別乗降者数、時刻表、公共交通圏域の一覧表を、CSV 形式で出力する

表 2-21 路線別運行本数属性情報

No.	属性名	説明	データ型
1	route_id	路線 ID	文字
2	count	運行本数	数値

表 2-22 停留所別乗降者数属性情報

No.	属性名	説明	データ型
1	date	日付	YYYYMMDD 形式
2	stop_id	停留所 ID	文字
3	count_geton	全日乗車人数	数値
4	count_getoff	全日降車人数	数値

【IF015】 データベース入出力 IF

- 本データインターフェースの概要
 - 【CO003】 フロントエンド処理と 【CO002】 DB サーバーの間のデータを入出力する
- 本インターフェースを利用する機能
 - 【FN007】 データベース管理
- データ詳細
 - 【SL001】 PostgreSQL、【SL002】 PostGIS に格納したテーブル、データ

【IF016】 ユーザー情報 IF

- 本データインターフェースの概要
 - ログインに必要な ID・パスワードを認証機能に受け渡す
- 本インターフェースを利用する機能
 - 【FN008】 ログイン
- データ詳細
 - ID はシステム側で振り分けた値
 - パスワードは、ユーザー側で設定した値。英大文字小文字、数字、記号を利用可能。ハッシュ化して取り扱う。

【IF017】 GTFS データ入力 IF

- 本データインターフェースの概要
 - データ入力機能で取り込んだデータをデータベースに受け渡す
- 本インターフェースを利用する機能
 - 【FN001】 GTFS データ (API) 入力機能
 - 【FN010】 GTFS データ入力機能
- データ詳細
 - GTFS データ

【IF018】 GTFS DB 入出力 IF

- 本データインターフェースの概要
 - データベースに格納した GTFS データを GTFS 編集機能に受け渡す
 - 編集した GTFS データをデータベースに受け渡す
- 本インターフェースを利用する機能
 - 【FN002】 GTFS 編集機能
- データ詳細
 - GTFS データ、編集済み GTFS データ

【IF019】 乗降実績 DB 入出力 IF

- 本データインターフェースの概要
 - データベースに格納した乗降集計データ、乗降実績データ、駅・バス停間 OD データを乗降実績の入出力機能に受け渡す
- 本インターフェースを利用する機能
 - 【FN004】 乗降実績データの入出力機能
- データ詳細
 - 乗降集計データ、乗降実績データ、駅・バス停間 OD データ

【IF020】可視化用 DB 出力 IF

- 本データインターフェースの概要
 - データベースに格納したデータを GTFS 可視化、乗降実績の可視化、交通分担率の変化シミュレーションに受け渡す
- 本インターフェースを利用する機能
 - 【FN003】道路ネットワークを用いた到達圏分析
 - 【FN005】乗降実績の可視化
 - 【FN006】交通分担率の変化シミュレーション
 - 【FN012】バッファ到達圏分析
 - 【FN013】運行頻度分析
 - 【FN014】OD の可視化
 - 【FN016】経路・時刻表機能
 - 【FN017】公共交通圏域機能
- データ詳細
 - GTFS データ、編集済み GTFS データ、乗降集計データ、乗降実績データ、駅・バス停間 OD データ

【IF021】静的データ IF

- 本データインターフェースの概要
 - データベースに格納したデータを GTFS 可視化、乗降実績の可視化、交通分担率の変化シミュレーションに受け渡す
- 本インターフェースを利用する機能
 - 【FN003】道路ネットワークを用いた到達圏分析
 - 【FN005】乗降実績の可視化
 - 【FN006】交通分担率の変化シミュレーション
 - 【FN012】バッファ到達圏分析
 - 【FN013】運行頻度分析
 - 【FN014】OD の可視化
 - 【FN016】経路・時刻表機能
 - 【FN017】公共交通圏域機能
- データ詳細
 - 静的ファイル、画像

【IF022】道路交通センサス入力 IF

- 本データインターフェースの概要
 - 道路交通センサスのデータを入力するためのデータインターフェース
- 本インターフェースを利用する機能
 - 【FN011】関連データ入力機能
- データ詳細

- 全国道路・街路交通情勢調査、一般交通量調査集計表「2.箇所別基本表及び時間帯別交通量表」の都道府県別箇所別基本表 CSV 版を使用。
- 以下の説明資料を参照
 - ◇ <https://www.mlit.go.jp/road/census/r3/data/pdf/kasyorep.pdf>

表 2-23 道路交通センサス使用する属性情報

No.	属性名	説明	データ型
1	交通調査基本区間番号	原則として「都道府県 (2桁)」+「道路種別 (1桁)」+「路線番号 (4桁)」+「順番号 (4桁)」から成る 11桁の番号 DRMの「交通センサス区間番号」とひも付けるために用いる	数値
2	区間延長 (km)	当該交通調査基本区間の延長 (道路中心線上の延長)	数値
3	上り 24 時間自動車類交通量合計 (台)	午前 7 時～翌日午前 7 時または午前 0 時～翌日午前 0 時まで上り方向に交通量観測地点を通過した自動車類の台数	数値
4	下り 24 時間自動車類交通量合計 (台)	午前 7 時～翌日午前 7 時または午前 0 時～翌日午前 0 時まで下り方向に交通量観測地点を通過した自動車類の台数	数値
5	24 時間自動車類交通量 (上下合計) 合計 (台)	午前 7 時～翌日午前 7 時または午前 0 時～翌日午前 0 時まで上下方向に交通量観測地点を通過した自動車類の台数	数値
6	混雑度	交通調査基本区間の交通容量に対する交通量の比	数値
7	昼間 12 時間平均旅行速度 (時間帯別交通量加重) 上り (km/h)	交通調査基本区間を上り方向に通過する自動車類の昼間 12 時間の平均速度	数値
8	昼間 12 時間平均旅行速度 (時間帯別交通量加重) 下り (km/h)	交通調査基本区間を下り方向に通過する自動車類の昼間 12 時間の平均速度	数値
9	交差点密度 (箇所/km) 信号交差点	当該交通調査基本区間等で構成する道路状況調査単位区間における信号あり・信号なし別の 1 km 当たりの交差点箇所数	数値
10	車線数	当該交通調査基本区間等で構成する道路状況調査単位区間の代表断面における車線数	数値

No.	属性名	説明	データ型
		車線数は、上下線合計（一方通行区間の場合を除く）	

【IF023】 関連データ入力 IF

- 本データインターフェースの概要
 - データ入力機能で取り込んだデータをデータベースに受け渡す
- 本インターフェースを利用する機能
 - 【FN011】 関連データ入力機能
- データ詳細
 - 人口統計データ等、OpenStreetMap データ、DRM データ、施設データ

【IF024】 運行記録（日報）帳票作成ツールへの GTFS 入力 IF

- 本データインターフェースの概要
 - 自治体職員等が現場の紙の運行記録（日報）を基に乗降実績を転記するための運行記録（日報）登録シートを作成するために GTFS を取り込むインターフェース
- 本インターフェースを利用する機能
 - 【FN015】 運行記録（日報）出力機能
- データ詳細
 - GTFS 形式に準拠したデータを、ローカル環境からアップロードし入力する。
 - 利用するデータは、【IF002】 GTFS 入力 IF（API）と同じ。
 - 以下の技術資料を参照
 - ◇ https://www.mlit.go.jp/sogoseisaku/transport/sosei_transport_tk_000112.html

【IF025】 乗降集計データ出力 IF

- 本データインターフェースの概要
 - 運行記録（日報）登録シートに転記された情報から乗降集計データを出力する
- 本インターフェースを利用する機能
 - 【FN015】 運行記録（日報）出力機能
- データ詳細

表 2-24 乗降集計データ属性情報

No.	属性名	説明	データ型
1	date	日付	YYYYMMDD 形式
2	agency_id	事業者 ID	文字
3	route_id	路線 ID	文字
4	trip_id	便 ID	文字
5	stop_id	停留所 ID	文字
6	Count_geton	乗車人数	数値

7	Count_getoff	降車人数	数値
---	--------------	------	----

【IF026】 施設データ入力 IF

- 本データインターフェースの概要
 - 施設データを入力するためのインターフェース
- 本インターフェースを利用する機能
 - 【FN011】 関連データ入力機能
- データ詳細

表 2-25 施設データ属性情報

No.	属性名	説明	データ型
1	タイプ	施設のタイプ	文字
2	名前	施設の名称	文字
3	緯度	施設の緯度	数値
4	経度	施設の経度	数値
5	備考	備考	文字

2-5. ユーザーインターフェース (UI)

2-5-1. 画面遷移図

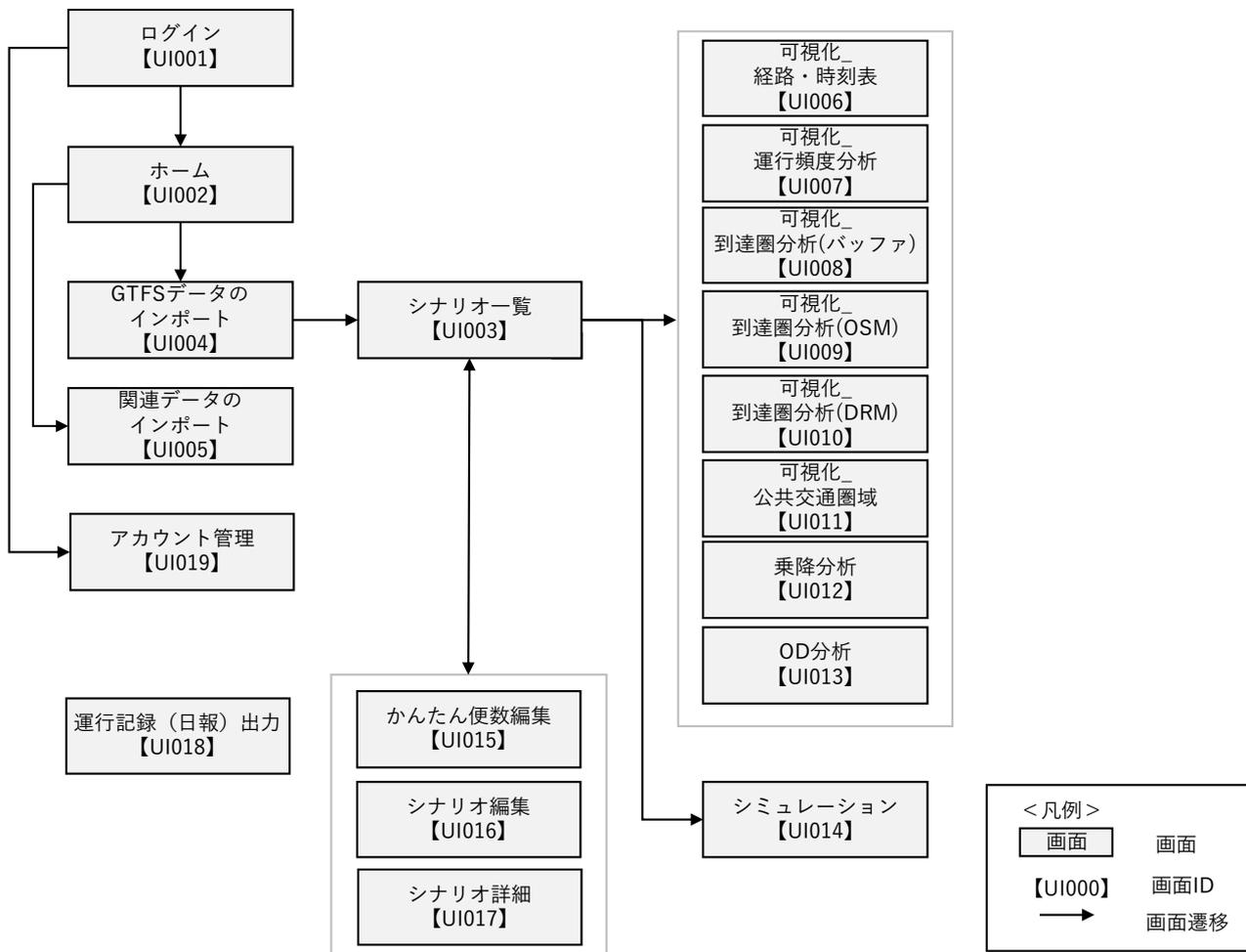


図 2-46 PC 用画面一覧

2-5-2. ユーザーインターフェース一覧

表 2-26 PC用画面一覧

※朱文字：新規開発・既存改修

ID	画面名	説明	画面を表示した機能 (ID)
UI001	ログイン	<ul style="list-style-type: none"> ● ユーザー名/パスワードを入力してログインする 	FN008
UI002	ホーム	<ul style="list-style-type: none"> ● 機能選択のダッシュボード ● 左メニューと中央タイルで各機能にアクセスする 	FN008
UI003	シナリオ一覧	<ul style="list-style-type: none"> ● インポート、アップロード、編集した GTFS データを一覧で確認する 	FN001
UI004	GTFS データインポート	<ul style="list-style-type: none"> ● GTFS データを、API を用いてインポート、又はローカル環境からアップロードする 	FN001 FN010
UI005	関連データのインポート	<ul style="list-style-type: none"> ● 施設データを CSV 形式でインポートする ● 学校、カフェ、公園、病院等の施設情報を登録する ● 乗降実績データをアップロードする ● 自動で乗降集計データと駅・バス停間 OD データが生成される 	FN004 FN011
UI006	可視化_経路・時刻表	<ul style="list-style-type: none"> ● 路線図や駅・停留所別の時刻表を可視化する 	FN016
UI007	可視化_運行頻度分析	<ul style="list-style-type: none"> ● 選択したシナリオで運行頻度を可視化する ● 地図、グラフ・表を出力する 	FN013
UI008	可視化_到達圏分析 (バッファ)	<ul style="list-style-type: none"> ● 選択したシナリオでバッファ到達圏分析を行う ● 地図、グラフ・表を出力する 	FN012
UI009	可視化_到達圏分析 (OSM)	<ul style="list-style-type: none"> ● 選択したシナリオでネットワーク分析を行う ● OSM 道路ネットワークを用いて到達圏を可視化する ● 地図、グラフ・表を出力する 	FN003
UI010	可視化_到達圏分析 (DRM)	<ul style="list-style-type: none"> ● 選択したシナリオでネットワーク分析を行う ● DRM 道路ネットワークを用いて到達圏を可視化する ● 地図、グラフ・表を出力する 	FN003
UI011	可視化_公共交通圏域	<ul style="list-style-type: none"> ● 公共交通圏域を可視化する ● 停留所から指定距離内の範囲における施設・人口を集計する ● 地図、グラフ・表を出力する 	FN017
UI012	乗降分析	<ul style="list-style-type: none"> ● 乗降集計データをアップロードする ● 路線の乗降分析：乗降実績を可視化し、路線・停留所・時間帯別の利用状況を分析する ● 区間の時間帯別利用者数：区間別・時間帯別の利用者数を 	FN004 FN005

