

施設送迎共同管理システム システム設計書

国土交通省 総合政策局 モビリティサービス推進課

No.004-01



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.....	8
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.....	59
2-2. システムコンポーネント（CO）.....	60
2-2-1. システムコンポーネント図.....	60
2-2-2. システムコンポーネント一覧.....	61
2-3. ハードウェア（HW）.....	62
2-3-1. ハードウェアアーキテクチャ.....	62
2-3-2. ハードウェア一覧.....	63
2-3-3. ハードウェアの詳細.....	64
2-4. データインターフェース（IF）.....	67
2-4-1. データアーキテクチャ.....	67
2-4-2. データインターフェース一覧.....	75
2-4-3. データインターフェースの詳細.....	76
2-5. ユーザーインターフェース（UI）.....	111
2-5-1. 画面遷移図.....	111
2-5-2. ユーザーインターフェース一覧.....	113
2-5-3. ユーザーインターフェースの詳細.....	116
3. 開発するシステム：非機能要件（NF）.....	148
3-1. 機能要件一覧.....	148
3-2. 非機能要件の詳細.....	149
4. 実証調査に利用するデータ（DT）.....	153
4-1. 実証調査に利用するデータ一覧.....	153
4-2. 実証調査に利用するデータの詳細.....	154
5. 用語集.....	158

1. 開発スコープ

1-1. 概要

複数施設が保有する送迎車両の共同運用と、配車・稼働の一元管理を実現する「施設共同送迎管理システム」（以下「本システム」）を開発する。

本システムでは、福祉・教育・観光の異なる分野にまたがる送迎ニーズと車両・運転手リソースを統合的に管理することにより、地域の移動手段の効率化・最適化を目指す。開発に先立ち、施設間での送迎委託・受託の管理機構、利用者プロフィールに基づく乗車制約への対応、既存業務フローとの整合性を担保した送迎予定作成、システムと個人の体験に基づく最適ルーティング作成及び実績集計機能の業務要件を整理し、対象ユーザー（施設管理者）の IT リテラシーに応じた UX で実装する。

管理者アプリでは、複数施設の送迎予定作成・ルーティング作成と委託・受託管理、利用者・運転手・車両・地点などのマスタ登録、実績集計とレポート出力機能を提供する。また、OSS としての公開を前提とし、地域内の小規模事業者でも導入可能な軽量インフラ構成を採用する。具体的には、バックエンドは AWS EC2 上の Docker 環境に構築し、データベースはマルチテナント対応の構成とする。フロントエンドには React.js を、マップ機能には Google Maps Platform (Directions API, Distance Matrix API 等) を利用し、ナビゲーション・地点登録・送迎ルート表示を実現する。

利用者アプリは、利用者本人又は家族が送迎予定と車両現在位置を確認できる機能を提供する。対象は福祉・教育・観光等の各分野全体であり、送迎リソースは施設間・施設種間で共有する前提である。本システムは、分野や運用主体ごとに異なる申込・配車のワークフローを実施できるようにする。

以下は本実証で検証する代表シナリオの例であり、運用範囲を限定するものではない。

- 福祉分野（例）：施設オペレーターが事前の通所予定を基に計画を作成し、必要に応じて他施設へ委託
- 教育分野（例）：習い事事業者が保護者からの依頼を受け、共同送迎担当事業者へ委託。委託結果を習い事事業者・保護者が確認
- 観光分野（例）：旅館オペレーターが観光客の申込を受け、共同送迎担当事業者へ委託し、共同送迎担当者が計画作成（観光協会が計画作成を代行する場合もあり）。委託結果を旅館オペレーターが確認し、共同配車で運行

システムの主な処理には、送迎予定の作成、他施設への送迎委託依頼の通知・承認、送迎ルート作成、送迎表の出力、利用者の乗車・降車記録の管理が含まれる。施設間の送迎共用では、送迎者の身体条件（例：介助が必要、杖あり等）や申し送り事項（例：鍵の位置、電気の消灯）などの非構造情報も考慮した上で、車両や運転手との適合性を評価し、適切な割当てを行う機構を実装する。なお、ドライバーアプリについては既存ソリューションを活用し、本開発対象外とする。

本システムは、利用者アプリ・管理者アプリ・共通バックエンドにより構成され、将来的な API 連携（例：旅館予約サイトとの送迎連携）やデータ外部出力（例：介護保険申請用の CSV）の拡張も視野に入れた設計とする。システム全体の開発においては、Ruby、TypeScript、MySQL を主要技術として採用し、初

施設送迎共同管理システム システム設計書

期導入フェーズでは福祉・観光・教育の各現場におけるユースケースでの実証評価を行い、要件適合性と運用現場での利便性を検証する予定である。

2. システムシーケンス図

1 送迎計画作成

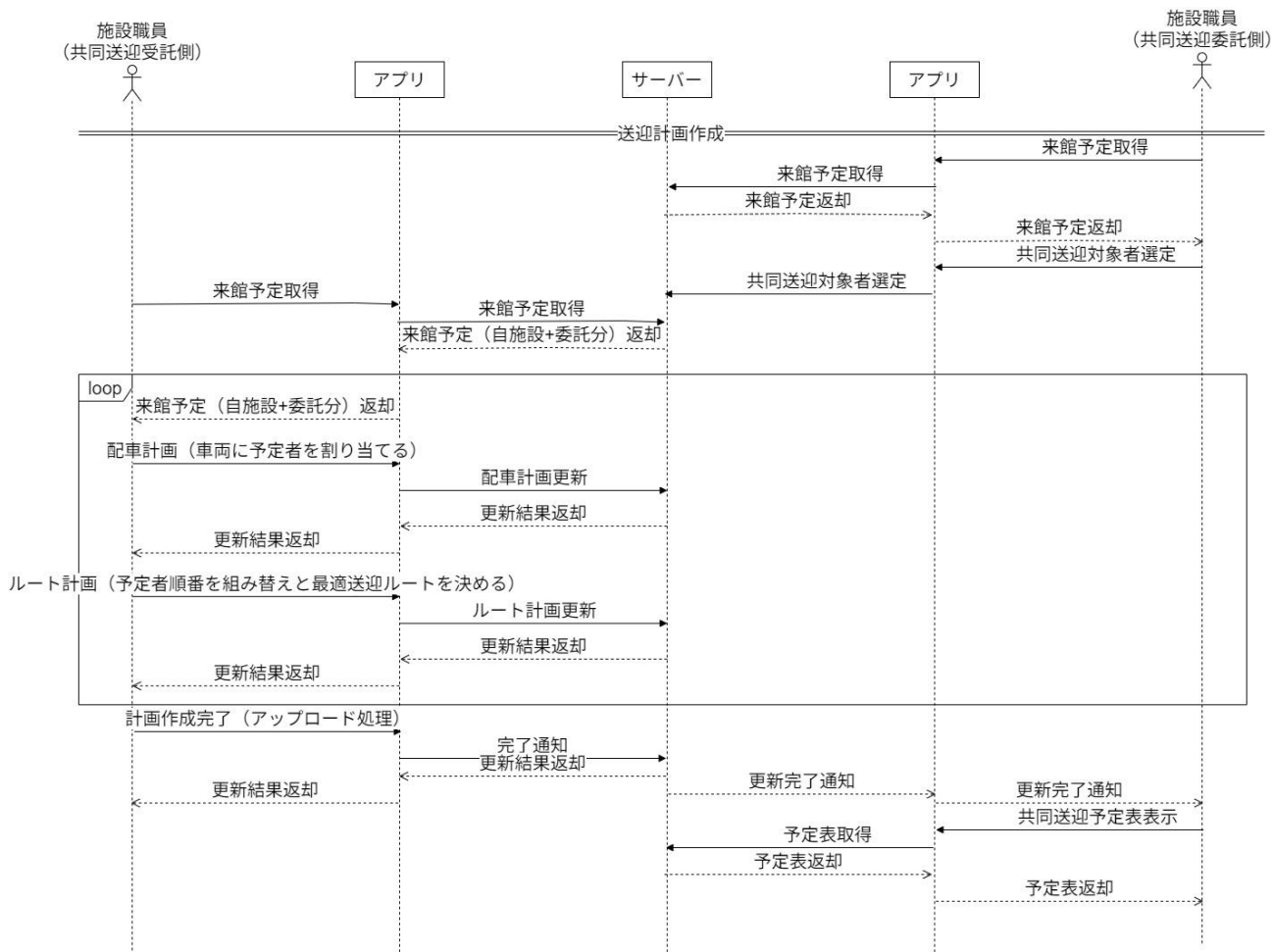


図 1-2 送迎計画作成

2 送迎実施

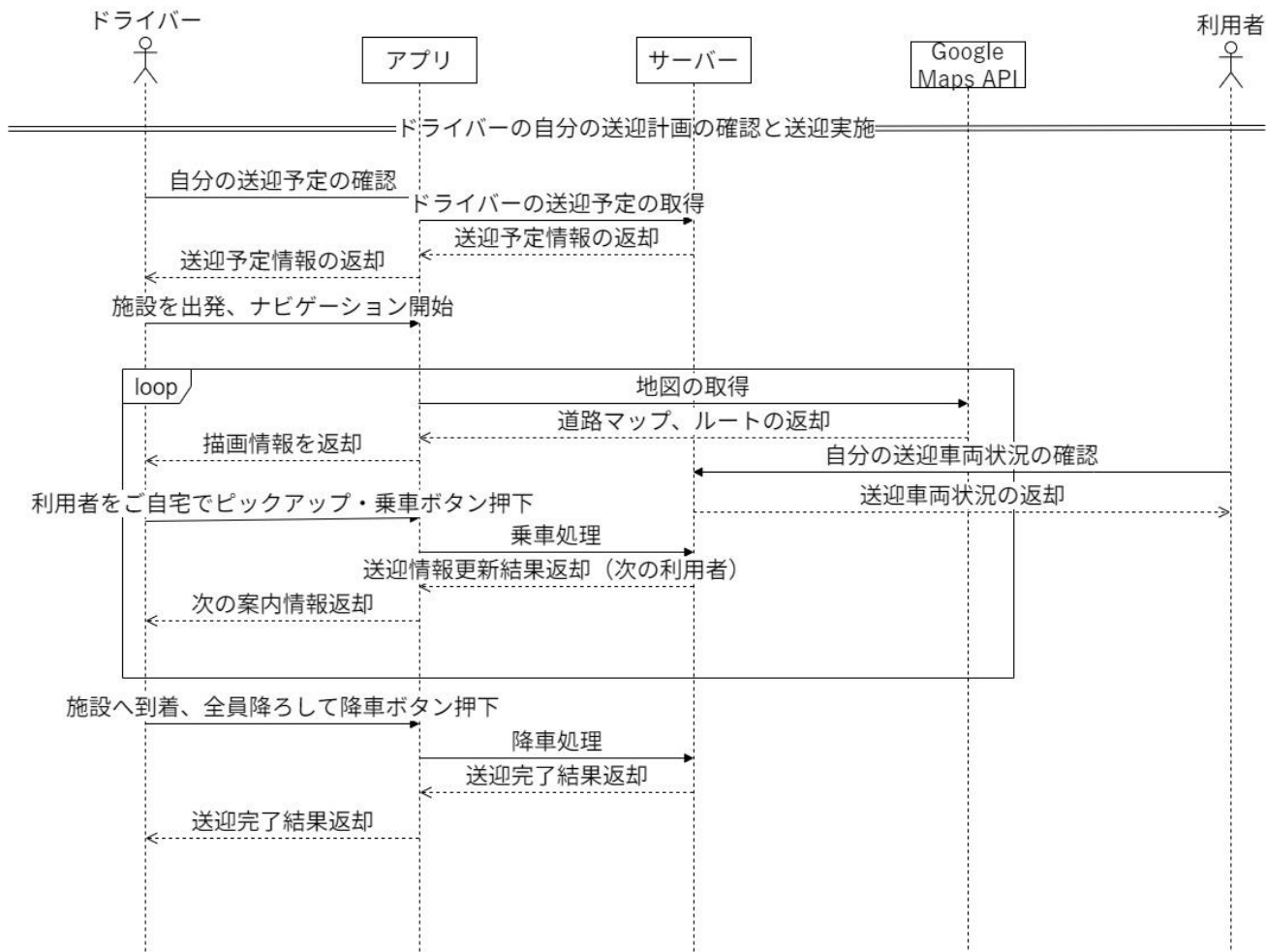
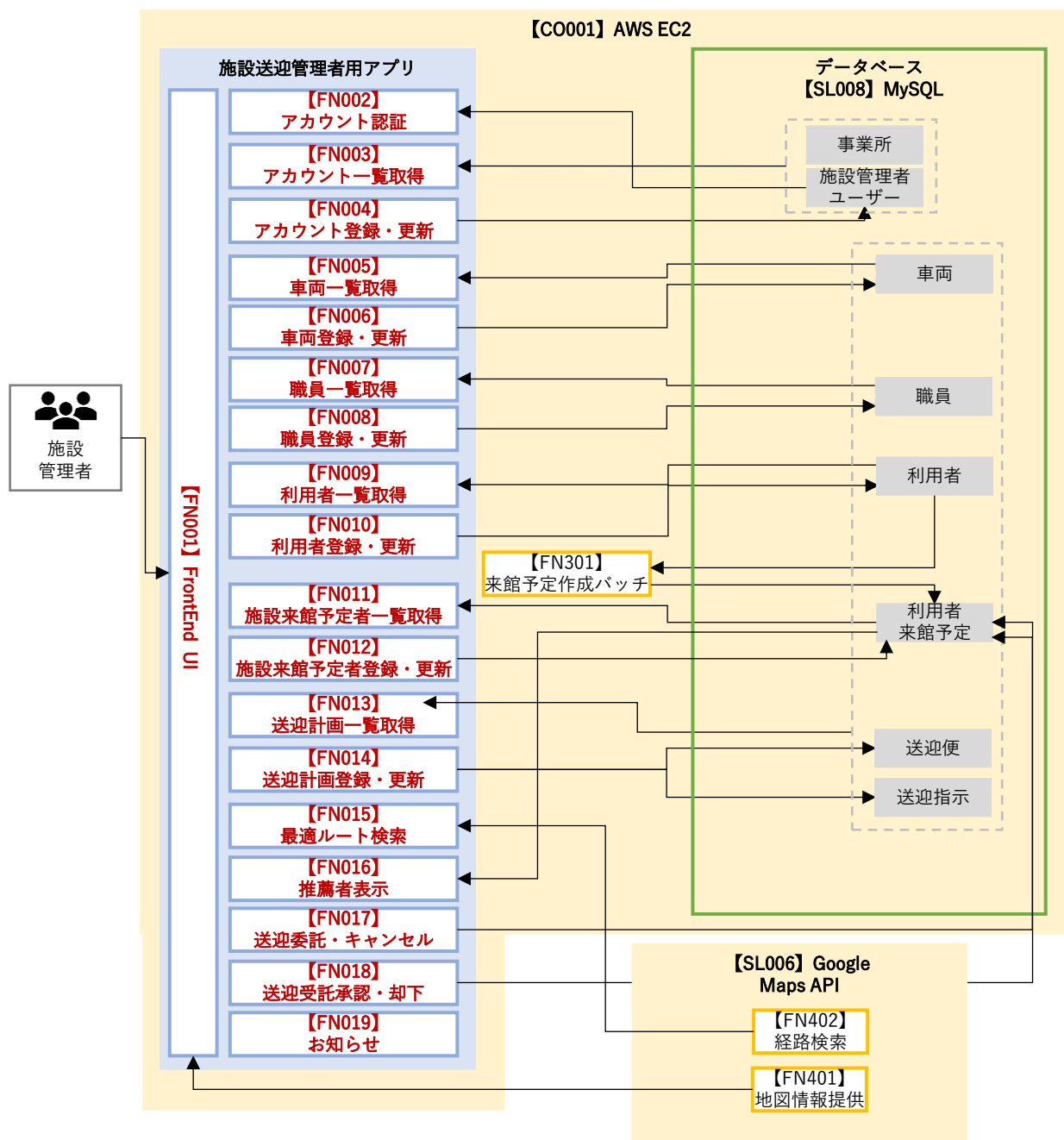


図 1-3 送迎実施

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）

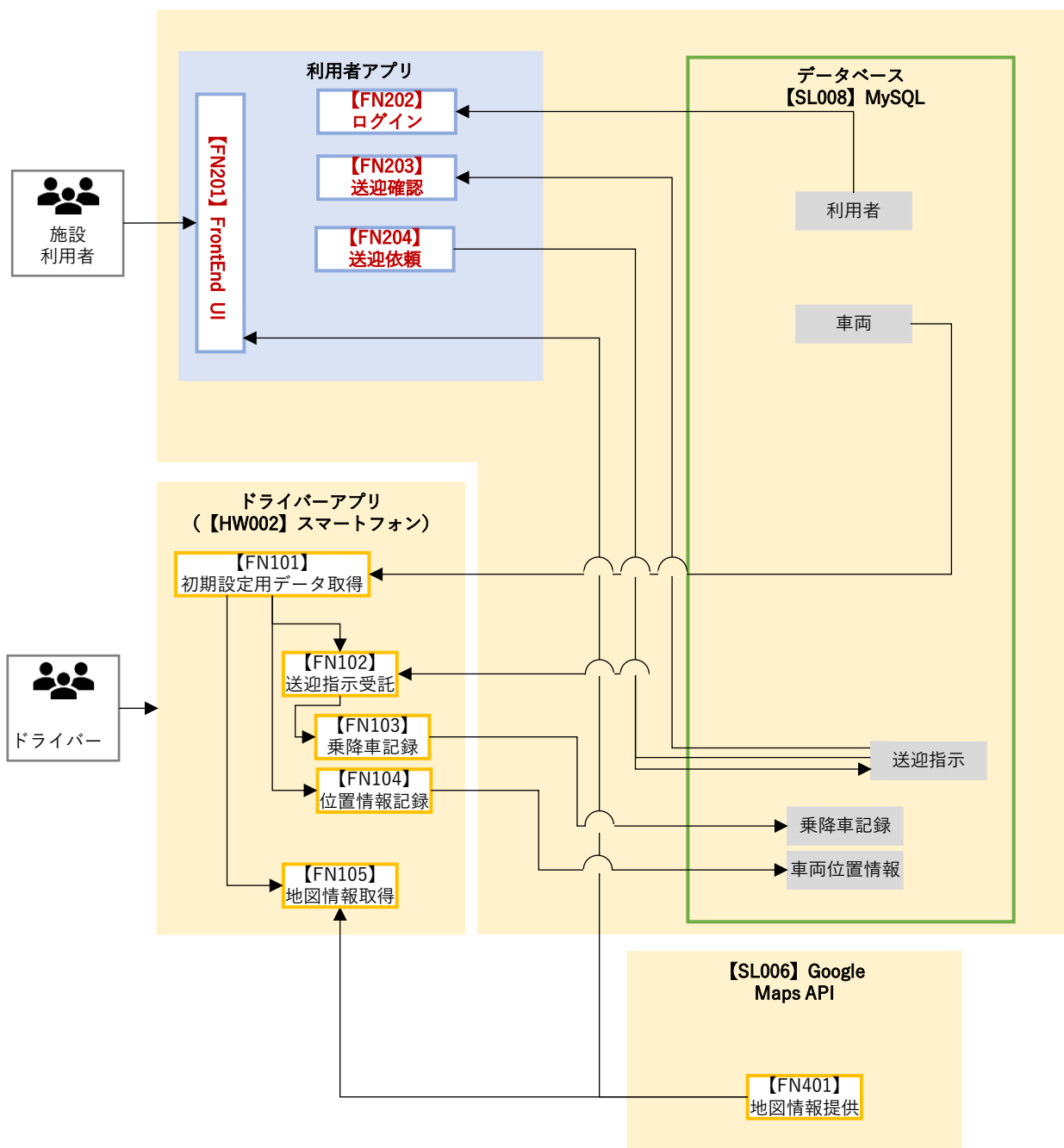
2-1-1. システムアーキテクチャ



凡例

- 既存のソフトウェア
- 開発したソフトウェア
- 既存機能
- 開発した機能
- データ
- ファイルストレージ
- データベース

図 2-1 システムアーキテクチャ（管理者アプリ）



凡例	既存のソフトウェア	開発したソフトウェア	既存機能	開発した機能	データ	ファイルストレージ	データベース
----	-----------	------------	------	--------	-----	-----------	--------

図 2-2 システムアーキテクチャ (利用者アプリ、ドライバーアプリ)

2-1-2. システム機能一覧

表 2-1 【HW001】 管理者アプリ機能一覧

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN001	FrontEnd UI	● 送迎計画を作成・管理する管理者向けのWebアプリケーション（フロントエンド）。Webブラウザ上で動作する
FN002	アカウント認証	● 入力されたアカウント情報を格納データと突き合わせ、施設管理者ユーザーの認証を行う機能
FN003	アカウント一覧取得	● ログイン可能な施設管理者ユーザーの情報を一括取得し返信する機能
FN004	アカウント登録・更新	● ログイン可能な施設管理者ユーザー情報を登録・更新する機能
FN005	車両一覧取得	● 送迎に活用する施設の車両の一覧を取得する機能
FN006	車両登録・更新	● 送迎に活用する施設の車両を登録・更新する機能
FN007	職員一覧取得	● 送迎に関わる施設職員（運転手・添乗員）の一覧を取得する機能
FN008	職員登録・更新	● 送迎に関わる施設職員（運転手・添乗員）を登録・更新する機能
FN009	利用者一覧取得	● 施設を利用する利用者情報一覧を取得する機能
FN010	利用者登録・更新	● 施設を利用する利用者を登録・更新する機能
FN011	施設来館予定者一覧取得	● 施設の指定日の来館予定者の一覧を取得する機能
FN012	施設来館予定者登録・更新・送迎委託情報登録	● 施設の指定日の来館予定者を登録・更新する機能 ● 委託施設から受託施設へ委託する来館予定者を指定する機能
FN013	送迎計画一覧取得	● 施設の指定日の送迎計画情報を取得する機能
FN014	送迎計画登録・更新	● 施設の指定日の送迎計画を登録・更新する機能
FN015	最適ルート検索	● 送迎便の最適ルートを検索する機能
FN016	同乗者推薦	● 送迎便に追加する利用者（同乗者）を推薦する機能
FN017	送迎委託・キャンセル	● 利用者を指定して他施設へ送迎委託する機能 ● 委託した送迎依頼をキャンセルする機能
FN018	送迎受託承認・却下	● 他施設・利用者から委託された送迎を受入れる（承認）又は受けられた送迎をキャンセルする機能
FN019	お知らせ	● 配車リクエストやキャンセル等のアクション実行時に、対象施設のオペレーターに通知する機能

表 2-2 【HW002】 ドライバーアプリ機能一覧

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN101	初期設定用データ取得	● ドライバーアプリを起動する車両情報を特定するため、事業所情報から車両情報一覧を取得する機能
FN102	送迎指示情報取得	● アプリで送迎指示を組み立てるために、指定日指定車両の送迎情報を取得する機能
FN103	乗降車記録	● 運転手が利用者を乗車・降車させた後に時間を記録する機能
FN104	車両位置情報記録	● 利用者に送迎車の位置を連絡するために、現在の車両の位置情報を記録する機能
FN105	地図情報取得	● 現在の走行位置から地図情報を取得する機能

表 2-3 【HW003】 利用者アプリ機能一覧

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN201	FrontEnd UI	● 送迎を利用する施設利用者向けのWebアプリケーション（フロントエンド）。Web ブラウザ上で動作する
FN202	利用者認証	● 入力されたアカウント情報を格納データと突き合わせ、利用者ユーザーの認証を行う機能
FN203	送迎確認	● 利用者の送迎予定情報（時間、車両、車両の現在地）を取得する機能
FN204	送迎依頼	● 利用者が送迎を依頼する機能

表 2-4 【CO001】 AWS EC2 機能一覧

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN301	来館予定作成バッチ	● 未来の日付の送迎計画に備えて、利用者の来館予定を自動生成するバッチ処理機能

表 2-5 【SL006】 Google Maps API 機能

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN401	地図表示	● 位置情報から地図を表示する機能
FN402	経路検索	● 出発地、経由地、目的地の乗降ポイント間の経路を検索する機能

2-1-3. システム機能の詳細

以下に、システム機能の詳細を記す。なお、本業務において開発（新規・改修）を行うシステム機能は、機能名称を**朱文字**で示す。

【FN001】 FrontEnd UI <新規開発>

- 本システム機能の概要
 - 送迎計画を作成・管理する管理者向けの Web アプリケーション（フロントエンド）
 - Web ブラウザ上で動作し、Next.js で実装
 - 福祉・観光・教育の業種を包含する統一 UI で実現
 - 業種固有の対応項目について、他業種での利用時には入力不可や非表示の対応をする（特に福祉の利用者に関する項目は固有のものが多い）

- 本システム機能の入力・処理・出力のフローチャート

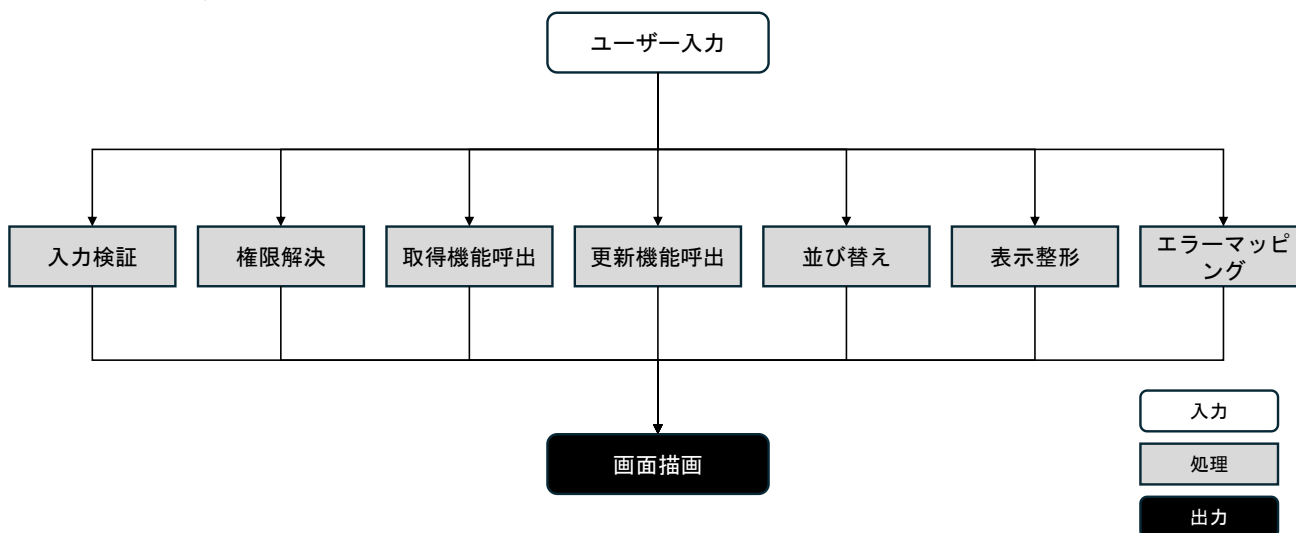


図 2-3 【FN001】のフローチャート

- 本システム機能の処理の詳細
 - 入力検証
 - ◇ 処理内容
 - 必須入力の有無、形式（例：日付 YYYY-MM-DD、数値範囲、ID 整合）を検証
 - 相関チェック（例：出発時間 ≤ 到着時間、施設 ID と事業所 ID の整合）
 - 検証 NG 時はエラー情報を生成
 - ◇ 利用するライブラリ
 - 【SL005】 Next.js
 - ◇ 利用するアルゴリズム
 - なし
 - 権限管理

- ◇ 処理内容
 - 権限（管理者/オペレーター等）に応じた操作可否・表示制御
 - フィルタ制約（所属事業所のみ参照可など）の適用
 - 権限 NG 時は権限不足エラーを返却
- ◇ 利用するライブラリ
 - 【SL005】 Next.js
- ◇ 利用するアルゴリズム
 - なし
- 取得機能呼出
 - ◇ 処理内容
 - 条件に基づく一覧/詳細の取得
 - ページング・ソート条件の付与
 - 取得結果を他処理へ受渡し
 - ◇ 利用するライブラリ
 - 【SL005】 Next.js
 - ◇ 利用するアルゴリズム
 - なし
- 更新機能呼出
 - ◇ 処理内容
 - ソート条件や編集内容の保存
 - サーバー応答に応じて保存結果を判定
 - 必要に応じて再取得（更新内容反映確認）
 - ◇ 利用するライブラリ
 - 【SL005】 Next.js
 - ◇ 利用するアルゴリズム
 - なし
- 並び替え
 - ◇ 処理内容
 - UI のドラッグ&ドロップ結果を順序に変換
 - 差分（変更行のみ）抽出、一覧の連番/桁空けなどの正規化
 - 更新機能呼出に渡す出力可能の抽出
 - ◇ 利用するライブラリ
 - 【SL005】 Next.js
 - ◇ 利用するアルゴリズム
 - なし
- 表示整形
 - ◇ 処理内容
 - 取得データを画面表示用 ViewModel へ整形（ページング情報、件数、ソート状態）
 - 地図表示用オブジェクト（ピン、ポリライン、ラベル）の組立て
 - 時刻・距離のフォーマット、空状態メッセージの決定
 - ◇ 利用するライブラリ

- 【SL005】 Next.js
- 【SL006】 Google Maps API (地図オーバーレイ 描画)
- ◇ 利用するアルゴリズム
 - なし
- エラーマッピング
 - ◇ 処理内容
 - HTTP/業務エラーを画面用メッセージに変換
 - フィールド起点のエラー表示 (該当行フォーカス等)
 - ◇ 利用するライブラリ
 - 【SL005】 Next.js
 - ◇ 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
 - ◇ ユーザー入力
 - データの内容
 - だれが：ログイン中のアカウントと役割・所属 (例：管理者/事業所 A)
 - いつ：対象の日付・時間 (例：8/30 の便)
 - なにを：行った操作 (検索、保存、並び替え、ダウンロード など)
 - どれを：対象の便・ルート、ページ番号、表示件数 など
 - 画面の状態：表示言語、タイムゾーン、画面サイズ
 - データの形式
 - リクエストパラメータ
 - 利用するデータインターフェース
 - なし
 - ◇ ユーザー入力
 - 出力
 - ◇ 画面描画
 - データの内容
 - 処理結果を反映した画面表示 (一覧・詳細・地図上のピン/経路・メッセージ表示)
 - データの形式
 - HTML
 - 利用するデータインターフェース
 - なし

