

デマンド型交通リアルタイム経路検索システム システム設計書



目次

1. 開発スコープ	- 1 -
1-1. 概要	- 1 -
1-1-1. GunMaaS システム (チケット認証、データ取得・統合モジュール)	- 1 -
1-1-2. デマンド型交通リアルタイム経路検索システム	- 2 -
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. GunMaaS システム (チケット認証、データ取得・統合モジュール)	- 7 -
2-1. 機能要件 (FN/SL/AL/CO/HW/IF/UI)	- 7 -
2-1-1. システム機能 (FN)	- 7 -
2-1-2. システムコンポーネント (CO)	- 33 -
2-1-3. ハードウェア (HW)	- 35 -
2-1-4. データインターフェース (IF)	- 41 -
2-1-5. ユーザインターフェース (UI)	- 78 -
2-2. 開発するシステム：非機能要件 (NF)	- 85 -
2-2-1. 非機能要件一覧	- 85 -
2-2-2. 非機能要件の詳細	- 86 -
3. デマンド型交通リアルタイム経路検索システム	- 87 -
3-1. 機能要件 (FN/SL/AL/CO/HW/IF/UI)	- 87 -
3-1-1. システム機能 (FN)	- 87 -
3-1-2. システムコンポーネント (CO)	- 131 -
3-1-3. ハードウェア (HW)	- 133 -
3-1-4. データインターフェース (IF)	- 136 -
3-1-5. ユーザインターフェース (UI)	- 156 -
3-2. 開発するシステム：非機能要件 (NF)	- 168 -
3-2-1. 非機能要件一覧	- 168 -
3-2-2. 非機能要件の詳細	- 169 -
4. 実証調査に利用するデータ (DT)	- 170 -
4-1. 実証調査に利用するデータ一覧	- 170 -
4-2. 実証調査に利用するデータの詳細	- 171 -
5. 用語集	- 172 -

1. 開発スコープ

1-1. 概要

1-1-1. GunMaaS システム (チケット認証、データ取得・統合モジュール)

交通モードの違いを利用者が意識することなく移動できる「マルチモーダルな移動体験の実現」の達成のため、前橋市をフィールドとし、域内の交通モードを網羅した ABT 方式(アカウント単位でチケットを管理する仕組み)による交通系 IC カードチケットサービスを実現する。

前橋市では、既に路線バスの「バス車載器システム」として、ABT 方式による交通系 IC カードチケットサービスが GunMaaS として実現されている。これにより、路線バス乗降時に交通系 IC カードをタッチするだけで精算が完了するというユーザ利便性の高さに加え、精算時に取得した OD データを可視化・分析することで区間ごとの乗降客数を時間別に把握でき、ダイヤ改正や路線改定に活用することが可能となっている。

一方で、地域住民の主たる公共交通である「上毛電気鉄道」においては、降車時に整理券を運転士に提示の上で現金支払いする運用が主となっており、精算対応に一定の時間を要するなどユーザ利便性に課題がある。さらに、地域住民は複数の公共交通を乗り継いで目的地まで移動するが、上毛電気鉄道の OD データが取得できないことから、交通モードを横断した「人の移動」の把握に至らず、結果域内全体でデータに基づく信頼性の高い公共交通計画の策定が困難な状況にある。

鉄道向けのチケットサービス代表例として自動改札機があるが、相応のコストを要するために地域鉄道での導入には高いハードルがある。そのため、本プロジェクトでは、路線バス用の「バス車載器システム」を上毛電気鉄道向けに導入することでチケットサービスを実現することを目指す。「バス車載器システム」は、自動改札機と比較して機能要件が限定されているため、構成を簡素化でき、その結果として比較的安価に導入することが可能である。他方、地域鉄道のチケットを路線バス用の車載器で行うためには、鉄道に特有の運用を考慮する必要がある。例えば、地域鉄道は一編成あたり複数の車両を運用するため、出入口も複数となる。また、駅には改札があるため、車両内でのチケットと改札におけるチケットを両立させる運用整理が求められる。このような鉄道固有の運用に対応するため、本プロジェクトではバス車載器システムの改修及びオペレーションの開発を行った。

この改修により上毛電気鉄道でバス車載器システムの運用が可能となるため、前橋市では全ての主要な交通モードにおいて交通系 IC カードでの乗車が可能となる。さらに、ABT 方式によるチケットサービスにより、域内の乗降実態をアカウント毎のデータによって網羅的に把握できるようになる。これにより、域内で「人の移動」に基づく公共交通計画の策定に加え、個人に着目したミクロな移動実態の分析なども可能となる。

1-1-2. デマンド型交通リアルタイム経路検索システム

現在の経路検索システムでは、デマンド交通のリアルタイム情報を経路検索に反映できない。定時定路線型交通である路線バス等では、ルートやダイヤを参照することで実際の運行状況と整合的な経路検索を提供することができるが、予約に応じて随時運行するデマンド交通では、参照可能なルートやダイヤが存在しないため、到着時間や所要時間、ルート、乗換等の経路案内を精度高く提供することは難しかった。また、デマンド交通の経路検索を簡易的に実現する手法として、仮のルート設定や時刻表設定を用いる方法もあるが、実際の運行とは異なる案内となるため、他の公共交通機関（鉄道やバス）と連携した複合的な経路検索を提供することは難しい。

現状において、デマンド交通が経路検索に反映されるケースは限定的であり、その多くはファーストワンマイルまたはラストワンマイル区間での補完的な利用にとどまっている。これらは、あらかじめデータベースに登録された地点間の所要時間や、最寄駅との接続関係といった静的情報を用いた簡易的な検索として実装されている。一方、鉄道や路線バス等の定時定路線型交通については、既存の経路検索エンジンを用いて独立に検索が行われており、デマンド交通区間と一体的に扱われているわけではない。その結果、現在提供されているデマンド交通を含む経路案内は、公共交通機関の経路検索結果に、ファースト／ラストワンマイル区間の情報を付加した形にとどまっている。

経路検索エンジンは、公共交通ネットワーク、検索アルゴリズム及びネットワークに紐づく各種データを基に経路を探索する仕組みである。データとは、鉄道・バス・航空機・船舶等の時刻表、運賃、乗換標準時間等の諸要素を指す。膨大な検索リクエストを極めて短時間で処理するため、エンジンはこれらのデータを内包したライブラリとして最適化されており、ここに収録されていない交通手段は検索対象から除外される構造となっている。デマンド交通を検索対象に組み込むには、既存の公共交通と同様、ネットワーク情報と運行データをライブラリ化する必要がある。