		<p>可視化する</p> <ul style="list-style-type: none"> ● 停留所別利用者数：停留所別の利用者数を可視化する ● 地図、グラフ・表を出力する 	
UI013	OD 分析	<ul style="list-style-type: none"> ● 駅・バス停間 OD データをアップロードする ● 停留所別利用者数：停留所別の OD 利用者数を可視化する ● OD 流動図：OD の移動の流れを可視化する ● OD 量の可視化：OD 量（起終点間の移動量）を可視化する ● 地図、グラフ・表を出力する 	FN004 FN014
UI014	シミュレーション	<ul style="list-style-type: none"> ● シミュレーションを実行する ● 地図、グラフ・表を出力する 	FN006
UI015	かんたん便数編集	<ul style="list-style-type: none"> ● 路線グループ、路線、運行パターンごとに運行本数を編集する 	FN002
UI016	シナリオ編集	<ul style="list-style-type: none"> ● 時刻表の変更：指定した路線の時刻表を変更する ● 路線作成：指定した路線（経路）を編集して、便を追加する ● 標柱変更：標柱・停留所を追加・変更し、それに伴う路線（経路）を編集する ● 運行日変更：運行日を追加・変更する 	FN002
UI017	シナリオ詳細	<ul style="list-style-type: none"> ● シナリオ一覧で選択した GTFS データの詳細を確認する 	FN002 FN003
UI018	運行記録（日報）出力	<ul style="list-style-type: none"> ● GTFS をインポートして、設定した日付・路線の運行記録（日報）手入力シートを出力する ● GTFS をインポートして、設定した年月の運行記録（日報）登録シートを出力する ● 乗降集計データを CSV 形式で出力する 	FN015
UI019	アカウント管理	<ul style="list-style-type: none"> ● ユーザー名／パスワード等、アカウント情報の管理を行う 	FN009

2-5-3. ユーザーインターフェースの詳細

ユーザーインターフェース（画面）の詳細を記す。なお、本業務において開発（新規・改修）を行うユーザーインターフェース（画面）を**朱文字**で示す。

全ての画面においてユーザーが迷わないよう、その画面で何ができるかを説明する文章を記載する。また、全画面で統一したアイコンとする。

システムの背景図には地理院タイルを使用しているため、本資料のシステムの画面キャプチャの背景図にも、地理院タイルを表示している (<https://maps.gsi.go.jp/development/ichiran.html>)。

1) 【HW001】 PC 用画面

【UI001】 ログイン

図 2-47 ログイン

- 本画面の概要
 - ユーザー名／パスワードを入力してログインする
 - ユーザー名／パスワードはシステム管理者が発行する
- 本画面から利用する機能
 - **【FN008】 ログイン**

【UI002】 ホーム

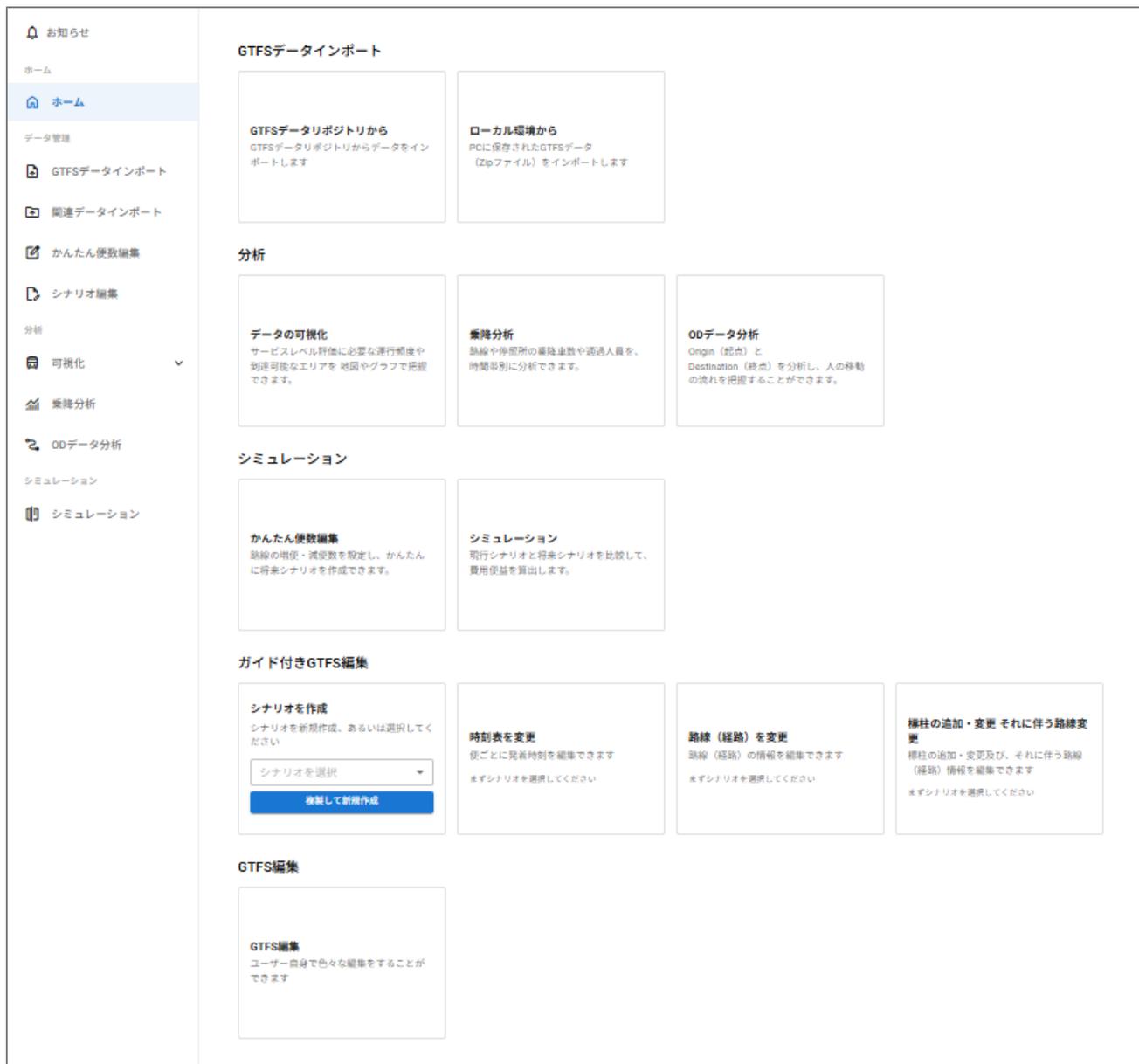


図 2-48 ホーム

- 本画面の概要
 - ログイン後に最初に表示されるダッシュボード画面
 - 左側メニューと中央タイル形式で全機能へアクセスできる
 - 各機能の概要説明が中央タイルに表示される
 - システム全体のナビゲーションハブとして機能する
- 本画面から利用する機能
 - 【FN001】 GTFS データ (API) 入力機能
 - 【FN002】 GTFS 編集機能
 - 【FN003】 道路ネットワークを用いた到達圏分析

- 【FN004】 乗降実績データの入出力機能
- 【FN005】 乗降実績の可視化
- 【FN006】 交通分担率の変化シミュレーション
- 【FN008】 ログイン
- 【FN009】 ログアウト
- 【FN010】 GTFS データ入力機能
- 【FN011】 関連データ入力機能
- 【FN012】 バッファ到達圏分析
- 【FN013】 運行頻度分析
- 【FN014】 OD の可視化
- 【FN016】 経路・時刻表機能
- 【FN017】 公共交通圏域機能

【UI003】シナリオ一覧

シナリオ一覧						
シナリオを複製して新規作成						
シナリオ名	フィールド名	開始日	終了日	シナリオ作成日時	シナリオ更新日時	シナリオ作成方法
シナリオ Chitetsu 自分	chitetsu_chitetsubus	2025/10/16	2026/10/15	2025/12/18 13:51	2025/12/18 14:58	ローカルからインポート
シナリオ 富山地方鉄道バス 自分	chitetsubus	2025/11/01	2026/10/31	2025/12/12 9:31	2025/12/12 18:16	富山地方鉄道バスを複製して編集
前橋_永井運輸GTFS編集 自分	AllLines-20251001	2025/11/01	2026/04/15	2025/12/11 20:23	2025/12/11 20:51	前橋_永井運輸を複製して編集
前橋_群馬中央バスGTFS編集 自分	GTFS-JP(gunmachuo)	2025/04/02	2026/03/31	2025/12/11 19:44	2025/12/11 20:07	前橋_群馬中央バスを複製して編集
前橋_阿部交通GTFS編集 自分	GTFS-JP(kan-etsu)	2025/12/02	2026/12/01	2025/12/11 18:47	2025/12/11 19:33	前橋_阿部交通を複製して編集
前橋_上信観光 自分	AllLines-20250401	2025/04/01	2026/03/31	2025/12/11 17:50	-	ローカルからインポート
前橋_群馬バス 新線路線 POCK_Sandbox (pock_sawano)	GTFS-JP_gunbus	2025/07/14	2026/03/31	2025/12/11 15:17	2025/12/11 15:49	前橋_群馬バス ガイド付きを複製して編集
前橋_群馬バス ガイド付き POCK_Sandbox (pock_sawano)	GTFS-JP_gunbus	2025/07/14	2026/03/31	2025/12/11 14:56	2025/12/11 15:10	前橋_群馬バス かんたん編集を複製して編集
前橋_群馬バス かんたん編集 POCK_Sandbox (pock_sawano)	GTFS-JP_gunbus	2025/07/11	2026/03/31	2025/12/11 14:44	2025/12/11 14:47	前橋_群馬バスを複製して編集
前橋_群馬バスGTFS編集 自分	GTFS-JP_gunbus	2025/08/01	2026/04/15	2025/12/11 14:25	2025/12/11 17:15	前橋_群馬バスを複製して編集

図 2-49 シナリオ一覧

- 本画面の概要
 - インポート、アップロード、編集した GTFS データのシナリオを一覧で表示する
 - GTFS データのシナリオを削除する
 - GTFS データのシナリオを複製して新規作成する
 - GTFS データとしてダウンロードする
 - ログインユーザーごとのシナリオを表示する
- 本画面から利用する機能
 - 【FN001】 GTFS データ (API) 入力機能
 - 【FN010】 GTFS データ入力機能

【UI004】 GTFS データインポート

GTFSデータインポート

[GTFSデータリポジトリから取得](#) ローカルからインポート

都道府県

未選択

事業者名

未選択

事業者名	都道府県	GTFSフィールド名	ライセンス	最新フィールド開始日	最新フィールド終了日	最終公開日	詳細
岩宇地域公共交通活性化協議会	北海道	しおかぜライン	CC BY 4.0 公開元: 岩宇地域公共交通活性化協議会	2025-10-01	2026-09-30	2025-09-22	詳細
上士幌町	北海道	上士幌町コミュニティバス	CC BY 4.0 公開元: 上士幌町	2025-12-01	2026-10-31	2025-12-02	詳細
上士幌町	北海道	上士幌町自動運転バス	CC BY 4.0 公開元: 上士幌町	2025-11-28	2026-10-31	2025-12-02	詳細
根室交通株式会社	北海道	根室交通株式会社	CC0 1.0 公開元: 根室交通株式会社	2025-08-07	2026-08-06	2025-08-07	詳細
階上町	青森県	階上町コミュニティバス	CC0 1.0 公開元: 階上町	2025-04-01	2026-03-31	2025-06-25	詳細
弘前市	青森県	弘南鉄道	CC BY 4.0 公開元: 弘前市	2025-06-11	2026-03-31	2025-11-06	詳細