【FN002】 アカウント認証<新規開発>

- 本システム機能の概要
 - 入力されたアカウント情報を格納データと突き合わせ、施設管理者ユーザーの認証を行う機能
 - アカウント情報は ID とパスワード

- 本システム機能の入力・処理・出力のフローチャート

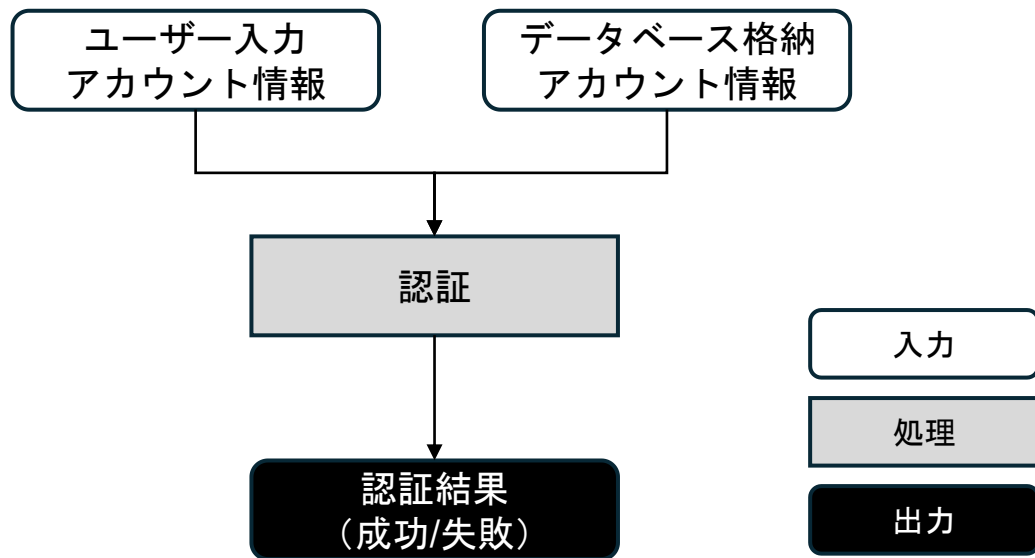


図 2-4 【FN002】のフローチャート

- 本システム機能の処理の詳細
 - 認証
 - ◇ 処理内容
 - IDとパスワードをデータベースに格納している情報と突合
 - ◇ 利用するライブラリ
 - 【SL009】 Devise
 - ◇ 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
 - ◇ ユーザー入力アカウント情報
 - データの内容
 - ID、パスワード
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - ◇ データベース格納アカウント情報
 - データの内容
 - ID、パスワード
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF002】施設管理者ユーザーデータ連携用 IF
 - 出力

- ◇ 認証結果
 - 内容
 - 認証結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN003】アカウント一覧取得<新規開発>

- 本システム機能の概要
 - 所属事業所の利用可能アカウントを一覧化し、所属・権限の確認（棚卸・監査）をする機能
 - ログインする施設管理者ユーザーの所属する事業所の施設管理者ユーザーを取得する
- 本システム機能の入力・処理・出力のフローチャート

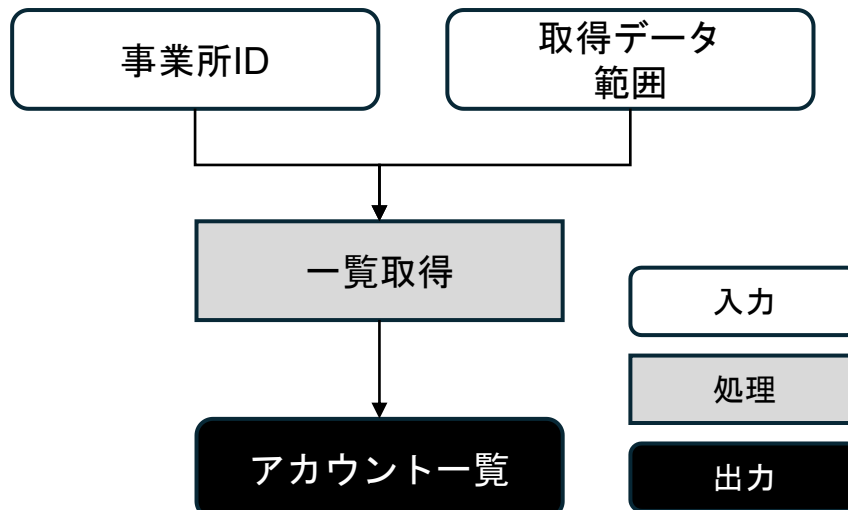


図 2-5 【FN003】のフローチャート

- 本システム機能の処理の詳細
 - 一覧取得
 - ◇ 処理内容
 - 格納データから、ログインする施設管理者ユーザーが所属する事業所の施設管理者ユーザーの一覧を取得する
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ 事業所 ID

- データの内容
 - ログインする施設管理者ユーザーが所属する事業所の ID 等
- データの形式
 - JSON
- 利用するデータインターフェース
 - なし
- ◇ 取得データ範囲
 - データの内容
 - 取得するデータ範囲 (例:先頭から 20 件)
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
- 出力
 - ◇ アカウント一覧
 - データの内容
 - ユーザーアカウントの一覧
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF002】施設管理者ユーザーデータ連携用 IF

【FN004】アカウント登録・更新<新規開発>

- 本システム機能の概要
 - 同一インスタンスを複数の施設管理者ユーザーが利用する前提で、法人・事業所単位で分離したセグメンテーション内でログイン可能な施設管理者ユーザー情報を登録・更新する機能
 - ID、パスワード、e-mail、法人 ID、事業者 ID、権限等の情報を格納
 - 指定された権限区分に応じて、閲覧権限、編集権限を付与
 - 送迎の委託・受託を行う他法人事業者が参照できる情報は最小限に限定（送迎を委託する車両・利用者のみ開示）その既定値（許可範囲/期間/開示プロファイル）は本機能で調整可能とする
- 本システム機能の入力・処理・出力のフローチャート

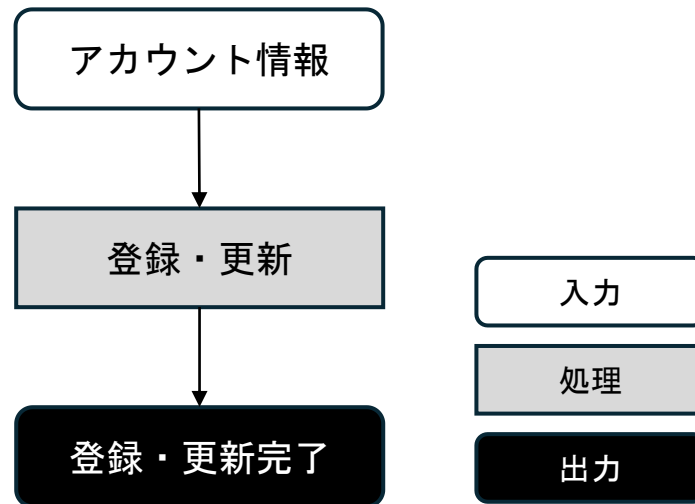


図 2-6 【FN004】のフローチャート

- 本システム機能の処理の詳細
 - 登録・更新
 - ◇ 処理内容
 - 入力パラメータの検証（必須・形式・長さ・権限整合性）
 - ID と e-mail の一意性確認
 - 法人 ID (party_id) ・事業所 (office_id) の整合性確認
 - パスワードのハッシュ化
 - 権限の付与・更新
 - アカウント登録・更新に必要な情報を基に施設管理者ユーザー情報を格納
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ アカウント情報
 - データの内容
 - アカウント情報 (ID、法人 ID、事業所 ID、e-mail、パスワード、名前、権限等)
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF002】施設管理者ユーザーデータ連携用 IF
 - 出力
 - ◇ 登録・更新完了
 - データの内容
 - 処理結果 (成功/失敗)
 - データの形式

- JSON
- 利用するデータインターフェース
- なし

【FN005】車両一覧取得<新規開発>

- 本システム機能の概要
 - 配車計画・稼働確認の前提となる所属事業所の利用可能車両を一覧化し参照する機能
 - ログインする施設管理者ユーザーに紐づく事業所 ID のデータを取得
 - 一括で取得する件数を指定可能
- 本システム機能の入力・処理・出力のフローチャート

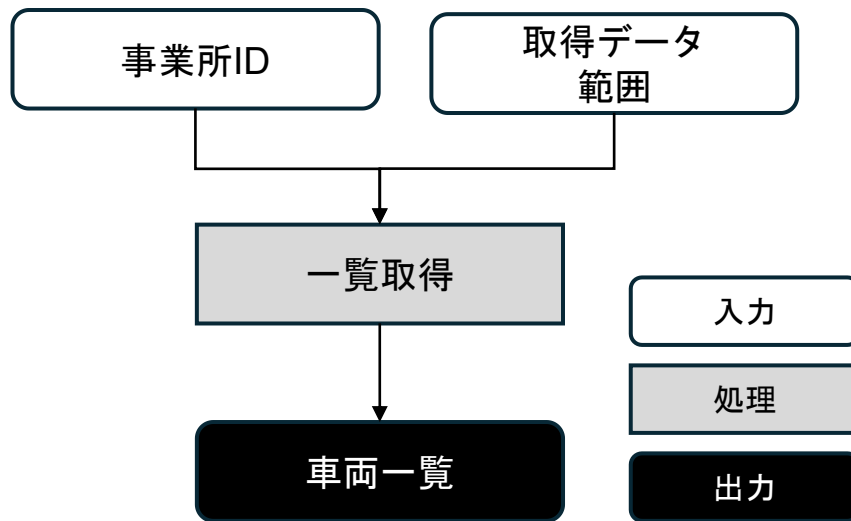


図 2-7 【FN005】のフローチャート

- 本システム機能の処理の詳細
 - 一覧取得
 - ◇ 処理内容
 - 格納データから、事業所で登録した車両の一覧を取得する
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ 事業所 ID
 - データの内容
 - ログインする施設管理者ユーザーが所属する事業所の ID 等
 - データの形式
 - JSON
 - 利用するデータインターフェース

- なし
- ◇ 取得データ範囲
 - データの内容
 - 取得するデータ範囲 (例:先頭から 20 件)
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
- 出力
 - ◇ 車両一覧
 - データの内容
 - 車両情報の一覧
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF004】車両データ一覧取得用 IF

【FN006】車両登録・更新<新規開発>

- 本システム機能の概要
 - 送迎に活用する車両を登録・更新する機能
 - 車両 (名前、車椅子の数、座席の数)、タイプ (ブランド) のデータを格納する
- 本システム機能の入力・処理・出力のフローチャート

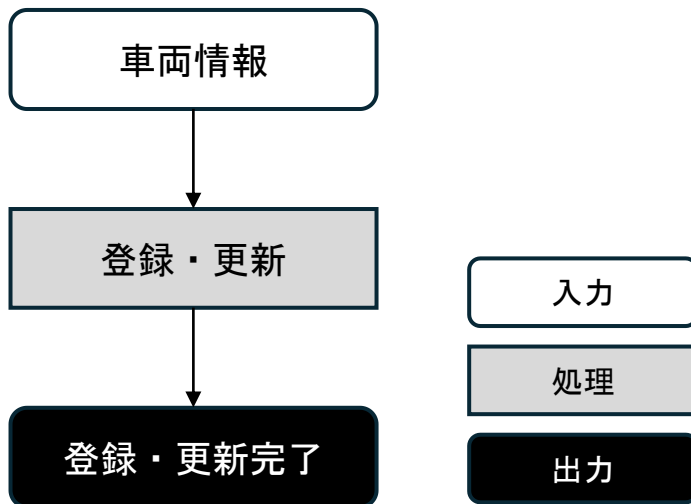


図 2-8 【FN006】のフローチャート

- 本システム機能の処理の詳細
 - 登録・更新
 - ◇ 処理内容

- 車両の登録・更新
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ 車両情報
 - データの内容
 - 車両の情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF005】 車両データ連携用 IF
 - ◇ 出力
 - ◇ 登録・更新
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN007】 職員一覧取得 <新規開発>

- 本システム機能の概要
 - 配車計画・乗務割当て（運転手/添乗員）の前提となる所属事業所の職員を一覧化し参照する機能
 - ログインする施設管理者ユーザーに紐づく事業所 ID のデータを取得
 - 一括で取得する件数を指定可能
 - 事業所職員が送迎計画を作成する際に運転手・添乗員情報として使用する
- 本システム機能の入力・処理・出力のフローチャート

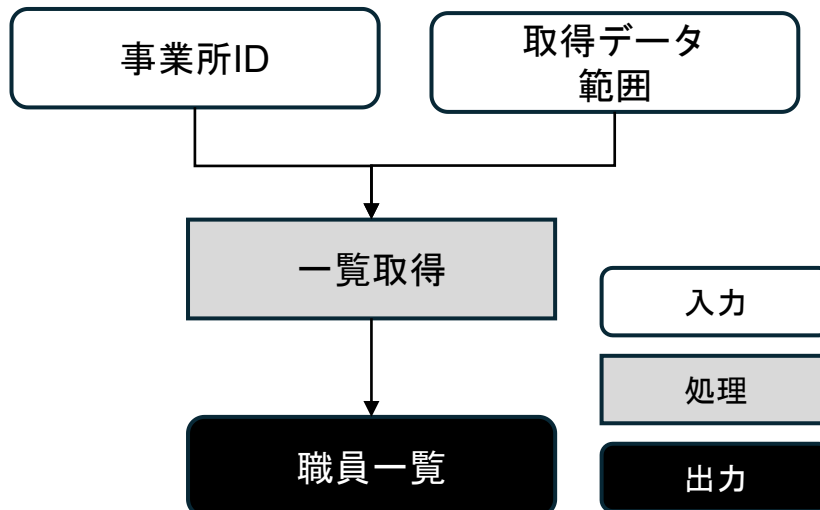


図 2-9 【FN007】のフローチャート

- 本システム機能の処理の詳細
 - 一覧取得
 - ◇ 処理内容
 - 格納データから、事業所で登録した職員の一覧を取得する
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ 事業所 ID
 - データの内容
 - ログインする施設管理者ユーザーが所属する事業所の ID 等
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - ◇ 取得データ範囲
 - データの内容
 - 取得するデータ範囲 (例:先頭から 20 件)
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - 出力
 - ◇ 職員一覧
 - データの内容
 - 職員情報の一覧

- データの形式
 - JSON
- 利用するデータインターフェース
 - 【IF006】職員データ一覧取得用 IF

【FN008】職員登録・更新<新規開発>

- 本システム機能の概要
 - 送迎に関わる施設職員（運転手・添乗員）を登録・更新する機能
 - 職員の氏名、運転可能な車両情報等を格納する
- 本システム機能の入力・処理・出力のフローチャート

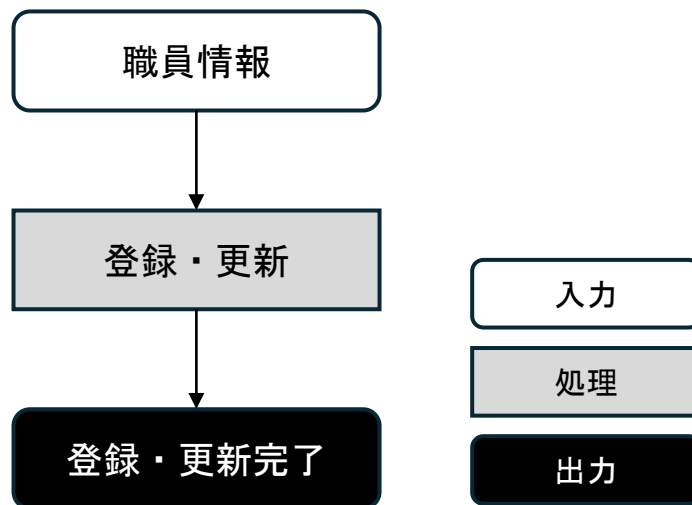


図 2-10 【FN008】のフローチャート

- 本システム機能の処理の詳細
 - 登録・更新
 - ◇ 処理内容
 - 職員の登録・更新
 - ◇ 利用するライブラリ
 - 【SL002】Ruby on Rails
 - ◇ 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
 - ◇ 職員情報
 - データの内容
 - 職員の情報
 - データの形式
 - JSON
 - 利用するデータインターフェース

- 【IF007】職員データ連携用 IF
- 出力
 - ◇ 登録・更新完了
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN009】利用者一覧取得<新規開発>

- 本システム機能の概要
 - 送迎計画の前提となる所属事業所の利用者を一覧化し参照する機能
 - ログインする施設管理者ユーザーに紐づく事業所 ID のデータを取得
 - 一括で取得する件数を指定可能
 - 本利用者情報を基に送迎計画を作成する
- 本システム機能の入力・処理・出力のフローチャート

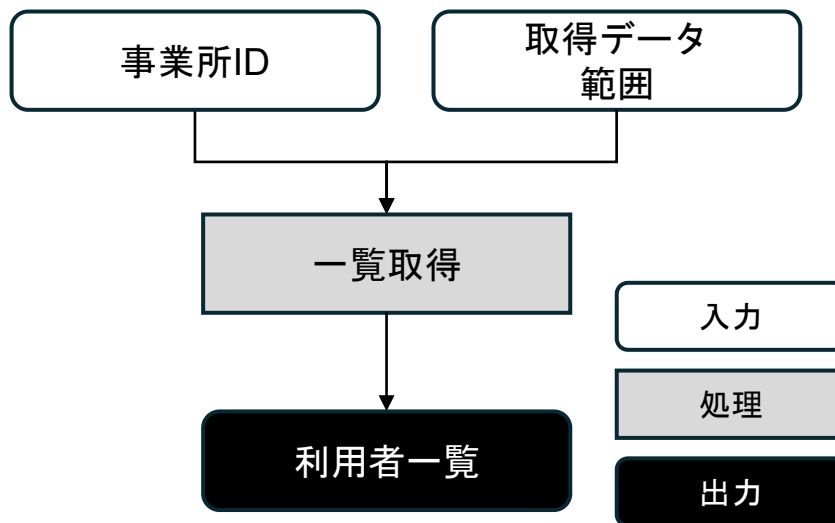


図 2-11 【FN009】のフローチャート

- 本システム機能の処理の詳細
 - 一覧取得
 - ◇ 処理内容
 - 格納データから、利用者の一覧を取得する
 - ◇ 利用するライブラリ
 - 【SL002】Ruby on Rails
 - ◇ 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
 - ◇ 事業所 ID
 - データの内容
 - ログインする施設管理者ユーザーが所属する事業所の ID 等
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - ◇ 取得データ範囲
 - データの内容
 - 取得するデータ範囲 (例:先頭から 20 件)
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
- 出力
 - ◇ 利用者一覧
 - データの内容
 - 利用者の一覧
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF008】利用者データ一覧取得用 IF

【FN010】利用者登録・更新<新規開発>

- 本システム機能の概要
 - 施設の利用者を登録・更新する機能
 - 利用者の基本情報 (氏名、住所等)、自宅の位置情報、施設に来館するパターン情報、乗り合わせ NG の利用者を格納する
- 本システム機能の入力・処理・出力のフローチャート

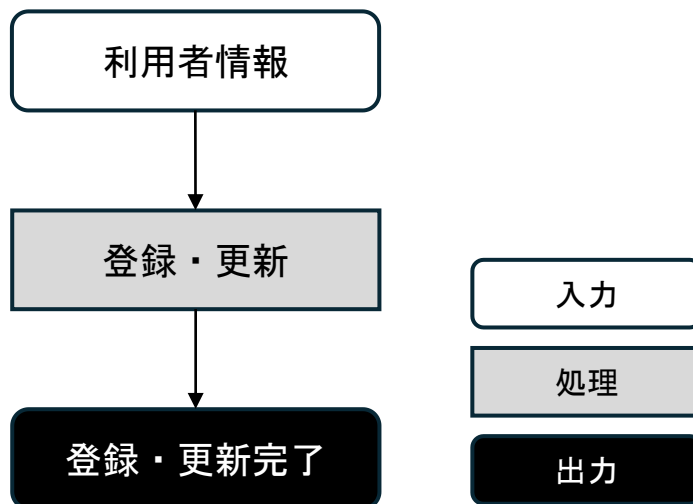


図 2-12 【FN010】のフローチャート

- 本システム機能の処理の詳細
 - 登録・更新
 - ◇ 処理内容
 - 利用者の登録・更新
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ 利用者情報
 - データの内容
 - 利用者の情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF009】 利用者データ連携用 IF
 - 出力
 - ◇ 登録・更新完了
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN011】施設来館予定者一覧取得<新規開発>

- 本システム機能の概要
 - 施設の指定日の来館予定者の一覧を取得する機能
 - ログインする施設管理者ユーザーに紐づく事業所 ID のデータを取得
 - 一括で取得する件数を指定可能
- 本システム機能の入力・処理・出力のフローチャート

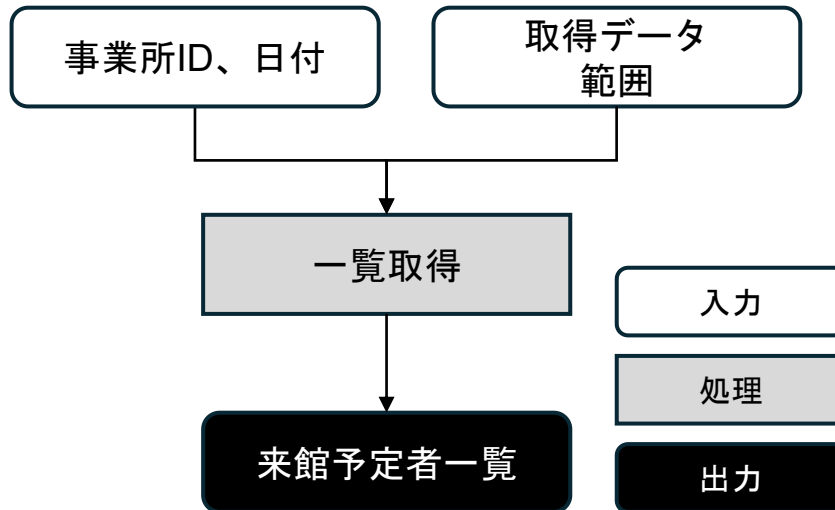


図 2-13 【FN011】のフローチャート

- 本システム機能の処理の詳細
 - 一覧取得
 - ◇ 処理内容
 - 格納データから、来館予定者の一覧を取得する
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ 事業所 ID、日付
 - データの内容
 - ログインする施設管理者ユーザーが所属する事業所の ID と、来館予定者を確認する日付等
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - ◇ 取得データ範囲
 - データの内容
 - 取得するデータ範囲 (例:先頭から 20 件)

- データの形式
 - JSON
- 利用するデータインターフェース
 - なし
- 出力
 - ◇ 来館予定者一覧
 - データの内容
 - 来館予定者の一覧
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF011】利用者来館予定データ一覧取得用 IF

【FN012】施設来館予定者登録・更新<新規開発>

- 本システム機能の概要
 - 施設の指定日の来館予定者を登録・更新する機能
 - 追加やキャンセルになった方を登録・更新する
- 本システム機能の入力・処理・出力のフローチャート

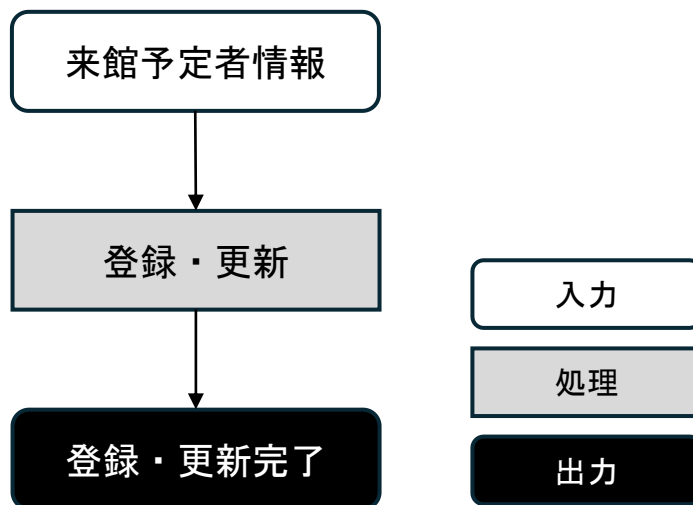


図 2-14 【FN012】のフローチャート

- 本システム機能の処理の詳細
 - 登録・更新
 - ◇ 処理内容
 - 利用者の登録・更新
 - ◇ 利用するライブラリ
 - 【SL002】Ruby on Rails
 - ◇ 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
 - ◇ 来館予定者情報
 - データの内容
 - 利用者の情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF012】利用者来館予定データ連携用 IF
 - 出力
 - ◇ 登録・更新
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN013】送迎計画一覧取得<新規開発>

- 本システム機能の概要
 - 施設の指定日の送迎計画一覧情報を取得する機能
 - 送迎便（指定日の送迎する便）、送迎指示（指定日の指定便に誰をどの順番で乗せるか）のデータを取得
- 本システム機能の入力・処理・出力のフローチャート

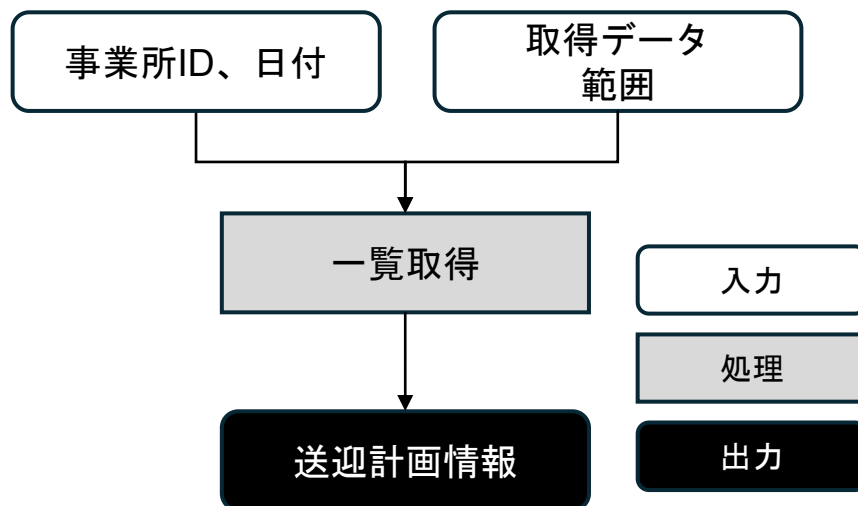


図 2-15 【FN013】のフローチャート

- 本システム機能の処理の詳細
 - 一覧取得
 - ◇ 処理内容

- 格納データから、施設の指定日の送迎計画情報を取得する
- ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
- ◇ 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
 - ◇ 事業所 ID
 - データの内容
 - ログインする施設管理者ユーザーが所属する事業所の ID、送迎計画情報を確認する日付等
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - ◇ 取得データ範囲
 - データの内容
 - 取得するデータ範囲（例:先頭から 20 件）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - 出力
 - ◇ 送迎計画情報
 - データの内容
 - 利用者の一覧
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF010】送迎計画データ取得用 IF

【FN014】送迎計画登録・更新<新規開発>

- 本システム機能の概要
 - 施設の指定日の送迎計画を登録・更新する機能
 - 送迎便（指定日の送迎する便）、送迎指示（指定日の指定便に誰をどの順番で乗せるか）のデータを格納
 - 車両アサイン（車両・運転手・添乗員）の登録・更新及び送迎便への費も付け
 - ルートアサイン（経路・立寄順）の登録・更新及び送迎便への適用（【FN015】最適ルート検索で算出した結果の受入れを含む）

- 本システム機能の入力・処理・出力のフローチャート

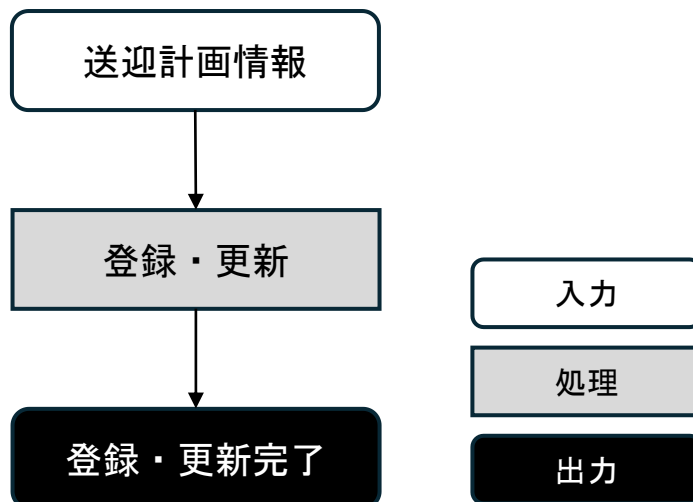


図 2-16 【FN014】のフローチャート

- 本システム機能の処理の詳細
 - 登録・更新
 - ◇ 処理内容
 - 送迎計画の登録・更新
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ 来館予定者情報
 - データの内容
 - 送迎計画の情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF013】送迎指示データ連携用 IF
 - 出力
 - ◇ 登録・更新
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN015】最適ルート検索<新規開発>

- 本システム機能の概要
 - 送迎するそれぞれの便に乗降する利用者情報（複数）を基に、始点（施設）～経由地（利用者の乗降地点）～終点（施設）の最適ルートと所要時間・距離を検索する機能
 - ◇ 1回の送迎便で通過する複数の地点の最適ルートを検索する
 - ◇ 地点の種類は、施設情報（施設位置、出発・到着に活用）、利用者情報（自宅位置、誘導経路位置）とし、位置情報を緯度・経度で指定
 - ◇ 地点の距離を計算し、Google Maps でルートを取得する
- 本システム機能の入力・処理・出力のフローチャート

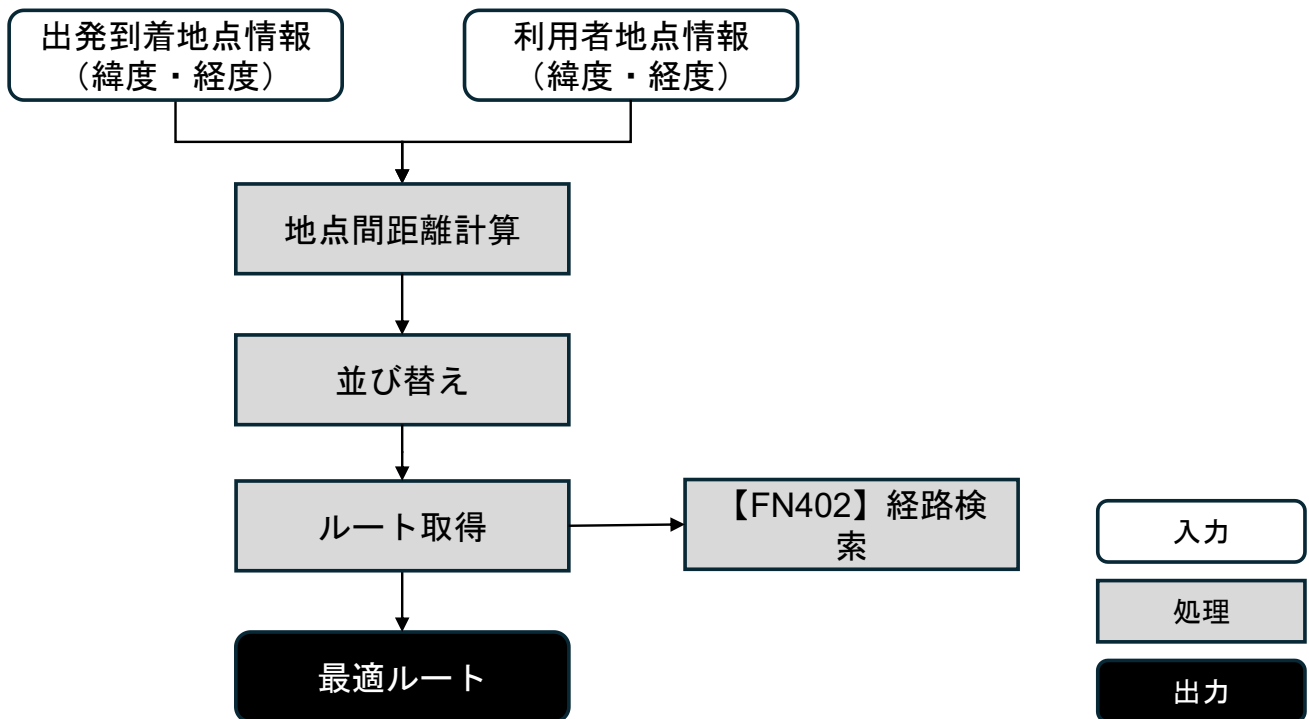


図 2-17 【FN015】のフローチャート

- 本システム機能の処理の詳細
 - 地点間距離計算
 - ◇ 処理内容
 - 出発地点からそれぞれの利用者の自宅位置の距離を計算
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - 並び替え
 - ◇ 処理内容
 - 利用者情報を以下の2種類で並び替え
 - 迎え（利用者の自宅からピックアップして最後に施設に降ろす場合）：遠い順
 - 送り（施設でまとめてピックアップして順に利用者の自宅に降ろす場合）：近い順
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails

- ルート取得
 - ◇ 処理内容
 - 並び替えられた利用者地点情報一覧を入力情報として Google Maps API 「【IF103】 ルート表示、誘導経路及びピンの制御」を使って地図上のルート情報を取得
 - ◇ 利用するライブラリ
 - 【SL002】 Ruby on Rails
 - 【SL006】 Google Maps API
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 出発到着地点情報（緯度・経度）
 - データの内容
 - 事業所の地点情報（複数）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF002】 施設管理者ユーザーデータ連携用 IF
 - ◇ 利用者地点情報（緯度・経度）
 - データの内容
 - 各利用者の自宅の地点情報（複数）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF009】 利用者データ連携用 IF
 - 出力
 - ◇ 最適ルート
 - データの内容
 - 最適ルート
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF102】 事業所及び乗降ポイント
 - 【IF103】 ルート表示、誘導経路及びピンの制御

【FN016】 同乗者推薦<新規開発>

- 本システム機能の概要
 - 送迎便に追加する利用者（同乗者）を推薦する機能
 - 指定された利用者の情報から近い位置にいる利用者を推薦表示

- 本システム機能の入力・処理・出力のフローチャート

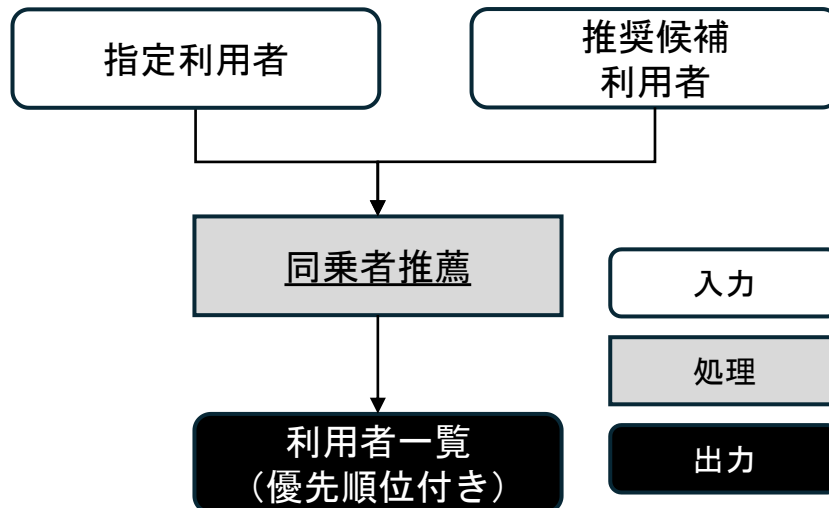


図 2-18 【FN016】 のフローチャート

- 本システム機能の処理の詳細

- 同乗者推薦

- ◇ 処理内容

- 指定された利用者の自宅の緯度・経度から、自宅が近い利用者を推薦表示
 - 指定条件
 - 指定された利用者の迎え時間又は送り時間から前後 30 分以内の範囲の、迎え時間又は送り時間を登録している他の利用者
 - 自宅の緯度・経度情報を基に指定された利用者との他の利用者との直線距離を算出し、直線距離が近い順に推奨

- ◇ 利用するライブラリ

- 【SL002】 Ruby on Rails

- ◇ 利用するアルゴリズム

- 【AL101】 推薦者表示アルゴリズム

- 本システム機能の入出力データの仕様

- 入力

- ◇ 指定利用者

- データの内容
 - 利用者の情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF009】 利用者データ連携用 IF

- 出力

- ◇ 登録・更新

- データの内容
 - 利用者一覧 (優先順位づけ)

- データの形式
 - JSON
- 利用するデータインターフェース
 - 【IF008】利用者データ一覧取得用 IF

【FN017】送迎委託・キャンセル<新規開発>

- 本システム機能の概要
 - 送迎委託する来館予定者を設定する
 - 送迎委託した来館予定者をキャンセルする
- 本システム機能の入力・処理・出力のフローチャート

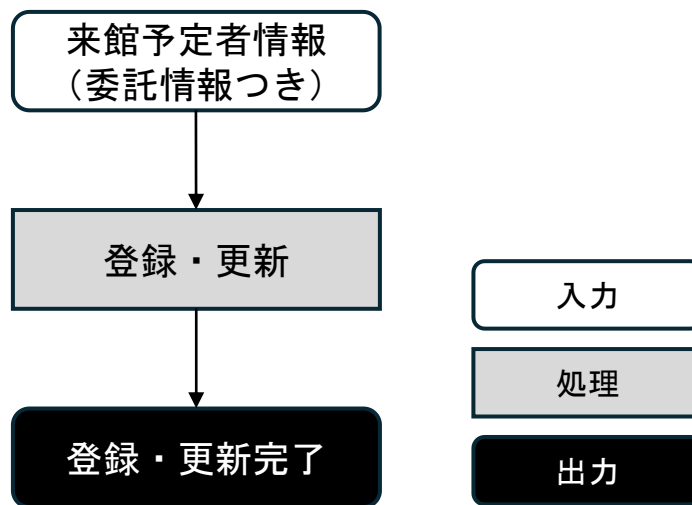


図 2-19 【FN017】のフローチャート

- 本システム機能の処理の詳細
 - 登録・更新
 - ◇ 処理内容
 - 利用者の登録・更新
 - 委託：送迎委託する利用者の情報に「送迎委託」フラグを立てる
 - キャンセル：送迎委託した利用者の情報の「送迎委託」フラグを下ろす
 - ◇ 利用するライブラリ
 - 【SL002】Ruby on Rails
 - ◇ 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
 - ◇ 来館予定者情報（委託情報つき）
 - データの内容
 - 利用者の情報
 - データの形式
 - JSON
 - 利用するデータインターフェース

- 【IF012】利用者来館予定データ連携用 IF
- 出力
 - ◇ 登録・更新
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN018】送迎受託承認・却下<新規開発>

- 本システム機能の概要
 - 送迎受託する来館予定者を承認する
 - 送迎受託した来館予定者を却下する
- 本システム機能の入力・処理・出力のフローチャート

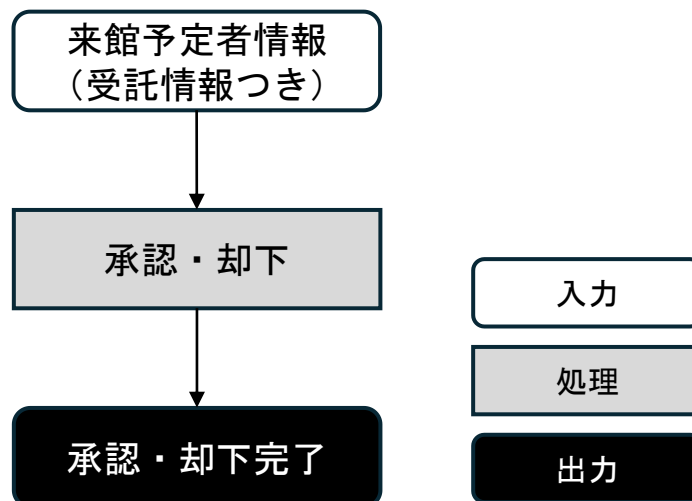


図 2-20 【FN018】のフローチャート

- 本システム機能の処理の詳細
 - 承認・却下
 - ◇ 処理内容
 - 利用者の登録・更新
 - 承認：送迎委託する利用者の情報から予定情報を登録する
 - 却下：送迎委託した利用者の情報の「送迎委託」フラグを下ろす
 - ◇ 利用するライブラリ
 - 【SL002】Ruby on Rails
 - ◇ 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
 - ◇ 来館予定者情報（受託情報つき）
 - データの内容
 - 利用者の情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF012】利用者来館予定データ連携用 IF
 - 出力
 - ◇ 承認・却下
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN019】お知らせ<新規開発>

- 本システム機能の概要
 - 送迎委託・受託の結果をそれぞれの担当者に通知し、情報を確認する
- 本システム機能の入力・処理・出力のフローチャート

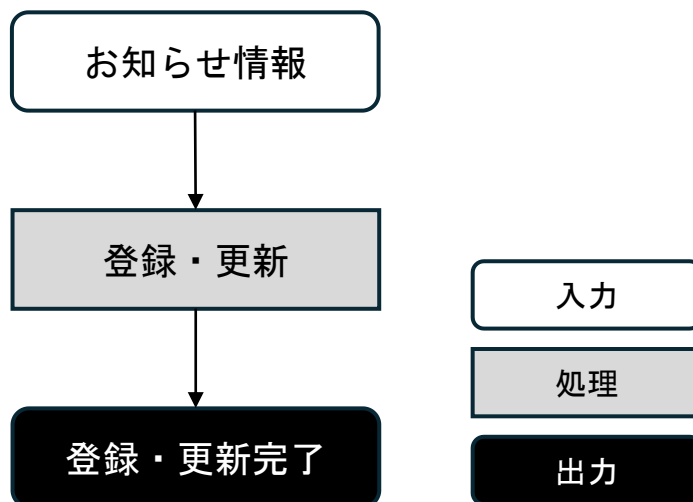


図 2-21 【FN019】のフローチャート

- 本システム機能の処理の詳細
 - 登録・更新
 - ◇ 処理内容
 - お知らせ情報の登録・更新
 - ◇ 利用するライブラリ

- 【SL002】 Ruby on Rails
 - ◇ 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
 - ◇ お知らせ情報
 - データの内容
 - お知らせ情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF016】 お知らせデータ連携用 IF
 - 出力
 - ◇ 登録・更新
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN101】 初期設定用データ取得

- 本システム機能の概要
 - ドライバーアプリを起動する際に基礎情報を取得する機能
 - 事業所情報から車両情報一覧を取得
- 本システム機能の入力・処理・出力のフローチャート

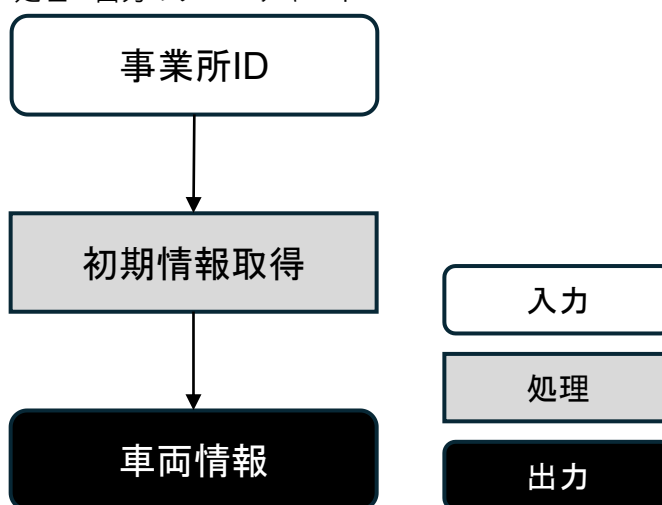


図 2-22 【FN101】 のフローチャート

- 本システム機能の処理の詳細
 - 初期情報取得
 - ◇ 処理内容
 - 初期情報（事業所 ID）から車両情報を取得
 - ◇ 利用するライブラリ
 - Delphi
 - ◇ 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
 - ◇ 事業所情報
 - データの内容
 - 事業所 ID
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - ◇ 車両情報
 - データの内容
 - 車両情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF004】車両データ一覧取得用 IF
 - 出力

【FN102】送迎指示受託

- 本システム機能の概要
 - 送迎指示テーブルの指示データを取得し、送迎便情報を画面描画するために必要な情報を取得する機能
 - 画面描画のためのデータ整形・状態更新を行う

- 本システム機能の入力・処理・出力のフローチャート

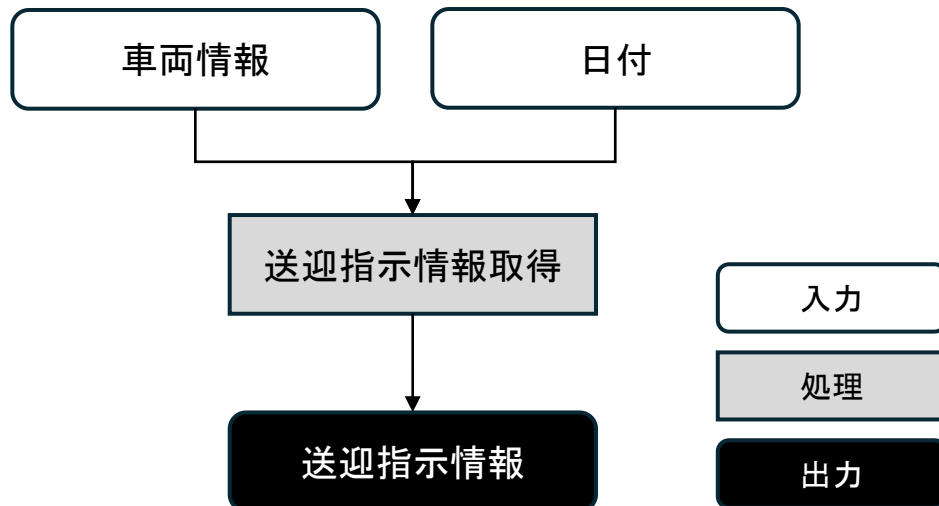


図 2-23 【FN102】のフローチャート

- 本システム機能の処理の詳細
 - 送迎指示情報取得
 - ◇ 処理内容
 - 指定日付と車両情報（車両 ID）を基に、送迎指示情報を取得
 - ◇ 利用するライブラリ
 - Delphi
 - ◇ 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
 - ◇ 車両情報
 - データの内容
 - 指定した車両の ID 等
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - ◇ 日付
 - データの内容
 - 指定日
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - 出力
 - ◇ 送迎指示情報
 - データの内容

- 送迎指示情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF013】送迎指示データ連携用 IF
- ◇ 処理結果
- データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN103】乗降車記録

- 本システム機能の概要
 - 運転手が利用者の乗車・降車を記録する機能
 - 利用者の状態（乗車・降車・キャンセル・後回し）、乗車時刻、降車時刻を格納する
- 本システム機能の入力・処理・出力のフローチャート

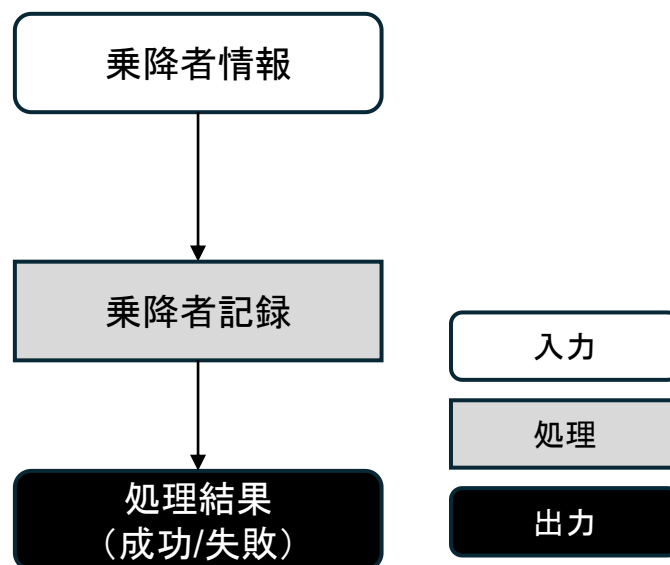


図 2-24 【FN103】のフローチャート

- 本システム機能の処理の詳細
 - 乗降者記録
 - ◇ 処理内容
 - 利用者の乗車・降車時間と状態をデータ格納する
 - 状態は以下のとおり
 - 「乗車」:この利用者の乗車確認し、次の地点へ移動指示する
 - 「降車」:この利用者の降車確認し、次の地点へ移動指示する

- 「キャンセル」:この利用者をスキップする（機能的には「乗車する」と同等）
- 「後回し」:この利用者の乗車確認を無視し、当該便の最後に再度訪問するようにする
- ◇ 利用するライブラリ
 - Delphi
- ◇ 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
 - ◇ 乗降者情報
 - データの内容
 - 利用者の乗車時刻、降車時刻等
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF014】乗降車記録データ連携用 IF
 - 出力
 - ◇ 処理結果
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN104】位置情報記録

- 本システム機能の概要
 - 利用者に送迎車の位置を連絡するために、現在の車両の位置情報を記録する機能

- 本システム機能の入力・処理・出力のフローチャート

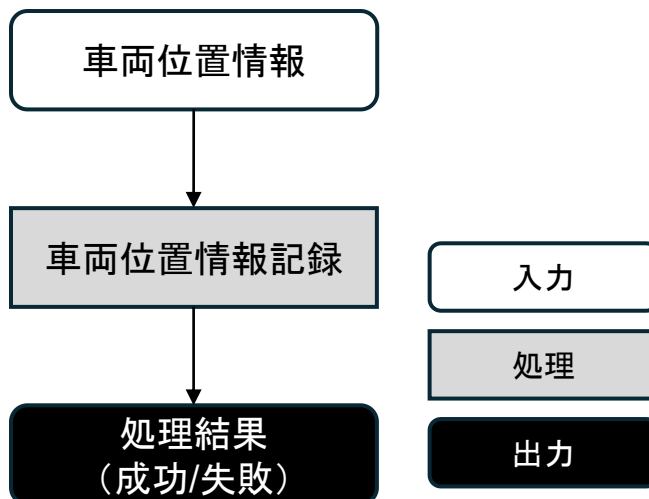
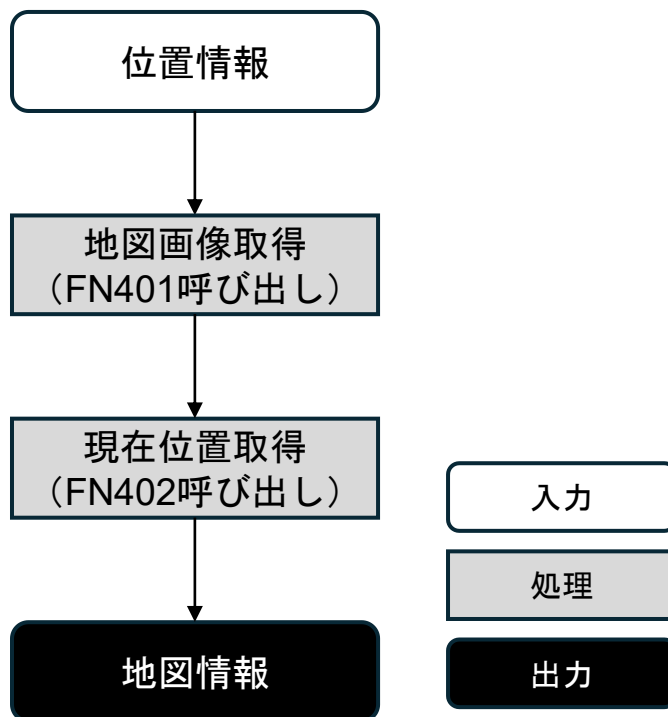


図 2-25 【FN103】 のフローチャート

- 本システム機能の処理の詳細
 - 車両位置情報記録
 - ◇ 処理内容
 - スマホの OS から位置情報（緯度・経度）を取得
 - 取得したデータを格納
 - ◇ 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
 - ◇ 車両位置情報
 - データの内容
 - 緯度・経度等
 - データの形式
 - HTTP
 - 利用するデータインターフェース
 - 【IF024】 車両位置データ連携用 IF
 - 出力
 - ◇ 処理結果
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN105】地図情報取得

- 本システム機能の概要
 - 現在の走行位置から地図情報を取得する機能
 - ※本機能が利用されるドライバーアプリでは地図の静止画像と現在位置を表示する。ナビゲーションに関しては Google Maps アプリを呼出している
- 本システム機能の入力・処理・出力のフローチャート



● 図 2-26 【FN104】のフローチャート

- 本システム機能の処理の詳細
 - 地図画像取得
 - ◇ 処理内容
 - 現在位置から地図画像を取得（「【FN401】地図表示」呼出し）
 - ◇ 利用するライブラリ
 - PHP
 - 【SL006】 Google Maps API
 - ◇ 利用するアルゴリズム
 - なし
 - 現在位置取得
 - ◇ 処理内容
 - 現在位置から現在位置のポイントを取得（「【FN402】経路検索」呼出し）
 - ◇ 利用するライブラリ
 - PHP
 - 【SL006】 Google Maps API
 - ◇ 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
 - ◇ 位置情報
 - データの内容
 - 現在位置（緯度・経度）等
 - データの形式
 - HTTP
 - 利用するデータインターフェース
 - テキスト
 - 出力
 - ◇ 地図情報
 - データの内容
 - 現在地周辺の地図情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF101】道路マップ

【FN201】 FrontEnd UI <改修機能>

- 本システム機能の概要
 - 利用者向けの Web アプリケーション（フロントエンド）
 - Web ブラウザ上で動作する

既存機能の改修内容

現行の利用者アプリは、PHP による直接実装などアーキテクチャが旧式であり、セキュリティリスクが高い。今回開発する Web アプリでは、セキュリティ対策の強化を目的に、堅牢で実績のある Ruby on Rails をベースとしたアーキテクチャに刷新する

- 本システム機能の入力・処理・出力のフローチャート

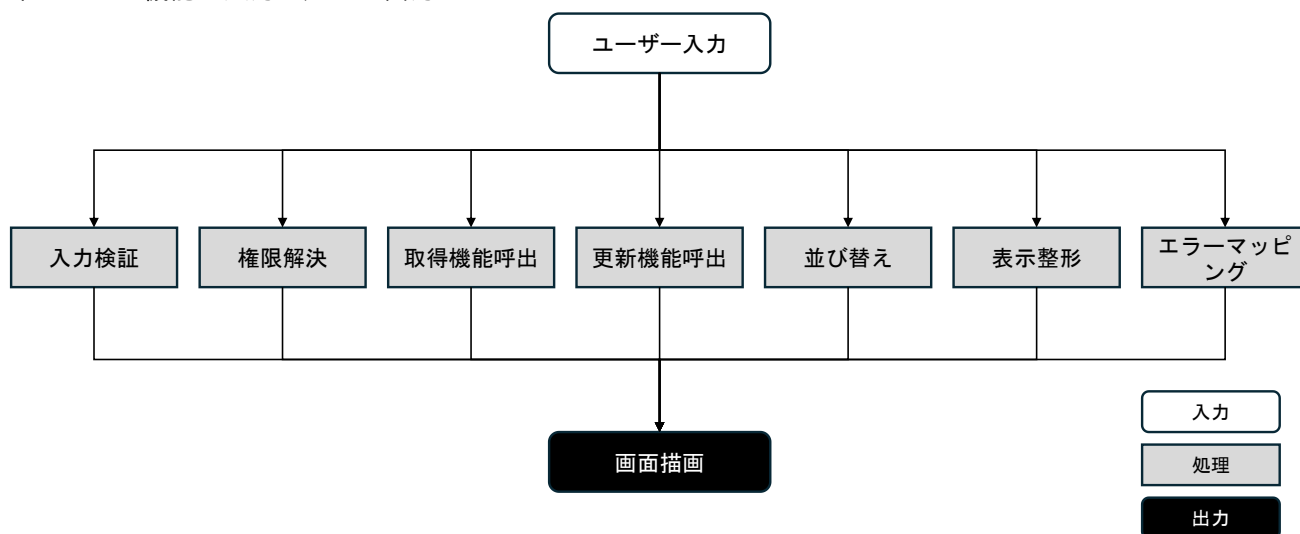


図 2-27 【FN201】 のフローチャート

- 本システム機能の処理の詳細
 - 入力検証
 - ◇ 処理内容
 - 必須入力の有無、形式（例：日付 YYYY-MM-DD、数値範囲、ID 整合）を検証
 - 関連チェック（例：出発時間 ≤ 到着時間開、施設 ID と事業所 ID の整合）
 - 検証 NG 時はエラー情報を生成
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - 【SL005】 Next.js <改修後>
 - ◇ 利用するアルゴリズム
 - なし
 - 権限解決
 - ◇ 処理内容
 - 権限（管理者/オペレーター等）に応じた操作可否・表示制御
 - フィルタ制約（所属事業所のみ参照可 など）の適用
 - 権限 NG 時は権限不足エラーを返却
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - 【SL005】 Next.js <改修後>
 - ◇ 利用するアルゴリズム
 - なし
 - 取得機能呼出
 - ◇ 処理内容
 - 条件に基づく一覧/詳細の取得
 - ページング・ソート条件の付与
 - 取得結果を他処理へ受渡し
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - 【SL005】 Next.js <改修後>
 - ◇ 利用するアルゴリズム
 - なし
 - 更新機能呼出
 - ◇ 処理内容
 - ソート条件や編集内容の保存
 - サーバー応答に応じて保存結果を判定
 - 必要に応じて再取得（更新内容反映確認）
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - 【SL005】 Next.js <改修後>
 - ◇ 利用するアルゴリズム
 - なし

- 並び替え
 - ◇ 処理内容
 - UIのドラッグ&ドロップ結果を順序に変換
 - 差分(変更行のみ)抽出、一覧の連番/桁空けなどの正規化
 - 更新機能呼出しに渡す出力可能な抽出
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - **【SL005】Next.js <改修後>**
 - ◇ 利用するアルゴリズム
 - なし
- 表示整形
 - ◇ 処理内容
 - 取得データを画面表示用 ViewModel へ整形(ページング情報、件数、ソート状態)
 - 地図表示用オブジェクト(ピン、ポリライン、ラベル)の組立て
 - 時刻・距離のフォーマット、空状態メッセージの決定
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - **【SL005】Next.js <改修後>**
 - **【SL006】Google Maps API**(地図オーバーレイ描画)
 - ◇ 利用するアルゴリズム
 - なし
- エラーマッピング
 - ◇ 処理内容
 - HTTP/業務エラーを画面用メッセージに変換
 - フィールド起点のエラー表示(該当行フォーカス等)
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - **【SL005】Next.js <改修後>**
 - ◇ 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
 - ◇ ユーザー入力
 - データの内容
 - だれが: ログイン中のアカウントと役割・所属(例: 管理者/事業所 A)
 - いつ: 対象の日付・時間(例: 8/30の便)
 - なにを: 行った操作(検索、保存、並び替え、ダウンロードなど)
 - どれを: 対象の便・ルート、ページ番号、表示件数など
 - 画面の状態: 表示言語、タイムゾーン、画面サイズ
 - データの形式
 - リクエストパラメータ

- 利用するデータインターフェース
 - 【IF009】 利用者データ連携用 IF
- 出力
 - ◇ 画面描画
 - データの内容
 - 処理結果を反映した画面表示（一覧・詳細・地図上のピン/経路・メッセージ表示）
 - データの形式
 - HTML
 - 利用するデータインターフェース
 - なし

【FN202】 利用者認証<改修機能>

- 本システム機能の概要
 - 入力されたアカウント情報を格納データと突き合わせ、アクセスする利用者ユーザーの認証を行う機能

既存機能の改修内容

現行の利用者アプリは、PHP による直接実装などアーキテクチャが旧式であり、セキュリティリスクが高い。今回開発する Web アプリでは、セキュリティ対策の強化を目的に、堅牢で実績のある Ruby on Rails をベースとしたアーキテクチャに刷新する

- 本システム機能の入力・処理・出力のフローチャート

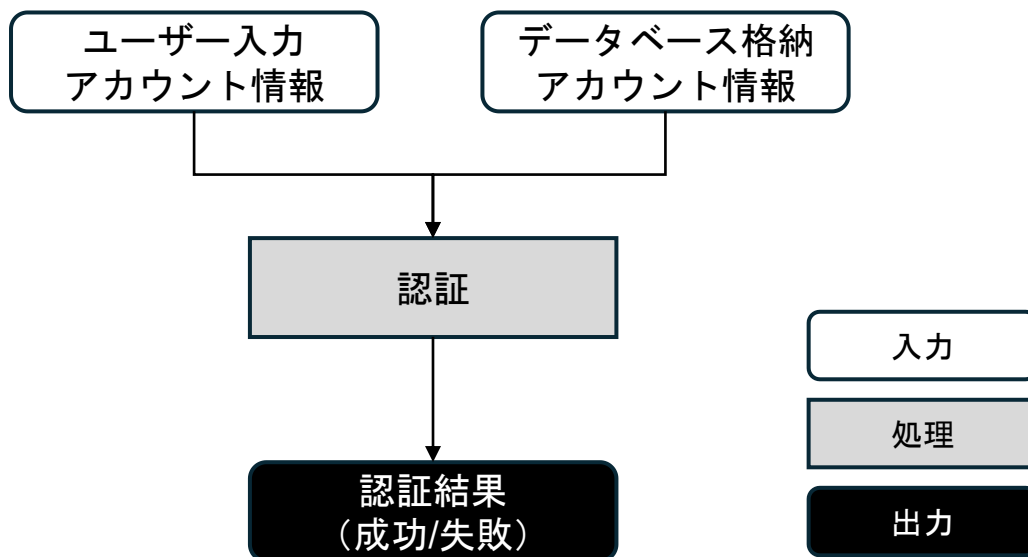


図 2-28 【FN202】 のフローチャート

- 本システム機能の処理の詳細
 - 認証
 - ◇ 処理内容
 - ID とパスワードをデータベースに格納している情報と突合
 - ◇ 利用するライブラリ

- 【SL009】 Devise
 - ◇ 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
 - ◇ ユーザー入力アカウント情報
 - データの内容
 - ID、パスワード
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF009】利用者データ連携用 IF
 - ◇ データベース格納アカウント情報
 - データの内容
 - ID、パスワード
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF009】利用者データ連携用 IF
 - 出力
 - ◇ 認証結果
 - 内容
 - 認証結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN203】送迎確認<改修機能>

- 本システム機能の概要
 - 利用者が乗車する送迎車両の予定情報を取得

既存機能の改修内容

現行の利用者アプリは、PHP による直接実装などアーキテクチャが旧式であり、セキュリティリスクが高い。今回開発する Web アプリでは、セキュリティ対策の強化を目的に、堅牢で実績のある Ruby on Rails をベースとしたアーキテクチャに刷新する

●

- 本システム機能の入力・処理・出力のフローチャート

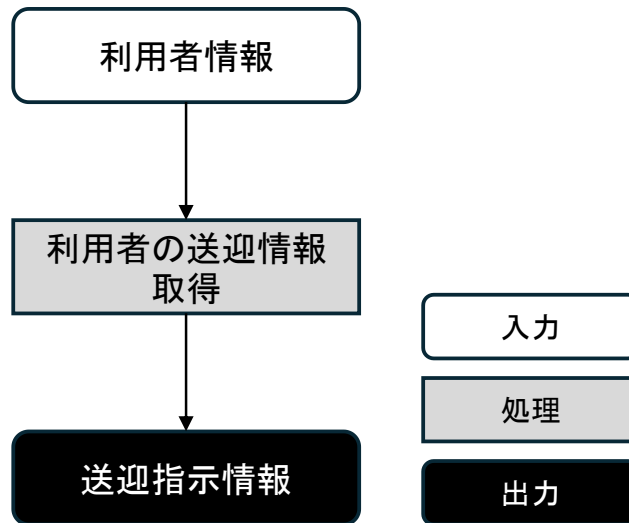


図 2-29 【FN203】のフローチャート

- 本システム機能の処理の詳細
 - 利用者情報
 - ◇ 処理内容
 - 利用者情報から送迎予定情報（時間、車両、車両の現在地）を取得
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - 【SL002】 Ruby on Rails <改修後>
 - ◇ 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
 - ◇ 利用者の送迎予定情報（時間、車両、車両の現在地）
 - データの内容
 - 利用者番号等
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - テキスト
 - 出力
 - ◇ 送迎指示情報
 - データの内容
 - 利用者の送迎する車両等の指示情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF013】送迎指示データ連携用 IF