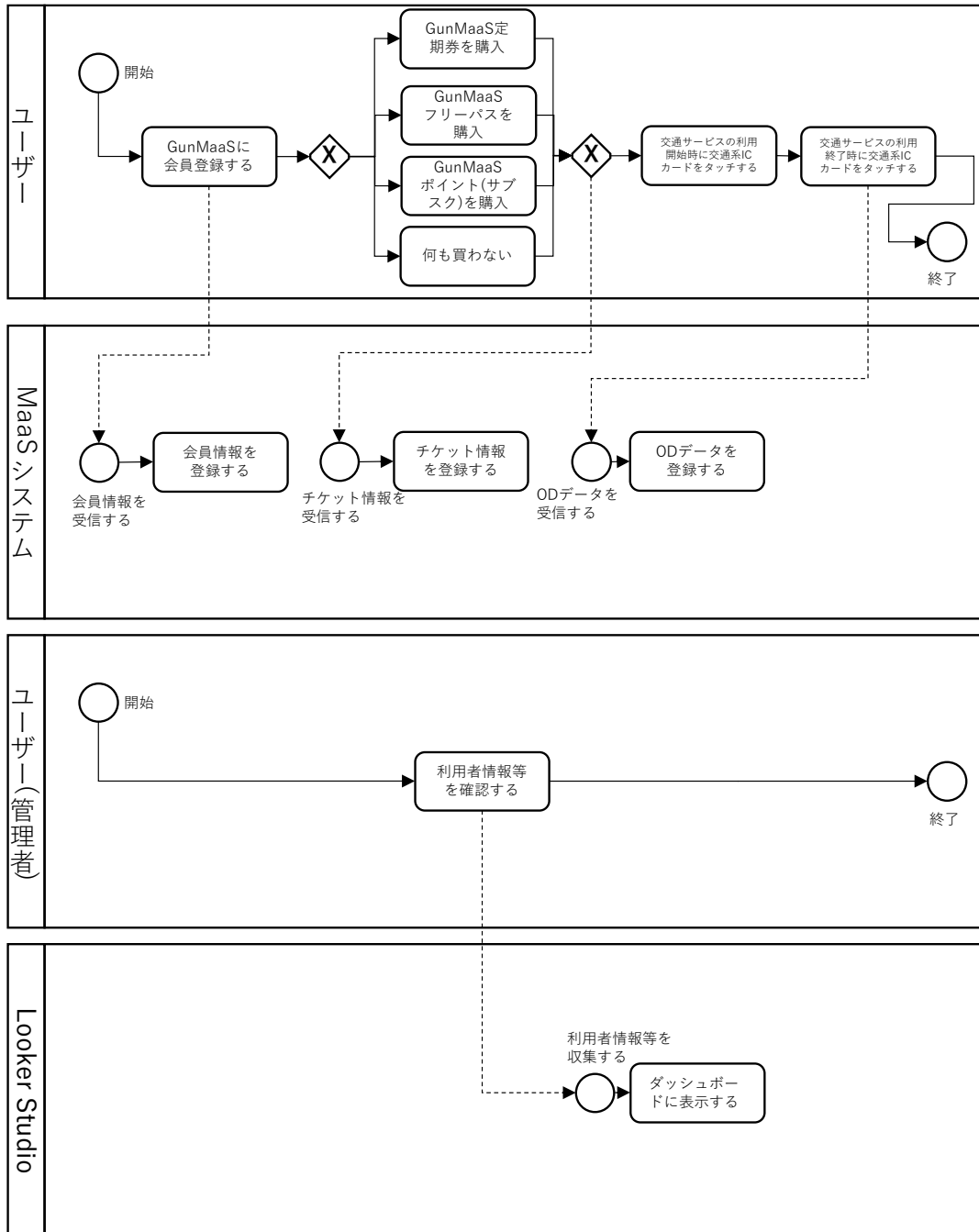
ダイヤ検索は、利用者の指定時刻に基づき、乗換時間等を考慮しながら最適な便を経路に割り当てる。しかし、エンジンは内包された静的データを参照しており、外部から動的なデータを逐次取り込む仕組みは想定されていない。また、デマンド交通の時刻情報は外部の配車システムへの照会により生成される動的なものであり、かつ利用者が限定されるパーソナルな性質を持つ。不特定多数が同一の時刻表を参照する従来の公共交通とは異なり、デマンド交通では特定の利用者の検索時のみ、その条件に合致するデータを適用する必要があるが、現行のエンジンは全利用者が共通のデータを参照することを前提として構築されている。

本開発では、経路検索エンジン自体の改修ではなく、出力された検索結果に対して外部から動的データを付与・更新する「後処理（ポストプロセッシング）」のアプローチを採用し、動的情報の反映を実現する。経路検索エンジンには、検索結果に対して前後便検索等を行うパーソナライズされた編集機能が備わっており、本開発ではこの仕組みを活用する。既存の経路検索結果を基点として、外部から動的な情報を付与・更新することで、デマンド交通の運行実態を反映した経路案内を実現する。