図 2-50 GTFS データインポート (API)

- 本画面の概要
 - GTFS データリポジトリや公共交通オープンデータセンター (ODPT) から API を用いて GTFS データをインポートする
 - GTFS データリポジトリのデータを一覧表示する
 - 都道府県、事業者名、フィールド名で一覧を絞り込み表示する
- 本画面から利用する機能
 - 【FN001】 GTFS データ (API) 入出力機能



図 2-51 GTFS データインポート（ローカル）

- 本画面の概要
 - GTFS データをローカル環境からアップロードする
- 本画面から利用する機能
 - 【FN010】GTFS データ入力機能

【UI005】 関連データのインポート



図 2-52 関連データインポート（施設データインポート）

- 本画面の概要
 - 施設データを CSV 形式でインポートする
 - 学校、カフェ、公園、病院等の施設情報を登録する
 - ログインユーザーごとにインポートデータをひも付ける
- 本画面から利用する機能
 - 【FN011】 関連データ入力機能

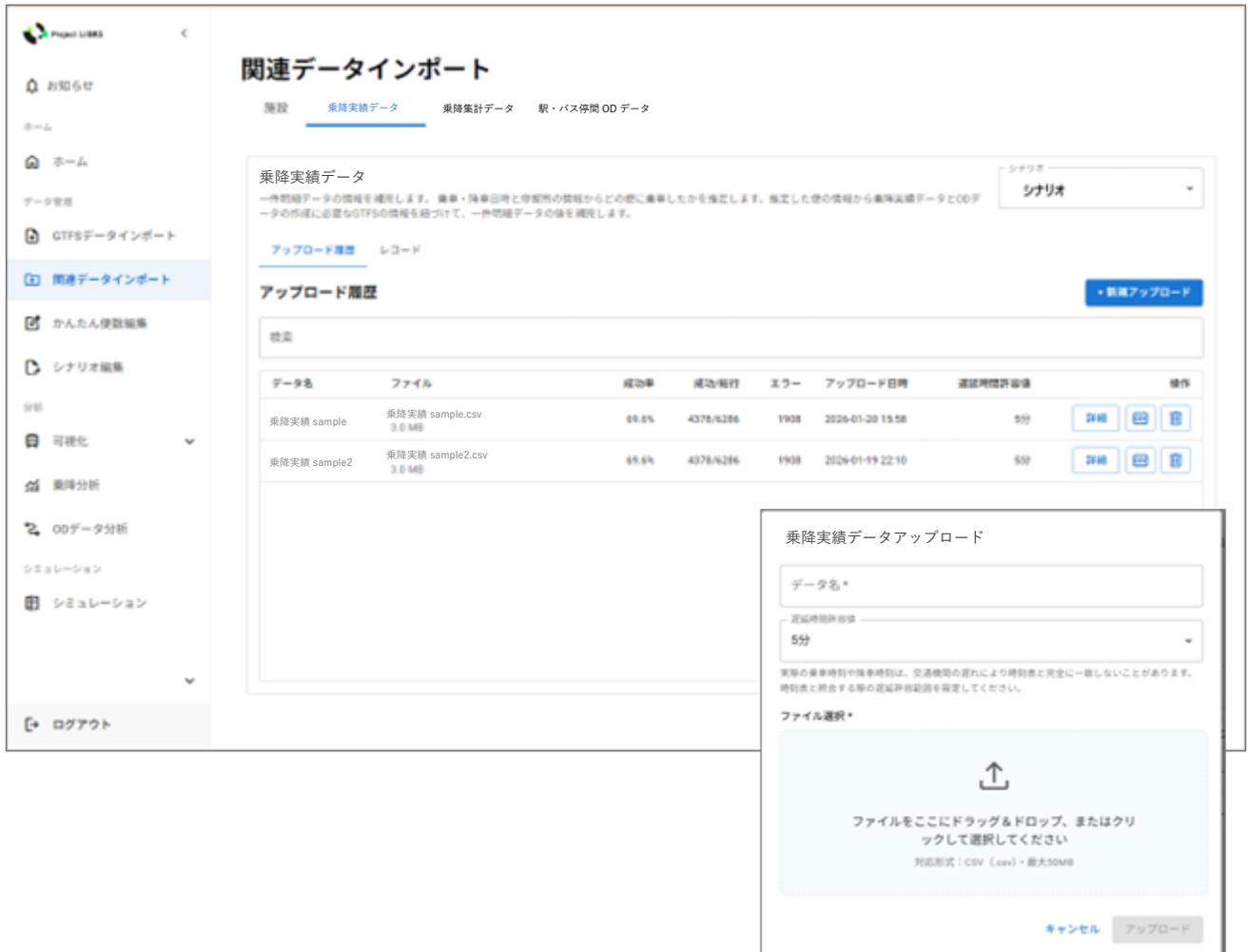


図 2-53 関連データインポート（乗降実績データインポート）

- 本画面の概要
 - 乗降実績データをアップロードする
 - GTFIS の情報と照合するための遅延時間許容値を指定する
 - 乗車・降車日時と停留所から、どの便に乗車したかを推定する
 - 推定した便から乗車実績データと駅・バス停間 OD データの作成に必要な GTFIS の情報をひも付けて、乗降実績データの値を補完する
 - アップロードされた乗降実績データを一覧表示する
 - アップロードされた乗降実績データのデータ名とファイル名を文字列で検索して絞り込み表示する
- 本画面から利用する機能
 - 【FN004】乗降実績データの入出力機能



図 2-54 関連データインポート (乗降集計データ作成)

- 本画面の概要
 - 乗降実績データから乗降集計データを CSV 形式で出力する
 - アップロードされた乗降実績データを一覧表示する
 - アップロードされた乗降実績データのデータ名とファイル名を文字列で検索して絞り込み表示する
- 本画面から利用する機能
 - 【FN004】乗降実績データの入出力機能



図 2-55 関連データインポート (駅・バス停間 OD データ作成)

- 本画面の概要
 - 乗降実績データから駅・バス停間 OD データを CSV 形式で出力する
 - アップロードされた乗降実績データを一覧表示する
 - アップロードされた乗降実績データのデータ名とファイル名を文字列で検索して絞り込み表示する
- 本画面から利用する機能
 - 【FN004】乗降実績データの入出力機能

【UI006】可視化_経路・時刻表



図 2-56 可視化_経路・時刻表

- 本画面の概要
 - 路線図や駅・停留所別の時刻表を可視化する
 - シナリオ・標柱／停留所・路線グループ・タイムレンジ・往復区分・運行日によって表示を絞り込む
 - 地図上で路線の塗分け、標柱／停留所の表示、ラベル表示を切り替える
- 本画面から利用する機能
 - 【FN016】経路・時刻表機能

【UI007】可視化_運行頻度分析



図 2-57 可視化_運行頻度分析

- 本画面の概要
 - シナリオ・路線グループ・往復区分・運行日・タイムレンジを選択すると、運行頻度図を表示する
 - 選択項目だけでなく、地図の選択に応じて自動で結果表示が変わる
 - 地図上に表示する情報（路線・停留所・標柱・関連データ等）を選択する
 - 地図、グラフ・表を出力する
- 本画面から利用する機能
 - 【FN013】運行頻度分析

【UI008】可視化_到達圏分析（バッファ）

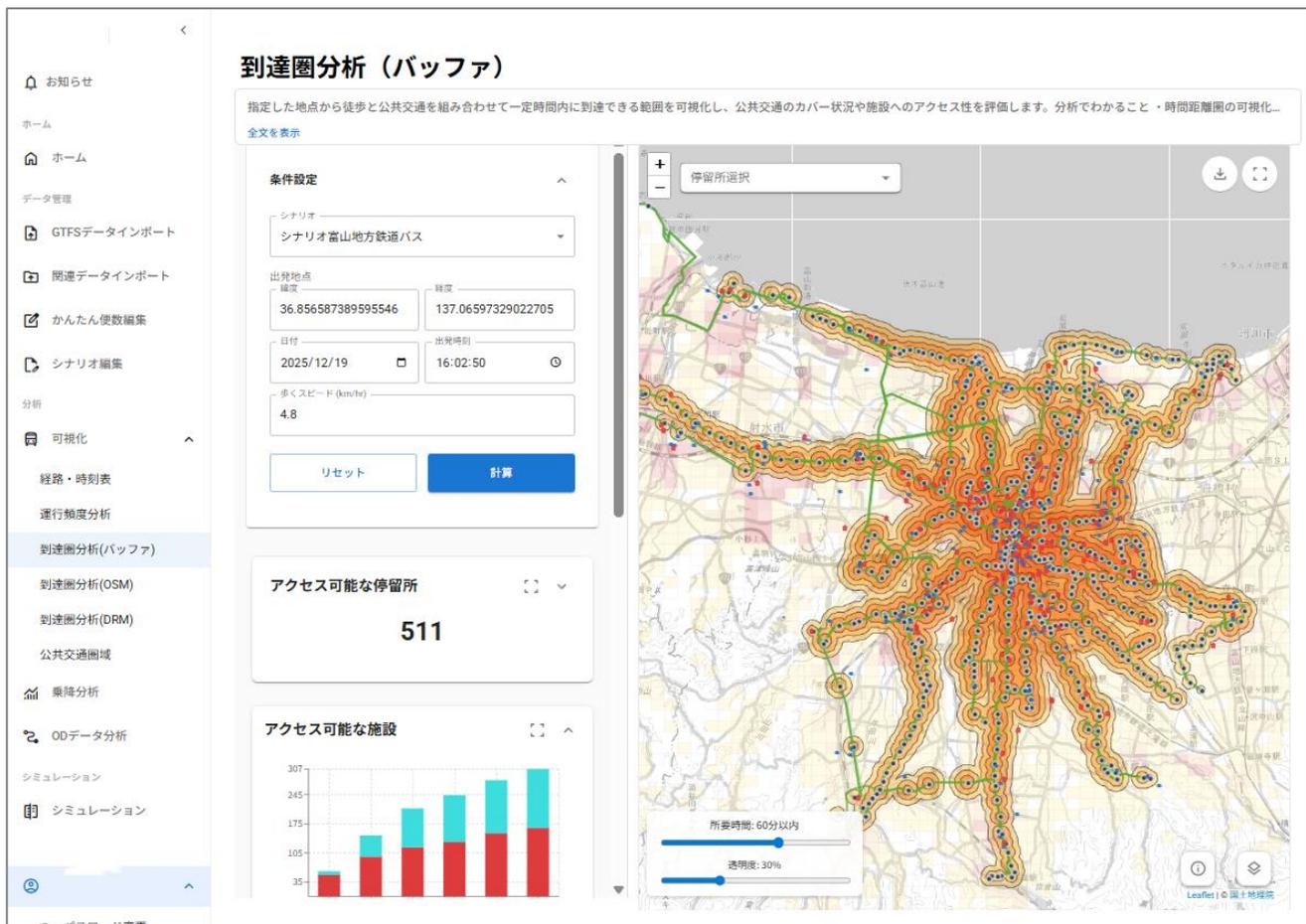


図 2-58 可視化_到達圏分析（バッファ）

- 本画面の概要
 - シナリオ・路線グループ・停留所・往復区分・運行日を選択した上で、ある地点からのバッファ到達圏分析を行う
 - 選択項目だけでなく、地図の選択に応じて自動で結果表示が変わる
 - 地図・グラフ・表を出力できる
 - 地図上に表示する情報（路線・停留所・標柱・関連データ等）を選択できる
 - 地図画面は下から、背景地図、バッファのポリゴン、路線、停留所、施設の順に重畳させて表示させる
 - 重畳表示するバッファのポリゴンは、透明度のスライダーによって、ユーザーが必要に応じて透明度を変更できる
 - 本画面から利用する機能

【UI009】可視化_到達圏分析 (OSM)