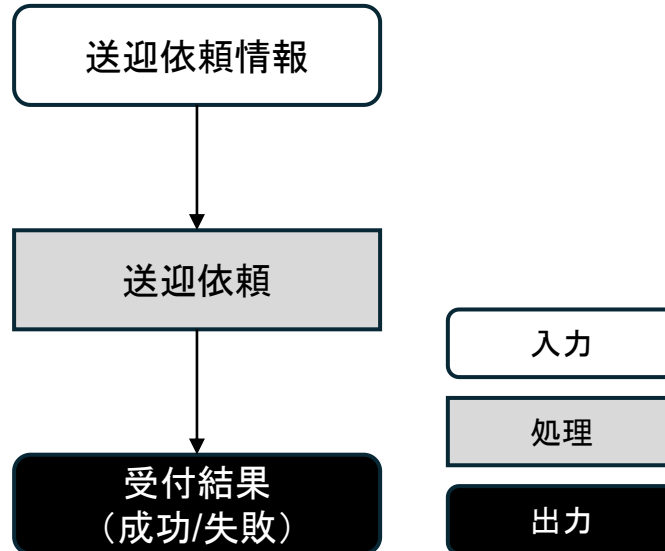
【FN204】送迎依頼<改修機能>

- 本システム機能の概要
 - 利用者が送迎を依頼する機能

既存機能の改修内容

現行の利用者アプリは、PHP による直接実装などアーキテクチャが旧式であり、セキュリティリスクが高い。今回開発する Web アプリでは、セキュリティ対策の強化を目的に、堅牢で実績のある Ruby on Rails をベースとしたアーキテクチャに刷新する

- 本システム機能の入力・処理・出力のフローチャート



● 図 2-30 【FN204】のフローチャート

- 本システム機能の処理の詳細
 - 送迎依頼
 - ◇ 処理内容
 - 利用者の希望情報（乗車位置、降車位置、同乗者人数、車椅子の有無）を取得し、データに格納する
 - 施設管理者ユーザーへ通知する
 - ◇ 利用するライブラリ
 - PHP<改修前>
 - 【SL002】Ruby on Rails <改修後>
 - ◇ 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
 - ◇ 送迎依頼情報
 - データの内容
 - 利用者の希望情報（乗車位置、降車位置、同乗者人数、車椅子の有無）
 - データの形式
 - JSON

- 利用するデータインターフェース
 - 【IF013】 送迎指示データ連携用 IF
- 出力
 - ◇ 受付結果
 - データの内容
 - 受付成功/失敗
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし

【FN301】 来館予定作成バッチ

- 本システム機能の概要
 - 翌月の来館予定を前月に確定する実務を自動化する定期バッチ機能
 - 利用者ごとの週間来館パターン（例：月・水・金）を基に、入力対象期間内の具体的な日付・時刻へ変換し、「利用者来館予定」テーブルへ追加・更新
 - 定期実行を前提とし、再実行時も同一期間の二重登録を行わない
- 本システム機能の入力・処理・出力のフローチャート

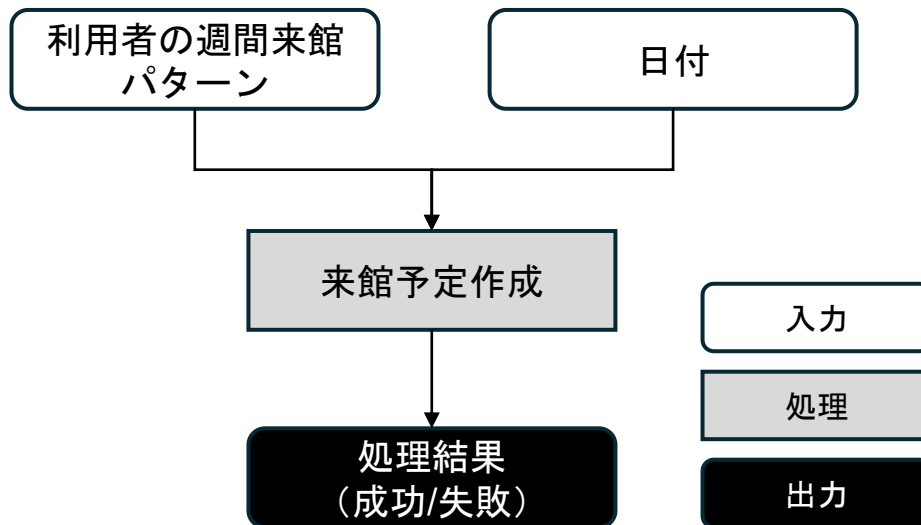


図 2-31 【FN301】 のフローチャート

- 本システム機能の処理の詳細
 - 来館予定作成
 - ◇ 処理内容
 - 指定日（未来日付、範囲指定可）の利用者ごとの週間来館パターンを取得
 - 指定範囲の未来日付の来館予定情報を作成し、データ格納
 - ◇ 利用するライブラリ
 - 【SL010】 Flask
 - ◇ 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
 - ◇ 日付
 - データの内容
 - 指定日（未来日付、範囲指定可）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - なし
 - ◇ 利用者の週間来館パターン
 - データの内容
 - 利用者データから週間来館パターンを抽出
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF009】利用者データ連携用 IF
 - 出力
 - ◇ 来館予定作成
 - データの内容
 - 処理結果（成功/失敗）
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF012】利用者来館予定データ連携用 IF

【FN401】地図表示

- 本システム機能の概要
 - 地図情報を取得して表示する機能
- 本システム機能の入力・処理・出力のフローチャート

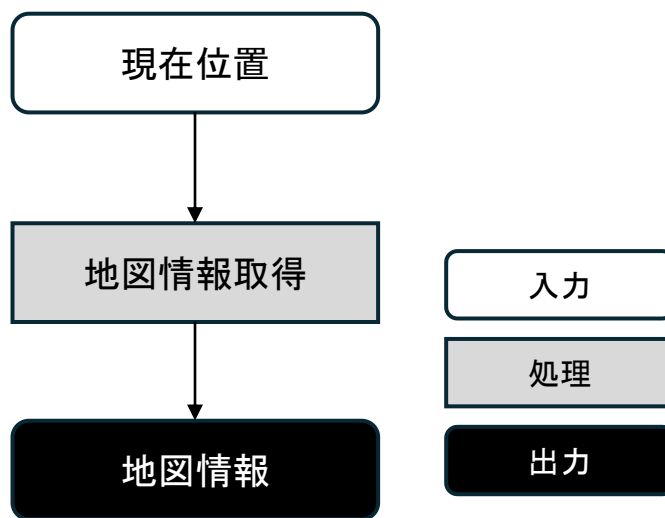


図 2-32 【FN401】のフローチャート

- 本システム機能の処理の詳細
 - 地図情報取得
 - ◇ 処理内容
 - 地図情報の取得
 - ◇ 利用するライブラリ
 - 【SL010】 Flask
 - 【SL006】 Google Maps API
 - ◇ 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
 - ◇ 現在位置
 - データの内容
 - テキスト
 - データの形式
 - HTTP
 - 利用するデータインターフェース
 - 【IF101】 道路マップ
 - 出力
 - ◇ 地図情報
 - データの内容
 - 地図情報
 - データの形式
 - HTTP

- 利用するデータインターフェース
 - 【IF101】 道路マップ

【FN402】 経路検索

- 本システム機能の概要
 - 出発地、経由地、目的地の乗降ポイント間の経路を検索する機能
 - 利用者の乗車・降車順に従い乗降ポイント（誘導経路位置情報）を指定する
- 本システム機能の入力・処理・出力のフローチャート

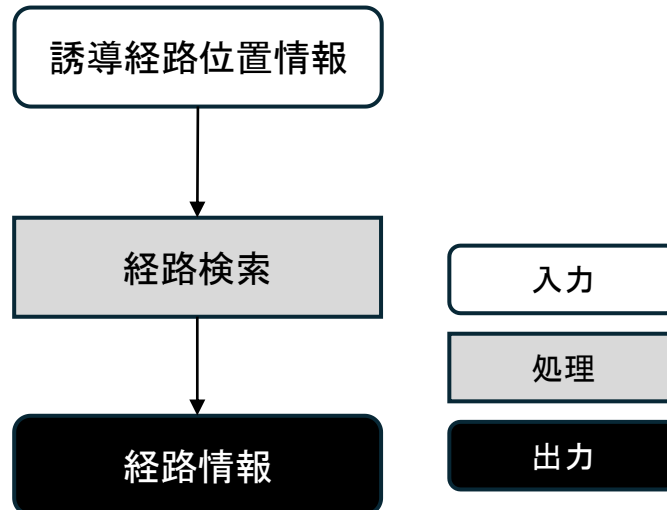


図 2-33 【FN402】 のフローチャート

- 本システム機能の処理の詳細
 - 経路検索
 - ◇ 処理内容
 - 誘導経路（ラストワンマイル（利用者の自宅付近）における通過経路）を指定して、ルートを検索する
 - ◇ 利用するライブラリ
 - 【SL010】 Flask
 - 【SL006】 Google Maps API
 - ◇ 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
 - ◇ 誘導経路位置情報
 - データの内容
 - 緯度・経度等
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF102】 事業所及び乗降ポイント

- 【IF103】 ルート表示、誘導経路及びピンの制御
- 【IF104】 地点間の移動時間や距離
- 出力
 - ◇ 地図情報
 - データの内容
 - 地図情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF102】 事業所及び乗降ポイント
 - 【IF103】 ルート表示、誘導経路及びピンの制御
 - 【IF104】 地点間の移動時間や距離

2-1-4. ソフトウェア・ライブラリ (SL) の詳細

表 2-6 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	Ruby	3.3.0	● オブジェクト指向のプログラミング言語
SL002	Ruby on Rails	7.1.3	● Ruby の Web アプリケーションフレームワーク
SL003	TypeScript	5.0.4	● 静的型付けのプログラミング言語
SL004	React	18.2.0	● オープンソースなフロントエンド JavaScript ライブラリ
SL005	Next.js	14.2.3	● React をベースにしたオープンソースの JavaScript フレームワーク
SL006	Google Maps API	-	● Google Maps の機能を API で利用可能
SL007	Docker	24.0	● コンテナサービス
SL008	MySQL	8.3.0.1	● データストレージ
SL009	Devise	4.9.3	● ユーザー認証
SL010	Flask	3.1.2	● Python の WSGI 系軽量 Web フレームワーク

システムコンポーネントの詳細を記す。なお、本業務において開発（新規・改修）を行うシステムコンポーネントを朱文字で示す。

【SL001】 Ruby

- ベンダー
 - コミュニティ開発（主要開発者：まつもとゆきひろ）
- 公式サイト
 - <https://www.ruby-lang.org/ja/>
- 本ソフトウェア・ライブラリの概要
 - まつもとゆきひろ氏によって設計されたオープンソースのオブジェクト指向スクリプト言語
 - シンプルで直感的な文法と高い表現力を特徴としており、開発者の生産性と満足度を重視している
- 主な機能
 - 完全なオブジェクト指向：数値・文字列など全てをオブジェクトとして扱う
 - ガベージコレクション：メモリ管理を自動化

【SL002】 Ruby on Rails

- ベンダー
 - David Heinemeier Hansson (DHH) によって開発され、以降オープンソースコミュニティ主導
- 公式サイト
 - <https://rubyonrails.org/>
- 概要

- Ruby で書かれた MVC アーキテクチャの Web アプリケーションフレームワーク
- 規約優先 (Convention over Configuration) 思想により、非冗長で高速な開発を実現
- 主な機能
 - Active Record による ORM
 - 内蔵ルーティング・セッション管理
 - RESTful 設計支援

【SL003】 TypeScript

- ベンダー
 - Microsoft
- 公式サイト
 - <https://www.typescriptlang.org/>
- 本ソフトウェア・ライブラリの概要
 - JavaScript に静的型付けを追加し、コンパイル時のエラー検知を強化したスーパーセット言語
 - 既存 JS との完全互換性も保持
- 主な機能
 - 静的型チェック
 - クラスベースのオブジェクト指向サポート

【SL004】 React

- ベンダー
 - Meta
- 公式サイト
 - <https://ja.react.dev/>
- 本ソフトウェア・ライブラリの概要
 - コンポーネントベースの UI ライブラリ
 - 仮想 DOM により高速レンダリングを実現し、宣言的な UI 構築を支援
- 主な機能
 - JSX による直感的な UI 記述
 - ライフサイクルと状態管理

【SL005】 Next.js

- ベンダー
 - Vercel
- 公式サイト
 - <https://nextjs.org/>
- 本ソフトウェア・ライブラリの概要
 - React ベースのフルスタック JavaScript フレームワーク
- 主な機能

- 自動ルーティング

【SL006】 Google Maps API

- ベンダー
 - Google (Google Maps Platform)
- 公式サイト
 - <https://developers.google.com/maps/documentation?hl=ja>
- 本ソフトウェア・ライブラリの概要
 - 地図表示、ジオコーディング、ルート案内、施設検索などをサポートする Web/API プラットフォーム
 - Web アプリに地理情報機能を統合可能
- 主な機能
 - Web アプリに地理情報機能を統合可能
 - プレイス検索・ルート描画

【SL007】 Docker

- ベンダー
 - Docker, Inc.
- 公式サイト
 - <https://www.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コンテナとして実行・管理するための仮想化基盤
 - 開発環境から本番環境まで一貫した動作保証を実現
- 主な機能
 - イメージによる環境の再現性確保
 - コンテナ単位でのアプリケーション分離・管理
 - Docker Hub を通じた公開・共有

【SL008】 MySQL

- ベンダー
 - Oracle Corporation
- 公式サイト
 - <https://www.mysql.com/>
- 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
 - 幅広い Web システムで採用される
- 主な機能
 - トランザクション (ACID) 対応
 - インデックス・ビューによる高速検索
 - ユーザー認証・アクセス制御

【SL009】 Devise

- ベンダー
 - オープンソース (主要開発者: José Valim / Plataformatec による開発、その後コミュニティ主導でメンテナンス)
- 公式サイト
 - <https://github.com/heartcombo/devise>
- 本ソフトウェア・ライブラリの概要
 - Ruby on Rails 向けの柔軟で拡張性の高い認証フレームワーク
 - 認証・認可に関する機能を包括的に提供し、ユーザー登録・ログイン・セッション管理・パスワードリセットなどを容易に実装可能
- 主な機能
 - データベース認証 (メールアドレス・パスワードなど)
 - パスワードリセット、アカウントロック、アカウント確認メール送信

【SL010】 Flask

- ベンダー
 - Pallets Projects
- 公式サイト
 - https://pypi.org/project/Flask/?utm_source=chatgpt.com
- 本ソフトウェア・ライブラリの概要
 - Python の WSGI 系軽量 Web フレームワーク
 - 拡張で API 機能を積み上げる“マイクロ”設計。Jinja2、クリック系 CLI など最小核を提供
- 主な機能
 - ルーティング、リクエスト/レスポンス、コンテキスト、拡張機構 (Blueprint/Extension)

2-1-5. 数理モデル・アルゴリズム（AL）の詳細

表 2-7 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	推薦者表示アルゴリズム	<ul style="list-style-type: none"> 送迎便に追加する推薦利用者を、時間帯と位置の近さに基づいて自動抽出するアルゴリズム 	【FN016】推薦者表示

【AL101】 推薦者表示アルゴリズム

- 概要
 - 送迎便に追加する推薦利用者を、時間帯と位置の近さに基づいて自動抽出するアルゴリズム
 - 選択された利用者の迎え/送り時刻を基準に、設定された時間幅（既定：±30 分）内の利用者から、緯度・経度の直線距離が近い順に上位 N 件（既定：10 件）を提示
 - 「休み」設定の利用者や、当該便（自車）に既に割当て済みの利用者は候補外とし、時間幅・件数は設定画面で変更可能
- 本アルゴリズムを利用した機能
 - 【FN016】 推薦者表示
- アルゴリズムの詳細
 - 選択された利用者の迎え時間又は送り時間から前後 30 分(Windows は設定画面パラメータ)の範囲が対象となり、上記の対象範囲の利用者との緯度・経度を使って直線距離が近い順に 10 件(Windows は設定画面パラメータ)表示させる
 - 休み、自車の利用者は対象外

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

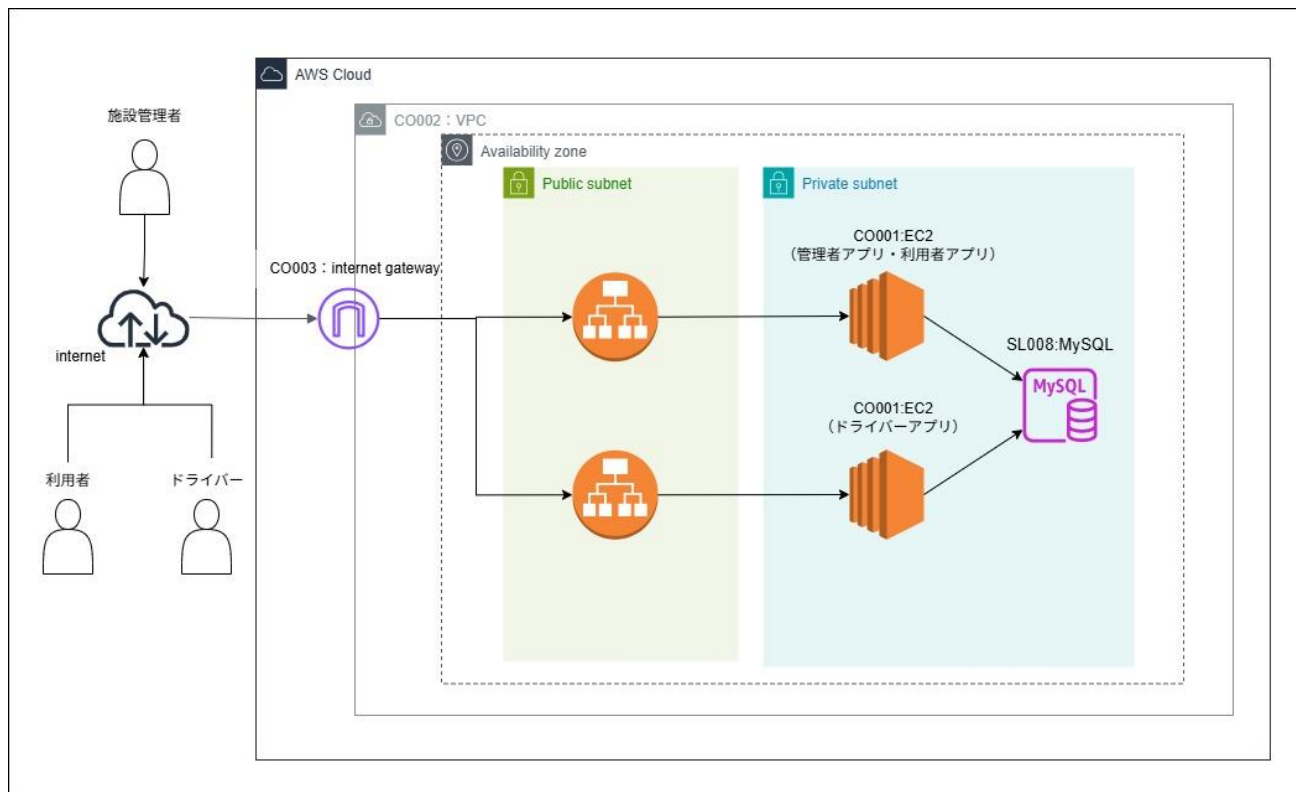


図 2-34 システムコンポーネント図

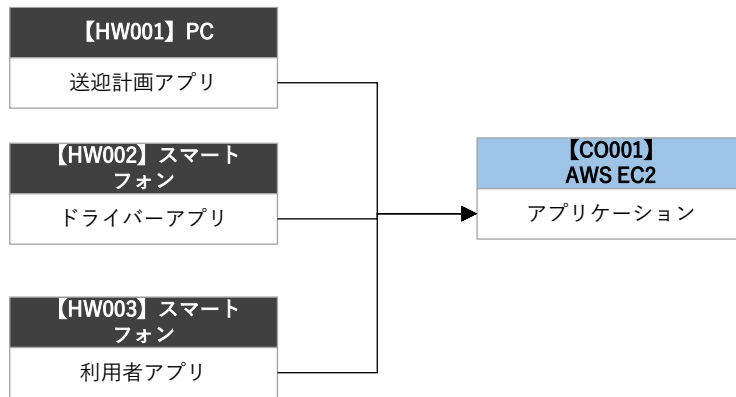
2-2-2. システムコンポーネント一覧

表 2-8 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	仮想サーバー	AWS EC2	<ul style="list-style-type: none"> アプリケーションの実行環境。システムの中核としてアプリケーション処理を担う
CO002	ネットワーク構成	AWS VPC	<ul style="list-style-type: none"> プライベート仮想ネットワーク。外部と隔離された安全な通信空間を提供し、各コンポーネントを接続する基盤
CO003	VPC コンポーネント	インターネットゲートウェイ	<ul style="list-style-type: none"> VPC のインターネット接続。パブリック通信を可能にし、外部サービスとの連携やアップデート等を実現
CO004	ロードバランサー	AWS ALB	<ul style="list-style-type: none"> アプリケーションレイヤーでの負荷分散。複数のサーバーにリクエストを分配し、処理性能と可用性を向上

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



凡例



図 2-35 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 2-9 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	-	-	<ul style="list-style-type: none"> Web アプリケーションの動作
HW002	スマートフォン	Sharp	SH-51C (ドコモ版)	<ul style="list-style-type: none"> ドライバーアプリケーションの動作
HW003	スマートフォン	-	-	<ul style="list-style-type: none"> 利用者ブラウザアプリケーションの動作

2-3-3. ハードウェアの詳細

ハードウェアの詳細を記す。なお、本業務において開発（新規・改修）を行うハードウェアを**朱文字**で示す。

【HW001】 PC

- 本ハードウェアの概要
 - 送迎管理システム(web)へブラウザからアクセスを行う
- ベンダー
 - -
- 仕様・スペック
 - Google Chrome 又は Microsoft Edge にアクセスが可能
 - Google Chrome 推奨
- イメージ



DELL Latitude 3550¹

¹ 公式 HP より抜粋：<https://www.dell.com/ja-jp/shop/dell%E3%81%AE%E3%83%8E%E3%83%BC%E3%83%88%E3%83%91%E3%82%BD%E3%82%B3%E3%83%B3/latitude-3550-%E3%83%8E%E3%83%BC%E3%83%88%E3%83%91%E3%82%BD%E3%82%B3%E3%83%B3/spd/latitude-15-3550-laptop>

【HW002】 スマートフォン：AQUOS wish2 SH-51C

- 本ハードウェアの概要
 - ドライバーアプリケーション(Android)用端末
- ベンダー
 - Sharp
- 仕様・スペック
 - サイズ: 約 147mm×約 71mm×約 8.9mm
 - 約 162g
 - OS : Android™ 12
- イメージ



AQUOS wish2 SH-51C²

² 公式 HP より抜粋：<https://jp.sharp/products/aquos-wish2/d/>

【HW003】スマートフォン

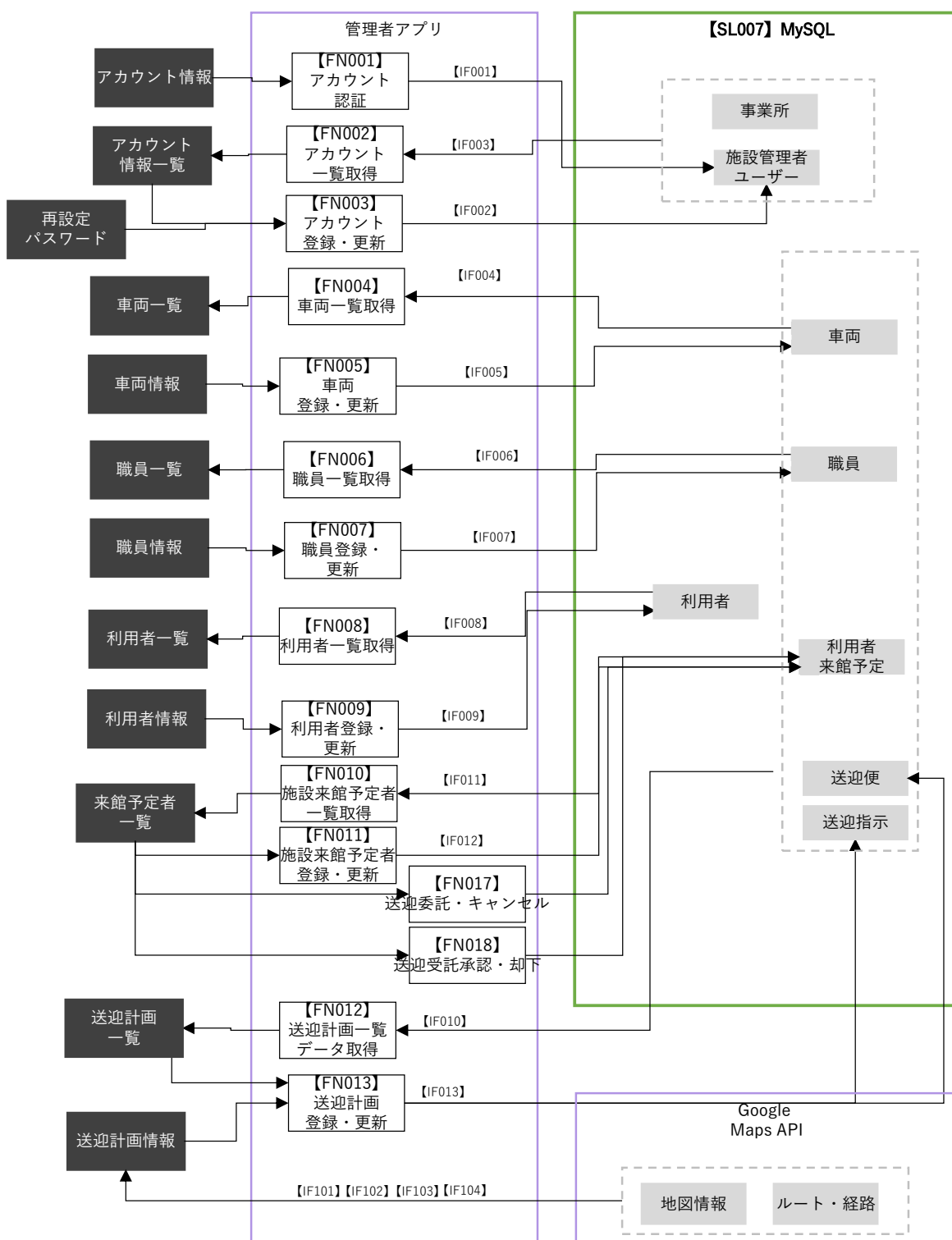
- 本ハードウェアの概要
 - 利用者アプリケーション(web)へブラウザからアクセスを行う
- ベンダー
 - Android
 - iOS
- 仕様・スペック
 - Chrome 又は safari にアクセスが可能
 - Chrome 推奨
- イメージ

NTT
docomo
AQUOS wish2
SH-51C



2-4. データインターフェース (IF)

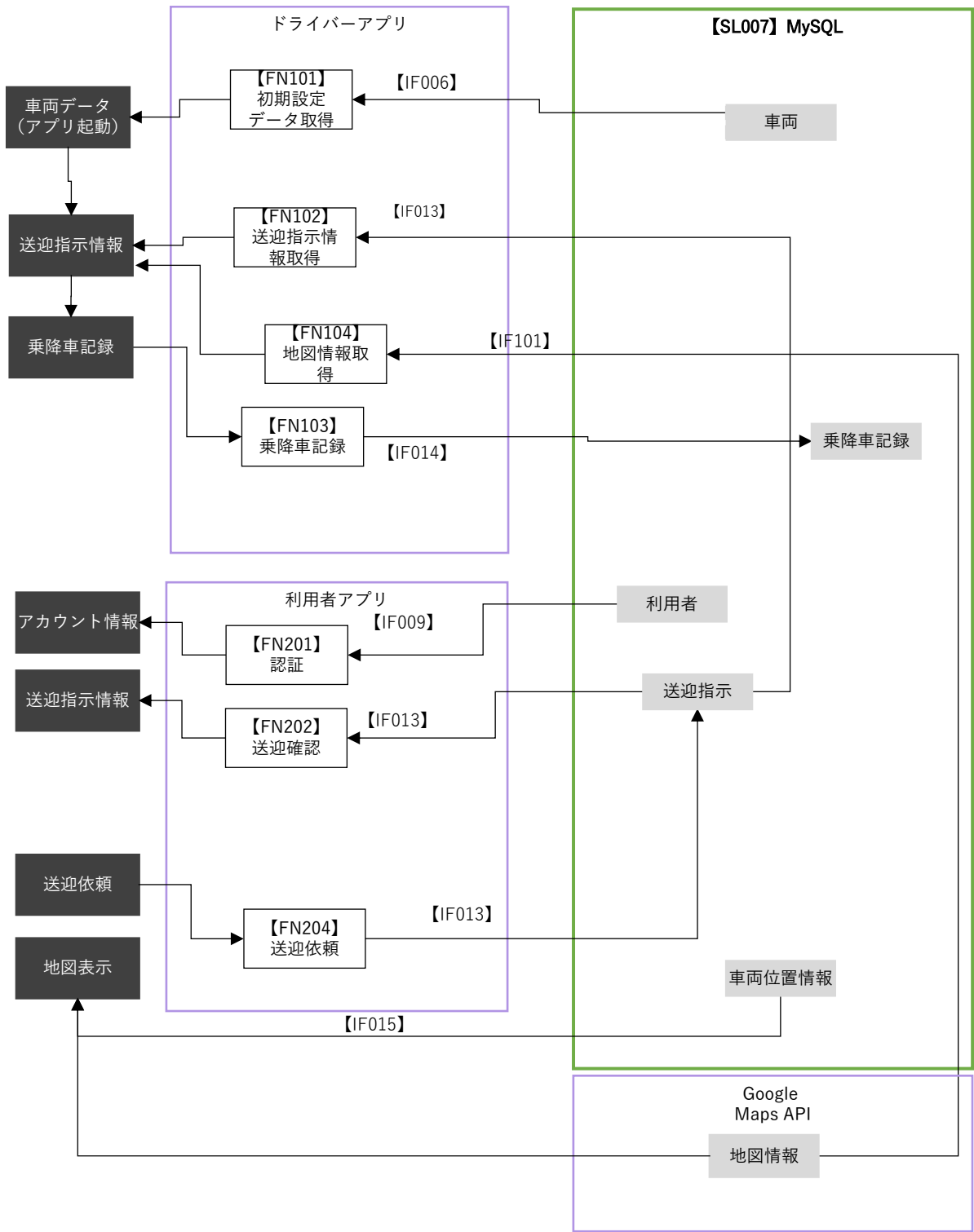
2-4-1. データアーキテクチャ



凡例



図 2-36 データアーキテクチャ (管理者アプリ)