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

【GunMaaSシステム（チケット認証、データ取得・統合モジュール）】



【デマンド型交通リアルタイム経路検索システム】

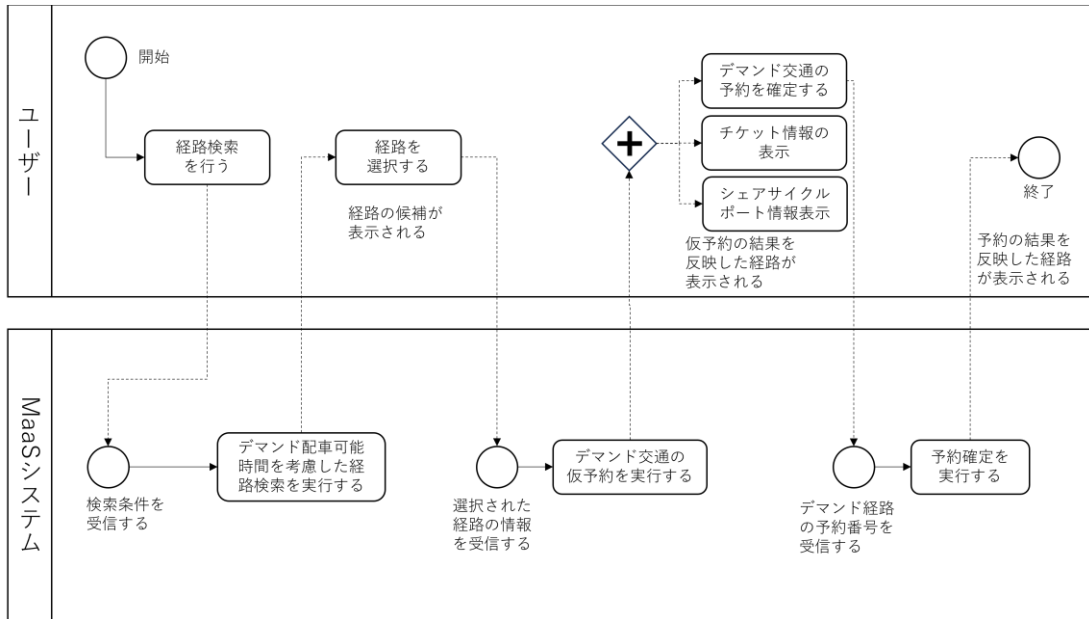


図 1-1 業務フローの全体像

2. システムシーケンス図

【GunMaaS システム (チケット認証、データ取得・統合モジュール)】

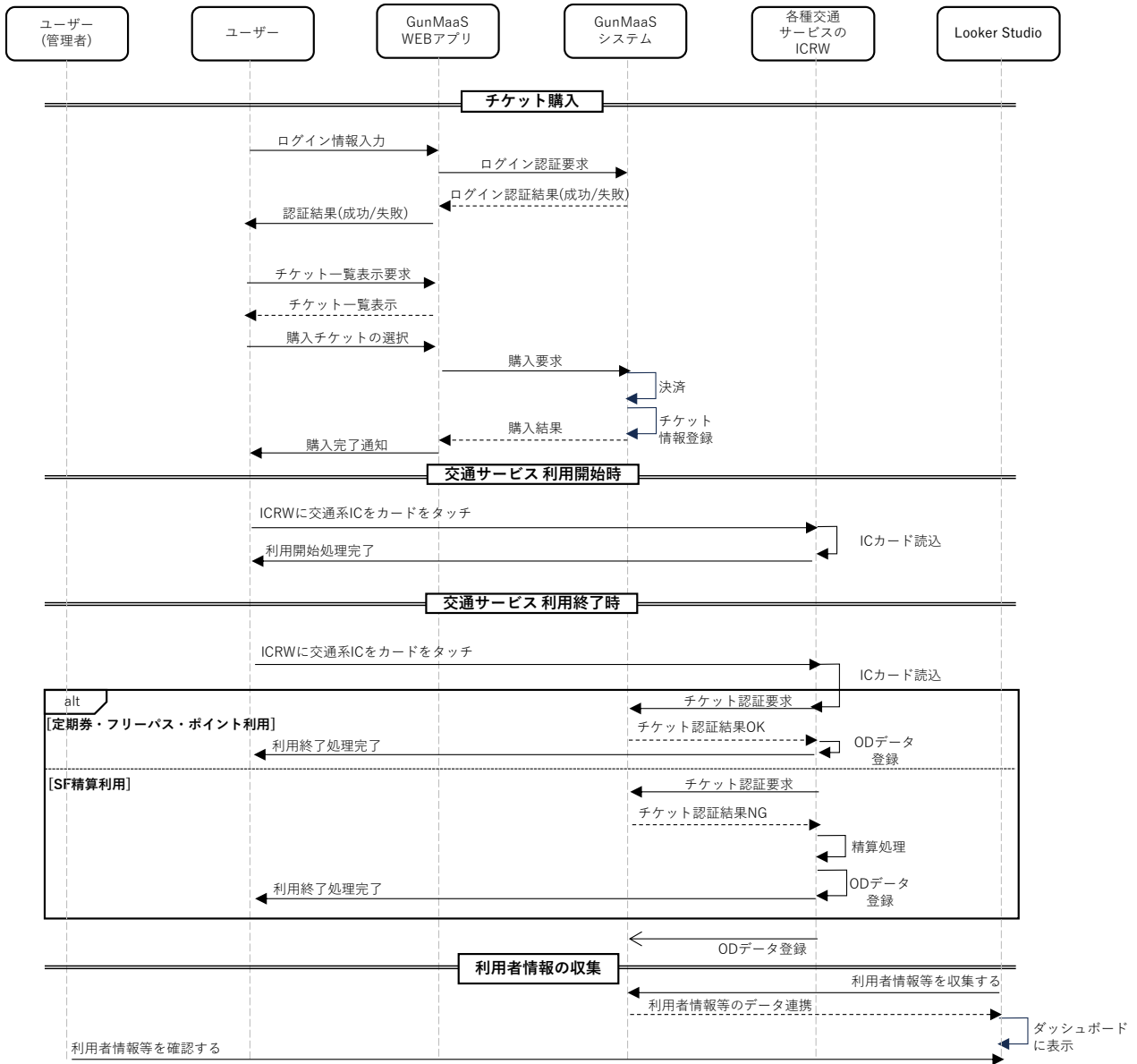


図 1-2 GunMaaS Web アプリへのログイン～チケット購入～交通サービス利用開始・終了時

【デマンド型交通リアルタイム経路検索システム】

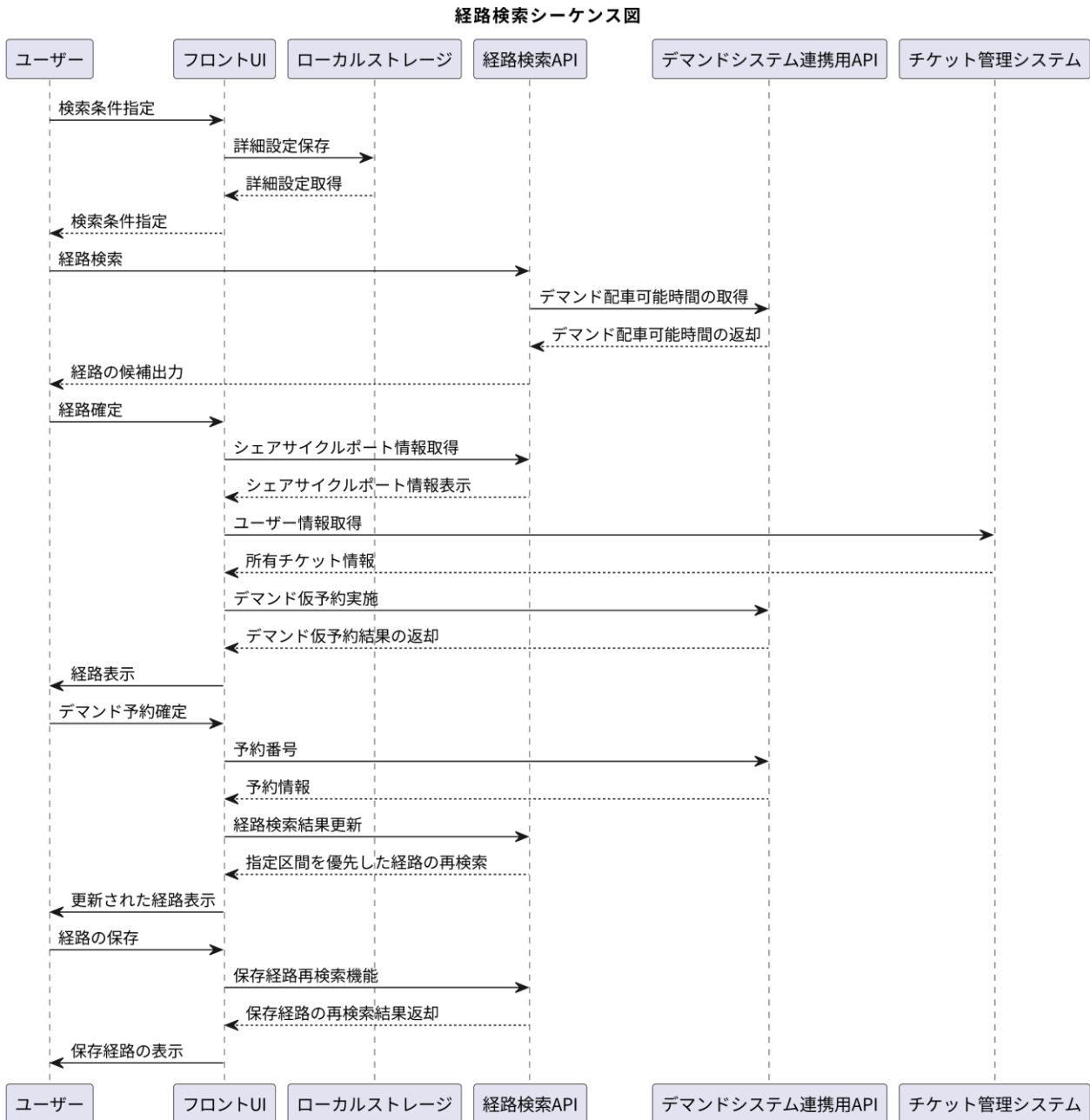


図 1-3 高度な経路検索、デマンド交通の仮予約、仮予約確定、保存経路の再検索

2. GunMaaS システム (チケット認証、データ取得・統合モジュール)

2-1. 機能要件 (FN/SL/AL/CO/HW/IF/UI)

2-1-1. システム機能 (FN)