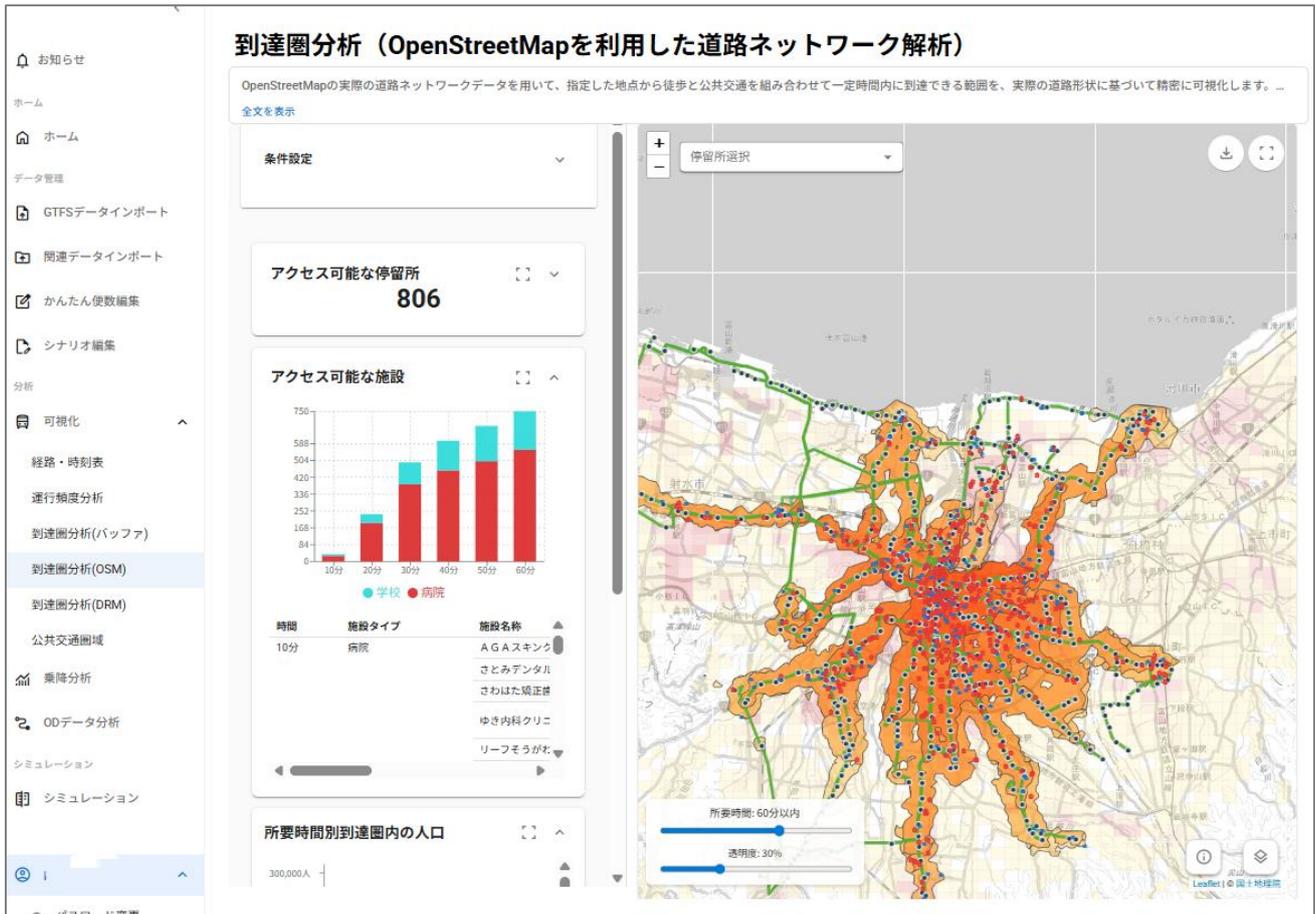


図 2-59 可視化_到達圏分析 (OSM)

- 本画面の概要
 - OSM 道路ネットワークを用いて到達圏を可視化する
 - シナリオ・路線グループ・停留所・往復区分・運行日を選択した上で、ある地点からの道路ネットワークを用いた到達圏分析を行う
 - 路線地図・グラフ・時刻表が表示される
 - 選択項目だけでなく、地図の選択に応じて自動で結果表示が変わる
 - 地図設定から、地図上に表示する情報（路線・停留所・標柱・関連データ等）を選択できる
 - 地図、グラフ・表を出力する
- 本画面から利用する機能
 - 【FN003】道路ネットワークを用いた到達圏分析

【UI010】可視化_到達圏分析 (DRM)

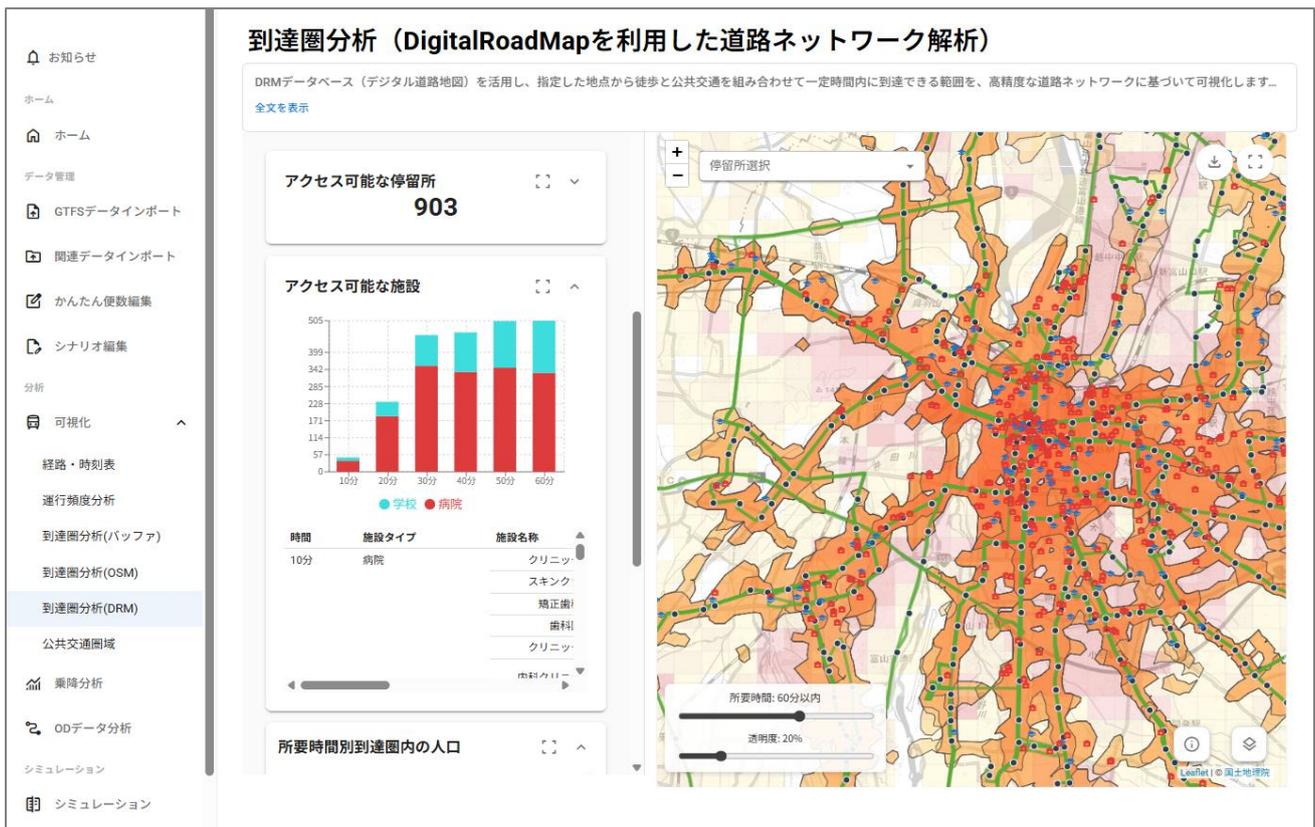


図 2-60 可視化_到達圏分析 (DRM)

- 本画面の概要
 - DRM 道路ネットワークを用いて到達圏を可視化する
 - シナリオ・路線グループ・停留所・往復区分・運行日を選択した上で、ある地点からの道路ネットワークを用いた到達圏分析を行う
 - 路線地図・グラフ・時刻表が表示される
 - 選択項目だけでなく、地図の選択に応じて自動で結果表示が変わる
 - 地図設定から、地図上に表示する情報（路線・停留所・標柱・関連データ等）を選択できる
 - 地図、グラフ・表を出力する
- 本画面から利用する機能
 - 【FN003】道路ネットワークを用いた到達圏分析

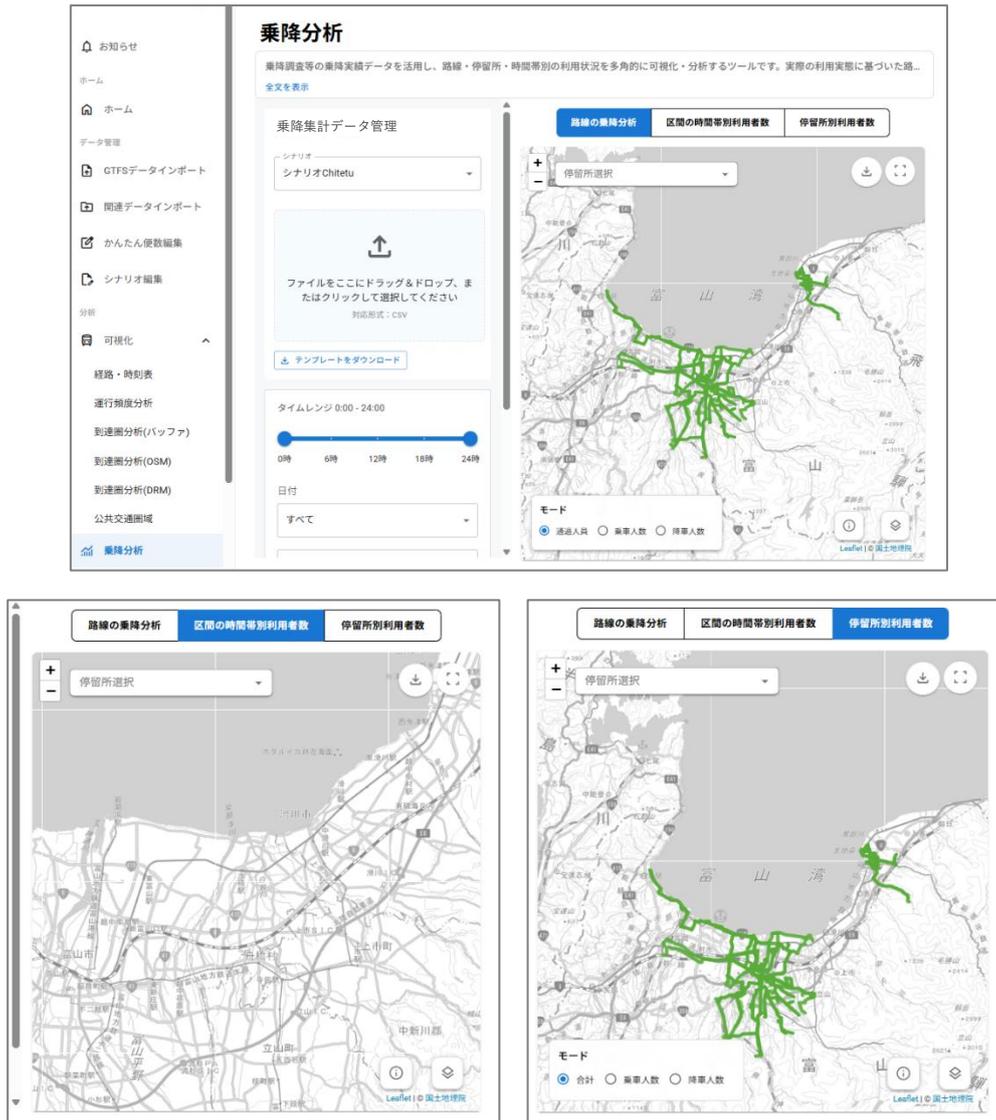
【UI011】可視化_公共交通圏域



図 2-61 可視化_公共交通圏域

- 本画面の概要
 - 公共交通圏域を可視化する
 - 停留所から指定距離内の範囲における施設・人口を集計する
 - 地図、グラフ・表を出力する
- 本画面から利用する機能
 - 【FN017】公共交通圏域機能