凡例 UI入力 データ **データベース** ファイルストレージ ソフトウェア

図 2-37 データアーキテクチャ (ドライバーアプリ、利用者アプリ)

- IF001 認証IF
- IF002 施設管理者ユーザーデータ連携用IF
- IF003 施設管理者ユーザーデータ一覧取得用IF
- IF016 お知らせデータ連携用IF

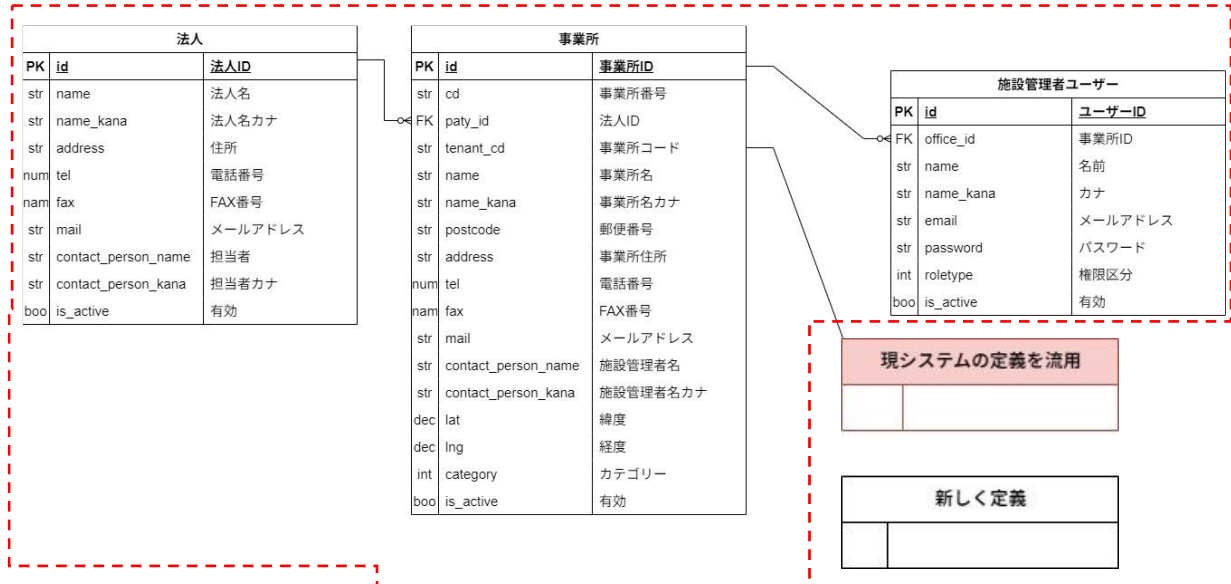


図 2-39 ER 図

(IF001 認証 IF、IF002 施設管理者ユーザーデータ連携用 IF、
IF003 施設管理者ユーザーデータ一覧取得用 IF)

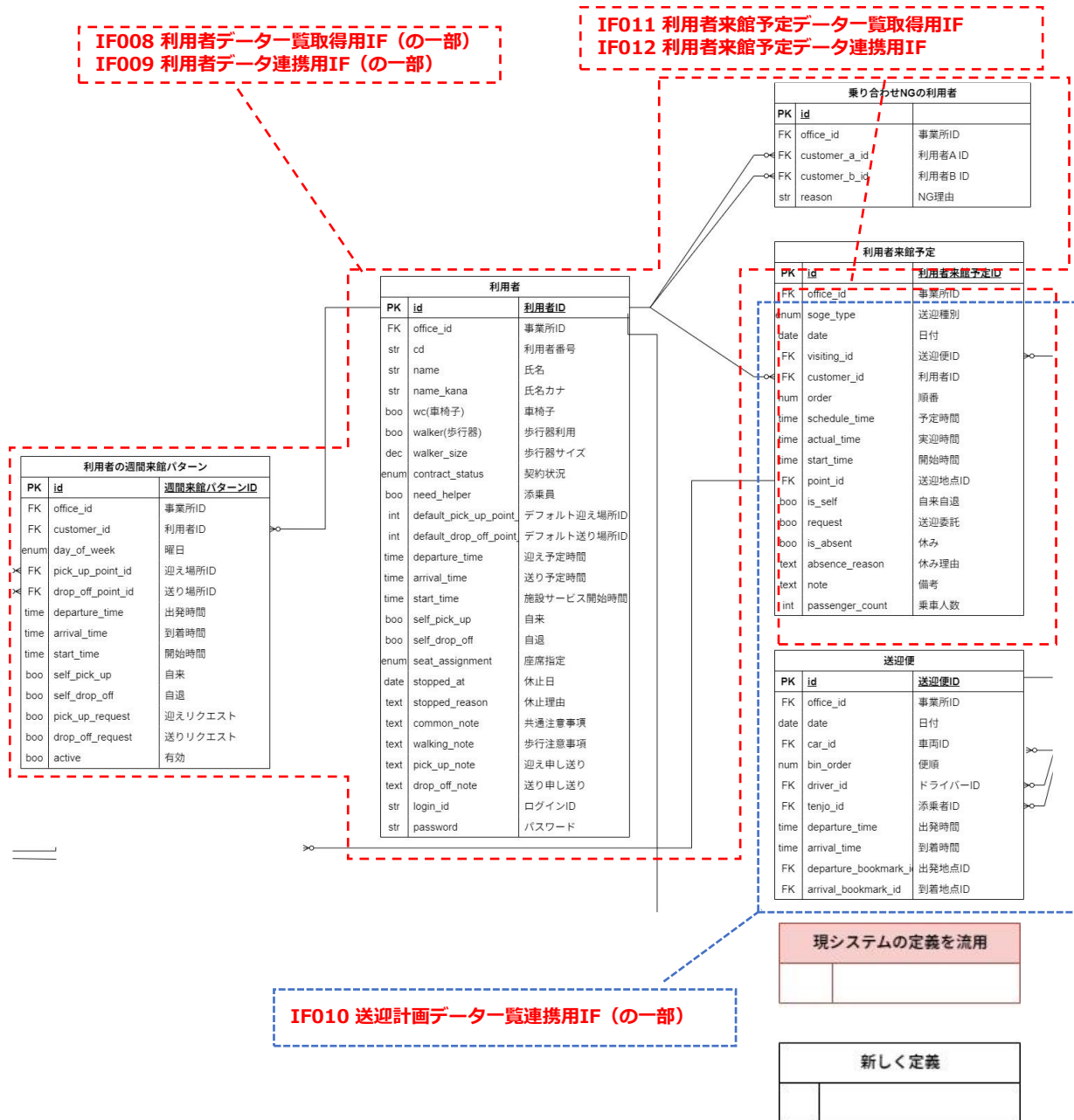


図 2-40 ER 図

(IF008 利用者データ一覧取得用 IF (一部)、IF009 利用者データ連携用 IF (一部)、IF011 利用者来館予定データ一覧取得用 IF、IF012 利用者来館予定データ連携用 IF)

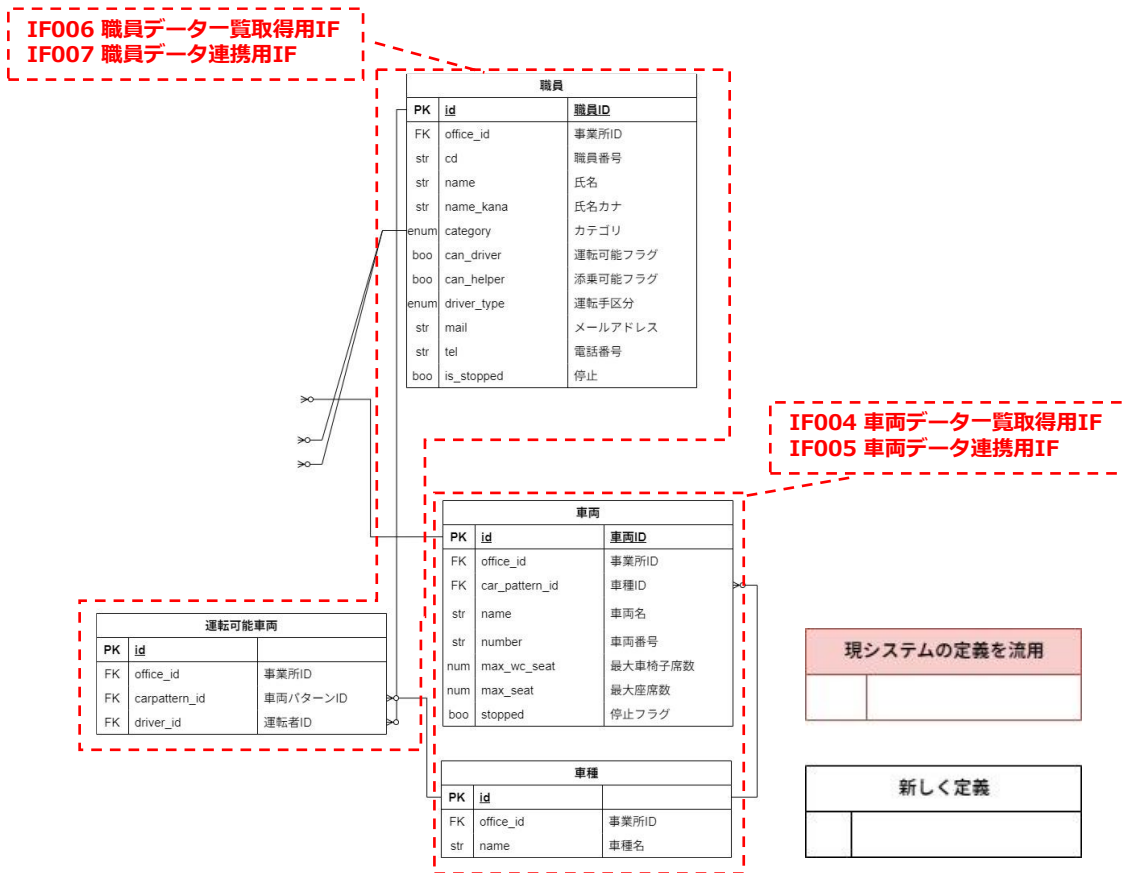


図 2-41 ER 図

(IF004 車両データ一覧取得用 IF、IF005 車両データ連携用 IF、IF006 職員データ一覧取得用 IF、IF007 職員データ連携用 IF)

IF008 利用者データ一覧取得用IF (の一部)
 IF009 利用者データ連携用IF (の一部)

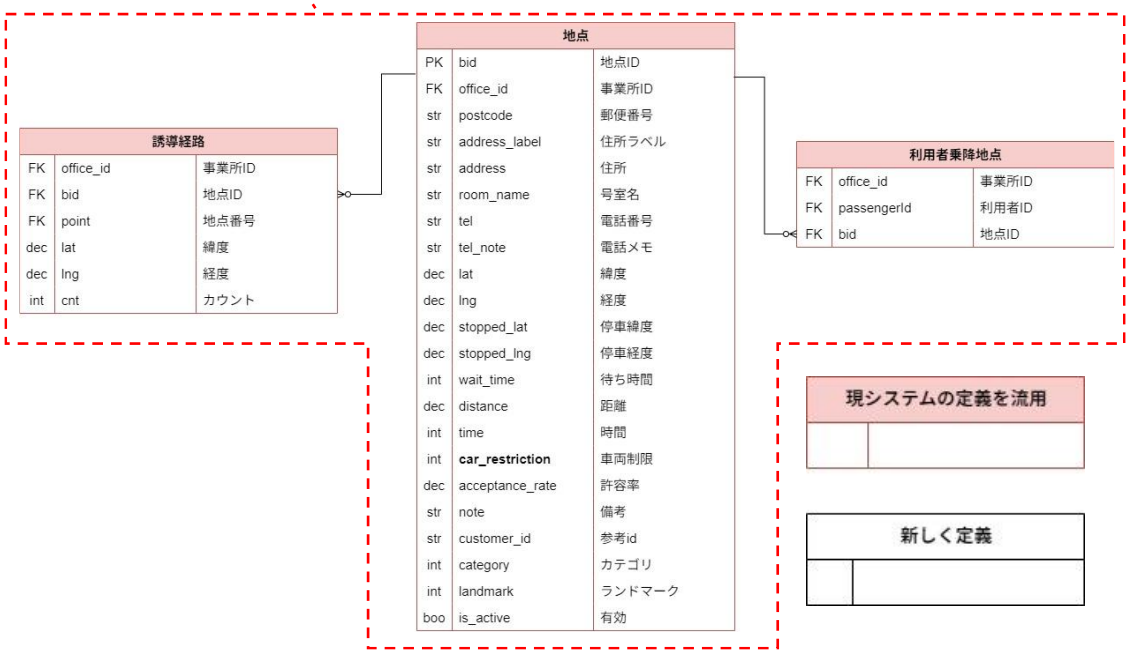


図 2-42 ER 図

(IF008 利用者データ一覧取得用 IF (一部)、IF009 利用者データ連携用 IF (一部))



図

2-43 ER 図

(IF013 送迎指示データ連携用 IF、IF014 乗降車記録データ連携用 IF、IF015 車両位置データ連携用 IF、IF010 送迎計画データ一覧連携用 IF (一部))

2-4-2. データインターフェース一覧

表 2-10 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	認証 IF	SL008	FN001
IF002	施設管理者ユーザーデータ連携用 IF	SL008	FN003
IF003	施設管理者ユーザーデータ一覧取得用 IF	FN002	SL008
IF004	車両データ一覧取得用 IF	FN004	SL008
IF005	車両データ連携用 IF	SL008	FN005
IF006	職員データ一覧取得用 IF	FN006	SL008
IF007	職員データ連携用 IF	SL008	FN007
IF008	利用者データ一覧取得用 IF	FN008	SL008
IF009	利用者データ連携用 IF	SL008	FN009
IF010	送迎計画データ一覧連携用 IF	FN012	SL008
IF011	利用者来館予定データ一覧取得用 IF	FN010	SL008
IF012	利用者来館予定データ連携用 IF	SL008	FN011
IF013	送迎指示データ連携用 IF	SL008	FN013
IF014	乗降車記録データ連携用 IF	SL008	FN103
IF015	車両位置データ連携用 IF	SL008	FN104
IF016	お知らせデータ連携用 IF	SL008	FN019
IF101	道路マップ	FN001 FN104 FN201	FN501
IF102	事業所及び乗降ポイント	FN001	FN501
IF103	ルート表示、誘導経路及びピンの制御	FN001	FN501
IF104	地点間の移動時間や距離	FN001	FN501

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】 認証 IF

- 本インターフェースの概要
 - アカウント認証情報を連携する IF
 - 施設管理者ユーザーがログインの際に利用する
- 本インターフェースを利用する機能
 - **【FN002】** アカウント認証
- 本 API の共通仕様
 - 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
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
E メールアドレス	email	ユーザーの E メールアドレス	文字列	○
パスワード	password	ユーザーのパスワード	文字列	○

```
{
  "user": {
    "email": ""
    "password": ""
  }
}
```

- レスポンスパラメータ

項目	名称	説明	値
メッセージ	messages	レスポンスメッセージ	文字列
ID	id	ユーザーの ID	数値
E メールアドレス	email	ユーザーの E メールアドレス	文字列

名前	name	ユーザーの名前	文字列
フリガナ	kana	ユーザーのフリガナ	文字列
作成日時	created_at	データの作成日時	日付
更新日時	updated_at	データの更新日時	日付

◇ 成功時

```
{
  "messages": [
    "ログインに成功しました"
  ],
  "user": {
    "id": "",
    "email": "",
    "name": "",
    "kana": "",
    "created_at": "",
    "updated_at": ""
  }
}
```

◇ 失敗時

```
{
  "success": false,
  "error": "メールアドレスまたはパスワードが違います"
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-11 認証 IF 項目

論理名	物理名	型(サイズ)
ユーザーID	id	Integer
email	email	String
パスワード	password	String
名前	name	String
名前カナ	kana	String
権限区分	roletype	String

※権限区分は、「システム管理者」「法人管理者」「一般」を想定

【IF002】 施設管理者ユーザーデータ連携用 IF

- 本インターフェースの概要
 - システムにログインする施設管理者ユーザーのデータを連携する IF
- 本インターフェースを利用する機能
 - **【FN004】** アカウント登録・更新
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ 登録: POST
 - ◇ 編集: PUT
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
ID	id	ユーザーの ID	数値	○
名前	name	ユーザーの名前	文字列	○
フリガナ	kana	ユーザーのフリガナ	文字列	
役割	roletype	ユーザーのロールタイプ	数値	○

```
{
  "user": {
    "id": ""
    "name": ""
    "kana": ""
    "roletype": ""
  }
}
```

- レスポンスパラメータ

項目	名称	説明	値
メッセージ	messages	レスポンスメッセージ	文字列

◇ 成功時

```
{
  "messages":["ユーザー を登録しました"]
}
```

◇ 失敗時

```
{
  " error_messages ":["エラーメッセージ"]
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-12 ユーザーデータ連携用 IF 項目

論理名	物理名	型(サイズ)
ユーザーID	id	Integer(PK)
法人 ID	party_id	Integer
事業所 ID	office_id	Integer(FK)
事業所緯度	lat	Decimal
事業所経度	lng	Decimal
email	email	String
パスワード	Password	String
名前	name	String
名前カナ	kana	String
権限区分	roletype	String

※権限区分は、「システム管理者」「法人管理者」「一般」を想定

【IF003】施設管理者ユーザーデータ一覧取得用 IF

- 本インターフェースの概要
 - システムにログインするユーザー一覧データを連携する IF
- 本インターフェースを利用する機能
 - **【FN003】** アカウント一覧取得
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8

➤ リクエストパラメータ

◇ Body

項目	名称	説明	値	必須
法人 ID	party_id	法人の ID	数値	○
事業所 ID	office_id	事業所の ID	数値	○

```
{
  "serch_params": {
    "party_id": ""
    "office_id": ""
  }
}
```

➤ レスポンスパラメータ

項目	名称	説明	値
ID	id	ユーザーの ID	数値
名前	name	ユーザーの名前	文字列
フリガナ	kana	ユーザーのフリガナ	文字列
役割	roletype	ユーザーのロールタイプ	数値

◇ 成功時

```
{
  "user": [
    {
      "id": ""
      "name": ""
      "kana": ""
      "roletype": ""
    },
  ]
}
```

◇ 認証エラー

```
{
  "success": false,
  "error": "Unauthenticated"
}
```

◇ サーバーエラー

```
{
  "status": "error",
```

```

"message": "サーバーエラーが発生しました"
}
    
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-13 ユーザーデータ一覧取得用 IF 項目

論理名	物理名	型(サイズ)
法人 ID	party_id	Integer
事業所 ID	office_id	Integer
ユーザー データ項目	Users 【IF002】施設管理者ユーザーデータ連携用 IF 参照	(リスト)

【IF004】車両データ一覧取得用 IF

- 本インターフェースの概要
 - 送迎車両一覧を取得するデータ連携用 IF
- 本インターフェースを利用する機能
 - 【FN005】車両一覧取得
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
法人 ID	party_id	法人の ID	数値	○
事業所 ID	office_id	事業所の ID	数値	○

```

{
  "serch_params": {
    "party_id": ""
    "office_id": ""
  }
}
    
```

➤ レスポンスパラメータ

項目	名称	説明	値
ID	id	車両の ID	数値
車両名	name	車両の呼び名	文字列
車両番号	number	車両の番号	文字列
最大車椅子数	max_wc_seat	車両の最大車椅子数	数値
最大座席数	max_seat	車両の最大座席数	数値

◇ 成功時

```
{
  "cars": [
    {
      "id": ""
      "name": ""
      "number": ""
      "max_wc_seat": ""
      "max_seat": ""
    },
  ]
}
```

◇ 認証エラー

```
{
  "success": false,
  "error": "Unauthenticated"
}
```

◇ サーバーエラー

```
{
  "status": "error",
  "message": "サーバーエラーが発生しました"
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-14 車両データ一覧取得用 IF 項目

論理名	物理名	型(サイズ)
法人 ID	party_id	Integer
事業所 ID	office_id	Integer
車両 ID	id	Integer
車両データ データ項目	car_data 【IF005】車両データ連携用 IF を参照	(リスト)

【IF005】 車両データ連携用 IF

- 本インターフェースの概要
 - 送迎に活用する車両データを連携する IF
- 本インターフェースを利用する機能
 - **【FN006】** 車両登録・更新
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ 登録: POST
 - ◇ 更新: PUT
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
ID	id	車両の ID	数値	○
車両名	name	車両の呼び名	文字列	○
車両番号	number	車両の番号	文字列	○
最大車椅子数	max_wc_seat	車両の最大車椅子数	数値	○
最大座席数	max_seat	車両の最大座席数	数値	○

```

{
  "car": {
    "id": ""
    "name": ""
    "number": ""
    "max_wc_seat": ""
    "max_seat": ""
  }
}
    
```

➤ レスポンスパラメータ

項目	名称	説明	値
メッセージ	messages	レスポンスメッセージ	文字列

◇ 成功時

```
{
  "messages":["車両 を登録しました"]
}
```

◇ 失敗時

```
{
  " error_messages ":["エラーメッセージ"]
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-15 車両データ連携用 IF 項目

論理名	物理名	型(サイズ)
車両 ID	id	Integer(PK)
法人 ID	party_id	Integer
事業所 ID	office_id	Integer(FK)
車種 ID	car_pattern_id	Integer(FK)
車両名	name	String
車両番号	number	String
最大車椅子数	max_wc_seat	Integer
最大座席数	max_seat	Integer

【IF006】 職員データ一覧取得用 IF

- 本インターフェースの概要
 - 施設職員の一覧を取得するデータ IF
- 本インターフェースを利用する機能
 - **【FN007】 職員一覧取得**
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード

- ◇ UTF-8
- リクエストパラメータ
- ◇ Body

項目	名称	説明	値	必須
事業所 ID	office_id	事業所の ID	数値	○
職員番号	cd	事業所の職員番号	文字列	

```
{
  "serch_params": {
    "office_id": ""
    "cd": ""
  }
}
```

- レスポンスパラメータ

項目	名称	説明	値
ID	id	職員の ID	数値
名前	name	職員の名前	文字列
フリガナ	kana	職員のフリガナ	文字列
運転可能区分	can_driver	職員が運転可能か真偽	数値
添乗可能区分	can_helper	職員が添乗可能か真偽	数値

- ◇ 成功時

```
{
  "staffs": [
    {
      "id": ""
      "name": ""
      "kana": ""
      "category": ""
      "can_driver": ""
      "can_helper": ""    },
  ]
}
```

- ◇ 認証エラー

```
{
  "success": false,
  "error": "Unauthenticated"
}
```

◇ サーバーエラー

```
{
  "status": "error",
  "message": "サーバーエラーが発生しました"
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-16 職員データ一覧取得用 IF 項目

論理名	物理名	型(サイズ)
法人 ID	party_id	Integer
事業所 ID	office_id	Integer
職員	staffs	(リスト)
データ項目	【IF007】職員データ連携用 IF 参照	

【IF007】職員データ連携用 IF

- 本インターフェースの概要
 - 車両を運転又は同乗する施設の従業員データを連携する IF
- 本インターフェースを利用する機能
 - 【FN008】職員登録・更新
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ 登録: POST
 - ◇ 更新: PUT
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
ID	id	職員の ID	数値	○
法人 ID	party_id	法人の ID	数値	○
事業所 ID	office_id	事業所の ID	数値	○

```
{
  "staffs": {
    "id": ""
    "party_id": ""
    "office_id": ""
    (以下職員データ項目)
  }
}
```

➤ レスポンスパラメータ

項目	名称	説明	値
メッセージ	messages	レスポンスメッセージ	文字列

◇ 成功時

```
{
  "messages":["職員を登録しました"]
}
```

◇ 失敗時

```
{
  " error_messages ":["エラーメッセージ"]
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-17 職員データ連携用 IF 項目

論理名	物理名	型(サイズ)
職員 ID	id	Integer(PK)
法人 ID	party_id	Integer
事業所 ID	office_id	Integer(FK)
氏名	name	String
氏名カナ	name_kana	String
カテゴリ	category	Integer
運転可能フラグ	can_driver	Boolean
添乗可能フラグ	can_helper	Boolean
運転手区分	driver_type	Integer
メールアドレス	mail	String
電話番号	tel	String
停止	is_stopped	Boolean
運転可能車両	can_driving_cars	(繰り返し)
車両パターン	car_pattern_id	Integer

※運転手区分：介護職員、専属運転手等の補足情報

【IF008】利用者データ一覧取得用 IF

- 本インターフェースの概要
 - 利用者の一覧を取得するデータ IF
- 本インターフェースを利用する機能
 - **【FN009】**利用者一覧取得
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
法人 ID	party_id	法人の ID	数値	○
事業所 ID	office_id	事業所の ID	数値	○

```
{
  "serch_params": {
    "party_id": ""
    "office_id": ""
  }
}
```

- レスポンスパラメータ

項目	名称	説明	値
ID	id	利用者 ID	文字列
法人 ID	party_id	法人の ID	数値
事業所 ID	office_id	事業所の ID	数値
利用者番号	customer_cd	事業所の利用者管理番号	文字列

- ◇ 成功時

```
{
  "customers": [
    {
```

```

    "id": ""
    "party_id": ""
    "office_id": ""
    "customers_cd": ""
    (以下利用者データ項目)
  },
]
}

```

◇ 認証エラー

```

{
  "success": false,
  "error": "Unauthenticated"
}

```

◇ サーバーエラー

```

{
  "status": "error",
  "message": "サーバーエラーが発生しました"
}

```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-18 利用者データ一覧取得用 IF 項目

論理名	物理名	型(サイズ)
法人 ID	party_id	Integer
事業所 ID	office_id	Integer
利用者	customers	(リスト)
「利用者の項目」	【IF009】利用者データ連携用 IF 参照	

【IF009】利用者データ連携用 IF

- 本インターフェースの概要
 - 施設の利用者データを連携する IF
- 本インターフェースを利用する機能
 - 【FN010】利用者登録・更新
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ 登録: POST
 - ◇ 更新: PUT
 - レスポンスデータ形式

- ◇ JSON
- 文字コード
 - ◇ UTF-8
- リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
ID	id	利用者 ID	文字列	○
事業所 ID	office_id	事業所の ID	数値	○
利用者番号	customer_cd	事業所の利用者管理番号	文字列	○

```
{
  "customers": {
    "id": ""
    "office_id": ""
    "customers_cd": ""
    (以下利用者データ項目)
  }
}
```

- レスポンスパラメータ

項目	名称	説明	値
メッセージ	messages	レスポンスメッセージ	文字列

◇ 成功時

```
{
  "messages":["利用者 を登録しました"]
}
```

◇ 失敗時

```
{
  " error_messages ":["エラーメッセージ"]
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-19 利用者データ連携用 IF 項目

論理名	物理名	型(サイズ)
利用者 ID	customer_id	Integer(PK)
法人 ID	party_id	Integer

施設送迎共同管理システム システム設計書

事業所 ID	office_id	String(FK)
利用者番号	customers_cd	String
email	email	String
パスワード	password	String
名前	name	String
名前カナ	name_kana	String
車椅子利用	wc	Boolean
歩行器利用	walker	Boolean
契約状況	contract_status	Integer
添乗員の必要	need_helper	Boolean
デフォルト迎え場所 ID	default_pick_up_point_id	Integer
デフォルト送り場所 ID	default_drop_off_point_id	Integer
迎え予定時間	departure_time	Time
送り予定時間	arrival_time	Time
施設サービス開始時間	start_time	Time
自来	self_pick_up	Boolean
自退	self_drop_off	Boolean
座席指定	seat_assignment	Enum
休止日	stopped_at	Date
休止理由	stopped_reason	Text
共通注意事項	common_note	Text
歩行注意事項	walking_note	Text
迎え申し送り事項	pick_up_note	Text
送り申し送り事項	drop_off_note	Text
利用者の週間来館パターン	visiting_patterns	(リスト)
来館パターン-ID	id	Integer(PK)
来館パターン-曜日	day_of_week	Integer
来館パターン-迎え場所 ID	pick_up_point_id	Integer
来館パターン-送り場所 ID	drop_off_point_id	Integer
来館パターン-出発時間	departure_time	Time
来館パターン-到着時間	arrival_time	Time
来館パターン-開始時間	start_time	Time
来館パターン-自来	self_pick_up	Boolean
来館パターン-自退	self_drop_off	Boolean
来館パターン-迎えリクエスト	pick_up_request	Boolean
来館パターン-送りリクエスト	drop_off_request	Boolean
乗り合わせ NG の利用者	customer_ngs	(リスト)
NG 利用者-NG 利用者 ID	customer_a_id	Integer

NG 利用者-NG 理由	reason	String
乗降地点	bindatad	(リスト)
乗降地点-郵便番号	postcode	String
乗降地点-住所ラベル	address_label	String
乗降地点-住所	address	String
乗降地点-号室名	room_name	String
乗降地点-電話番号	tel	String
乗降地点-電話メモ	tel_note	String
乗降地点-緯度	lat	Decimal
乗降地点-経度	lng	Decimal
乗降地点-停車緯度	stopped_lat	Decimal
乗降地点-停車経度	stopped_lng	Decimal
乗降地点-待ち時間	wait_time	Time
乗降地点-距離	distance	Decimal
乗降地点-時間	time	Integer
乗降地点-車両制限	car_restriction	Integer
乗降地点-許容率	acceptance_rate	Decimal
乗降地点-備考	note	String
乗降地点-参考 ID	customer_id	String
乗降地点-カテゴリ	category	Integer
乗降地点-ランドマーク	landmark	Integer
誘導経路	keiroposdata	(リスト)
誘導経路-地点番号	point	String
誘導経路-緯度	lat	Decimal
誘導経路-経度	lng	Decimal
誘導経路-カウント数	cnt	Integer

【IF010】送迎計画データ一覧連携用 IF

- 本インターフェースの概要
 - 送迎計画データを作成するための情報を一括取得するデータ IF
- 本インターフェースを利用する機能
 - 【FN012】送迎計画一覧取得
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード

- ◇ UTF-8
- リクエストパラメータ
- ◇ Body

項目	名称	説明	値	必須
事業所 ID	office_id	事業所の ID	数値	○
日付	date	日付 (yyyyMMdd)	日付	○

```
{
  "serch_params": {
    "office_id": ""
    "date": ""
  }
}
```

- レスポンスパラメータ

項目	名称	説明	値
事業所 ID	office_id	事業所の ID	数値
日付	date	日付 (yyyyMMdd)	日付
来館利用者リスト	visitings_customers	来館予定の利用者リスト	オブジェクト
ID	id	予定 ID	数値
送迎タイプ	soge_type	送迎タイプ	数値
送迎便 ID	visiting_id	送迎便 ID	数値
送迎便リスト	visitings	送迎便のリスト	オブジェクト
車両 ID	car_id	車両の ID	数値
便順	bin_order	送迎便の順番	数値
運転手 ID	driver_id	職員の ID	数値

- ◇ 成功時

```
{
  "office_id": ""
  "date": ""
  " visitings_customers ": [
    {
      "id": ""
      "soge_type": ""
      "visiting_id": ""
      (以下利用者来館予定データ項目)
    }
  ]
}
```

```

    },
    "visitings": [
    {
        "car_id": ""
        "bin_order": ""
        "driver_id": ""
        (以下送迎便データ項目)
    },
    ]
}