2-1-1-a. システムアーキテクチャ

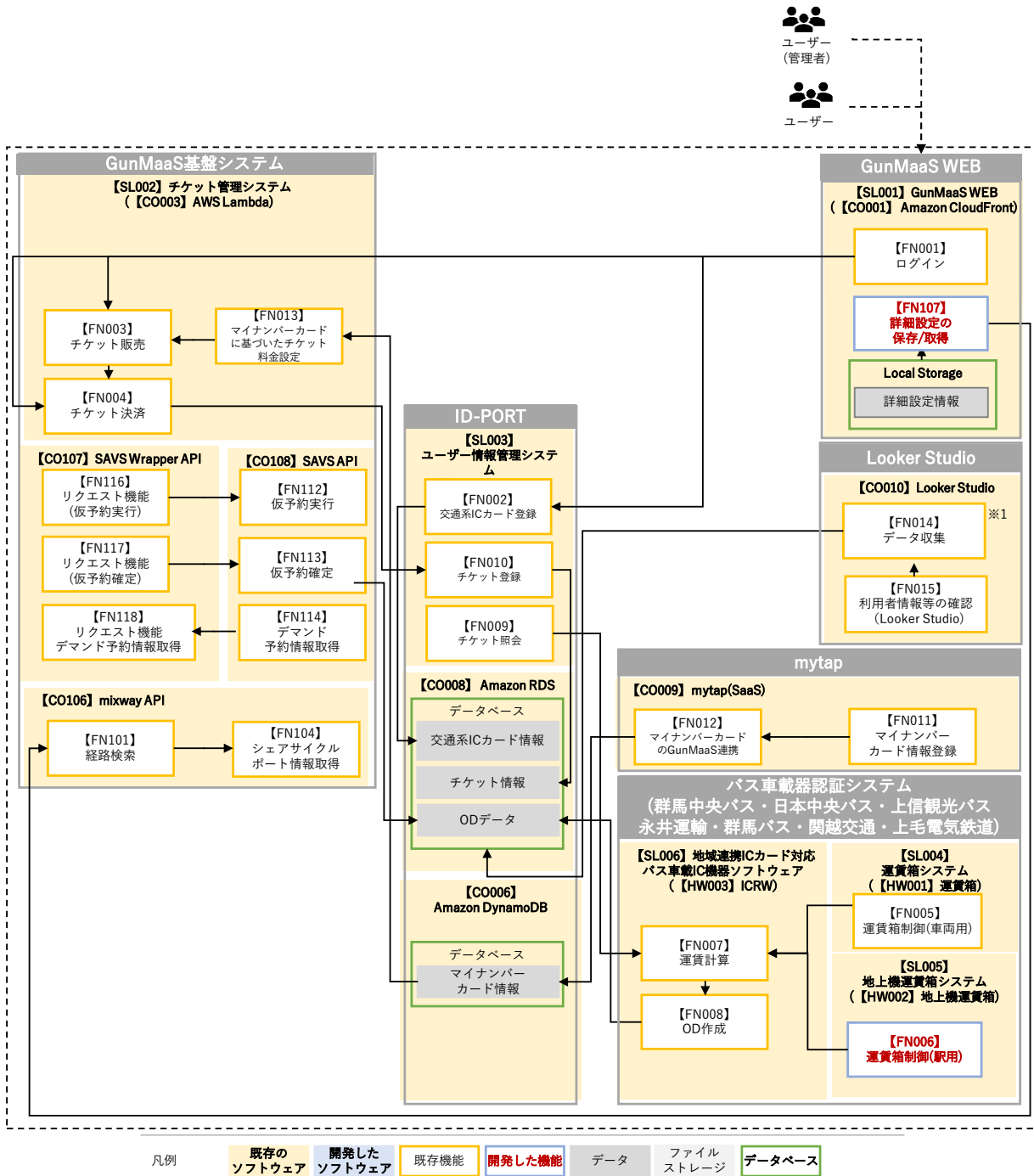


図 2-1 システムアーキテクチャ

※ 1 取得対象のデータは、バス利用ログ・デマンド利用ログ・タクシー利用ログを含みます。

2-1-1-b. システム機能一覧

表 2-1 機能一覧

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN001	ログイン	● ユーザが GunMaaS へログインをする際にログイン情報を認証する機能
FN002	交通系 IC カード登録	● GunMaaS のアカウント情報に交通系 IC カードの ID を登録する機能 ● 交通系 IC カードのタッチ機能を用いてシームレスな UX を提供する機能
FN003	チケット販売	● GunMaaS でチケットを販売する機能。ユーザがスマホからチケットの購入を行うことができる機能
FN004	チケット決済	● GunMaaS 内においてデジタルチケットの購入手続に伴うクレジットカード等の決済機能を提供する機能
FN005	運賃箱制御(車両用)	● バス・鉄道の車両で運賃箱と接続された IC リーダライタを稼働させるために必要な系統・停留所・運賃情報テーブル(運賃箱用)を出力する機能
FN006	運賃箱制御(駅用)	● 鉄道の地上駅で運賃箱と接続された IC リーダライタを稼働させるために必要な系統・停留所・運賃情報テーブル(運賃箱用)を出力する機能
FN007	運賃計算	● GunMaaS のチケット認証の運賃計算を実施する機能
FN008	OD 作成	● 移動データ収集のために、交通系 IC カードの OD データを作成する機能
FN009	チケット照会	● 専用端末への交通系 IC カードタッチによる GunMaaS のチケット認証を提供する機能
FN010	チケット登録	● GunMaaS のチケット登録機能。購入したチケットを登録し、アクティベートにより使用できるようにする機能
FN011	マイナンバーカード情報登録	● mytap を通じてマイナンバーカード情報を取得する機能
FN012	マイナンバーカード情報の GunMaaS 連携	● mytap で登録したマイナンバーカード情報を GunMaaS に連携する機能
FN013	マイナンバーカードに基づいたチケット料金設定	● デジタルチケットの購入手続の際に、連携されたマイナンバーカード情報を元に料金を更新する機能
FN014	データ収集	● 各交通モードの利用データ・経路検索データ・ユーザの属性情報を収集する機能
FN015	利用者情報等の確認	● 交通モードの利用データをダッシュボードに表示する機能

2-1-1-c. 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】ログイン

- 本システム機能の概要
 - ユーザが GunMaaS へログインをする際に、ログイン情報を認証する機能。
- 本システム機能の入力・処理・出力のフローチャート

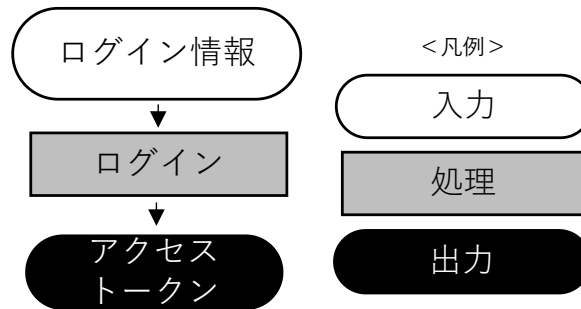


図 2-2 【FN001】のフローチャート

- 本システム機能の処理の詳細
 - ログイン
 - ◇ 処理内容
 - ユーザが入力したログイン情報を認証する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ ログイン情報
 - データの内容
 - ログイン ID、パスワード
 - データの形式
 - -
 - 利用するデータインターフェース
 - -
 - 出力

- ◇ アクセストークン
 - データの内容
 - GunMaaS にログイン済みであることを証明する情報
 - データの形式
 - -
 - 利用するデータインターフェース
 - -

【FN002】交通系 IC カード登録

- 本システム機能の概要
 - ユーザが Web 画面上で入力した交通系 IC カードの IC カード番号を登録する機能。
- 本システム機能の入力・処理・出力のフローチャート

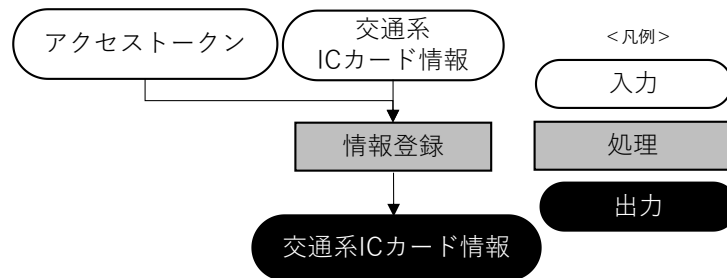


図 2-3 【FN002】のフローチャート

- 本システム機能の処理の詳細
 - 情報登録
 - ◇ 処理内容
 - ユーザの入力情報に基づき、交通系 IC カードの ID 情報を ID-PORT に登録する。この交通系 IC カードの情報はアカウント情報に紐づく形で登録される。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 交通系 IC カード情報
 - データの内容
 - 交通系 IC カードのカード ID 情報
 - データの形式
 - -
 - 利用するデータインターフェース
 - -

- ◇ アクセストークン
 - データの内容
 - GunMaaS にログイン済みであることを証明する情報
 - データの形式
 - -
 - 利用するデータインターフェース
 - -
- 出力
 - ◇ 交通系 IC カード情報
 - データの内容
 - ID-PORT に登録するための交通系 IC カードのカード ID 情報
 - データの形式
 - -
 - 利用するデータインターフェース
 - -

【FN003】 チケット販売

- 本システム機能の概要
 - GunMaaS サイト上で、フリーパスチケットやシェアサイクルの利用券、定期券やサブスクリプションチケットなどの交通チケットの販売を行う。チケットの購入を行う際には登録されているマイナンバーカード情報により、割引の適用が行われる機能。
- 本システム機能の入力・処理・出力のフローチャート

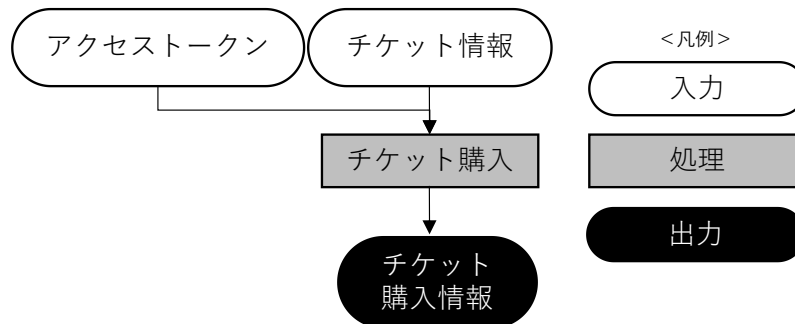


図 2-4 【FN003】 のフローチャート

- 本システム機能の処理の詳細
 - チケット購入
 - ◇ 処理内容
 - GunMaaS サイト上でログインしているユーザが選択しているチケットのチケット情報に対してチケット購入の処理を行い、チケット購入情報を出力する。
 - ◇ 利用するライブラリ

-

◇ 利用するアルゴリズム

● -

● 本システム機能の入出力データの仕様

➤ 入力

◇ チケット情報

- データの内容
 - 割引情報を含むチケットの情報
- データの形式
 - HTTP リクエスト
- 利用するデータインターフェース
 - 【IF013】 チケット料金更新 API