【UI012】乗降分析

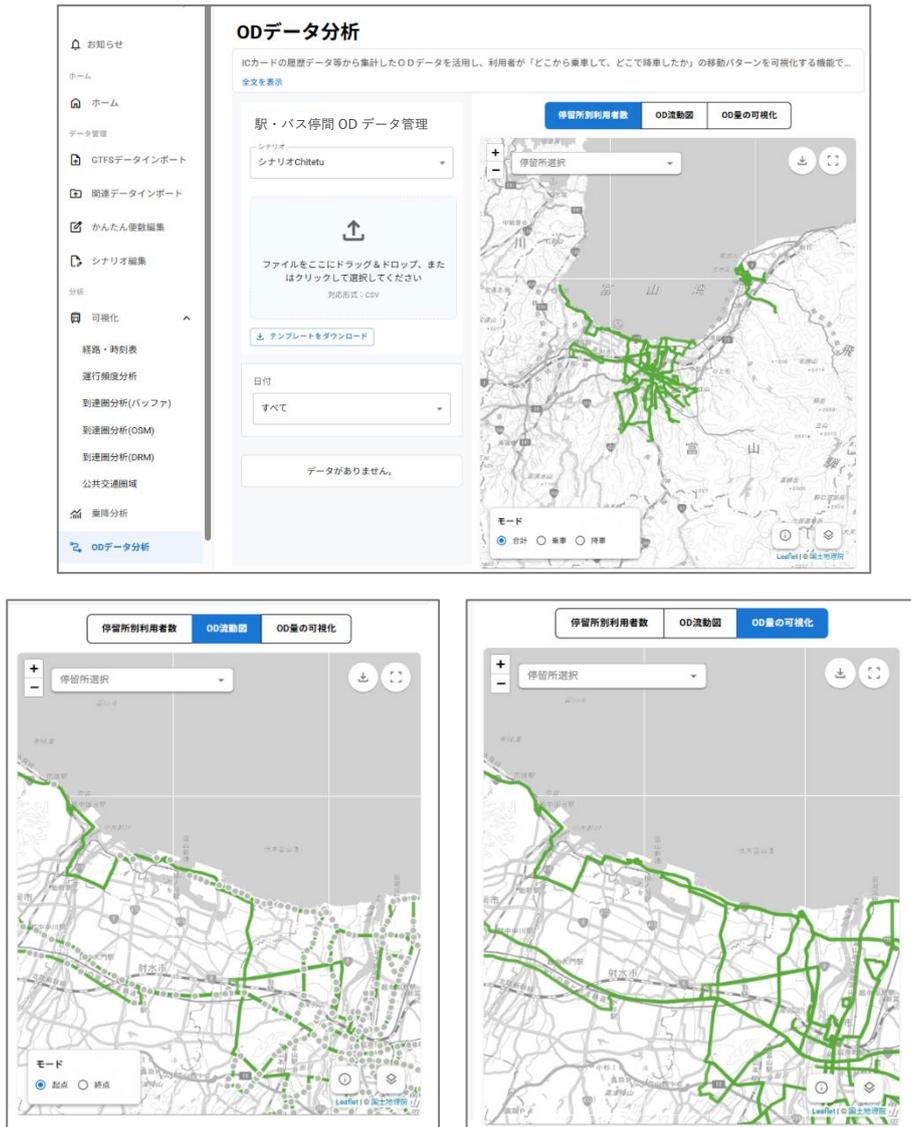


上段：路線の乗降分析、下段左：区間の時間帯別利用者数、下段右：停留所別利用者数

図 2-62 乗降分析

- 本画面の概要
 - 乗降集計データをアップロードする
 - シナリオ・路線グループ・運行日・タイムレンジを選択し、利用する乗降集計データを選択すると、駅別乗降実績が可視化される
 - 選択項目だけでなく、地図の選択に応じて自動で結果表示が変わる
 - 地図・グラフ・表を出力できる
 - 地図設定から、地図上に表示する情報（路線・停留所・標柱・関連データ等）を選択できる
- 本画面から利用する機能
 - 【FN004】乗降実績データの入出力機能
 - 【FN005】乗降実績の可視化

【UI013】 OD 分析



上段：停留所別利用者数、下段左：OD 流動図、下段右：OD 量の可視化

図 2-63 OD 分析

- 本画面の概要
 - 駅・バス停間 OD データをアップロードする
 - シナリオ・運行日を選択し、利用する駅・バス停間 OD データを選択すると、OD 輸送量が可視化される
 - 選択項目だけでなく、地図の選択に応じて自動で結果表示が変わる
 - 地図・グラフ・表を出力できる
 - 地図設定から、地図上に表示する情報（路線・停留所・標柱・関連データ等）を選択できる
- 本画面から利用する機能
 - 【FN004】乗降実績データの入出力機能
 - 【FN014】OD の可視化

【UI014】 シミュレーション



図 2-64 シミュレーションシナリオ一覧画面



図 2-65 シミュレーション詳細

- 本画面の概要
 - 交通分担率の変化シミュレーションを行う
 - 変化前後のシナリオを並べて表示することで、選択したシナリオ間で結果を比較できる
 - 地図、グラフ、表を出力する
- 本画面から利用する機能
 - 【FN006】交通分担率の変化シミュレーション

【UI015】かんたん便数編集



図 2-66 かんたん便数編集

- 本画面の概要
 - 路線グループ、路線、運行パターンごとに運行本数を簡易に編集する
 - 便数の編集
 - ◇ シミュレーションに必要な増減便を行うため、指定した既存のルートに対して、運行本数の増減を×、+、-を操作することで設定する
- 本画面から利用する機能
 - 【FN002】GTFS 編集機能

【UI016】シナリオ編集

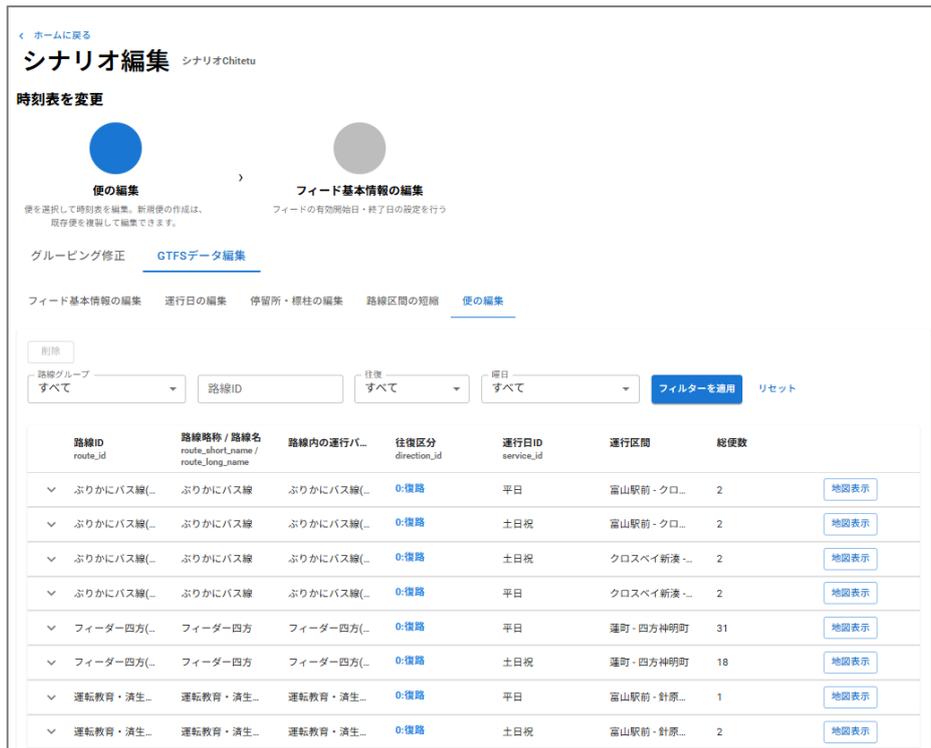


図 2-67 シナリオ編集（「時刻表を変更」のガイド）



図 2-68 シナリオ編集（「路線（経路）を作成して、便を追加」のガイド）



図 2-69 シナリオ編集（「標柱の追加・変更及び、それに伴う路線（経路）の変更」のガイド）

- 本画面の概要
 - GTFS の路線・停留所・便情報の編集
 - ◇ ユーザーが使いやすいUIとなるよう配慮した画面とするため、ユーザーが選択する編集メニューに応じて編集の操作フローを表示する
 - ◇ ユーザーが迷わないよう、ガイド付きの編集メニューは、操作フローの各プロセスを選択することで、そのプロセスで編集操作が必要となる画面に遷移させる
 - ◇ ガイド付きの編集メニューは、ユーザーがよく実施する「時刻表を変更」「路線（経路）の変更」「標柱の追加・変更及び、それに伴う路線（経路）の変更」とする
- 本画面から利用する機能
 - 【FN002】GTFS 編集機能



上段：便の編集、下段：時刻表の変更

図 2-70 シナリオ編集（時刻表の変更）

- 本画面の概要
 - 指定した路線の時刻表を変更する
 - 始発停留所の到着・出発時刻を指定すると、それ以降の停留所の時刻が自動で入力される
- 本画面から利用する機能
 - 【FN002】GTFS 編集機能



図 2-71 シナリオ編集（路線区間の短縮）

- 本画面の概要
 - 開始ポイント・終点ポイントを指定し、既存の路線区間を短縮する
- 本画面から利用する機能
 - 【FN002】 GTFS 編集機能

シナリオ編集 シナリオChitetsu

グループ修正 [GTFSデータ編集](#)

フィールド基本情報の編集 運行日の編集 [停留所・標柱の編集](#) 路線区間の短縮 便の編集

[新規標柱作成](#)

停留所ID	停留所名称	停留所緯度	停留所経度	
SG00235	富山駅前	36.700527	137.213117	地図表示
標柱ID stop_id	標柱名称 stop_name	緯度 stop_lat	経度 stop_lon	
101_01	富山駅前	36.700638	137.213592	編集 複製 削除
101_03	富山駅前	36.700962	137.213486	編集 複製 削除
101_04	富山駅前	36.701038	137.212940	編集 複製 削除
101_05	富山駅前	36.700862	137.212855	編集 複製 削除
101_06	富山駅前	36.700707	137.212892	編集 複製 削除
101_07	富山駅前	36.700561	137.213016	編集 複製 削除
101_08	富山駅前	36.699688	137.213219	編集 複製 削除
101_11	富山駅前	36.699761	137.212934	編集 複製 削除
SG00217	富山市役所前	36.695476	137.213083	地図表示
SG00150	城址公園前	36.692637	137.212608	地図表示

標柱の追加・変更及び、それに伴う路線（経路）の変更

標柱の新規作成

停留所ID SG00590 停留所名称 道の駅新湊（上り）

標柱ID stop_id 999_0

※ 同じIDを登録することはできません。必ず変更してください。

標柱名称 stop_name 道の駅新湊（上り）

緯度 stop_lat 36.781268

経度 stop_lon 137.088585

標柱番号 stop_code

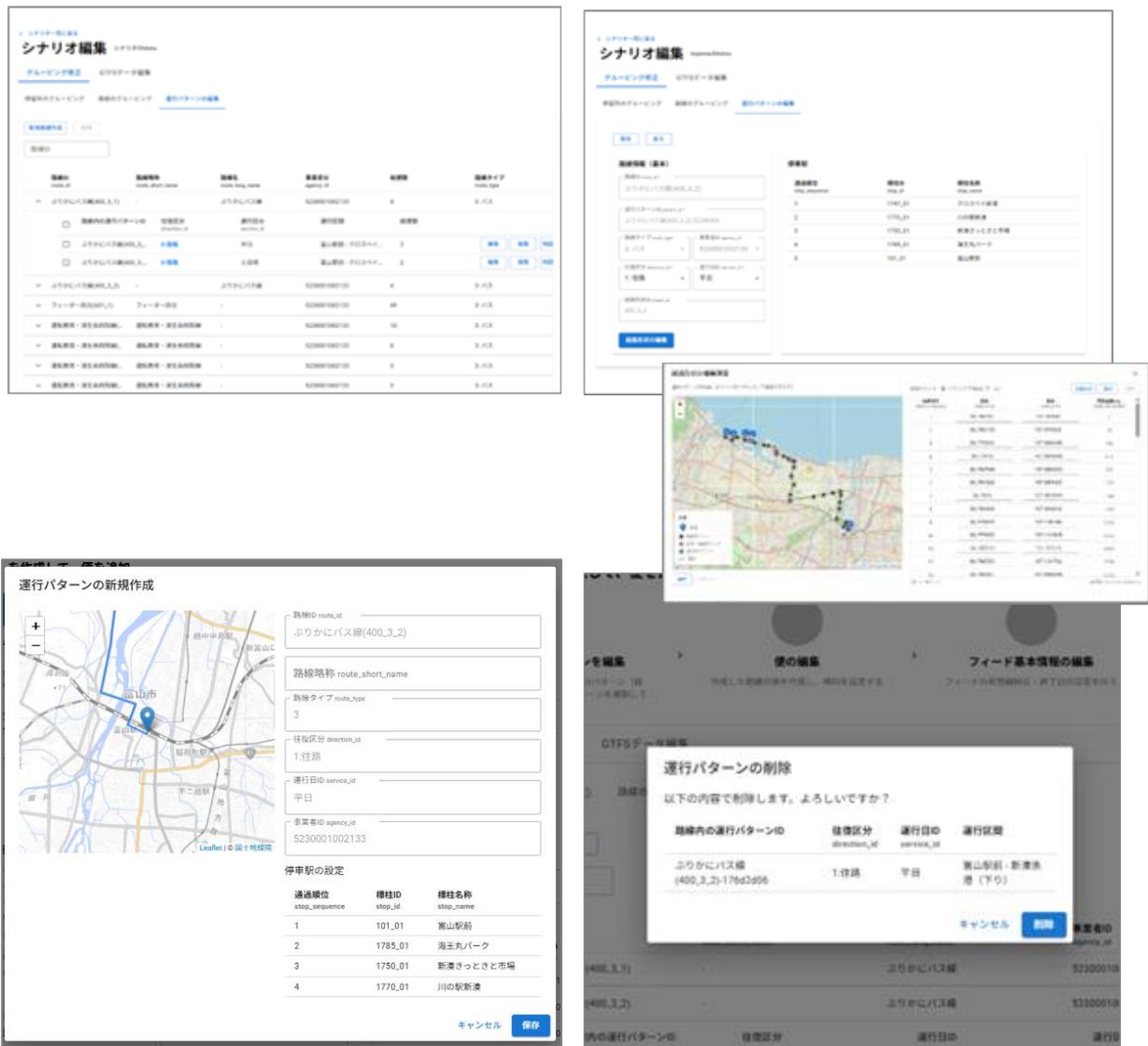
翻訳を追加 [翻訳を追加](#)