```

◇ 認証エラー

```

{
    "success": false,
    "error": "Unauthenticated"
}

```

◇ サーバーエラー

```

{
    "status": "error",
    "message": "サーバーエラーが発生しました"
}

```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-20 送迎計画データ一覧連携用 IF 項目

論理名	物理名	型(サイズ)
事業所 ID	office_id	Integer
日付	date	Date
利用者来館予定	visitings_customers	(リスト)
来館予定-利用者来館予定 ID	id	Integer(PK)
来館予定-送迎種別	soge_type	String
来館予定-日付	date	String
来館予定-送迎便 ID	visiting_id	String(FK)
来館予定-利用者 ID	customer_id	String(FK)
来館予定-順番	order	Integer
来館予定-予定時間	schedule_time	Time
来館予定-実際時間	actual_time	Time
来館予定-開始時間	start_time	Time
来館予定-送迎地点 ID	point_id	Integer(FK)
来館予定-自来・自退	is_self	Boolean
来館予定-送迎委託	request	Boolean
来館予定-休み	is_absent	Boolean

来館予定-休み理由	absence_reason	String
来館予定-備考	note	String
送迎便	visitings	(リスト)
送迎便-送迎便 ID	bin_id	Integer(PK)
送迎便-車両 ID	car_id	Integer(FK)
送迎便-便順	bin_order	Integer
送迎便-運転手 ID	driver_id	Integer(FK)
送迎便-添乗者 ID	tenjo_id	Integer(FK)
送迎便-出発時間	departure_time	Time
送迎便-到着時間	arrival_time	Time
送迎便-出発地点 ID	departure_bookmark_id	Integer(FK)
送迎便-到着地点 ID	arrival_bookmark_id	Integer(FK)

【IF011】利用者来館予定データ一覧取得用 IF

- 本インターフェースの概要
 - 来館予定者の一覧を取得するデータ IF
- 本インターフェースを利用する機能
 - **【FN011】施設来館予定者一覧取得**
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
事業所 ID	office_id	事業所の ID	数値	○
日付	date	日付 (yyyymmdd)	日付	○

```
{
  "serch_params": {
    "office_id": ""
    "date": ""
  }
}
```

```
}
}
```

➤ レスポンスパラメータ

項目	名称	説明	値
ID	id	利用者予定の ID	数値
事業所 ID	office_id	事業所の ID	数値
送迎タイプ	soge_type	送迎タイプ	数値
日付	date	日付 (yyyymmdd)	日付

◇ 成功時

```
{
  "visitings_customers": [
    {
      "id": ""
      "office_id": ""
      "soge_type": ""
      "date": ""
      (以下利用者来館予定データ項目)
    },
  ]
}
```

◇ 認証エラー

```
{
  "success": false,
  "error": "Unauthenticated"
}
```

◇ サーバーエラー

```
{
  "status": "error",
  "message": "サーバーエラーが発生しました"
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-21 利用者来館予定一覧データ一覧取得用 IF 項目

論理名	物理名	型(サイズ)
施設 ID	id	Integer
日付	date	Date
利用者来館予定	visitings_customers	List

【IF012】利用者来館予定データ連携用 IF

- 本インターフェースの概要
 - 利用者の1日の来館予定データを連携する IF
- 本インターフェースを利用する機能
 - 【FN012】施設来館予定者登録・更新
 - 【FN017】送迎委託・キャンセル
 - 【FN018】送迎受託承認・却下
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ 登録: POST
 - ◇ 更新: PUT
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
ID	id	予定の ID	数値	○
送迎タイプ	soge_type	送迎タイプ	数値	○
日付	date	日付 (yyyymmdd)	日付	○
利用者 ID	customer_id	利用者の ID	数値	○

```

{
  "visitings_customers": {
    "id": ""
    "soge_type": ""
    "date": ""
    "customer_id": ""
    (以下、利用者来館予定データ項目)
  }
}
    
```

➤ レスポンスパラメータ

項目	名称	説明	値
メッセージ	messages	レスポンスメッセージ	文字列

◇ 成功時

```
{
  "messages":["利用者来館予定を登録しました"]
}
```

◇ 失敗時

```
{
  " error_messages ":["エラーメッセージ"]
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-22 利用者来館予定データ連携用 IF 項目

論理名	物理名	型(サイズ)
利用者来館予定 ID	id	Integer(PK)
送迎種別	soge_type	String
日付	date	String
送迎便 ID	visiting_id	String(FK)
利用者 ID	customer_id	String(FK)
順番	order	Integer
予定時間	schedule_time	Time
実際時間	actual_time	Time
開始時間	start_time	Time
送迎地点 ID	point_id	Integer(FK)
自来・自退	is_self	Boolean
送迎委託	request	Boolean
休み	is_absent	Boolean
休み理由	absence_reason	String
備考	note	String

【IF013】送迎指示データ連携用 IF

- 本インターフェースの概要
 - 送迎便における個々の利用者の送迎案内・指示・利用者向け注意情報を連携する IF
- 本インターフェースを利用する機能
 - 【FN102】送迎指示情報取得
- 本 API の共通仕様
 - プロトコル

- ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
事業所 ID	office_id	事業所の ID	数値	○
日付	date	日付 (yyyymmdd)	日付	○
車両 ID	car_id	車両の ID	数値	○

```
{
  "serch_params": {
    "office_id": ""
    "date": ""
    "car_id": ""
  }
}
```

- レスポンスパラメータ

項目	名称	説明	値
事業所 ID	office_id	事業所の ID	数値
日付	date	日付 (yyyymmdd)	日付
車両番号	car_no	車両の番号	文字列
車両 ID	car_id	車両の ID	数値

- ◇ 成功時

```
{
  "mergedata": {
    "office_id": ""
    "date": ""
    "car_no": ""
    "car_id": ""
    (以下送迎指示データ項目)
  }
}
```

}

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-23 送迎指示データ連携用 IF 項目

論理名	物理名	型(サイズ)
事業所 ID	office_id	String(PK)
日付	date	Date(PK)
車両番号	car_no	String
車両 ID	car_id	Integer(PK)
車両名	car_name	String
定員	seat	Integer
車椅子定員	wc_seat	Integer
乗車人数	passenger_count	Integer
車椅子人数	wc_passenger_count	Integer
運転手	driver_name	String
添乗員	tenjo_name	String
便順	bin_order	Integer
表示順	order	Integer
出発時間	departure_time	Time
予到着時間	arrival_time	Time
予定時間	schedule_time	Time
利用者番号	customer_id	String(PK)
利用者名	customer_name	String
フリガナ	customer_kana	String
送迎区分	soge_type	Integer(PK)
設定時刻	setting_time	Time
開始時刻	start_time	Time
乗車利用者 ID	pick_up_id	Integer
乗車利用者名	pick_up_name	String
乗車利用者住所	pick_up_address	String
降車利用者 ID	drop_off_id	Integer
降車利用者名	drop_off_name	String
降車利用者住所	drop_off_address	String
乗車緯度	pick_up_lat	Decimal
乗車経度	pick_up_lng	Decimal
誘導経路緯度	route_lat	Decimal
誘導経路経度	route_lng	Decimal
降車緯度	drop_off_lat	Decimal
降車経度	drop_off_lng	Decimal

注意事項	note	Text
車両制限	car_restriction	Integer
車椅子	wc	Boolean
歩行器サイズ	walker_size	Decimal
休み	is_absent	Boolean
休み理由	absence_reason	Text
待ち時間	waittime	Integer
自車区分	is_self	Boolean

【IF014】乗降車記録データ連携用 IF

- 本インターフェースの概要
 - 利用者が送迎便に乗車・降車することの予定・実績データを連携する IF
- 本インターフェースを利用する機能
 - 【FN103】乗降車記録
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ 登録: POST
 - ◇ 更新: PUT
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
事業所 ID	office_id	事業所の ID	数値	○
日付	date	日付 (yyyymmdd)	日付	○
利用者 ID	customer_id	利用者の ID	数値	○
送迎タイプ	soge_type	送迎タイプ	数値	○

```
{
  "bindata": {
    "office_id": ""
    "date": ""
    "customer_id": ""
  }
}
```

```

“soge_type”: “”
(以下乗降者記録データ項目)
}
}
    
```

➤ レスポンスパラメータ

項目	名称	説明	値
メッセージ	messages	レスポンスメッセージ	文字列

◇ 成功時

```

{
  "messages":["乗降車記録を登録しました"]
}
    
```

◇ 失敗時

```

{
  " error_messages ":["エラーメッセージ"]
}
    
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-24 乗降車記録データ連携用 IF 項目

論理名	物理名	型(サイズ)
事業所 ID	office_id	String(PK)
日付	date	Date(PK)
利用者 ID	customer_id	String(PK)
送迎種別	soge_type	String(PK)
車両 ID	carId	Integer(PK)
出発時間	departure_time	Time
乗車時刻	pick_up_time	Time
降車時刻	drop_off_time	Time
車両ステータス	carStatus	String
更新日時	updatetimec	Time
オペレーターステータス	userStatus	String
更新日時	updatetimeu	Time
乗車人数	passenger_count	Integer

【IF015】 車両位置データ連携用 IF

- 本インターフェースの概要
 - 送迎中の車両の位置情報を連携する IF
- 本インターフェースを利用する機能
 - 【FN102】 送迎指示情報取得
 - 【FN105】 地図情報取得
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ 登録: POST
 - ◇ 更新: PUT
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
車両 ID	id	車両の ID	文字列	○
日付	date	日付 (yyyyMMdd)	日付	○
時間	time	時間 (hh:mm:ss)	日付	○
運転手	driever	運転手名	文字列	

```

{
  "nowpos": {
    "id": ""
    "date": ""
    "time": ""
    "driver": ""
    (以下車両位置データ項目)
  }
}
    
```

➤ レスポンスパラメータ

項目	名称	説明	値
メッセージ	messages	レスポンスメッセージ	文字列

◇ 成功時

```
{
  "messages":["車両位置を登録しました"]
}
```

◇ 失敗時

```
{
  " error_messages ":["エラーメッセージ"]
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-25 車両位置データ連携用 IF 項目

論理名	物理名	型(サイズ)
車両位置 ID	id	Integer(PK)
日付	date	Date
時間	time	String
運転手	driver	String
緯度	Posx	Decimal
経度	Posy	Decimal
休憩中	rest	Boolean
バッテリー	battery	Integer
出発時間	departure_time	Time
事業所 ID	office_id	String
ステータス	status	String

【IF016】お知らせデータ連携用 IF

- 本インターフェースの概要
 - 配車リクエストやキャンセル等のアクションデータを格納し、お知らせとして確認するデータの連携 IF
- 本インターフェースを利用する機能
 - 【FN019】お知らせ情報取得
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド

- ◇ GET
- レスポンスデータ形式
 - ◇ JSON
- 文字コード
 - ◇ UTF-8
- リクエストパラメータ
 - ◇ Body

項目	名称	説明	値	必須
事業所 ID	office_id	事業所の ID	数値	○

```
{
  "serch_params": {
    "office_id": ""
  }
}
```

- レスポンスパラメータ

項目	名称	説明	値
ID	id	ユーザーの ID	数値
事業所 ID	office_id	事業所の ID	数値
メッセージ	messages	レスポンスメッセージ	文字列
カテゴリ	category	メッセージのステータスカテゴリ	数値
確認時間	read_at	メッセージを確認した時間	日付

◇ 成功時

```
• {
•   "notifications": {
•     "id": ""
•     "office_id": ""
•     "message": ""
•     "category": ""
•     "read_at": ""
•   }
• }
```

◇ 失敗時

```
{
  " error_messages ":["エラーメッセージ"]
}
```

- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-26 お知らせデータ連携用 IF 項目

論理名	物理名	型(サイズ)
お知らせ ID	id	Integer(PK)
事業所 ID	office_id	Integer
メッセージ	comment	String
カテゴリ	category	Integer
確認した時間	read_at	Datetime

【IF101】 道路マップ

- 本インターフェースの概要
 - 送迎計画作成時や送迎ルート案内時に表示するデジタル地図データ
- 本インターフェースを利用する機能
 - 【FN401】 地図表示
 - 【FN402】 経路検索
- サンプル・イメージ



図 2-44 Google Maps のサンプル・イメージ

- データ形式
 - ベクターデータ（内部的にベクトルタイル形式：Protocol Buffer 形式 .pbf などを利用）
 - API 利用時は JSON 又は XML 形式で返される（例：Directions API）
- 出所
 - Google が独自に収集・保有する地図データ

【IF102】 事業所及び乗降ポイント

- 本インターフェースの概要
 - 送迎計画作成時や送迎ルート案内時に表示するデジタル地図データ (Google Maps API)
 - 施設及び利用者の住所情報を変換した JSON データ
 - GET https://maps.googleapis.com/maps/api/geocode/json?address=住所情報&key=YOUR_API_KEY
- 本インターフェースを利用する機能
 - 【FN402】 経路検索

表 2-27 事業所及び乗降ポイント項目

No.	日本語名称	項目長	書式・選択肢など	必須
1	status	-	API の結果ステータス OK：有効な結果が返された状態 ZERO_RESULTS：リクエストは成功したが、該当する結果が見つからなかった（住所が存在しないなど） OVER_DAILY_LIMIT：無料利用枠や課金プランの制限を超えた又は API キーが無効・認可されていない場合	○
2	results	-	結果の配列（マッチする住所が複数ある場合も）	○
3	formatted_address	-	正規化された住所表記 (例) 日本、〒150-0002 東京都渋谷区渋谷	○
4	geometry.location.lat	16	緯度。(例) "lat": 35.6593829802915	○
5	geometry.location.lng	17	経度。(例) "lng": 139.7030019802915	○
6	location_type	-	位置の精度 (例) "location_type": "ROOFTOP" ROOFTOP：住所が正確に特定されていて、位置情報が建物の屋根などの具体的なポイントにマッピング RANGE_INTERPOLATED：番地不明な住所の近似位置表示 GEOMETRIC_CENTER：行政区域や広域エリアの代表点表示 APPROXIMATE：大まかな地域指定	○
7	以下省略			

- データ形式
 - JSON
- 出所
 - Google が保持する地理空間データベース (Geocoding API)
- 変換方法
 - 処理内容・手順
 - ◇ 住所情報をインプットに、Geocoding API を呼出し、結果を緯度・経度欄に転記
 - 利用したソフトウェア
 - ◇ Google Geocoding API

【IF103】 ルート表示、誘導経路及びピンの制御

- 本インターフェースの概要
 - 送迎計画作成時に表示するデジタル地図データ
 - Google Maps JavaScript API の google.maps.Map クラスを利用して、マーカーにひも付ける google.maps.Map(htmlElement, mapOptions, mapTypeId);
 htmlElement: マップを表示する DOM 要素 (例: document.getElementById("map"))
 mapOptions: 地図の初期表示設定 (緯度・経度、ズーム、地図タイプなど)
 mapTypeId: 地図の種類 (例: roadmap、satellite、hybrid、terrain)
- サンプル・イメージ



図 2-45 Google Maps のサンプル・イメージ

- 本インターフェースを利用する機能
 - 【FN402】 経路検索

表 2-28 ルート表示、誘導経路及びピンの制御項目

No.	日本語名称	項目長	書式・選択肢など	必須
1	マーカー追加			
2	position	-	緯度・経度 (例) position: { lat: 35.6812, lng: 139.7671 }	○
3	map	-	地図オブジェクト (例) map: map	○
	title	-	表示文字 (例) title: "東京駅"	○
4	ルート描画			
5	origin	-	出発地 (例) { lat: 35.6812, lng: 139.7671 }	○
6	destination	-	目的地 (例) { lat: 35.6586, lng: 139.7454 }	○
7	travelMode	-	移動手段 (例) google.maps.TravelMode.DRIVING DRIVING: 車 (デフォルト) WALKING: 徒歩	○

			BICYCLING：自転車（一部エリア限定） TRANSIT：公共交通機関（バス・電車など）	
--	--	--	---	--

- データ形式
 - JSON
- 出所
 - Google Maps JavaScript API
- 変換方法
 - 処理内容・手順
 - ◇ 緯度・経度情報をインプットに、Maps JavaScript API を呼出し、結果を HTML/JS 表示

JavaScript 例

```

<html>
  <head>
    <style>
      #map { height: 100%; width: 100%; }
      html, body { height: 100%; margin: 0; padding: 0; }
    </style>
    <script src='https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY'>
    </script>
    <script>
      function initMap() {
        var location = { lat: 35.6895, lng: 139.6917 };
        var map = new google.maps.Map(document.getElementById('map'), {
          zoom: 12,
          center: location
        });
        var marker = new google.maps.Marker({
          position: location,
          map: map
        });
      }
    </script>
  </head>
  <body onload='initMap()'>
    <div id='map'></div>
  </body>
</html>

```

- 利用したソフトウェア
 - ◇ Maps JavaScript API

【IF104】 地点間の移動時間や距離

- 本インターフェースの概要
 - 送迎計画作成に際し最適ルート計算の基となる、送迎施設と乗降ポイント間の移動時間や距離
 - GET https://maps.googleapis.com/maps/api/distancematrix/json?origins=出発地&destinations=目的地&mode=移動手段&key=YOUR_API_KEY
- 本インターフェースを利用する機能
 - 【FN402】 経路検索

表 2-29 地点間の移動時間や距離項目

No.	日本語名称	項目長	書式・選択肢など	必須
1	status	-	API の結果ステータス OK：有効な結果が返された状態 NOT_FOUND：出発値、目的値の住所・位置情報が無効 ZERO_RESULTS：指定された移動手段で到達不可能 (例) 徒歩で海を渡ろうとした等	○
2	distance	-	距離 text：人が読みやすい形式 (例) "text": "6.5 km" value：距離の数値 (メートル単位、例 "value": 6500)	○
3	duration	-	所要時間 text：人間が読みやすい表現 (例) "text": "20 分" value：秒単位の所要時間 (例) "value": 1200	○

- データ形式
 - API 利用時は JSON 又は XML 形式で返される (例：Distance Matrix API)
- 出所
 - Google が独自に収集・保有する地図・交通データ

2-5. ユーザーインターフェース (UI)

2-5-1. 画面遷移図

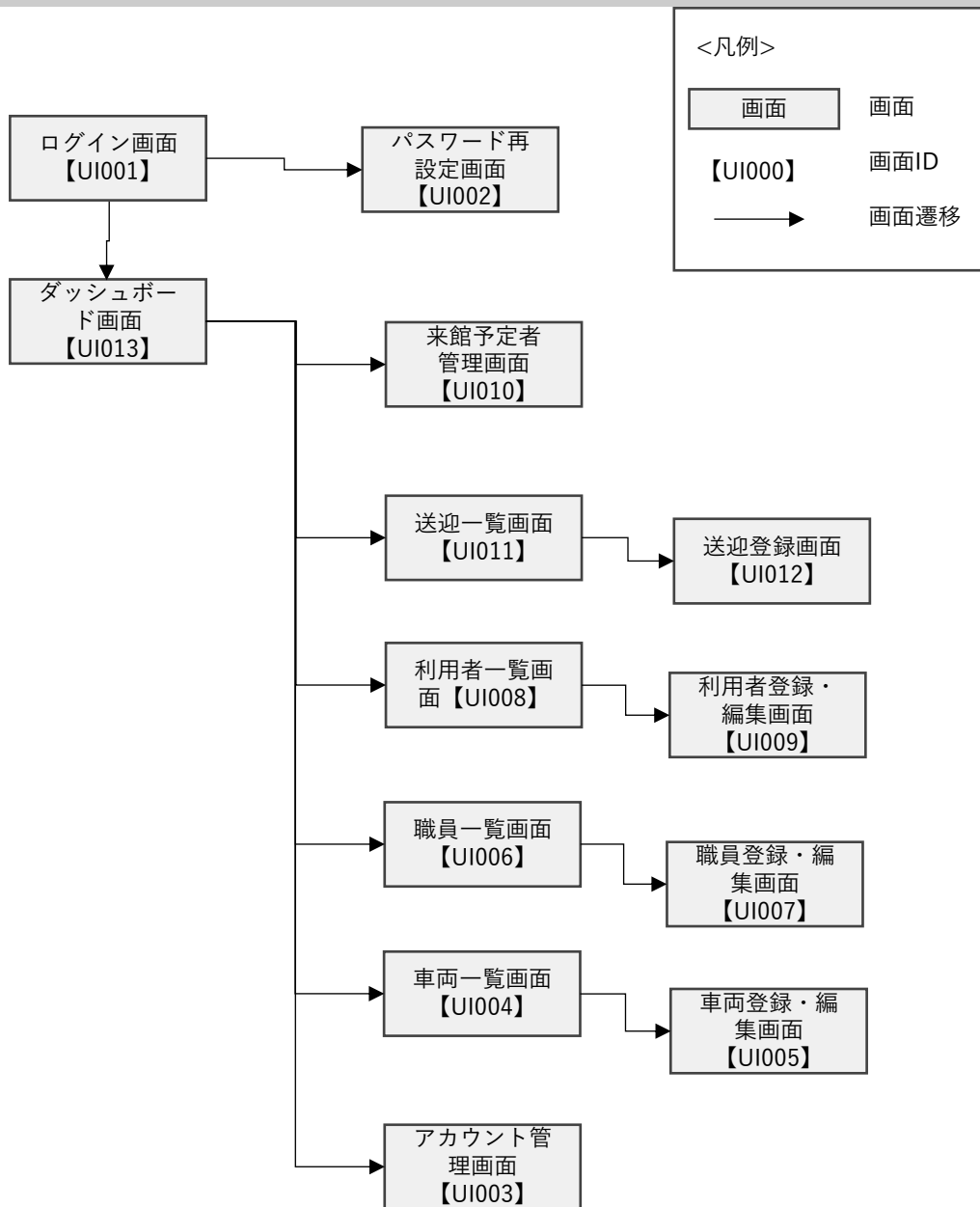


図 2-46 【HW001】 PC 用画面遷移図

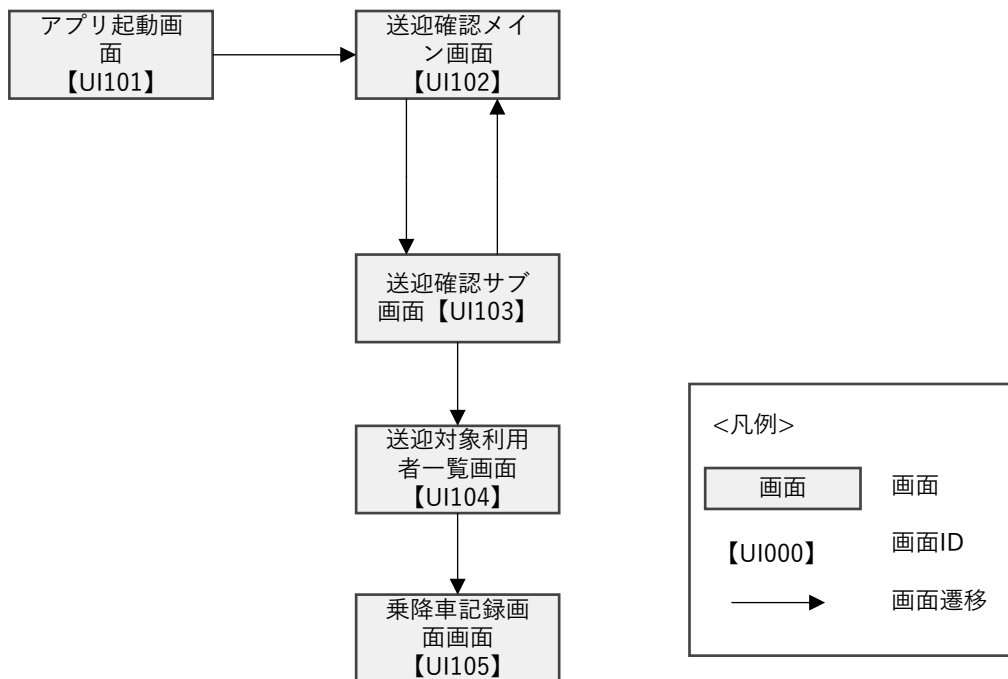


図 2-47 【HW002】 ドライバーアプリ（スマートフォン）用画面遷移図

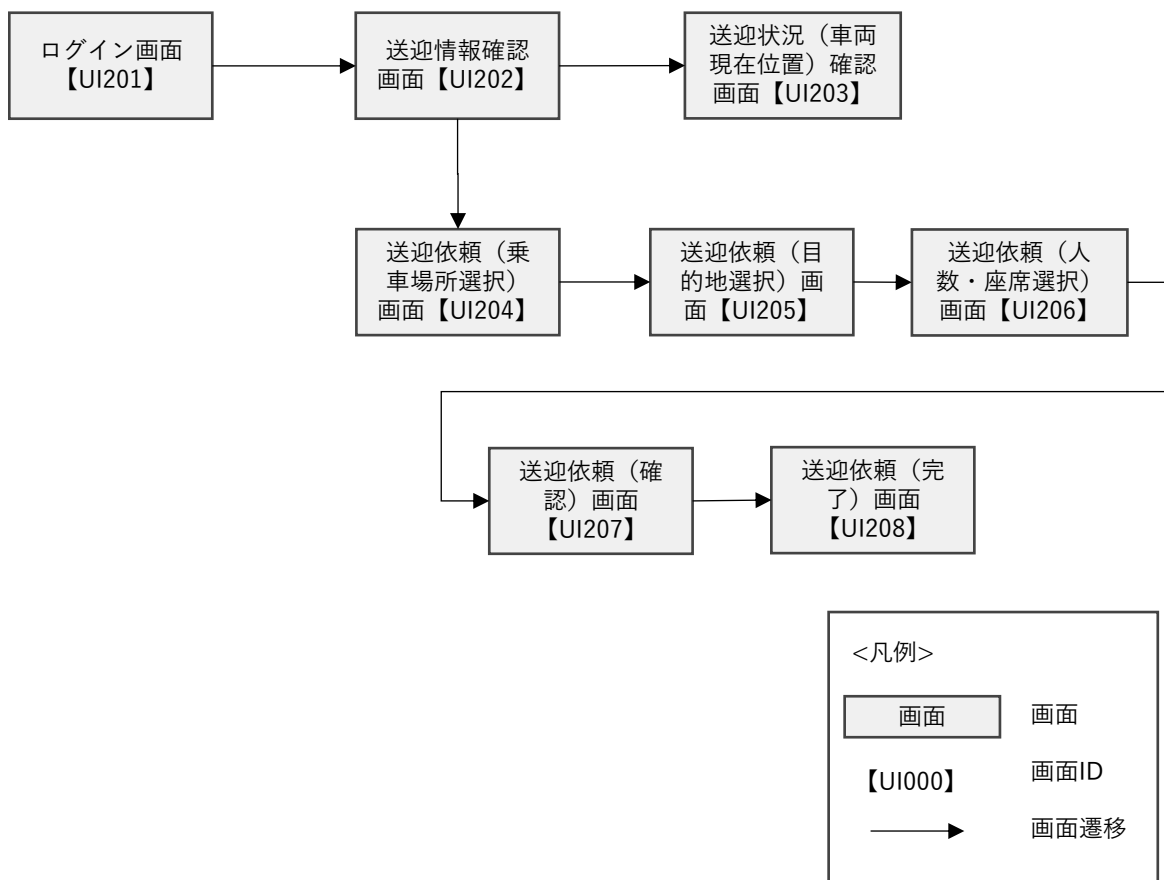


図 2-48 【HW003】 利用者アプリ（スマートフォン）用画面遷移図

2-5-2. ユーザーインターフェース一覧

表 2-30 【HW001】 管理者アプリ画面一覧

※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	ログイン画面 (PC)	<ul style="list-style-type: none"> 施設管理者ユーザーがシステムにアクセスするために本人確認を行う認証用の画面 	FN001
UI002	パスワード再設定画面 (PC)	<ul style="list-style-type: none"> 施設管理者ユーザーがログインパスワードを再設定するために使用する画面 	FN001
UI003	アカウント管理画面 (PC)	<ul style="list-style-type: none"> 管理者アプリを利用する施設管理者ユーザーの登録・編集などを行うための管理用の画面 	FN001
UI004	車両一覧画面 (PC)	<ul style="list-style-type: none"> 送迎に活用する車両一覧を確認する画面 	FN001
UI005	車両登録・編集画面 (PC)	<ul style="list-style-type: none"> 送迎に活用する車両を登録・編集する画面 	FN001
UI006	職員一覧画面 (PC)	<ul style="list-style-type: none"> 送迎に関わる職員（運転手・添乗員）の一覧を表示する画面 	FN001
UI007	職員登録・編集画面 (PC)	<ul style="list-style-type: none"> 職員を登録・編集する画面 	FN001
UI008	利用者一覧画面 (PC)	<ul style="list-style-type: none"> 施設の利用者一覧を表示する画面 	FN001
UI009	利用者登録・編集画面 (PC)	<ul style="list-style-type: none"> 施設の利用者情報を登録・編集する画面 	FN001
UI010	送迎予約一覧画面 (PC)	<ul style="list-style-type: none"> 指定日の来館予定利用者の一覧を表示し、登録・編集を行う画面 本画面から、送迎委託する利用者を指定する 	FN001
UI011	送迎計画画面 (PC)	<ul style="list-style-type: none"> 指定日の来館予定者に対して送迎の配車計画を立て、車両ごとの順番設定や最適ルートの計算、計画の印刷を行うための画面 	FN001
UI012	送迎ルート確認・編集画面 (PC)	<ul style="list-style-type: none"> 指定された日付・便の車両における利用者とその乗車順を表示し、地図上でルートを確認・修正するための画面 	FN001
UC013	ダッシュボード画面 (PC)	<ul style="list-style-type: none"> 施設管理者ユーザーがログイン後に確認する最初の画面 	FN001

表 2-31 【HW002】 ドライバーアプリ画面一覧

※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)
UI101	アプリ起動画面 (スマートフォン)	● 運転手が日付、事業所、車両名を選び、送迎ナビゲーションアプリを起動する画面	FN101
UI102	送迎確認メイン画面 (スマートフォン)	● 指定日・指定車両の送迎指示内容を確認し、便ごとの送迎人数やルートのナビ開始、GPS 表示、送迎依頼の受入れ可否などを操作するための画面	FN102
UI103	送迎確認サブ画面 (スマートフォン)	● 選択した利用者の住所・顔写真・注意事項・乗車一覧を確認し、必要に応じて位置情報表示や経路確認を行うための補助画面	FN102
UI104	送迎対象利用者一覧画面	● 指定日・指定便の送迎対象利用者を一覧で確認するための画面	FN102
UI105	乗降車記録画面 (スマートフォン)	● 利用者ごとに乗降車時間を記録する	FN103

表 2-32 【HW003】 利用者アプリ画面一覧

※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)
UI201	ログイン画面	● 利用者がシステムにアクセスするために本人確認を行う認証用の画面	FN201
UI202	送迎状況確認画面	● 利用者の送迎予定を確認する画面	FN201
UI203	送迎状況 (車両現在位置) 確認画面	● 利用者が乗車する送迎車両の現在位置を地図上で確認する画面	FN201
UI204	送迎依頼 (乗車場所選択) 画面	● 利用者が送迎依頼をするときに乗車場所を選択する場面	FN201
UI205	送迎依頼 (目的地選択) 画面	● 利用者が送迎依頼をするときに目的地を選択する画面	FN201
UI206	送迎依頼 (人数・座席選択) 画面	● 利用者が送迎依頼をするときに同乗者や車椅子が乗れる座席を選択する画面	FN201

UI207	送迎依頼（確認）画面	<ul style="list-style-type: none"> ● 利用者が指定した送迎依頼内容を確認する画面 	FN201
UI208	送迎依頼（完了）画面	<ul style="list-style-type: none"> ● 利用者による送迎依頼を受け付けたことを伝える画面 	FN201

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。

1) 【HW001】 PC 用画面

【UI001】 ログイン画面（PC）

 **COMmmONS**
by MLIT

共同送迎管理システム

メールアドレス

パスワード 表示

[→ ログイン](#)

- ・国土交通省が進める地域交通DXプロジェクト「COMmmONS(コモンズ)」である「地域施設送迎のリソースシェア推進プロジェクト」で開発した参考画面です。
- ・「共同送迎管理システム」は、複数の施設管理者が施設送迎（車両を用いて自施設の利用者の送迎を無償で行うもの）を共同して行うためのシステムです。

➤ [COMmmONSについてはこちら](#)

➤ [本システムの取り組みについてはこちら](#)

Copyright©2026 by MLIT. All Rights Reserved.