◇ アクセストークン

- データの内容
 - GunMaaS にログイン済みであることを証明する情報
- データの形式
 - -
- 利用するデータインターフェース
 - -

➤ 出力

◇ チケット購入情報

- データの内容
 - 購入手続に際して必要となるチケットの券種、枚数等のチケット情報
- データの形式
 - JSON
- 利用するデータインターフェース
 - 【IF001】 チケット販売 API

【FN004】 チケット決済

- 本システム機能の概要
 - GunMaaS サイト上で、ユーザが選択したチケットに基づいて作成されたチケット購入情報により購入手続を行う機能。
- 本システム機能の入力・処理・出力のフローチャート

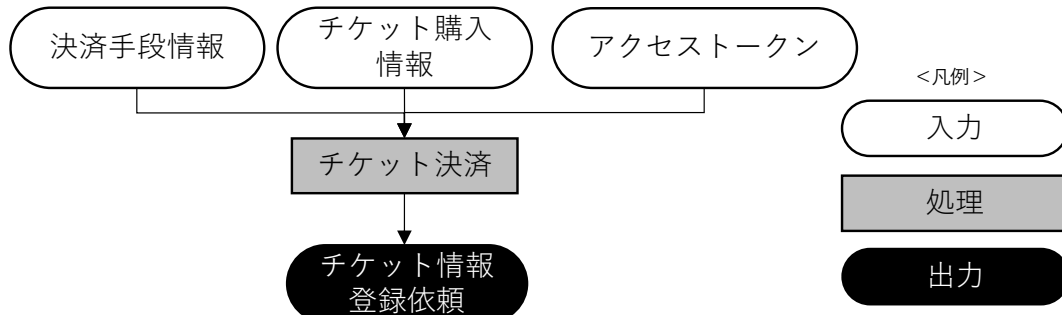


図 2-5 【FN004】 のフローチャート

- 本システム機能の処理の詳細
 - チケット決済
 - ◇ 処理内容
 - チケットの購入情報を基にチケットの決済処理を行い、チケット情報登録依頼を出力する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ チケット購入情報
 - データの内容
 - 購入手続に際して必要となるチケットの券種、枚数等のチケット情報
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF001】 チケット販売 API
 - ◇ アクセストークン
 - データの内容
 - GunMaaS にログイン済みであることを証明する情報
 - データの形式
 - -

- 利用するデータインターフェース
 - -
- ◇ 決済手段情報
 - データの内容
 - 決済手段に関連する情報
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF001】チケット販売 API
- 出力
 - ◇ チケット情報登録依頼
 - データの内容
 - ID-PORT にチケットを登録するための情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF002】チケット登録 API

【FN005】運賃箱制御(車両用)

- 本システム機能の概要
 - バス・鉄道の車両にて、運賃箱と接続された IC リーダライタにおいて、交通系 IC カードのタッチ処理を稼働させるために必要な系統・停留所・運賃情報テーブル(運賃箱用)を IC リーダライタに出力する機能。
- 本システム機能の入力・処理・出力のフローチャート

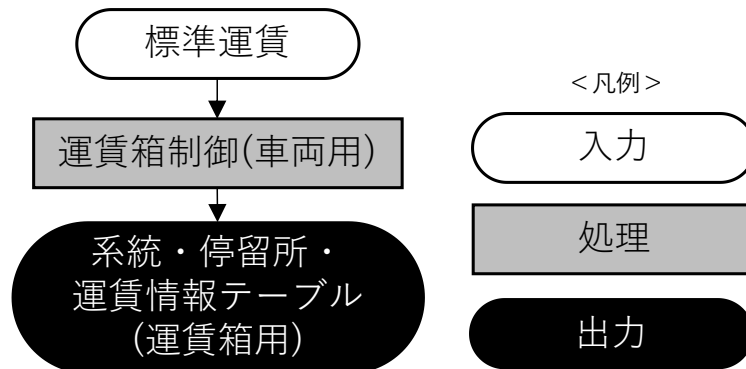


図 2-6 【FN005】のフローチャート

- 本システム機能の処理の詳細
 - 運賃箱制御(車両用)
 - ◇ 処理内容
 - 標準運賃から運賃計算の処理に必要な系統・停留所・運賃テーブル(運賃箱用)を出力して

IC リーダライターに送信する

- ◇ 利用するライブラリ
 - -
- ◇ 利用するアルゴリズム
 - -

● 本システム機能の入出力データの仕様

- 入力
 - ◇ 標準運賃
 - データの内容
 - 標準運賃
 - データの形式
 - バイナリ
 - 利用するデータインターフェース
 - 【IF005】標準運賃取得
- 出力
 - ◇ 系統・停留所・運賃情報テーブル（運賃箱用）
 - データの内容
 - 系統・停留所・運賃情報
 - データの形式
 - バイナリ
 - 利用するデータインターフェース
 - 【IF007】操作部-車載 IC 機器間

【FN006】運賃箱制御(駅用) <改修機能>

- 本システム機能の概要
 - 鉄道の地上駅にて、運賃箱と接続された IC リーダライターにおいて、交通系 IC カードのタッチ処理を稼働させるために必要な系統・停留所・運賃情報テーブル(運賃箱用)を IC リーダライターに出力する機能。

既存機能の改修内容

上毛電気鉄道では、全体最適を見据えたコスト調整のため、バス用のシステムを改修して鉄道に導入する。バス用の車載器である運賃箱は、運賃計算の処理に必要な「系統情報」「停留所情報」「運賃情報」のテーブル情報を出力するため、標準運賃データの入力が必要となる。通常、バス車両の運賃箱は、運賃箱と運賃表示器が接続されており、当該表示器から標準運賃データを取得できる。

今回、上毛電気鉄道の車両に導入する運賃箱は、同様の仕組みを導入している。一方で、今回新規導入となる地上駅に設置する運賃箱の場合は、運賃表示器に相当する標準運賃データを取得する機器が存在しないため、運賃箱内部で標準運賃データを取得可能とするよう、ソフトウェアの改修が必要となる。

- 本システム機能の入力・処理・出力のフローチャート

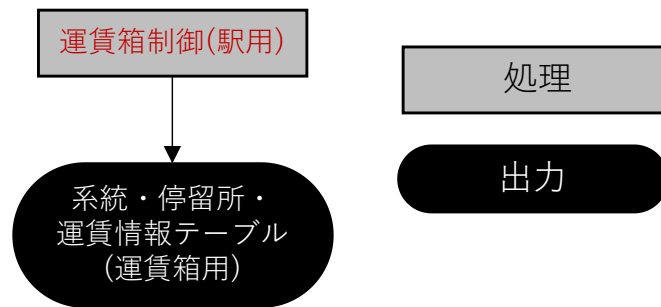


図 2-7 【FN006】のフローチャート（改修後）

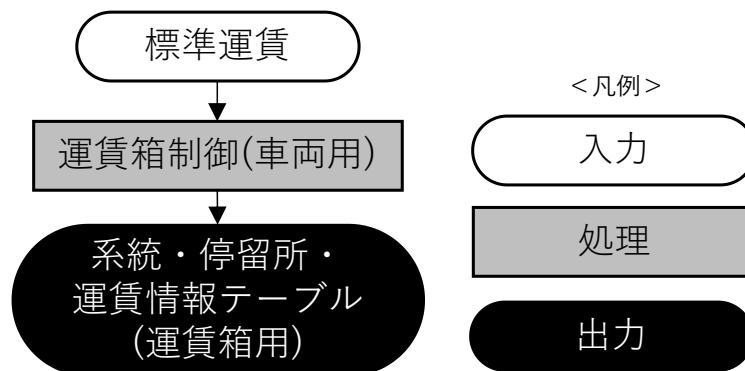


図 2-8 【FN006】のフローチャート（改修前）

- 本システム機能の処理の詳細

- 運賃箱制御(駅用)

- ◇ 処理内容

- 内部処理として標準運賃から運賃計算の処理に必要な系統・停留所・運賃テーブル(運賃箱用)を出力し、ICリーダライターに送信するため、運賃箱制御(駅用)の処理を行う。
 - 電源起動時に埋め込まれた「系統情報」「停留所情報」「運賃情報」を固定値として一括で出力する。