[キャンセル](#) [新規作成](#)

上段：停留所・標柱の編集、下段：標柱の新規作成

図 2-72 シナリオ編集（標柱の追加・変更）

- 本画面の概要
 - 標柱・停留所を追加・変更する
 - 停留所の編集
 - ◇ 標柱・停留所を新規作成・削除する
 - ◇ 停留所名、停留所 ID を入力する
 - ◇ 地図上で標柱の位置をクリックし緯度・経度を指定する
- 本画面から利用する機能
 - 【FN002】GTFS 編集機能

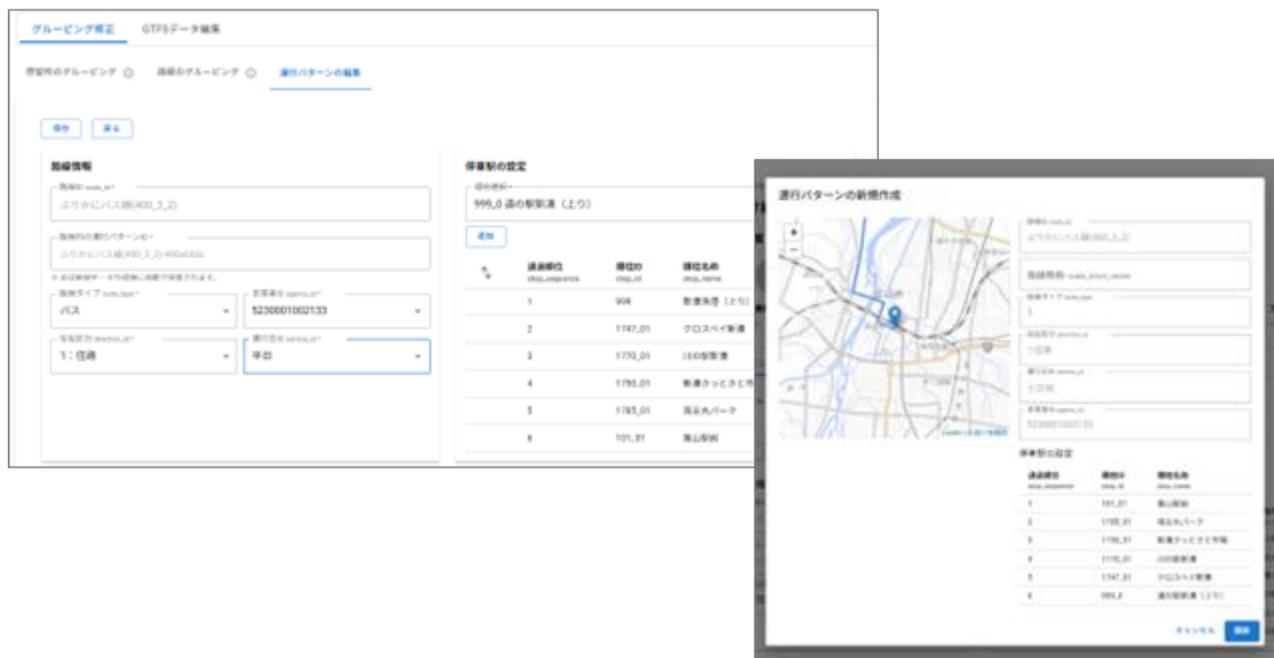


上段：運行パターンの編集、上段右：運行パターンの編集詳細、経路形状の編集

下段左：運行パターンの新規作成、下段右：運行パターンの削除

図 2-73 シナリオ編集（ルート編集）

- 本画面の概要
 - 指定した路線（経路）の情報を編集する
 - 運行パターンを追加・削除、編集する
 - 指定した運行パターンの経路形状を編集する
 - ◇ 新規の経路形状を作成及び任意の経路を削除する
 - ◇ 新規の経路形状の作成にあたっては、既存の経路形状を参照して停留所の追加削除を可能とする
 - ◇ 新規の経路形状の作成にあたっては、標柱の位置から経路形状の自動生成を可能とする
- 本画面から利用する機能
 - 【FN002】 GTFS 編集機能



上段：運行パターンの編集、下段左：停留所の設定、下段右：便の追加
 図 2-74 シナリオ編集（便の新規作成・変更）

- 本画面の概要
 - 標柱・停留所を新規作成・変更し、それに伴う路線（経路）を編集する
 - 便の編集
 - ◇ 選択したルートに含まれる便の新規作成を行う
- 本画面から利用する機能
 - 【FN002】GTFS 編集機能



上段：運行パターンの編集、下段：運行パターンの削除

図 2-75 シナリオ編集（便の削除）

- 本画面の概要
 - 標柱・停留所を削除する
 - 便の編集
 - ◇ 選択したルートに含まれる便を削除する
- 本画面から利用する機能
 - 【FN002】GTFS 編集機能



図 2-76 シナリオ編集（停留所のグルーピング）

- 本画面の概要
 - 複数の標柱を一つの停留所にグループ化する
- 本画面から利用する機能
 - 【FN002】 GTFB 編集機能



図 2-77 シナリオ編集（路線のグルーピング）

- 本画面の概要
 - 複数の路線を一つの路線グループにまとめる

- 本画面から利用する機能
 - 【FN002】 GTFS 編集機能



図 2-78 シナリオ編集（フィード基本情報の編集）

- 本画面の概要
 - GTFS のフィード基本情報を編集する
- 本画面から利用する機能
 - 【FN002】 GTFS 編集機能

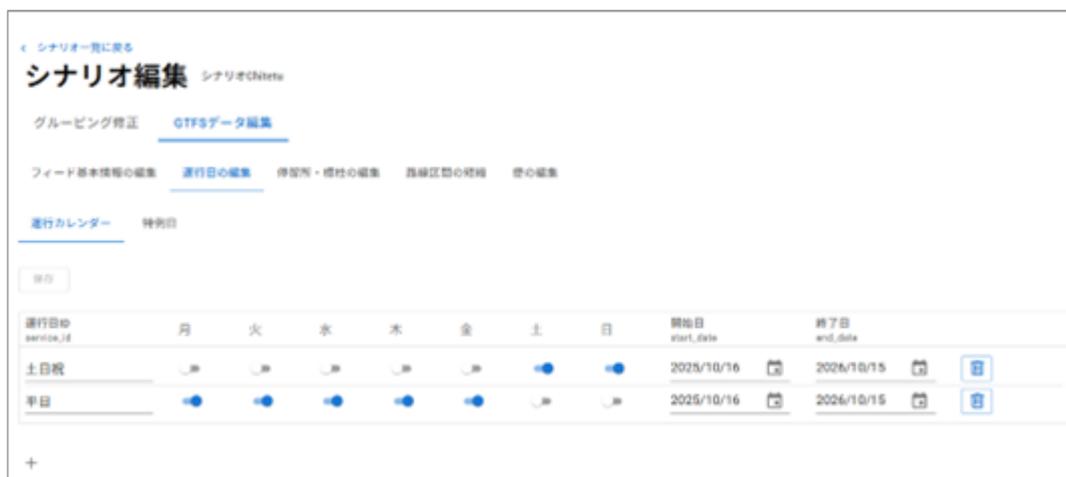


図 2-79 シナリオ編集（運行日の編集）

- 本画面の概要
 - 運行日を編集する
- 本画面から利用する機能
 - 【FN002】 GTFS 編集機能

【UI017】シナリオ詳細



図 2-80 シナリオ詳細

- 本画面の概要
 - シナリオ一覧で選択した GTFS データの詳細を確認する
- 本画面から利用する機能
 - 【FN002】GTFS 編集機能

【UI018】 運行記録（日報）出力

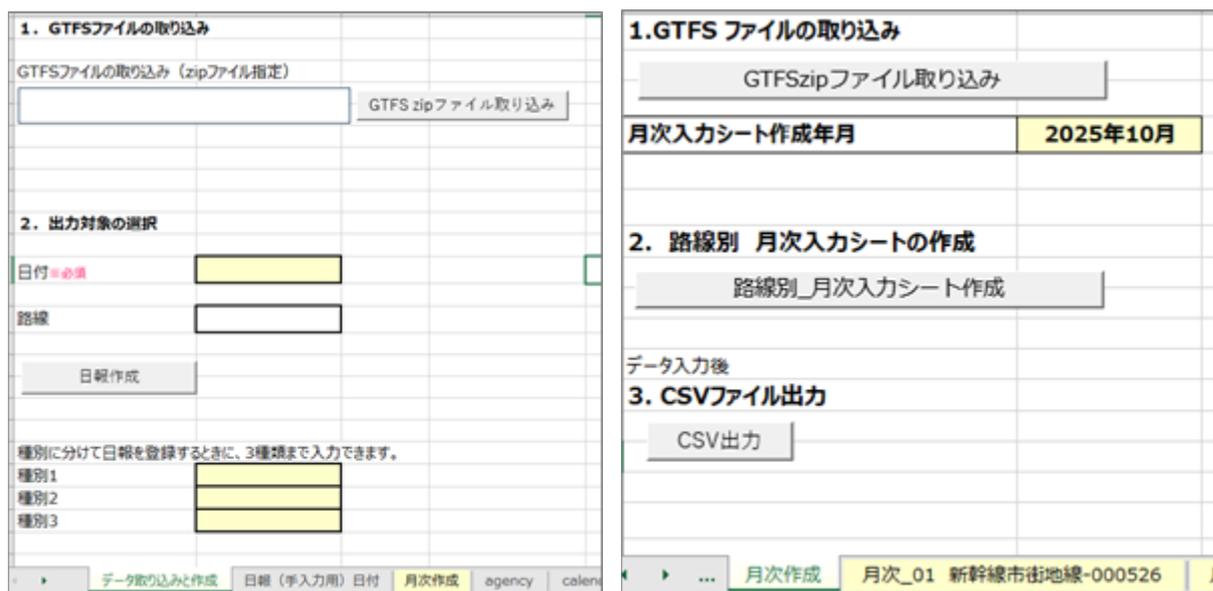


図 2-81 運行記録（日報）出力

- 本画面の概要
 - GTFS データをインポートする
 - 設定した日付・路線の運行記録（日報）手入力シートを出力する
 - 設定した年月の運行記録（日報）登録シートを出力する
 - 運行記録（日報）登録シートに転記された情報から乗降集計データを CSV 形式で出力する
- 本画面から利用する機能
 - 【FN015】 運行記録（日報）出力機能