図 2-49 ログイン画面

- 本画面の概要
 - 施設管理者ユーザーがシステムにアクセスするために本人確認を行う認証用の画面
 - ID（メールアドレス）とパスワードを入力して認証する
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN001】 アカウント認証

【UI002】 パスワード再設定画面（PC）

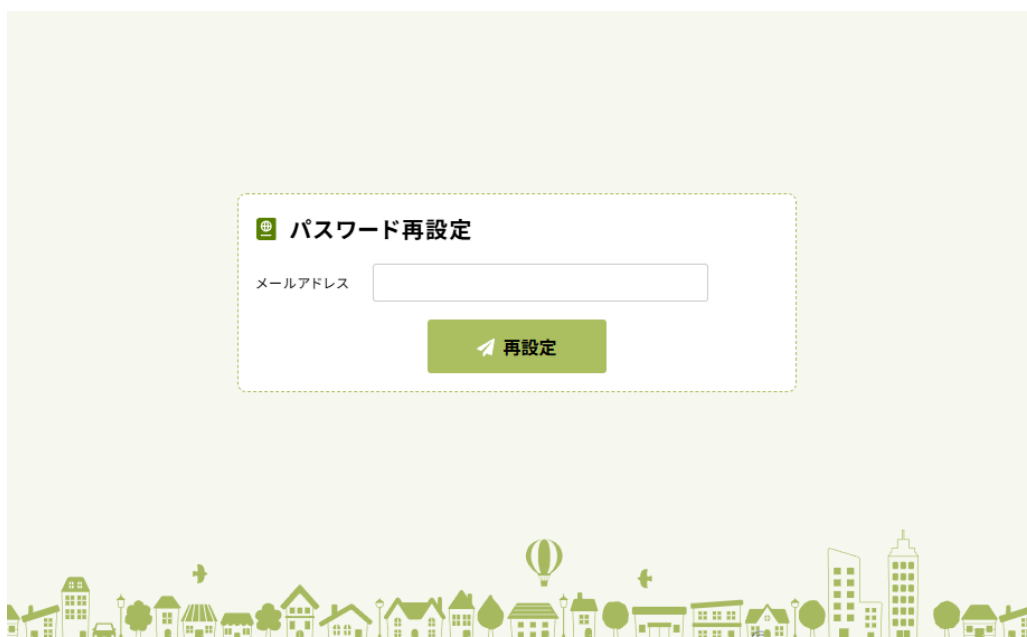


図 2-50 パスワード再設定画面

- 本画面の概要
 - 施設管理者ユーザーがログインパスワードを再設定するために使用する画面
 - メールアドレスを入力することで、該当者のメールアドレスにパスワードリセット用メールが送信される
 - 施設管理者ユーザーはメールアドレスでパスワードをリセットする
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN003】 アカウント登録・更新

【UI003】アカウント管理画面（PC）

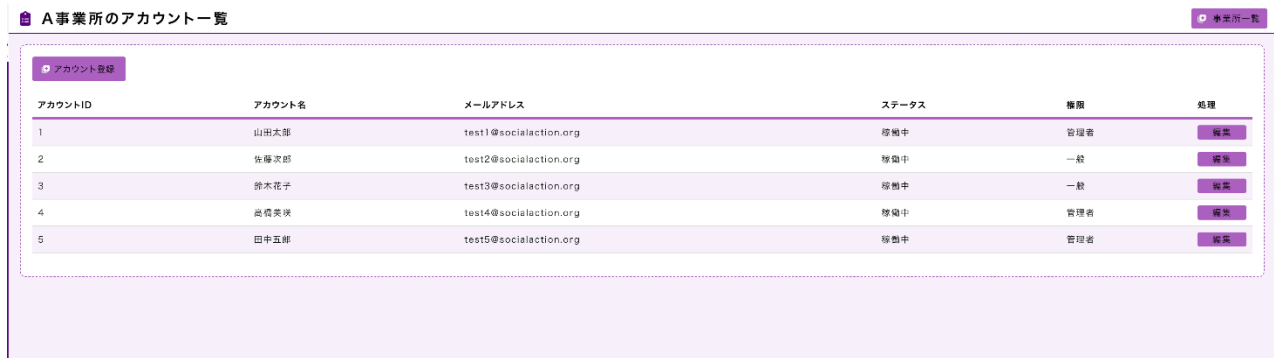


図 2-51 アカウント管理画面

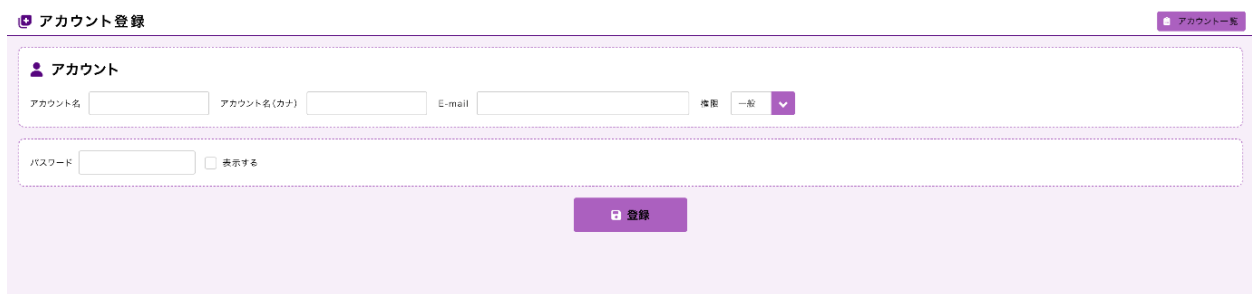


図 2-52 アカウント管理画面（登録・更新モーダルウィンドウ）

- 本画面の概要
 - 管理者アプリを利用する施設管理者ユーザーの登録・編集などを行うための管理用の画面
 - 事業所にアクセス可能なアカウントの一覧を表示
 - 指定アカウントを登録・更新する
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN003】 アカウント一覧取得
 - 【FN004】 アカウント登録・更新

【UI004】車両一覧画面（PC）

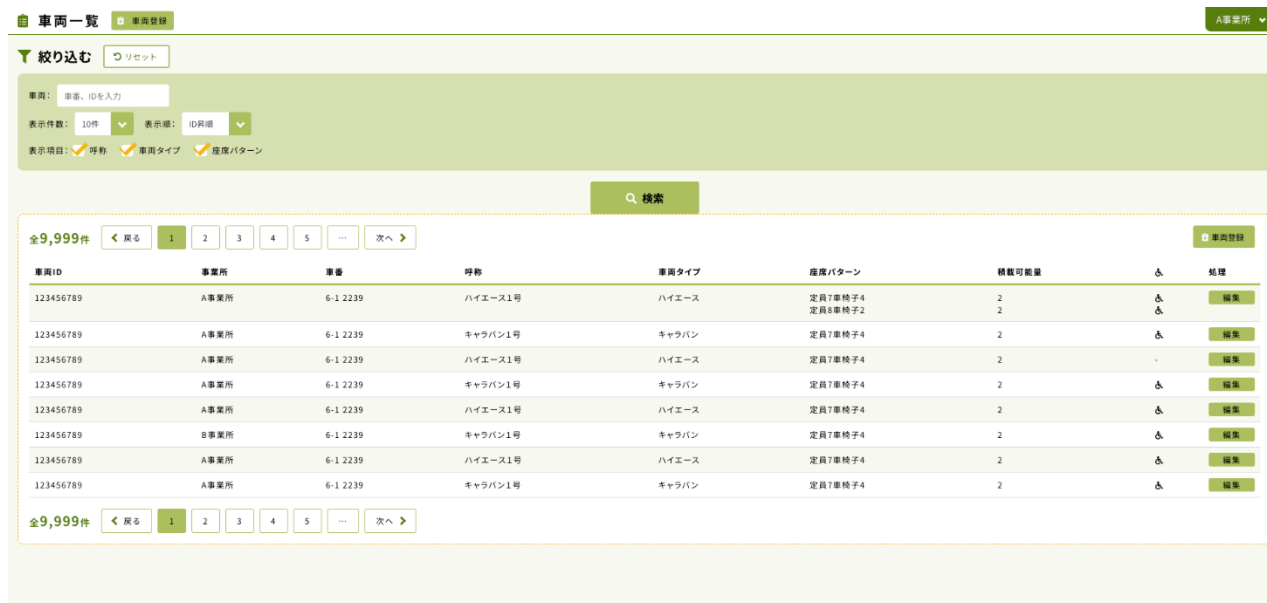


図 2-53 車両一覧画面

- 本画面の概要
 - 送迎に活用する事業所ごとの車両一覧を確認する画面
 - 車両名、車両タイプ（ハイエース、Voxy等）、座席パターン（座席○席、車椅子○）席等を確認可能
 - 本画面から車両を指定して登録・更新画面へ遷移
- 本画面から利用する機能
 - 【FN001】FrontEnd UI
 - 【FN005】車両一覧取得

【UI005】車両登録・編集画面 (PC)

車両編集 車両一覧

車両

事業所* アンドリハデモ 車両名* セレナ 車両番号* 77-88 定員* 6

車椅子数* 0 車両の出発・到着地点: NSIデモ

車両パターン セレナ

車両パターン(セレナ)

車種* セレナ

通常座席数(車椅子0台の場合) 6 積載可能量 0

+ 車椅子座席を追加する

車両制限*

- 車両制限なし
- 車両制限なし
- 軽のみ
- 軽の助手席
- 軽のゲート不可
- ハイエース不可
- 軽かゲート
- 軽車両

図 2-54 車両登録・編集画面

- 本画面の概要
 - 送迎に活用する事業所ごとの車両を登録・編集する画面
 - 車両名、座席定員、車椅子の座席数、車両の出発地点等を登録・更新する
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN006】 車両登録・更新

【UI006】職員一覧画面（PC）



図 2-55 職員一覧画面

- 本画面の概要
 - 送迎に関わる事業所ごとの職員（運転手・添乗員）の一覧を表示する画面
 - 職員の写真、名前、送迎可否、添乗可否を確認する
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN007】 職員一覧取得

【UI007】職員登録・編集画面（PC）

更新日:2023/02/18 退職日:2023/02/18

職員

職員ID 職員名

職員名(カナ) 略名

職種

職員画像

画像をここにドラッグ&ドロップまたはクリックして選択

送迎区分 添乗区分

運転手タイプ メールアドレス 電話番号

運転車両

- キャラG4
- キャラG2
- セレナG
- セレナ

図 2-56 職員登録・編集画面

- 本画面の概要
 - 職員を登録・編集する画面
 - 職員の名前、カナ、送迎（運転）の可否、添乗の可否、運転可能な車両を登録・更新する
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN008】 職員登録・更新

【UI008】利用者一覧画面（PC）



図 2-57 利用者一覧画面

- 本画面の概要
 - 施設の利用者一覧を表示する画面
 - 利用者の氏名、カナ、住所、乗車できる車両や現在の利用状況を確認する
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN009】 利用者一覧取得

【UI009】利用者登録・編集画面 (PC)

🏠 利用者登録
👤 利用者一覧
🕒 利用者履歴
⚠️ 乗り合わせ設定
A事業所 ▼

👤 利用者

利用者ID* 利用者名

利用者名(カナ) 契約識別

歩行状態 車椅子

歩行器の種類 歩行器の大きさ

車椅子の使用 添乗員 座席指定

📍 住所ラベルA 地点A

+ 送迎地点を追加する

利用者画像




✏️ 編集
🗑️ 削除



画像をここにドラッグ&ドロップまたは
クリックして選択

📍 地点A
📍 地点登録
👤 地点一覧

〒 ➡️ 〒→住所

住所 ➡️ 住所→〒

マンション名・号室

電話番号

地点緯度 地点経度

停車緯度 停車経度

距離 km 時間 分

待ち時間 分 許容率 %

誘導経路登録
 単独ルート登録(迎え)
 単独ルート登録(送り)

車両制限

車両制限なし
 軽のみ
 軽の助手席
 軽のゲート不可
 ゲートのみ
 ハイエース不可
 軽かゲート
 軽車両

地点画像

画像をここにドラッグ&ドロップまたは
クリックして選択

電話メモ

地点備考

送迎時間*	送迎地点	送迎車種	乗務時間*	終了時間*	送り地点	送り車種	送迎リクエスト	送りリクエスト	自乗	他乗
7:00	---	---	7:00	7:30	---	---	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> 日	---	ワンボックスカー	7:00	7:30	---	ワンボックスカー	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> 月	---	ワンボックスカー	7:00	7:30	---	ワンボックスカー	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> 火	---	ワンボックスカー	7:00	7:30	---	ワンボックスカー	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> 水	---	ワンボックスカー	7:00	7:30	---	ワンボックスカー	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> 木	---	ワンボックスカー	7:00	7:30	---	ワンボックスカー	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> 金	---	ワンボックスカー	7:00	7:30	---	ワンボックスカー	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> 土	---	ワンボックスカー	7:00	7:30	---	ワンボックスカー	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

利用者						運転手					
<input type="checkbox"/>	利用者ID	顔写真	利用者名	登録日	解除日	<input type="checkbox"/>	運転手ID	顔写真	運転手名	登録日	解除日
<input type="checkbox"/>	123456789		真鍋智彦	2023/04/15	2023/04/15	<input type="checkbox"/>	123456789		真鍋智彦	2023/04/15	2023/04/15
<input type="checkbox"/>	123456789		真鍋智彦	2023/04/15	2023/04/15	<input type="checkbox"/>	123456789		真鍋智彦	2023/04/15	2023/04/15
<input type="checkbox"/>	123456789		真鍋智彦	2023/04/15	2023/04/15	<input type="checkbox"/>	123456789		真鍋智彦	2023/04/15	2023/04/15
<input type="checkbox"/>	123456789		真鍋智彦	2023/04/15	2023/04/15	<input type="checkbox"/>	123456789		真鍋智彦	2023/04/15	2023/04/15

図 2-58 利用者登録・編集画面

- 本画面の概要
 - 施設の利用者情報を登録・編集する画面
 - ◇ 利用者情報：氏名や身体情報、乗降地点、週間来館パターン、乗り合わせてはいけない利用者の組合せ
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN010】 利用者登録・更新

【UI010】送迎予約一覧画面（PC）

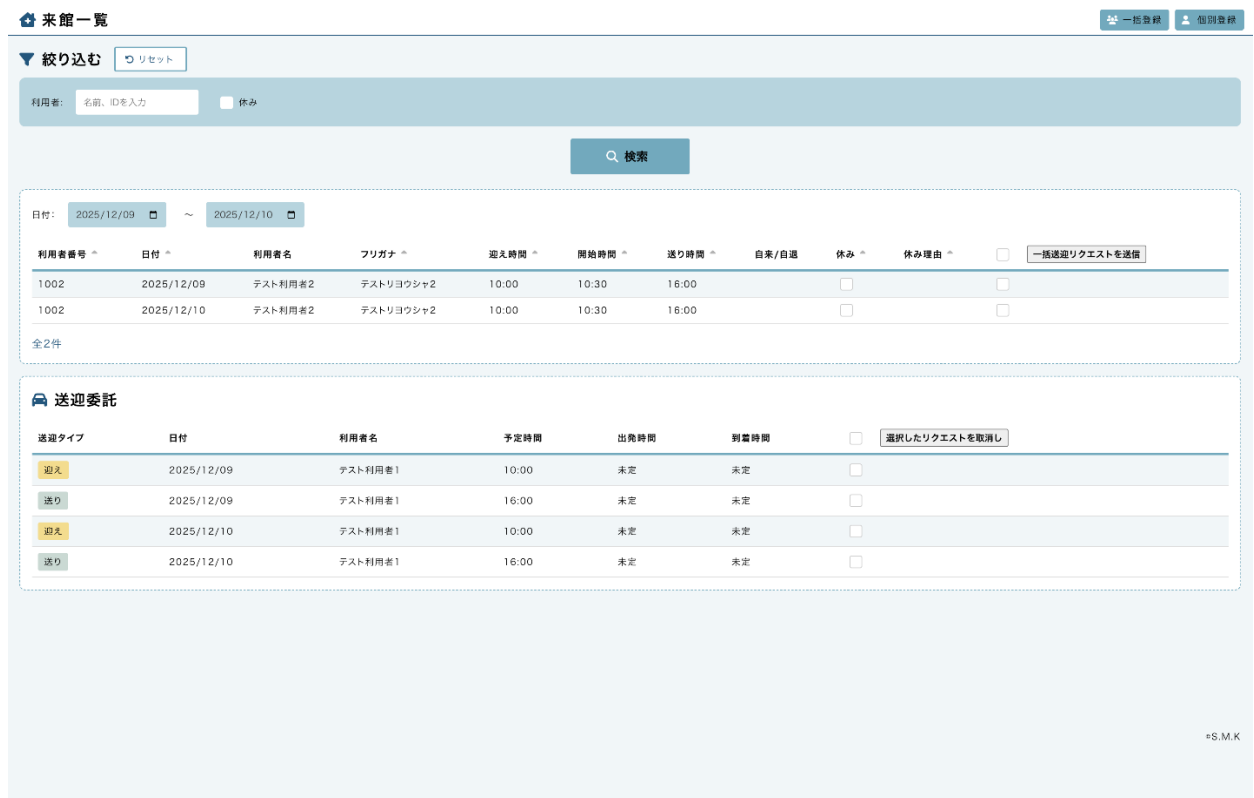


図 2-59 送迎予約一覧画面（一覧表示）



図 2-60 送迎予約一覧画面（登録・編集モーダルウィンドウ）



図 2-61 送迎予約一覧画面 (送迎委託一括リクエスト・キャンセル)

- 本画面の概要
 - 指定日の送迎予約者の一覧を表示し、登録・編集を行う画面
 - 本画面から、送迎委託する利用者を指定して依頼・キャンセルする
 - ◇ 一括で依頼・キャンセルすることも可能
 - 本画面にて、他施設から送迎委託された利用者や、個人で送迎依頼をした利用者を表示する
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN011】 施設来館予定者一覧取得
 - 【FN012】 施設来館予定者登録・更新
 - 【FN017】 送迎委託・キャンセル

【UI011】送迎計画画面 (PC)



図 2-62 送迎一覧画面 (右ウィンドウ来館予定者表示)

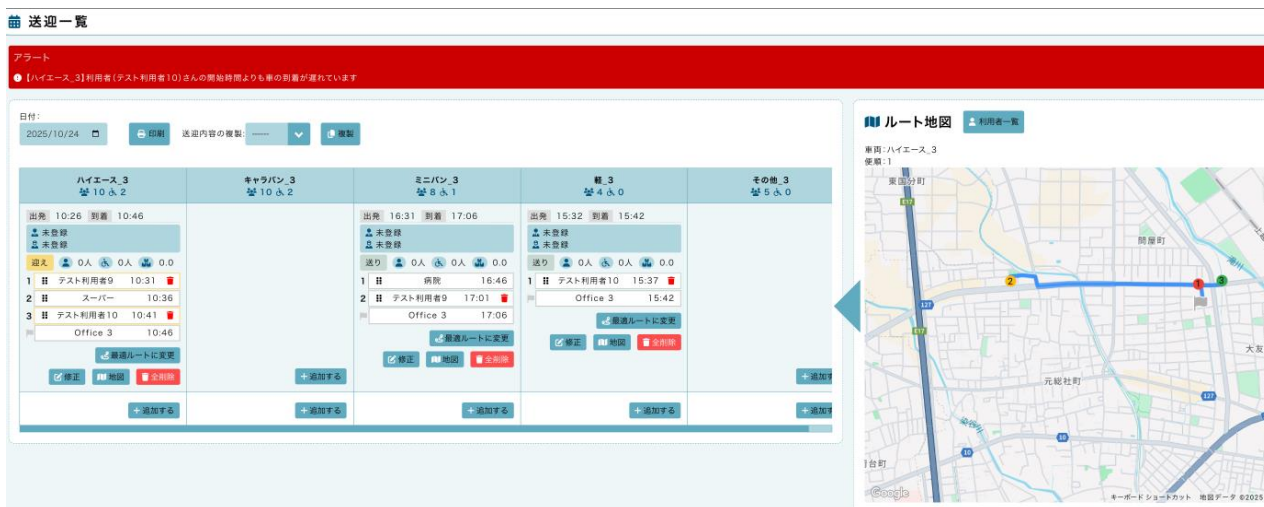


図 2-63 送迎一覧画面 (右ウィンドウ地図表示)

送迎一覧

日付: 2026/01/07 印刷 送迎内容の複製: 複製

1号車 1号車 (3434) 7 区 4	2号車 2号車 (52) 7 区 4	3号車 3号車 (1473) 8 区 2	5号車 4号車 (328) 5 区 1	6号車 5号車 (2239) 5 区 2	7号車 6号車 (123) 5 区 2
出発: 11:15 到着: 12:05 未登録 未登録 迎え: 1人 0人 0.3 1 # リヨウシャ 11:30 利用者2 12:00 2 # ヒカリアレ 11:40 創春館 12:00 最適ルートに変更 修正 地図 全削除					
+追加する	+追加する	+追加する	+追加する	+追加する	+追加する
+追加する	+追加する	+追加する	+追加する	+追加する	+追加する
+追加する	+追加する	+追加する	+追加する	+追加する	+追加する
+追加する	+追加する	+追加する	+追加する	+追加する	+追加する

↑ 確定

図 2-64 送迎一覧画面（ドロップダウン枠の指定）

送迎予約

予約						
	送迎	乗車場所	降車場所	名前	乗車人数	予定時間
☰	迎え	SMK	テスト利用者2自宅	テスト利用者2	1人	15:02
☰	迎え	SMK	テスト利用者1自宅	テスト利用者1	1人	16:07
☰	送り	テスト利用者1自宅	SMK	テスト利用者1	1人	21:33
☰	送り	テスト利用者2自宅	SMK	テスト利用者2	1人	21:33

図 2-65 送迎一覧画面（利用者の表示）



図 2-66 送迎一覧画面（整合性がないデータの対処）

送迎一覧（委託）

日付: 2025/11/04 印刷

ハイエース_2 🚗 10 名 2	キャラバン_2 🚗 10 名 2	ミニバン_2 🚗 8 名 1
出発 12:17 到着 12:27 未登録 未登録 混在 👤 7人 ♿ 0人 🗑️ 0.0 旅館2 12:22 ▲ **** 4人 旅館1 12:22 ▲ 上田2 3人 観光協会 12:27 + 追加する	出発 09:09 到着 09:24 未登録 未登録 混在 👤 10人 ♿ 0人 🗑️ 0.0 旅館1 09:14 ▲ 上田智香 6人 旅館2 09:19 ▲ **** 4人 観光協会 09:24 + 追加する	+ 追加する

図 2-67 送迎一覧画面（送迎委託した場合の便の表示）

- 本画面の概要
 - 指定日の送迎予約者に対して送迎の配車計画を立て、車両ごとの順番設定や最適ルートの計算、計画の印刷を行うための画面
 - 左側のサブウィンドウに車両ごとの便を表示する
 - 便ごとに運転手・添乗員を設定する
 - 右側のサブウィンドウは2種類の画面構成
 - サブウィンドウの1種類目は利用者一覧画面
 - 当日の利用者を表示する
 - ◇ 送迎を委託する施設の場合は、当画面において自施設が委託した便・利用者のみ表示され、それ以外は非表示となる
 - 当日利用者をドラッグ・ドロップして左側ウィンドウの便の枠に入れて便を作る
 - ◇ 既に便に利用者がある場合、同じ便で送迎する推薦者を表示する
(利用時間、利用者自宅からの距離、相性を考慮)
 - サブウィンドウの2種類目は地図表示
 - ◇ 左側のサブウィンドウで指定された便の走行ルートを表示 (便の利用者順番)
 - ◇ 左側のサブウィンドウで「最適ルート」ボタンをクリックすることにより推奨する最短ルートを表示 (便の現在の利用者順番を無視した最短ルートを表示)
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN013】 送迎計画一覧取得
 - 【FN014】 送迎計画登録・更新
 - 【FN015】 最適ルート検索
 - 【FN016】 同乗者推薦

【UI012】送迎ルート確認・編集画面 (PC)

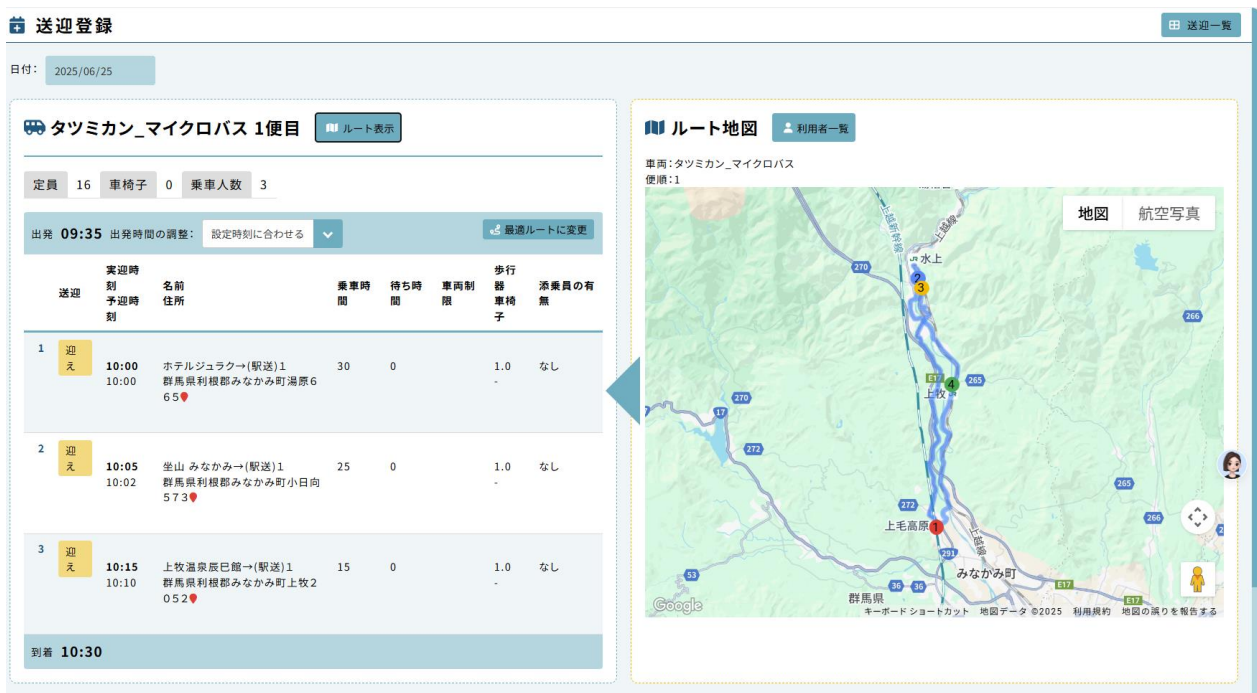


図 2-68 送迎登録画面

- 本画面の概要
 - 指定された日付・便の車両における利用者とその乗車順を表示し、地図上でルートを確認・修正するための画面
 - 左側のサブウィンドウで、指定された便の利用者の一覧が送迎順に表示される
 - 送迎順に合わせて、右側のサブウィンドウで地図と推奨送迎ルートが表示される
 - 送迎ルートを確認しながら、左側のサブウィンドウで順番を入替え、送迎ルートを確認する
 - 「最適ルート」ボタンをクリックすることにより推奨する最短ルートを表示（便の現在の利用者順番を無視した最短ルートを表示）
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN013】 送迎計画一覧取得
 - 【FN014】 送迎計画登録・更新
 - 【FN015】 最適ルート検索

【UI013】ダッシュボード画面 (PC)

🏠 **ダッシュボード**

日付: 09/11/2025

統計

予約者数
2人

送迎表

👤 運転手 🚗 車両

🟡 迎え 🟢 送り

2025/9/11
木曜日

	0:00	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	
坐山ドライバ																									
ジュラクドライバ																									
辰巳頼ドライバ																									

マイクログバス

送迎予約

利用者: フリガナ、IDを入力 🔍 検索 🔄 リセット

送迎委託 (10/23 ~ 11/23)

送迎タイプ	日付	利用者名	予定時間	出発時間	到着時間	<input type="checkbox"/>	選択したリクエストを取消し
送り	2025/10/23	ワカメ	11:29	未定	未定	<input type="checkbox"/>	
迎え	2025/10/23	カツオ	11:32	未定	未定	<input checked="" type="checkbox"/>	承認済み
迎え	2025/10/23	カツオ	15:30	未定	未定	<input type="checkbox"/>	
迎え	2025/10/23	上田	16:28	未定	未定	<input type="checkbox"/>	
送り	2025/10/23	サザエ	18:29	未定	未定	<input type="checkbox"/>	

画面収録を停止するには、🛑 をクリックします

🏠 **ダッシュボード**

日付: 2025/12/03

統計

未乗者数
4人

休み
0人

送迎受託 (12/03 ~ 01/03)

承認待ち

送迎タイプ	日付	事業所名	住所	利用者名	利用者名カナ	乗車地点	降車地点	予定時間	開始時間	🕒	🔧	委託を承認
迎え	2025/12/03	ジュラク	東京都中央区日本橋2-11-2 太陽生命日本橋ビル15階	いいい	いいい	東京都中央区日本橋2-11-2 太陽生命日本橋ビル15階	群馬県前橋市大友町3-24-1	18:03	00:00	-	🔧	
迎え	2025/12/03	ジュラク	東京都中央区日本橋2-11-2 太陽生命日本橋ビル15階	aaa	aaaa	東京都中央区日本橋2-11-2 太陽生命日本橋ビル15階	群馬県前橋市大友町3-24-1	18:46	00:00	-	🔧	
迎え	2025/12/03	ジュラク	東京都中央区日本橋2-11-2 太陽生命日本橋ビル15階	bbbb	bbbbbb	東京都中央区日本橋2-11-2 太陽生命日本橋ビル15階	群馬県前橋市大友町3-24-1	19:46	00:00	-	🔧	

🏠 **ダッシュボード**

日付:
2025/11/27 📅

🚨 未確認のお知らせ

- 2025/11/25 【アンドリハデモ】から、豊岡一郎様の送迎リクエストが許可されました
- 2025/11/25 【アンドリハデモ】から、港花子様の送迎リクエストが許可されました
- 2025/11/25 【アンドリハデモ】から、八条次朗様の送迎リクエストが許可されました