- ◇ 利用するライブラリ

- -

- ◇ 利用するアルゴリズム

- -

- 本システム機能の入出力データの仕様

- 入力

- ◇ 入力なし ※電源起動がトリガー

- データの内容
 - -
 - データの形式
 - -
 - 利用するデータインターフェース

- -
- 出力
 - ◇ 系統・停留所・運賃情報テーブル(運賃箱用)
 - データの内容
 - 系統・停留所・運賃情報
 - データの形式
 - バイナリ
 - 利用するデータインターフェース
 - 【IF007】操作部-車載 IC 機器間

【FN007】運賃計算

- 本システム機能の概要
 - GunMaaS のチケット認証結果に応じた運賃計算を実施する機能
- 本システム機能の入力・処理・出力のフローチャート

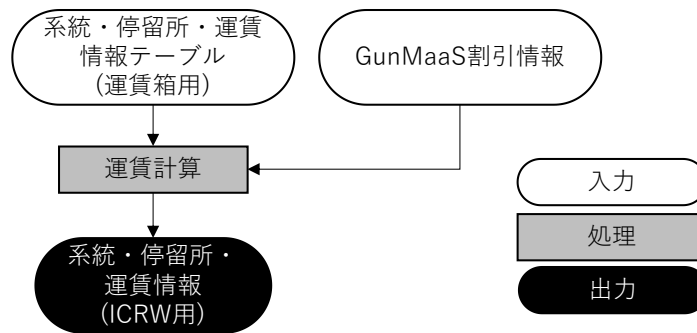


図 2-9 【FN007】のフローチャート

- 本システム機能の処理の詳細
 - 運賃計算
 - ◇ 処理内容
 - 系統・停留所・運賃情報テーブル(運賃箱用)に GunMaaS 割引情報を適用し、運賃計算を実施し、結果を系統・停留所・運賃情報 (ICRW 用) として出力する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 系統・停留所・運賃情報テーブル(運賃箱用)
 - データの内容
 - 運賃テーブルに定められた運賃
 - データの形式

- バイナリ
- 利用するデータインターフェース
 - 【IF007】 操作部-車載 IC 機器間
- ◇ GunMaaS 割引情報
 - データの内容
 - GunMaaS で設定した割引率または割引額
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF006】 パス情報照会 API
- 出力
 - ◇ 系統・停留所・運賃情報(ICRW 用)
 - データの内容
 - 運賃計算で確定した処理内容を OD 作成処理で OD データとして生成するための内部処理
 - データの形式
 - 内部処理
 - 利用するデータインターフェース
 - 【IF008】 OD 作成処理要求

【FN008】 OD 生成

- 本システム機能の概要
 - 本業務の対象の交通系 IC カードのシステムでは、IC リーダライターにタッチをすることにより乗車駅と降車駅をそれぞれ取得する。そのデータに基づき運賃の計算を行う。また利用実績データの収集を行うため OD データの作成を行う機能。
- 本システム機能の入力・処理・出力のフローチャート

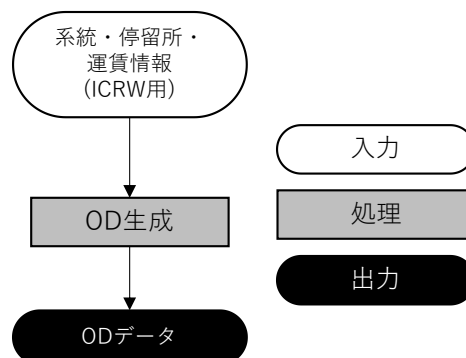


図 2-10 【FN008】 のフローチャート

- 本システム機能の処理の詳細
 - OD 生成

- ◇ 処理内容
 - 系統・停留所・運賃情報を元に、GunMaaS のオンラインチケットを利用した GunMaaS 会員の OD データを生成する。
- ◇ 利用するライブラリ
 - -
- ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 系統・停留所・運賃情報(ICRW 用)
 - データの内容
 - OD 生成要求
 - データの形式
 - 内部処理
 - 利用するデータインターフェース
 - 【IF008】 OD 作成処理要求
 - 出力
 - ◇ OD データ
 - データの内容
 - OD データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF009】 利用結果通知 API

【FN009】 チケット照会

- 本システム機能の概要
 - 専用端末への交通系 IC カードタッチによる GunMaaS のチケット認証機能を提供する。GunMaaS のチケットの登録がされた交通系 IC カードがタッチされた際は、そのチケットの情報に基づき割引情報の照会を行う機能。
- 本システム機能の入力・処理・出力のフローチャート

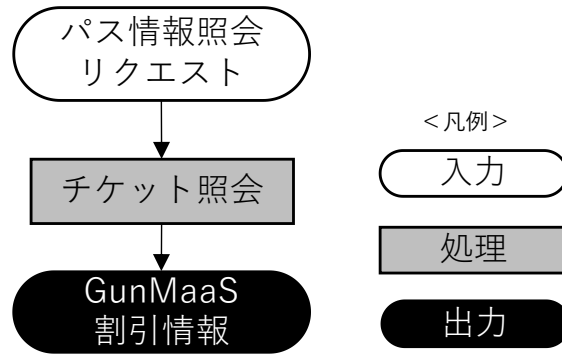


図 2-11 【FN009】 のフローチャート

- 本システム機能の処理の詳細
 - チケット照会
 - ◇ 処理内容
 - 交通系 IC カードが専用端末にタッチされた際に交通系 IC カードの IC カード番号の情報に紐づく GunMaaS チケットの情報を照会する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ パス情報照会リクエスト
 - データの内容
 - パス報を紹介するための API リクエスト
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF006】パス情報照会 API
 - 出力
 - ◇ GunMaaS 割引情報
 - データの内容
 - GunMaaS 割引情報
 - データの形式

- JSON
- 利用するデータインターフェース
 - 【IF006】パス情報照会 API

【FN010】チケット登録

- 本システム機能の概要
 - GunMaaS サイト上で購入されたチケットの登録依頼処理が行われた際に、そのチケットの登録処理を行う。交通系 IC カードでタッチ処理をした際にチケット情報として参照されるチケットのみ登録処理が行われる機能。
- 本システム機能の入力・処理・出力のフローチャート

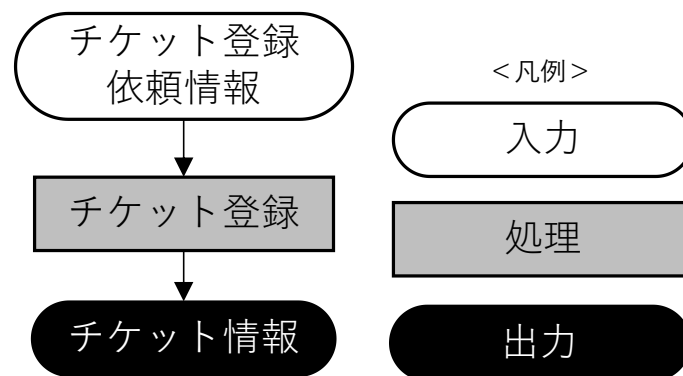


図 2-12 【FN010】のフローチャート

- 本システム機能の処理の詳細
 - チケット登録
 - ◇ 処理内容
 - チケット登録依頼情報に基づき、チケット情報についてチケットの登録処理を行う。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ チケット登録依頼情報
 - データの内容
 - ID-PORT へチケット登録の処理を依頼する情報
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF002】チケット登録 API
 - 出力

- ◇ チケット情報
 - データの内容
 - ID-PORT に録するチケット情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF002】チケット登録 API

【FN011】マイナンバーカード情報登録

- 本システム機能の概要
 - mytap アプリを通じて、マイナンバーカードの情報を取得し、その取得したマイナンバー情報を mytap 用に登録する機能。
- 本システム機能の入力・処理・出力のフローチャート

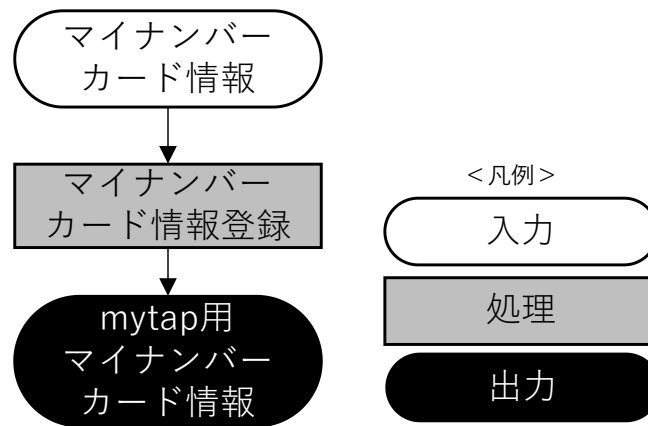


図 2-13 【FN011】のフローチャート

- 本システム機能の処理の詳細
 - マイナンバーカード情報登録
 - ◇ 処理内容
 - mytap アプリによりマイナンバーカード情報をマイナンバーカードから取得し、mytap 用マイナンバーカード情報として登録を行う。mytap 用に登録した後に ID-PORT に登録され GunMaaS で使用する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ マイナンバーカード情報