【UI019】アカウント管理

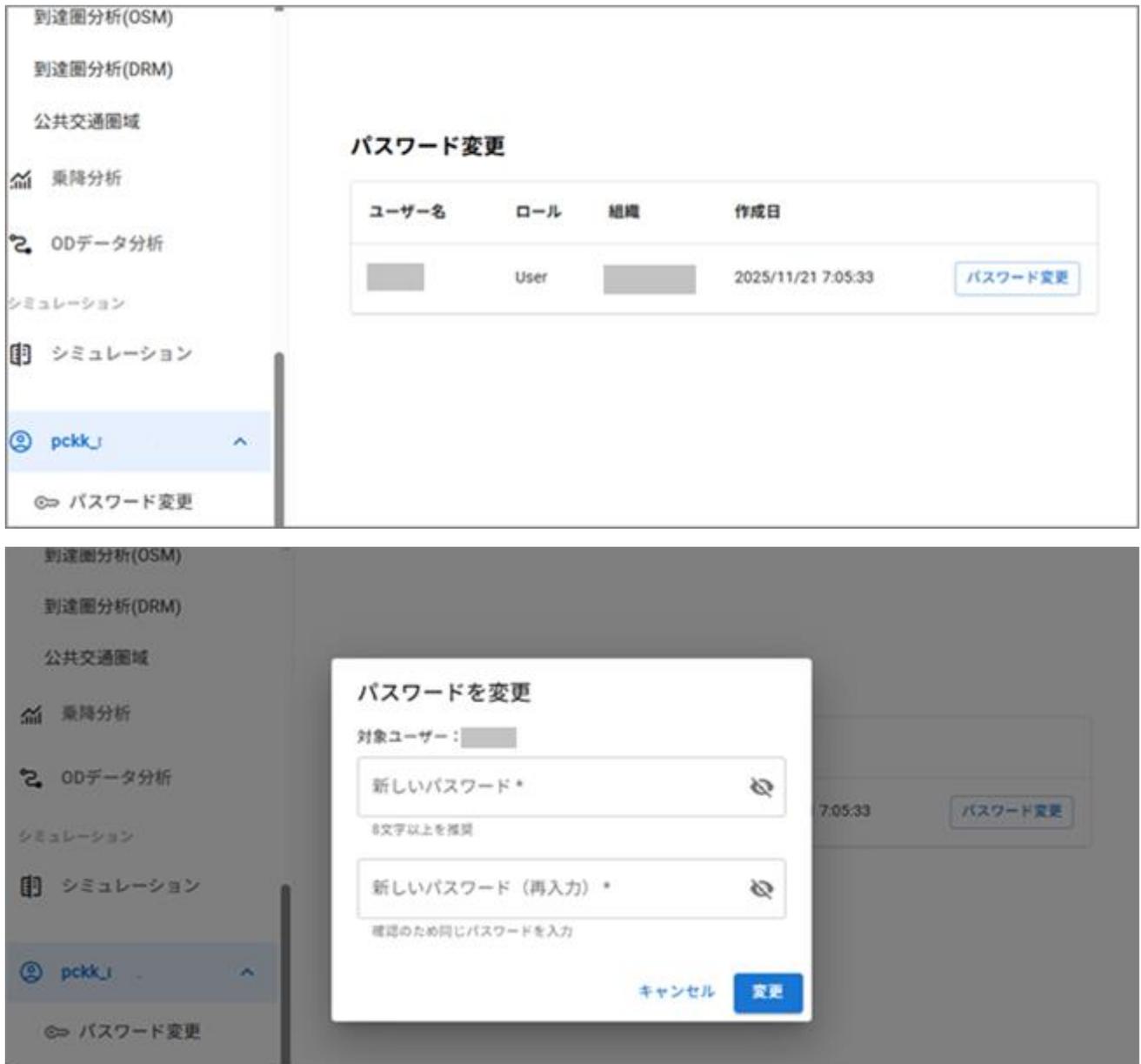


図 2-82 アカウント管理

- 本画面の概要
 - ユーザー名/パスワード等、アカウント情報の管理を行う
- 本画面から利用する機能
 - 【FN009】ログアウト

3. 開発するシステム：非機能要件（NF）

3-1. 非機能要件一覧

表 3-1 非機能要件一覧

カテゴリ	ID	非機能項目	要件詳細
可用性	NF002 NFR002	OSS 利用	● 本システムは可能な限り OSS ライブラリ等を用いて構築する
	NF003 NFR003	OSS 提供	● 開発成果は可能な限り OSS として提供する
	NF007	システムの連続稼働時間	● 分析及びシミュレーションを実施するため、4 時間以上稼働する
	NF008	安定動作時間	● ワークショップで実業務と同等に使用するため、2 時間以上の安定動作時間を確保する
	NF009	システム復旧時間	● 業務停止が伴う障害が発生した際に、3 営業日以内の復旧を担保可能とする。
	NF010	データの保管期間	● ワークショップ終了後の試行期間である 1 週間のデータ保管期間を担保する
性能・拡張性	NF005 NFR005	スケーラビリティと拡張性	● システムは将来のデータ量の増加や新機能の追加に対応できるように設計する
	NF011	データの読込速度	● インプットとなる GTFS データの平均読込時間を 3 分以内とする
	NF012	データのアップロード速度	● データ (10MB 程度を想定) のアップロードを、一般的な 5G 回線で 1 分以内に行う
	NF013	処理実行速度	● 一般的な 5G 回線で、分析を実行から結果表示までを、3 分以内に行う
運用・保守性	NF001 NFR001	クラウド環境の構築	● システムはクラウド環境で構築し、セキュリティ対策及び認証システムを備える
	NF006 NFR006	データのバックアップ	● データの定期的なバックアップを取得する
	NF014	セキュリティ	● 対象となるデータ及びシステム保守のため、IPA の安全な Web サイトの作り方に従う
	NF015	セキュリティ	● HTTPS 通信を使用する
	NF016	認証	● 利用ユーザーを特定するために、ユーザーID、8 桁以上のパスワードを設定したアカウントを発行する
	NF017	認証	● ネットワークを経由して送信するパスワード等については第三者に漏えいしないよう暗号化を実施する

ユーザビリティ	NF004 NFR004	UI/UX への配慮	<ul style="list-style-type: none"> ● 本システムの主たるユーザーを特定した上で、簡素かつ明快な UI/UX を設計する ● デザインルールを統一化し、分かりやすい UI とする
---------	-----------------	------------	------------------------------------------------------------------------------------------------------------------------------------

3-2. 非機能要件の詳細

【NF001】クラウド環境の構築

- 本非機能要件の概要
 - システムはクラウド環境で構築し、セキュリティ対策及び認証システムを備える。
- 設定理由
 - システムをクラウド環境上で構築し、クラウドサービスで提供されているセキュリティ対策や認証システムを活用することで、実装作業の省力化を行うため。

【NF002】OSS 利用

- 本非機能要件の概要
 - 本システムは可能な限り OSS ライブラリ等を用いて構築する。
- 設定理由
 - 既存ライブラリを構築することで、開発コストの削減と迅速な機能の実装が可能となるため。
 - OSS によりソースコードが公開されているため、透明性とセキュリティの検証が可能であるため。
 - 無償で利用・改変・再配布が許可されることで、柔軟な開発が実現できるため。

【NF003】OSS 提供

- 本非機能要件の概要
 - 開発成果は可能な限り OSS として提供する。
- 設定理由
 - OSS として提供することにより、外部の開発者や企業による利用・改善が促進されるため。

【NF004】UI/UX への配慮

- 本非機能要件の概要
 - 本システムの主たるユーザーを特定した上で、簡素かつ明快な UI/UX を設計する。
 - デザインルールを統一化し、分かりやすい UI とする。
- 設定理由
 - 主たるユーザーを特定したうえで、そのユーザーが使用しやすいよう簡素かつ明快な UI/UX を設計するため。
 - ユーザーが直感的かつ効率的に操作できるよう、学習コストや操作ミスを低減するため。

【NF005】スケーラビリティと拡張性

- 本非機能要件の概要
 - システムは将来のデータ量の増加や新機能の追加に対応できるように設計する。
- 設定理由
 - 将来的なユーザー数や機能追加に柔軟に対応できるため、再設計や大規模改修のコストを抑えつつ、継続的な成長が可能になるため。

【NF006】データのバックアップ

- 本非機能要件の概要
 - データの定期的なバックアップを取得する。
- 設定理由
 - システム障害や誤操作によるデータ損失に備えて、情報の安全性を確保するため、リスク管理の観点から必要となるため。

【NF007】システムの連続稼働時間

- 本非機能要件の概要
 - 分析及びシミュレーションを実施するため、4時間以上稼働する。
- 設定理由
 - 今回の実証実験で開発するシステムを用いて地域公共交通計画策定における一連の分析作業を実施するため、実証実験の実施時間である4時間の連続稼働時間を確保する。

【NF008】安定動作時間

- 本非機能要件の概要
 - ワークショップで実業務と同等に使用するため、2時間以上の安定動作時間を確保する
- 設定理由
 - 実証調査で実施するワークショップの実施時間中の安定動作を実現するため。

【NF009】システム復旧時間

- 本非機能要件の概要
 - 業務停止が伴う障害が発生した際に、3営業日以内の復旧を担保可能とする。
- 設定理由
 - 実効性あるコストでリスクを低減できるバランス点であるため。

【NF010】データの保管期間

- 本非機能要件の概要
 - ワークショップ終了後の試行期間である1週間のデータ保管期間を担保する。
- 設定理由
 - 実証調査期間中、最低限必要な保管期間を保持するため。

【NF011】データの読込速度

- 本非機能要件の概要
 - インプットとなるGTFSデータの平均読込時間を3分以内とする。
- 設定理由
 - 一般的なサーバー構成で実現可能な範囲、かつ、ユーザーの作業効率を確保し、インフラコストとのバランスを考慮した妥当な数値であるため。

【NF012】データのアップロード速度

- 本非機能要件の概要
 - データ（10MB程度を想定）のアップロードを、一般的な5G回線で1分以内に行う。
- 設定理由
 - 一般的なサーバー構成で実現可能な範囲かつユーザーの作業効率を確保し、インフラコストとのバランスを考慮した妥当な数値であるため。

【NF013】処理実行速度

- 本非機能要件の概要
 - 一般的な5G回線で、分析を実行から結果表示までを、3分以内に行う。
- 設定理由
 - 一般的な回線利用で実現可能な範囲かつユーザーの作業効率を確保し、インフラコストとのバランスを考慮した妥当な数値であるため。

【NF014】セキュリティ

- 本非機能要件の概要
 - 対象となるデータ及びシステム保守のため、IPAの「安全なウェブサイトの作り方」に従う。
- 設定理由
 - IPAによる公的ガイドラインで求められる要件をセキュリティ要件とするため。

【NF015】セキュリティ

- 本非機能要件の概要
 - HTTPS通信を使用する。
- 設定理由
 - TLS（HTTPS）通信を必須とし、通信の機密性・完全性を担保し、モダンブラウザとの互換性やIPAガイドラインへ準拠するため。

【NF016】認証

- 本非機能要件の概要
 - 利用ユーザーを特定するために、ユーザーID、8桁以上のパスワードを設定したアカウントを発行する。
- 設定理由
 - 不正ログイン・情報漏えいリスクを低減するため。

【NF017】認証

- 本非機能要件の概要
 - ネットワークを経由して送信するパスワード等については第三者に漏えいしないよう暗号化を実施する。

- 設定理由
 - TLS 等の暗号化通信を必須とすることで、盗聴・改ざん・なりすまし等の攻撃リスクを排除するため。

4. 実証調査に利用するデータ (DT)

4-1. 実証調査に利用するデータ一覧

表 4-1 実証調査に利用するデータ一覧

※朱文字：本実証で変換・作成するデータ

ID	データ名称	データ形式	出所	データを利用する ID
DT001	GTFS データ	TXT	GTFS データリポジトリ	FN001 FN010
DT002	運行記録 (日報) (紙)	紙	実証調査の協力自治体	FN004
DT003	運行記録データ (Excel 等)	Excel 等	実証調査の協力自治体	FN004
DT004	乗降集計データ	CSV	—	FN004
DT005	乗降実績データ (標準未準拠)	CSV	実証調査の協力事業者	FN004
DT006	駅・バス停間 OD データ	CSV	乗降集計データ (CSV)、乗降実績データから作成する	FN004
DT007	乗降実績データ	CSV	乗降実績データ標準仕様書 (鉄道・バス) に準拠して作成されたデータ	FN004

4-2. 実証調査に利用するデータの詳細

実証調査に利用するデータの詳細を記す。なお、本業務において変換・生成を行うデータを**朱文字**で示す。

【DT001】GTFS データ

- 本データの概要
 - 公共交通機関のスケジュールや地理情報を標準化された形式で提供するためのフォーマット
- サンプル・イメージ
 - データ定義

表 4-2 GTFS データのデータ定義

情報	ファイル名	必須任意別	システムで利用
事業者情報	agency.txt	必須	—
事業者追加情報	agency_jp.txt	任意	—
提供情報	feed_info.txt	必須	○
停留所・標柱情報	stops.txt	必須	○
経路情報	routes.txt	必須	○
便情報	trips.txt	必須	○
営業所情報	office_jp.txt	任意	—
停車パターン情報	pattern_jp.txt	任意	—
通過時刻情報	stop_times.txt	必須	○
運行区分情報	calendar.txt	条件付き必須	○
運行日情報	calendar_dates.txt	条件付き必須	○
運賃属性情報	fare_attributes.txt	必須	—
運賃定義情報	fare_rules.txt	必須	—
描画情報	shapes.txt	任意	○
運行間隔情報	frequencies.txt	任意	—
乗換情報	transfers.txt	任意	—
翻訳情報	translations.txt	必須	—

- 本データの形式
 - ZIP 形式、CSV 形式
- 出所
 - GTFS データリポジトリより取得する (<https://gtfs-data.jp/>)
 - 交通事業者の HP より取得する (<https://www.kotoden.co.jp/publichtm/gtfs/index.html>)

【DT002】 運行記録（日報）（紙）

- 本データの概要
 - 公共交通機関の乗客の乗降実績を記録した日報形式の紙データ
 - 提供元により様式は異なる
- サンプル・イメージ
 - ◇ バスの「どの便」の「どの停留所」で「何人が乗降した」が記載されている
 - ◇ バス運転手がある場で手書きにて記載する
 - ◇ データ定義の詳細を表 4-3 データ定義のサンプルに示す

表 4-3 データ定義のサンプル

NO	日本語名称	項目長	必須	書式・選択肢など
1	日付	-	○	文字(YYYYMMDD,8桁)
2	天候	-	○	晴、曇、雨、霧、雪
3	路線名	-	○	○○線
4	便名	-	○	1便、2便、3便、朝便、夕便
5	行先	-	○	○○線（朝1便のみ）△行き
6	発停留所	-	○	文字
7	経由	-	○	文字
8	着停留所	-	○	文字
9	発車時刻	-	○	文字(HHMM,4桁)
10	到着時刻	-	○	文字(HHMM,4桁)
11	停留所名	-	○	文字
12	乗車人数	-	○	数値
13	降車人数	-	○	数値

【DT003】 運行記録データ（Excel等）

- 本データの概要
 - 公共交通機関の乗客の乗降実績を記録したデータ
 - 提供元により様式は異なる
- サンプル・イメージ
 - データ定義
 - ◇ バスの、「どの便」の「どの停留所」で「何人が乗降した」を記載したデータ
 - ◇ データ定義詳細を表 4-4 乗降実績定義一覧、表 4-5 乗降実績定義のデータ定義のサンプルに示す