送迎タイプ	日付	利用者名	予定時間	出発時間	到着時間		<input type="checkbox"/> 送迎リクエストを取消し
迎え	2026/01/05	リヨウシャ	18:04	17:04	17:14	承認済み	<input type="checkbox"/> キャンセルリクエスト
送り	2026/01/05	リヨウシャ	20:04	20:24	20:34	承認済み	<input type="checkbox"/> キャンセルリクエスト
迎え	2026/01/07	リヨウシャ	12:00	未定	未定		<input type="checkbox"/> 承認待ち
送り	2026/01/07	リヨウシャ	12:10	未定	未定		<input type="checkbox"/> 承認待ち

図 2-69 ダッシュボード画面

- 本画面の概要
 - 施設管理者ユーザーがログイン後に確認する最初の画面
 - 当日の送迎予定者、送迎表、送迎一覧を確認する
 - 他施設へ委託する「送迎委託」をキャンセルする
 - 他施設から受託した「送迎委託」を承認・却下する
 - 他施設で実施した「送迎委託」リクエストのアクション結果をお知らせで確認する
- 本画面から利用する機能
 - 【FN001】 FrontEnd UI
 - 【FN011】 施設来館予定者一覧取得
 - 【FN013】 送迎計画一覧取得
 - 【FN017】 送迎委託・キャンセル
 - 【FN018】 送迎受託承認・却下
 - 【FN019】 お知らせ

2) 【HW002】スマートフォン（ドライバーアプリ）

【UI101】アプリ起動画面（スマートフォン）



図 2-70 アプリ起動画面

- 本画面の概要
 - 送迎ナビゲーションアプリを起動時に表示する画面
 - 運転手は画面上から、担当する日付、車両名を選択する
 - アプリは専用アプリとしてインストール時に事業所データのみ閲覧可能
- 本画面から利用する機能
 - 【FN101】初期設定データ取得

【UI102】送迎確認メイン画面（スマートフォン）

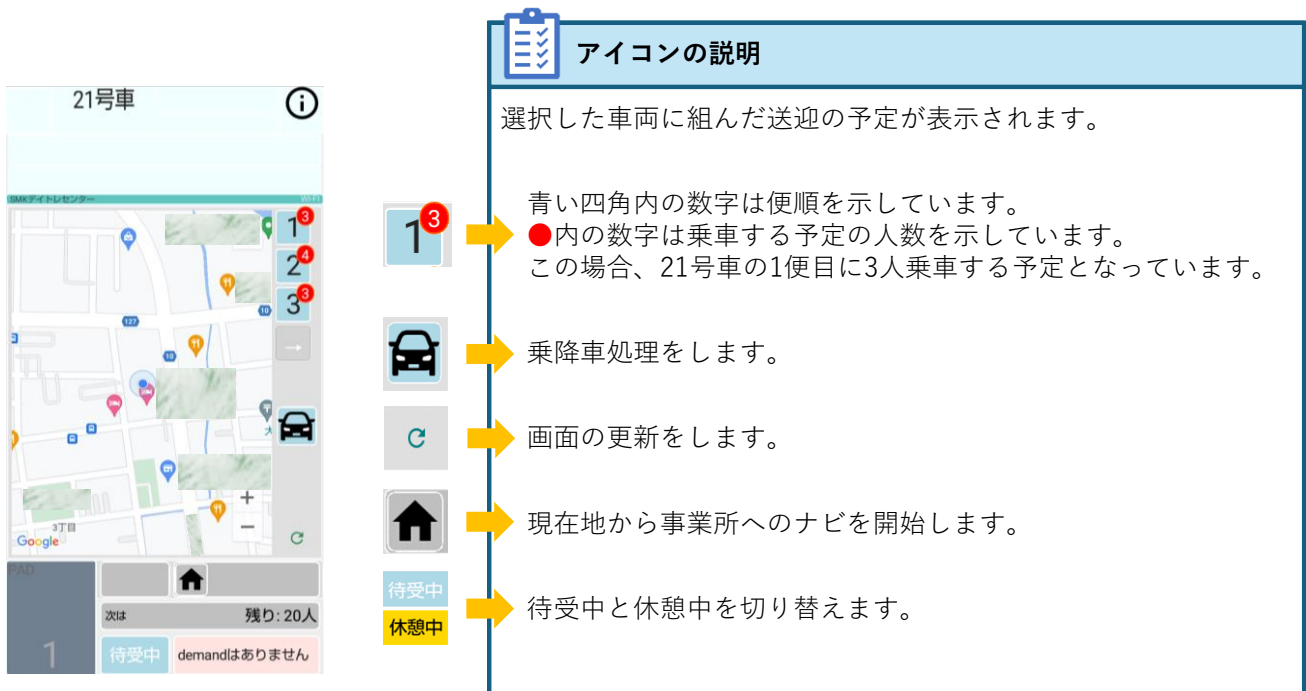


図 2-71 送迎確認メイン画面

- 本画面の概要
 - 指定日・指定車両の送迎指示内容を確認し、便ごとの送迎人数やルートのナビ開始、GPS 表示、送迎依頼の受入れ可否などを操作するための画面
 - 本画面で指定日の車両の便、送迎人数を確認
 - 本画面から乗降者処理、事業所までのナビゲーション起動、追加での送迎依頼を受ける状態（待受中=受ける、休憩中=受けない）を選択
- 本画面から利用する機能
 - 【FN102】送迎指示情報取得
 - 【FN104】地図情報取得

【UI103】送迎確認サブ画面（スマートフォン）

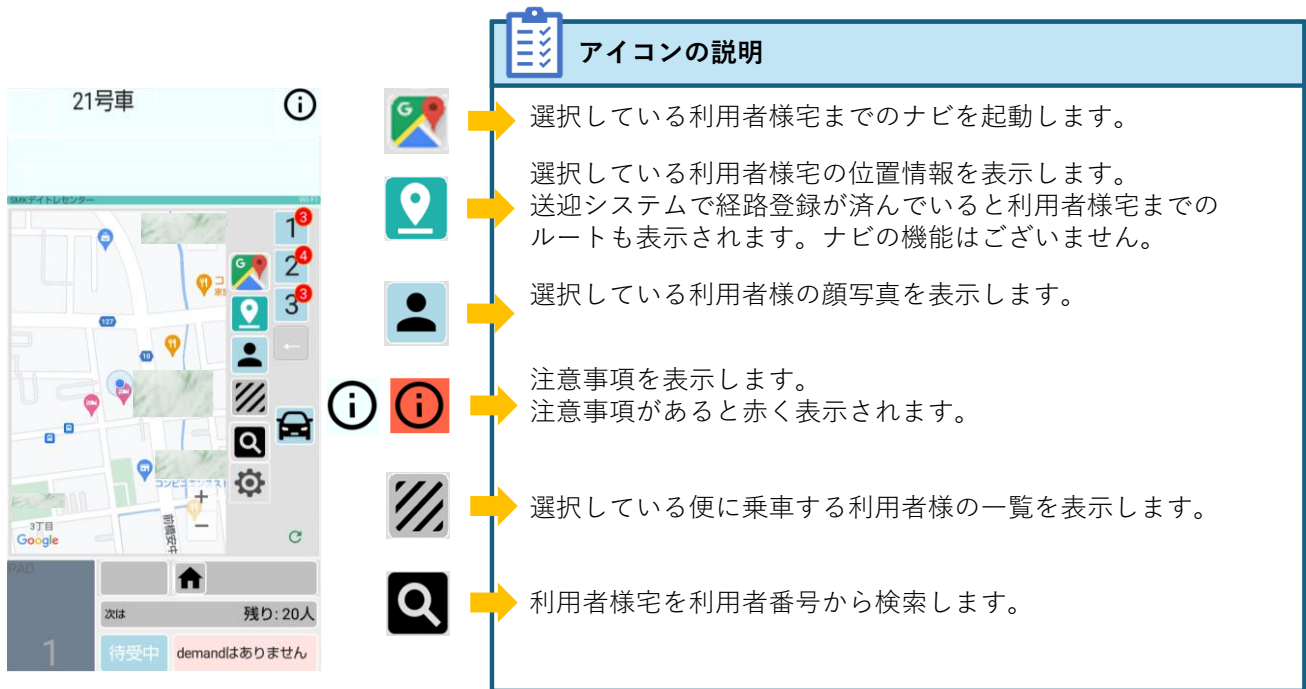


図 2-72 送迎確認サブ画面

- 本画面の概要
 - 選択した利用者の住所・顔写真・注意事項・乗車一覧を確認し、必要に応じて位置情報表示や経路確認を行うための補助画面
 - 本画面からナビゲーションアプリ（Google マップアプリ）の呼出し、利用者の位置情報表示、利用者の顔写真、利用者の注意事項（申し送り事項等）、利用者一覧画面へ遷移する
- 本画面から利用する機能
 - 【FN102】送迎指示情報取得
 - 【FN104】地図情報取得

【UI104】送迎対象利用者一覧画面（スマートフォン）



図 2-73 送迎対象利用者一覧画面

- 本画面の概要
 - 指定日・指定便の送迎対象利用者を一覧で確認するための画面
 - 利用者と予定時刻、乗車時刻、降車時刻が表示される
- 本画面から利用する機能
 - 【FN102】送迎指示情報取得

【UI105】乗降車記録画面

The image displays two tablet screens side-by-side, representing the boarding and alighting recording interfaces. Both screens feature a header bar with a title and a '閉じる' (Close) button. The left screen is titled '乗車の登録です。' (Boarding registration) and includes a 'デモ本 洋' label, a yellow '乗車する。' button, a light blue '後回しにする' button, a pink '休み登録' button, and a light blue 'まとめて処理する' button at the bottom. The right screen is titled '降車の登録です。' (Alighting registration) and includes an 'SMKデイトレセンター' label, a yellow '降車する。' button, a light blue '後回しにする' button, a pink '休み登録' button, a dark red '登録データ削除' button, and a light blue 'まとめて処理する' button at the bottom. Both screens are labeled 'PAD' and '[FORM12]' at the bottom.

図 2-74 乗降者記録画面

- 本画面の概要
 - 利用者の乗降車を記録する画面
 - 利用者の乗車処理・降車処理を行う
- 本画面から利用する機能
 - 【FN103】乗降車記録

3) 【HW003】スマートフォン（利用者アプリ）

【UI201】ログイン画面



図 2-75 ログイン画面

- 本画面の概要
 - 利用者がシステムにアクセスするために本人確認を行う認証用の画面
- 本画面から利用する機能
 - 【FN101】認証

【UI202】送迎状況確認画面 <改修機能>



図 2-76 送迎状況確認画面

- 本画面の概要
 - 利用者の送迎予定を確認する画面
 - 迎え（利用者の自宅から施設へ）、送り（施設から利用者の自宅へ）の情報を表示
 - 送迎依頼ができる
- 本画面から利用する機能
 - 【FN203】送迎確認

【UI203】送迎状況（車両現在位置）確認画面

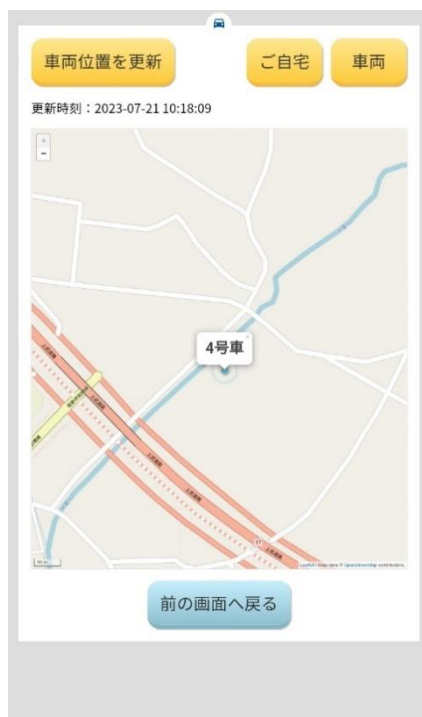


図 2-77 送迎状況確認画面

- 本画面の概要
 - 利用者が乗車する送迎車両の現在位置を地図上で確認する画面
 - 地図情報と送迎車両の位置を表示
 - 車両位置を更新することにより車両の位置情報を更新
- 本画面から利用する機能
 - 【FN203】送迎確認
 - 【FN401】地図表示

【UI204】送迎依頼（乗車場所選択）画面



図 2-78 送迎依頼（乗車場所選択）画面

- 本画面の概要
 - 利用者が送迎依頼をするときに乗車場所を選択する画面
 - あらかじめ利用者マスタに登録された乗降地点情報を表示（最大 5 箇所まで）
- 本画面から利用する機能
 - 【FN204】送迎依頼

【UI205】送迎依頼（目的地選択）画面



図 2-79 送迎依頼（目的地選択）画面

- 本画面の概要
 - 利用者が送迎依頼をするときに目的地を選択する画面
 - あらかじめ利用者マスタに登録された乗降地点情報を表示（最大 5 箇所まで）
 - 背景が反転しているボタンは乗車位置（前画面で選択済み）
- 本画面から利用する機能
 - 【FN204】送迎依頼

【UI206】送迎依頼（人数・座席選択）画面



図 2-80 送迎依頼（人数・座席選択）画面

- 本画面の概要
 - 利用者が送迎依頼をするときに同乗者や車椅子が乗れる座席を選択する場面
 - 自分（座席又は車椅子）と添乗者（座席）を指定
- 本画面から利用する機能
 - 【FN204】送迎依頼

【UI207】送迎依頼（確認）画面

4. この内容でお申込みしますか？

▼乗車場所
ヤオコー 関根店

▼降車場所
フレッセイ 荒牧店

▼人数
通常座席 1人

車を探す

1つ前に戻る

図 2-81 送迎依頼（確認）画面

- 本画面の概要
 - 利用者が指定した送迎依頼内容を確認する画面
 - 乗車位置、降車位置、指定した座席を選択
 - ボタンをタップすることにより、車両を検索
- 本画面から利用する機能
 - 【FN204】送迎依頼

【UI208】送迎依頼（完了）画面

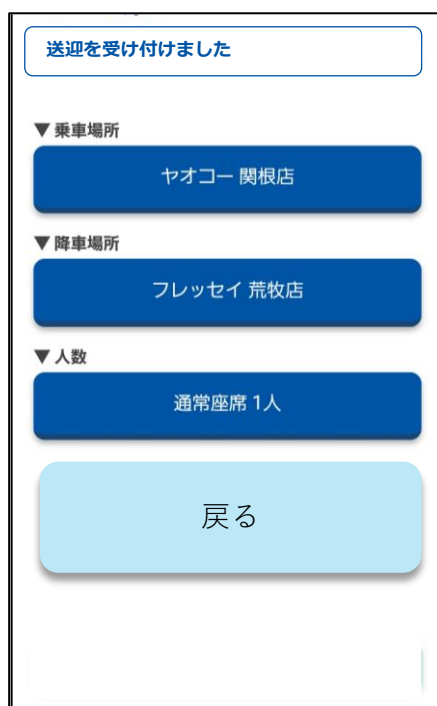


図 2-82 送迎依頼（完了）画面

- 本画面の概要
 - 利用者による送迎依頼を受け付けたことを伝える画面
 - 依頼された情報をサーバーへ送信
- 本画面から利用する機能
 - 【FN204】送迎依頼

3. 開発するシステム：非機能要件（NF）

3-1. 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	NF001	システムの連続稼働時間	<ul style="list-style-type: none"> 送迎管理（委託/受託）システムは、日中 8 時～18 時の 10 時間程度稼働すること。ドライバーアプリは、朝・夕の時間帯において、各 2 時間以上の安定動作時間を確保すること
	NF002	安定動作時間	<ul style="list-style-type: none"> 日中 8 時～18 時の時間帯にシステム停止を行わない。モジュール入替えやバックアップ作業は上記時間を除いた時間帯に実施すること
	NF003	システム復旧時間	<ul style="list-style-type: none"> 業務停止が伴う障害が発生した際に、インスタンス停止の検知からコンテナの復旧まで 3 時間以内に対応可能なマニュアル等を準備すること
	NF004	データの保管期間	<ul style="list-style-type: none"> 連続的なデータ更新がサーバー上で行われるため、通常的に更新が発生するトランザクションデータは過去 3 年間分に遡って検索・利用できること
性能・拡張性	NF005	データの読込速度	<ul style="list-style-type: none"> システム利用者が不快なくシステム運用を行うことを想定し、送迎管理システムのデータ検索・照会は通常時は 5 秒以内、ピーク時は 10 秒以内で行うこと。ドライバーアプリの送迎確認、マップ表示は通信状態を考慮して 15 秒以内で行うこと
	NF006	システムの処理実行速度	<ul style="list-style-type: none"> 送迎計画シミュレーションの実行から結果表示まで 20 秒以内で実施すること
	NF007	画面描画のフレームレート	<ul style="list-style-type: none"> ドライバーアプリは、マップ表示及び操作（パン・ズーム・回転）中において、1 秒間当たり 30 フレーム（30fps）以上（注）の描画性能を維持できること 注：ユーザーがストレスなく操作可能な基準
	NF008	拡張性	<ul style="list-style-type: none"> 現行のシステム構成において、【NF005】～【NF007】に定義された要件に対し、最大 50 施設（1 日当たり 5 台×2 便の送迎）まで対応可能とすること 想定規模を超える場合は、状況を監視し、必要に応じてスケールアップ可能な構成とすること
運用・保守性	NF009	セキュリティ	<ul style="list-style-type: none"> 対象となるデータ及びシステムの保守のために、IPA 発行「安全なウェブサイトの作り方」にのっとりセキュリティルールを設定すること
	NF010	認証	<ul style="list-style-type: none"> 利用ユーザーを特定するためにユーザー名、メールアドレス、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

3-2. 非機能要件の詳細

【NF001】システムの連続稼働時間

- 本非機能要件の概要
 - 送迎管理（委託/受託）システムは、日中 8 時～18 時の 10 時間程度稼働すること。ドライバーアプリは、朝・夕の時間帯において、各 2 時間以上の安定動作時間を確保すること
- 設定期間理由
 - 送迎計画は前日に毎日作成を行うため、日中の業務時間帯にいつでもシステムが利用できることが要求される。また、当日休みや変更の連絡対応を行えるよう、8 時にはシステムが稼働している状態を確保する。福祉事業者や観光においては連続して送迎対応を行う必要があるため、2 時間連続してシステムが稼働している状態を確保する。

【NF002】安定動作時間

- 本非機能要件の概要
 - 日中 8 時～18 時の時間帯にシステム停止を行わない。モジュール入替えやバックアップ作業は上記の時間を除いた時間帯に実施すること
- 設定期間理由
 - NF001 の理由から、保守メンテナンスやバックアップ等、システムの安定稼働を維持するために必要な作業については、システム停止を行える時間帯をあらかじめ定めることにより、利用者との不要なトラブルを回避する。

【NF003】システム復旧時間

- 本非機能要件の概要
 - 業務停止に伴う障害が発生した際に、インスタンス停止の検知からコンテナの復旧まで 3 時間以内に対応可能なマニュアル等を準備すること
- 設定期間理由
 - 送迎管理システムは、アプリケーションとその依存関係をパッケージ化するコンテナ化（Docker + ECS/EKS）を採用することで、復旧時の起動時間を短縮することが可能となる。また、万が一に備え、定期的にコンテナのレプリケーション（複製）を永続ストレージ（Amazon RDS、Amazon S3）に格納することで、AWS 上にコンテナを配置し直すことで、障害耐性を高めることが可能になる。手動対応が前提のため、担当者が不在時、検知から障害の原因切り分け等の初動フェーズまでの余裕値を設けるため、「最悪ケースの保証値」として 3 時間を設定。

【NF004】データの保管期間

- 本非機能要件の概要
 - 連続的なデータ更新がサーバー上で行われるため、通常的に更新が発生するトランザクションデータは過去 3 年間分に遡って検索・利用できること

- 設定理由
 - 福祉施設、教育(学童)に関しては、毎週月曜日など、利用パターンが決定しており、送迎計画作成の時間短縮を図るため、過去に作成した送迎計画を再利用する。また、送迎実績を監査に用いることができるため、前年度に遡ってデータの確認が行えることが望ましい。あわせて、統計データによる経年比較を行えるよう、3年分のデータ蓄積を可能とする。

【NF005】データの読込速度

- 本非機能要件の概要
 - システム利用者が不快なくシステム運用を行うことを想定し、送迎管理システムのデータ検索・照会通常時は5秒以内、ピーク時は10秒以内で行うこと。ドライバーアプリの送迎確認、マップ表示は通信状態を考慮して15秒以内で行うこと
- 設定理由
 - システム利用者が不快なく運用できるように、レスポンスタイムの目安は通常時5秒以内が一般的である。ただし、3年間分のデータ蓄積を行えるよう設計すること、Webシステムのため、通信遅延を考慮して、通常時とピーク時に幅を持たせた目標とする。ドライバーアプリは、位置情報を定期的に発信する仕組みを考慮し、電波状況が悪い地域でも15秒以内に送受信を終えるようにする

【NF006】システムの処理実行速度

- 本非機能要件の概要
 - 送迎計画シミュレーションの実行から結果表示まで20秒以内で実施すること
- 設定理由
 - 福祉施設の共用送迎については、4施設（福祉施設）の1日当たり利用者300人、車両20台を想定しており（実施計画書P48.実施エリア一覧）、往路・復路でそれぞれ150名分20台の送迎計画のシミュレーション行える必要がある。その組合せは11,175通りある。利用者間の距離、時間の計算はGoogle Maps PlatformのAPIを用いるが、呼出し回数が加速度的に増えることによる利用頻度の抑制を行う設計にする。【NF004】にも挙げたように、実際には過去の計画を再利用するため、車両単位でのシミュレーションで対応する。4名乗車の場合6通り、7名乗車の場合21通り、9名乗車の場合36通り、の最適ルートの計算と結果表示を行う。なお、距離と時間で最適ルートが異なること、Aさん→Bさん、Bさん→Aさんで移動時間が変わるケースを考慮すると、実際はそれぞれ倍の組合せ計算（API呼出し）が必要となる。シミュレーション実行は前日に実施することを基本とすることから、素早いレスポンスが必ずしも必要ないため、ユーザビリティとしても20秒は許容可能と判断し、本要件とした。なお、観光/教育(学童)については1日の想定利用人数、車両が少ないため、包含する。

【NF007】画面描画のフレームレート

- 本非機能要件の概要
 - ドライバーアプリは、マップ表示及び操作（パン・ズーム・回転）中において、1秒間当たり30フレーム（30fps）以上(注)の描画性能を維持できること
- 設定理由

- FPS の目安として、以下が推奨されている
- 30fps 以上 ユーザーがストレスなく操作可能な基準
- 60fps 以上 スマホアプリで一般的な快適操作レベル
- 90fps 以上 VR やゲーミングなど特殊用途向け。高品質保証
- 運転手は運転操作時に使用しない想定であるため、運転時以外にストレスなく操作可能な基準として、30fps を確保する

【NF008】 拡張性

- 本非機能要件の概要
 - 現行のシステム構成において、NF005～NF007 に定義された要件に対し、最大 50 施設（1 日当たり 5 台×2 便の送迎）まで対応可能とすること
 - 想定規模を超える場合は、状況を監視し、必要に応じてスケールアップ可能な構成とすること
- 設定理由
 - 本システムは今後の利用施設数や運用形態の拡大を想定しており、初期導入時点での想定規模（3 施設・1 日 5 台×2 便）に問題なく対応可能であることを前提とする。将来的な増加（50 施設）に対しても、システムが柔軟に対応できる構成としておくことで、再設計や全面更改のリスクを回避し、保守性・投資効率を確保するため。

【NF009】 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の保守のために、IPA 発行「安全なウェブサイトの作り方」にのっとりセキュリティルールを設定すること
- 設定理由
 - 最低限盛り込むべきセキュリティ要件として、①認証・認可（ユーザー認証、パスワード強度、権限管理）、②通信の安全（通信の暗号化、セッションIDの盗聴防止）、③入力の検証（XSS対策、SQL インジェクション対策、CSRF 対策）、④ログと監査（アクセスログ、監査イベントログ）が想定できる
 - 今回の実証実験で開発するシステムでは、以下のセキュリティ要件を定める
 1. アプリケーションは、全ての通信を HTTPS により暗号化すること
 2. システム管理者とシステム利用者で操作できる機能を制限すること
 3. ユーザー認証には ID・パスワードを用い、パスワードは 3 種類・8 文字以上を設定こと
 4. 入力値は全てサーバー側でバリデーションし、SQL インジェクション・XSS 等の攻撃に耐性を持つこと
 5. リクエストヘッダの確認を行い、POST リクエストの正当性を検証すること
 6. 認証後のセッションには有効期限を設定し、一定時間操作がない場合は自動ログアウトすること
 7. アカウントロックの機能により、ブルートフォース攻撃を防止すること
 8. エラーメッセージは内部構造を推測されないよう簡易な内容とすること
 9. アクセスログ、操作ログ、エラーログを記録すること

【NF010】 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名、メールアドレス、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
 - 【NF008】で挙げたセキュリティ方針、「2. システム管理者とシステム利用者が操作できる機能を制限すること」、「3. ユーザー認証には ID・パスワードを用い、パスワードは3種類・8文字以上を設定こと」に準拠する。利用ユーザーが特定できることで、「9. アクセスログ、操作ログ、エラーログを記録すること」にも準拠できる。なお、パスワードの平文保存・簡易なハッシュ（MD5等）はIPAでも明確に禁止されている。

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	運転手及び従業者情報	Excel	実証先事業者（福祉施設、旅館）	FN002
DT002	車両情報	Excel	実証先事業者（福祉施設、旅館）	FN003
DT003	利用者住所情報	Excel	実証先事業者（福祉施設、習い事施設）	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】 運転手及び従業者情報

- 本データの概要
 - 本システムで職員データ（運転手・添乗員）として利用する情報
 - 実証先が提供する運転手、従業員の名前
 - 同性同名を考慮し、Staff_Id で一意に管理（改姓後も同一 ID で管理可）
- データ定義

表 4-2 Staff 情報のサンプル・イメージ

No.	日本語名称	項目長	必須	書式・選択肢など
1	施設 CD	10	○	0000000001
2	職員 CD	255	○	1001
3	職員名	255	○	山田太郎
4	職員名 (カナ)	255	○	ヤマダタロウ
5	職種	1	○	1：介護 2：看護 3：相談員 4：訓練士 5：ドライバー 6：経理 7：他
6	送迎区分	1	○	0：不可（未設定） 1：可能
7	添乗区分	1	○	0：不可（未設定） 1：可能
8	運転手タイプ	1		1：介護職員 2：専属ドライバー 3：応戦 4：退職 5：送迎から除外 null：送迎不可（未設定）

- データ形式
 - Excel
- 出所
 - 実証先事業者（福祉施設、旅館）

【DT002】車両情報

- 本データの概要
 - 本システムで車両データとして利用する情報
 - 実証先が提供する車両情報
 - 同型車両、同ナンバープレートの保有を考慮し、car_Id で一意に管理
- データ定義

表 4-3 car 情報のサンプル・イメージ

No.	日本語名称	項目長	必須	書式・選択肢など
1	施設 CD	10	○	0000000001
2	車両 ID	int	○	1
3	車両番号	255	○	23-45
4	登録名	255	○	ハイエース
5	定員	Int	○	10
6	最大車椅子座席数	int	○	2
7	車両パターン	int	○	1：ハイエース 2：キャラバン 3：ミニバン 4：軽 5：その他

- データ形式
 - Excel
- 出所
 - 実証先事業者（福祉施設、旅館）

【DT003】利用者住所情報

- 本データの概要
 - 本システムで利用者の情報（利用者、利用者週間来館パターン、利用者乗り合わせ）として活用する情報
 - 実証先が提供する利用者情報
 - 同性同名者を考慮し、customer_id で一意に管理
- データ定義

表 4-4 customer 情報のサンプル・イメージ

No.	日本語名称	項目長	必須	書式・選択肢など
1	施設 CD	10	○	0000000001
2	利用者 ID	Int	○	1001
3	利用者名	255	○	岡山花子
4	利用者名（カナ）	255	○	オカヤマハナコ
5	歩行器の大きさ	2,1	○	0.3（小） 0.5（中） 0.7（大） 1.0（特大）
6	車椅子の使用	Int	○	0：なし 1：あり
7	住所ラベル B	40	○	
8	郵便番号	7	○	3710847
9	住所	120	○	群馬県前橋市〇〇町〇〇丁目〇〇番地
10	マンション名・号 室	40		
11	電話番号	13		086-232-2405
12	地点緯度	9,6	○	住所から変換（Google Maps / WGS84(EPWG:4326)・10 進度）（例 36.393227）
13	地点経度	9,6	○	住所から変換（Google Maps / WGS84(EPWG:4326)・10 進度）（例 139.044281）
14	停車緯度	9,6		マップのピン位置から変換（Google Maps / WGS84(EPWG:4326)・10 進度）（例 36.392418）
15	停車経度	9,6		マップのピン位置から変換（Google Maps / WGS84(EPWG:4326)・10 進度）（例 139.044377）
16	距離	Int	○	単位：km 施設の緯度・経度と地点緯度・経度から変換（例 0.1）
17	時間	Int	○	単位：分 施設の緯度・経度と地点緯度・経度から変換（例 10）
	住所 C～住所 E			
	以下省略			No.8～No.17 と同様

施設送迎共同管理システム システム設計書

- データ形式
 - Excel
- 出所
 - 実証先事業者（福祉施設、旅館）

5. 用語集

用語	読み仮名	定義・説明
共用送迎	きょうどうそうげい	福祉・観光・教育分野の複数の施設が連携し、利用者の同乗及び車両の共用を行う送迎方式
要介護・要支援者	ようかいご・ようしえんしゃ	介護保険法に基づき厚生労働省が定める区分により、介護サービスの提供対象となる者。要介護は5段階、要支援は2段階に区分される
通所型施設	つうしょがたしせつ	利用者（要介護者又は要支援者）が通所し、機能訓練等を行う施設。多くの施設において、利用者の自宅と施設間の送迎が実施される
送迎ルート	そうげいーと	複数の利用者を各自宅から施設まで送迎するために設定される経路。地図上に表示される
誘導経路	ゆうどうけいろ	ラストワンマイル（利用者の自宅付近）における通過経路。Google MapsやOpenStreetMapのルート検索ではラストワンマイルの精度が悪く、細道やマンション駐車場の停車位置等を示すために設定する
委託	いたく	共用送迎の実施を他者に依頼すること又はその依頼主体
受託	じゅたく	共用送迎の依頼を受け、実施すること又はその実施主体
自来	じらい	福祉施設が提供する送迎車両を利用せず、利用者が自ら（家族の車両等により）来館すること
自退	じたい	福祉施設が提供する送迎車両を利用せず、利用者が自ら（家族の車両等により）退館すること
ゲート車	げーとしゃ	車椅子が乗せられるように改造した車両のこと。リフト車と同じ
リフト車	りふとしゃ	車椅子が乗せられるように改造した車両のこと。ゲート車と同じ
施設管理者ユーザー	しせつかんりしゃゆーざー	施設において送迎計画の作成及び更新を担当する責任者
職員	しょくいん	施設職員のうち、送迎業務に従事する運転手及び添乗員
利用者	りようしゃ	送迎車両に乗降する者。福祉施設においては介護認定を受けた高齢者（要介護者又は要支援者）、観光施設においては観光客、教育施設においては学童等を指す



施設送迎共同管理システム 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

モビリティサービス推進課

受託者: 一般社団法人ソーシャルアクション機構

ソーシャルムーバー株式会社

EXA INNOVATION STUDIO株式会社

株式会社Cuon

株式会社岡山トヨタシステムサービス