- データの内容
 - マイナンバーカードから取得できる情報
- データの形式
 - -
- 利用するデータインターフェース
 - -
- 出力
 - ◇ mytap 用マイナンバーカード情報
 - データの内容
 - TOPIC 社で登録するマイナンバーカード情報
 - データの形式
 - -
 - 利用するデータインターフェース
 - -

【FN012】 マイナンバーカード情報の GunMaaS 連携

- 本システム機能の概要
 - mytap アプリで mytap 上に登録されたマイナンバーカードの情報を ID-PORT に登録するためにデータ処理を行い、ID-PORT に登録を行う。ID-PORT に登録を行うことで GunMaaS に連携される機能。
- 本システム機能の入力・処理・出力のフローチャート

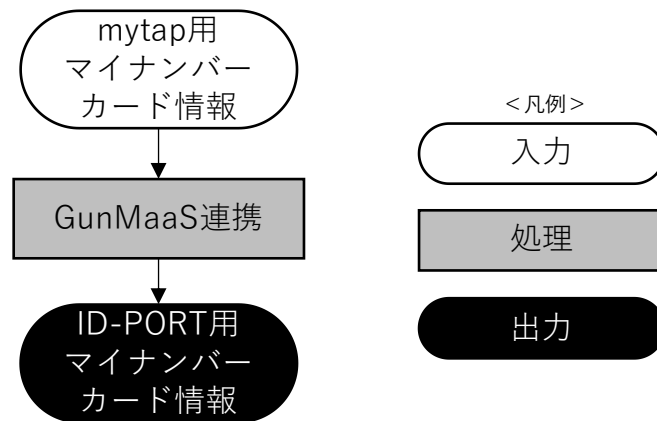


図 2-14 【FN012】 のフローチャート

- 本システム機能の処理の詳細
 - GunMaaS 連携
 - ◇ 処理内容
 - mytap 用マイナンバーカード情報を GunMaaS に連携する処理を行い、ID-PORT 用マイナンバーカード情報を出力する。

- ◇ 利用するライブラリ
 - -
- ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ mytap 用マイナンバーカード情報
 - データの内容
 - mytap 用マイナンバーカード情報
 - データの形式
 - -
 - 利用するデータインターフェース
 - -
 - 出力
 - ◇ ID-PORT 用マイナンバーカード情報
 - データの内容
 - ID-PORT に登録するためのマイナンバーカード情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF004】マイナンバーカード登録情報連携 API

【FN013】マイナンバーカードに基づいたチケット料金設定

- 本システム機能の概要
 - GunMaaS サイト上でのデジタルチケットの購入手続の際に、連携されているマイナンバーカードの情報を基に料金を更新する。チケット購入の際はこの更新されたチケット情報に基づく料金にて決済を行う機能。
- 本システム機能の入力・処理・出力のフローチャート

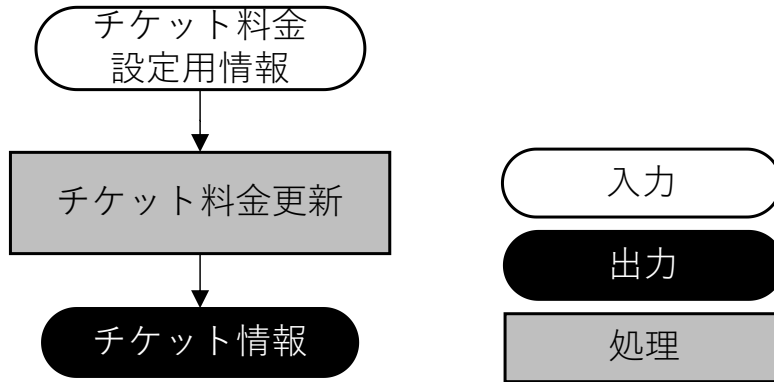


図 2-15 【FN013】 のフローチャート

- 本システム機能の処理の詳細
 - チケット料金更新
 - ◇ 処理内容
 - チケット情報にチケット料金更新処理を行う。料金更新が行われたチケット情報を出力する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ チケット料金設定用情報
 - データの内容
 - マイナンバーカードに基づいたチケット料金設定を行うために必要な情報
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF012】 チケット料金設定用情報取得 API
 - 出力
 - ◇ チケット情報
 - データの内容
 - 割引情報を含めたチケット情報
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF013】 チケット料金更新 API

【FN014】データ収集

- 本システム機能の概要
 - 各交通モードの利用データ・経路検索データ・ユーザの属性情報を収集する。収集するデータは GunMaaS のユーザのデータのみであり、GunMaaS のユーザでない利用者のデータについては収集を行わない機能。
- 本システム機能の入力・処理・出力のフローチャート

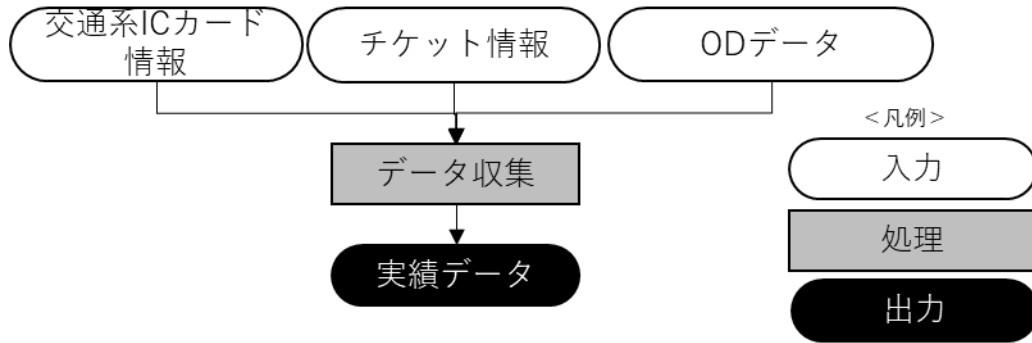


図 2-16 【FN014】のフローチャート

- 本システム機能の処理の詳細
 - データ収集
 - ◇ 処理内容
 - 交通系 IC カードの情報、GunMaaS において利用されたチケット情報、OD データを収集し、実績データとして出力する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 交通系 IC カード情報
 - データの内容
 - 交通系 IC カード情報
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF010】データ収集 API
 - ◇ チケット情報
 - データの内容
 - 割引情報を含むチケット情報
 - データの形式

- HTTP リクエスト
- 利用するデータインターフェース
 - 【IF010】データ収集 API
- ◇ OD データ
 - データの内容
 - OD データ
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF010】データ収集 API
- 出力
 - ◇ 実績データ
 - データの内容
 - ダッシュボード上で表示するための実績データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF010】データ収集 API

【FN015】利用者情報等の確認

- 本システム機能の概要
 - 各交通モードの利用データ、経路検索データ、ユーザの属性情報をダッシュボード上に表示する。表示されるデータは GunMaaS ユーザのデータのみであり、GunMaaS のユーザではない利用者のデータについては収集を行わないため含まれない機能。
- 本システム機能の入力・処理・出力のフローチャート

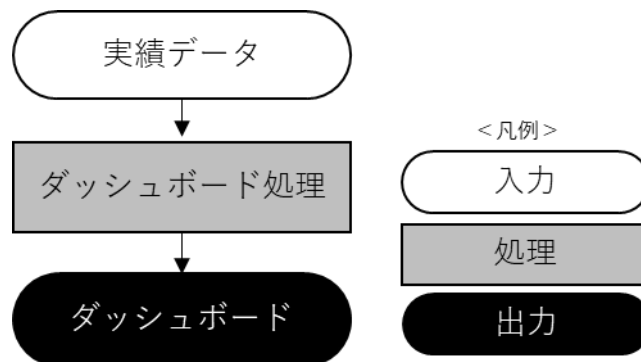


図 2-17 【FN015】のフローチャート

- 本システム機能の処理の詳細
 - ダッシュボード処理
 - ◇ 処理内容

- 実績データからダッシュボード処理を行い、Looker Studio のダッシュボードに出力する。
出力されるデータの形式は PDF 又は CSV 等とする。
- ◇ 利用するライブラリ
 - -
- ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 実績データ
 - データの内容
 - ダッシュボード上で表示するための実績データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF011】利用者情報等のダッシュボード連携 API
 - 出力
 - ◇ ダッシュボード
 - データの内容
 - Looker Studio にて各実績を出力
 - データの形式
 - PDF、CSV 等
 - 利用するデータインターフェース
 - -