表 4-4 乗降実績定義一覧

NO	自治体名	路線名	期間
1	文字（自治体名）	〇〇線	文字（元号 YMM,6 桁）～文字（元号 YMM,6 桁）

表 4-5 乗降実績定義のデータ定義のサンプル

NO	日本語名称	項目長	必須	書式・選択肢など
1	路線名	-	○	〇〇線
2	行先		○	〇〇線（朝 1 便のみ）△行き
3	日付	-	○	文字(YYYYMMDD,8 桁)
4	停留所名	-	○	文字
5	乗車人数	-		数値
6	降車人数	-		数値
7	天候	-		晴、曇、雨、霧、雪
8	便名	-		1 便、2 便、3 便、朝便、夕便
9	発停留所	-	○	文字
10	経由	-	○	文字
11	着停留所	-	○	文字

- 本データの形式
 - Excel 形式
- 出所
 - 自治体より Excel データを取得する

【DT004】乗降集計データ

- 本データの概要
 - 公共交通機関の乗客の乗降実績を記録した電子データ
 - 各便の停留所の停留所別乗降人数のデータ
- サンプル・イメージ
 - データ定義

表 4-6 乗降集計データのデータ定義

属性名	説明	属性の型
date	日付	文字 (YYYYMMDD,8桁)
agency_id	事業者 ID	文字
route_id	路線 ID	文字
trip_id	便 ID	文字
stop_id	停留所 ID	文字
stop_sequence	通過順位	数値
count_geton	乗車人数	数値
count_getoff	降車人数	数値

- 本データの形式
 - CSV 形式
- 出所
 - 各自治体より CSV データを取得する

【DT005】乗降実績データ（標準未準拠）

- 本データの概要
 - 交通系 IC カードの利用実績を 1 件ごとに記録した個別の履歴データ
 - 「モビリティ・データ標準化プロジェクト」にて策定した乗降実績データ標準仕様書（鉄道・バス）の仕様に未準拠のデータとして整理
- サンプル・イメージ
 - データ定義

表 4-7 乗降実績データ（標準未準拠）のデータ定義

チケット情報(ticket_info) 乗降実績(ridership_records)						
No	フィールド名	物理名	必須			データ型
			鉄道	バス (IC)	バス (現金)	
1	乗降実績 ID	ridership_record_id	○	○	○	Integer
2	IC カード識別コード	ic_card_agency_identification_code		○	○	String
3	IC カード発行事業者コード	ic_card_issuer_code				String

4	ICカード発行事業者名	ic_card_issuer_name				String
5	ICカード機能区分	ic_card_feature_type				Enum
6	券種エリアコード	ticket_type_area_code				String
7	券種区分	ticket_type				Enum
8	券種名	ticket_type_name				String
9	券有効開始日	ticket_valid_start_date				Date
10	券有効終了日	ticket_valid_end_date				Date
11	交通モードコード	transportation_mode_code				String
12	ICカード利用明細ID	ic_card_usage_detail_id				Integer
13	運行事業者コード	operating_agency_code				String
14	運行事業者名	operating_agency_name				String
15	営業所コード	serviced_office_code		○	○	String
16	営業所名	serviced_office_name				String
17	系統ID	route_pattern_id				String
18	系統番号	route_pattern_number		○	○	String
19	路線名	service_line_name				String
20	経路名	route_name				String
21	便コード	trip_code				String
22	ダイヤ番号	timetable_number				String
23	車両番号	vehicle_number				String
24	処理区分	operation_type				Enum
25	処理詳細区分	operation_detail_type				Enum
26	乗車エリアコード	boarding_area_code				String
27	乗車停留所連番	boarding_station_sequence				Integer
28	乗車駅(停留所)コード	boarding_station_code	○	○	○	String
29	乗車駅(停留所)名	boarding_station_name				String
30	乗車日時	boarding_at				Date-time
31	乗車経路ID	Boarding_route_id				String
32	乗継エリアコードリスト	transfer_area_code_list				Array <String>
33	乗継駅(停留所)コードリスト	transfer_station_code_list				Array <String>
34	降車エリアコード	alighting_area_code				String
35	降車停留所連番	alighting_station_sequence				Integer
36	降車駅(停留所)コード	alighting_station_code	○	○	○	String
37	降車駅(停留所)名	alighting_station_name				String
38	降車日時	alighting_at				Date-time

39	精算日時	payment_at	○	○	○	Date-time
40	大人障がい者利用者数	adult_challenged_passenger_count				Integer
41	大人利用者数	adult_passenger_count		○	○	Integer
42	小児障がい者利用者数	child_challenged_passenger_count				Integer
43	小児利用者数	child_passenger_count		○	○	Integer
44	利用者分類区分	passenger_classification_type				Enum
45	支払い区分	Payment_type				Enum

- 本データの形式

CSV 形式

- 出所

- 交通事業者等よりデータを取得する

【DT006】 駅・バス停間 OD データ

- 本データの概要
 - 公共交通機関の出発地 (Origin) と目的地 (Destination) の組み合わせごとに、利用者数を集計したデータ
 - IC カード履歴等から生成される
- サンプル・イメージ
 - データ定義

表 4-8 No.1 の駅・バス停間 OD データのデータ定義

属性名	説明	属性の型
date	日付	文字 (YYYYMMDD,8 桁)
agency_id	事業者 ID	文字
route_id	路線 ID	文字
stopid_geton	乗車停留所 ID	文字
stopid_getoff	降車停留所 ID	文字
count	利用者数	数値

表 4-9 No.2 の駅・バス停間 OD データのデータ定義

No.	列名	データ型 / 長	備考
1	id	bigint	連番
2	year	integer	統計年度
3	key_code	character varying(20)	図形と集計データのリンクコード
4	hyosyo	integer	地域階層レベル
5	cityname	character varying(50)	市区町村名
6	name	character varying(50)	町丁・字等名
7	t001082001	integer	総数、年齢「不詳」含む
8	t001082017	integer	総数 15 歳未満
9	t001082018	integer	総数 15～64 歳
10	t001082019	integer	総数 65 歳以上
11	t001082020	integer	総数 75 歳以上

- 本データの形式
 - CSV 形式

- 出所
 - 交通事業者等よりデータを取得する

【DT007】乗降実績データ

- 本データの概要
 - 「乗降実績データ標準仕様書（鉄道・バス）」に準拠した項目
 - 参考：https://www.mlit.go.jp/commmmons/document/005/commmmons_doc_005-01_ver01.pdf
- サンプル・イメージ
 - データ定義

表 4-10 乗降実績データのデータ形式

No	フィールド名	物理名	必須			データ型
			鉄道	バス (IC)	バス (現金)	
1	IC カード識別コード	ic_card_agency_identification_code	○	○		String
2	IC カード発行事業者コード	ic_card_issuer_code				String
3	IC カード発行事業者名	ic_card_issuer_name				String
4	IC カード機能区分	ic_card_feature_type				Enum
5	券種エリアコード	ticket_type_area_code				String
6	券種区分	ticket_type				Enum
7	券種名	ticket_type_name				String
8	券有効開始日	ticket_valid_start_date				Date
9	券有効終了日	ticket_valid_end_date				Date
10	乗降実績 ID	ridership_record_id	○	○	○	Integer
11	交通モードコード	transportation_mode_code				String
12	IC カード利用明細 ID	ic_card_usage_detail_id				Integer
13	運行事業者コード	operating_agency_code				String
14	運行事業者名	operating_agency_name				String
15	営業所コード	serviced_office_code		○	○	String
16	営業所名	serviced_office_name				String
17	系統 ID	route_pattern_id				String
18	系統番号	route_pattern_number		○	○	String
19	路線名	service_line_name				String
20	経路 ID	route_id				String
21	経路名	route_name				String
22	便コード	trip_code				String
23	ダイヤ番号	timetable_number				String

24	車両番号	vehicle_number				String
25	処理区分	operation_type				Enum
26	処理詳細区分	operation_detail_type				Enum
27	乗車エリアコード	boarding_area_code				String
28	乗車停留所連番	boarding_station_sequence				Integer
29	乗車駅（停留所）コード	boarding_station_code	○	○	○	String
30	乗車駅（停留所）名	boarding_station_name				String
31	乗車日時	boarding_at				Date-time
32	乗継エリアコードリスト	transfer_area_code_list				Array <String>
33	乗継駅（停留所）コードリスト	transfer_station_code_list				Array <String>
34	降車エリアコード	alighting_area_code				String
35	降車停留所連番	alighting_station_sequence				Integer
36	降車駅（停留所）コード	alighting_station_code	○	○	○	String
37	降車駅（停留所）名	alighting_station_name				String
38	降車日時	alighting_at				Date-time
39	精算日時	payment_at	○	○	○	Date-time
40	大人障がい者利用者数	adult_challenged_passenger_count				Integer
41	大人利用者数	adult_passenger_count		○	○	Integer
42	小児障がい者利用者数	child_challenged_passenger_count				Integer
43	小児利用者数	child_passenger_count		○	○	Integer
44	利用者分類区分	passenger_classification_type				Enum

- 本データの形式
 - CSV 形式
- 出所
 - 交通事業者等よりデータを取得する

用語集

用語	定義・説明
API	データや機能を外部とやり取りする仕組み。
DRM (デジタル道路地図)	一般財団法人日本デジタル道路地図協会が有償で提供するデジタル形式の道路地図。
GIS	地理情報をキャプチャ、保存、操作、分析、管理、提示するためのシステム。
GTFS Schedule (静的)	公共交通の時刻表等の静的情報を提供する標準フォーマット。stops.txt、routes.txt、trips.txt、stop_times.txt、calendar.txt、calendar_dates.txt、shapes.txt、feed_info.txtなどで構成。
GTFS Realtime (動的)	運行状況などリアルタイム情報を提供する仕様。本業務では対象外。
GTFS-JP 拡張	GTFSの日本向け拡張のファイル群(*_jp.txt)。
OD 輸送量	出発地から目的地まで (Origin-Destination (起点-終点)) の輸送量。
OSM (OpenStreetMap)	世界の無料編集可能な地図を作成する共同プロジェクト。道路情報等も提供している。
OSS (オープンソース)	誰でも確認、変更、強化できるソースコードを持つソフトウェア。
運行頻度	特定の交通手段 (バスや電車等) が一定の時間内に運行される回数を示す指標。
運行パターン	運行回数、間隔、行き先表示など、交通サービスやその路線・系統がもつ運行の種類。
駅・バス停間 OD データ	出発地 (Origin) と目的地 (Destination) の組み合わせごとに、利用者数を集計したデータ。 本システムに取り込むための CSV 形式のデータ。
公共交通圏域	公共交通を使って移動をスムーズに行えるよう設定されたサービス提供範囲。
交通事業者	交通サービスを提供する事業者。
交通分担率	バスや電車等の各交通手段の利用割合。
サービスレベル	提供されるサービスの品質。
施設データ	病院や学校など、交通サービスが運行するエリア内の施設。
実証調査	本業務で実施する地域公共交通計画策定支援ツール「LINKS Mobilys」や、モビリティ・データの有用性を検証するための調査。
渋滞緩和効果	交通渋滞を減少させるための施策や対策の効果。

乗降実績データ	主に IC カードを利用したチケットの仕組みにより、IC カードシステム及び運賃箱システムから出力される鉄道及びバスの乗降実績を記録したデータ。 「モビリティ・データ標準化プロジェクト」にて策定した乗降実績データ標準仕様書（鉄道・バス）の仕様に沿ったデータ。
乗降実績データ（標準未準拠）	交通系 IC カードの利用 1 件ごとに記録される個別の履歴データ。乗降日時、乗車地点、降車地点、利用路線、運賃などの情報を含む。 「モビリティ・データ標準化プロジェクト」にて策定した乗降実績データ標準仕様書（鉄道・バス）の仕様に未準拠のデータ。
乗降集計データ	各便の停留所の停留所別乗車人数・降車人数のデータ。 本システムに取り込むための CSV 形式のデータ。
スケーラビリティ	システムやネットワークが拡張される能力。
断面輸送量	バスや電車の特定の地点における輸送量。
地域公共交通計画策定支援ツール「LINKS Mobilys」	本業務で作成する地域公共交通計画の策定を支援するためのツール。
停留所	バスが停車する場所。複数の標柱をまとめたバス停のまとめり。
到達圏分析	特定の地点から到達可能な範囲を分析する手法。
地域公共交通計画	特定の地域内の公共交通機関の計画。
標柱	バスを乗り降りする位置を示すポール。
モビリティ・データ	乗降実績や駅・バス停間 OD データ等の、人や物の移動に関連するデータ。
路線	バスや電車等の交通機関が通る経路に関するデータ。
路線グループ	複数の路線を名称などにより 1 つにまとめたグループ。



Project LINKS



COMmmONS
By MLIT

地域公共交通計画策定支援ツール「LINKS Mobilys」
システム設計書

Ver1.0

発行日: 2026年3月

委託者: 国土交通省 総合政策局

情報政策課、モビリティサービス推進課

受託者: パシフィックコンサルタンツ株式会社