2-1-1-d. 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

【マルチモーダルな移動体験の実現】

ID	名称	バージョン	内容
SL001	GunMaaS Web	-	● GunMaaS ユーザーとのインターフェースとなるフロントエンドシステム
SL002	チケット管理システム	-	● GunMaaS のチケットの販売・管理機能を担うシステムの機能群
SL003	ユーザ情報管理システム	-	● 交通系 IC カード情報やマイナンバーカード情報の登録・管理を担うシステムの機能群
SL004	運賃箱システム	-	● 上毛電気鉄道の車両に設置した IC 端末にて交通系 IC カードを認証するための運賃箱・地上機システム
SL005	地上機運賃箱システム	-	● 上毛電気鉄道の駅に設置した IC 端末にて交通系 IC カードを認証するための運賃箱・地上機システム
SL006	地域連携 IC カード対応バス車載 IC 機器ソフトウェア	-	● IC リーダライタで交通系 IC カードを処理するための共通ソフトウェア

ソフトウェアライブラリの詳細を記す。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】 GunMaaS Web

- ベンダー
 - 東日本旅客鉄道株式会社
- 公式サイト
 - <https://g3m.jp>
- 概要
 - GunMaaS のフロントエンド
- 主な機能
 - ログイン、チケットの販売・管理、交通系 IC カード情報の登録機能を担う GunMaaS のユーザとのインターフェース

【SL002】 チケット管理システム

- ベンダー
 - 株式会社ジェイアール東日本企画
- 公式サイト
 - <https://www.wallabee.net/>
- 概要
 - チケット管理システム
- 主な機能
 - GunMaaS のチケットの販売・管理機能を担う

【SL003】 ユーザ情報管理システム

- ベンダー
 - JR 東日本メカトロニクス株式会社
- 公式サイト
 - <https://www.jrem.co.jp/product/idport.html>
- 概要
 - ユーザ情報管理システム
- 主な機能
 - GunMaaS 会員の交通系 IC カード情報やマイナンバーカード情報の登録・管理機能を担う

【SL004】 運賃箱システム

- ベンダー
 - レシップ株式会社
- 公式サイト

➤ <https://www.lecip.co.jp/lecip/products/pdf/p04-sougou.pdf>

● 概要

➤ ICカードシステム対応型運賃箱ソフトウェア

● 主な機能

➤ ICリーダライタを通じて、交通系ICカードの認証・決済サービスに対応する。【HW001】運賃箱上で動作するソフトウェアである。

【SL005】地上機運賃箱システム

● ベンダー

➤ レシップ株式会社

● 公式サイト

➤ -

● 概要

➤ ICカードシステム対応型地上機ソフトウェア

● 主な機能

➤ 地上機のICリーダライタを通じて、交通系ICカードの認証・決済サービスに対応する。【HW002】地上機運賃箱上で動作するソフトウェアである。

【SL006】地域連携ICカード対応バス車載IC機器ソフトウェア

● ベンダー

➤ 東日本旅客鉄道株式会社

● 公式サイト

➤ -

● 概要

➤ 交通系IC認証決済ソフトウェア

● 主な機能

➤ 地域連携ICカード導入事業者にて、交通系ICカードの認証・決済サービスを提供する。【HW003】ICリーダライタ上で動作する機器一体型のソフトウェアである。

2-1-1-e. 数理モデル・アルゴリズム (AL) の詳細

-

2-1-2. システムコンポーネント (CO)

2-1-2-a. システムコンポーネント図

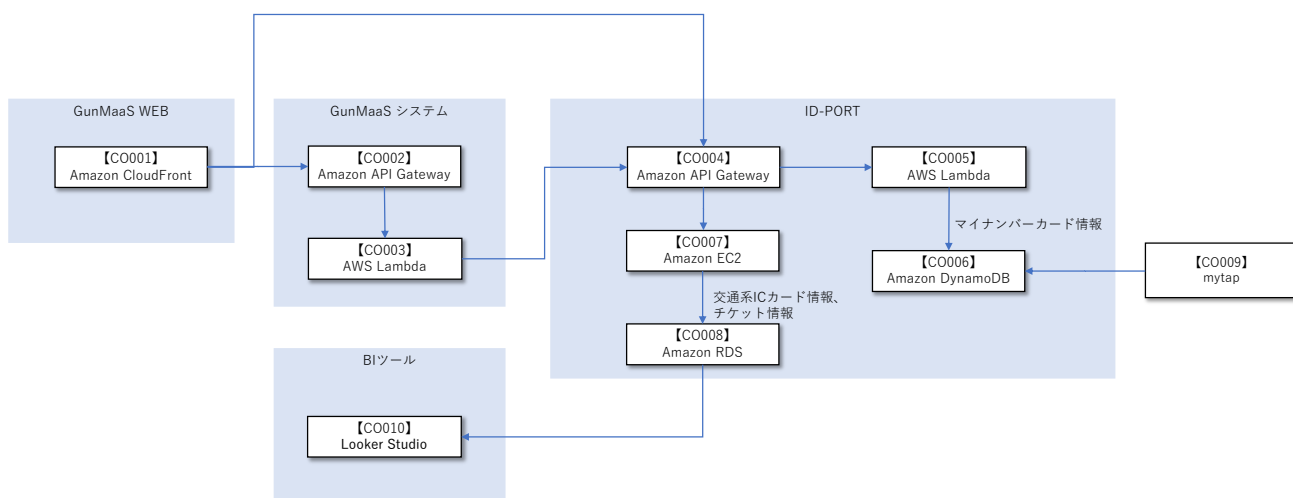


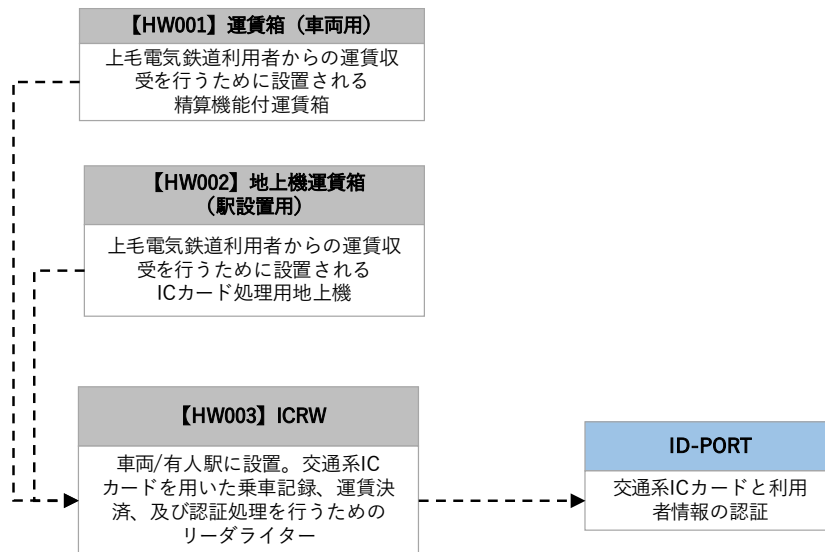
図 2-18 システムコンポーネント図

2-1-2-b. システムコンポーネント一覧

表 2-3 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	CDN	Amazon CloudFront	● Web ブラウザ上で動作する、(HTML・CSS・JavaScript による) フロントエンドアプリケーション。チケット販売や決済、ID-PORT の登録機能の呼出しなどを行う。
CO002	API サーバ	Amazon API Gateway	● チケット販売、チケット決済の動的コンテンツの配信を行う。
CO003	サーバレスコンピューティングサービス	AWS Lambda	● チケット販売、チケット決済の動的コンテンツの管理、実行を行う。
CO004	API サーバ	Amazon API Gateway	● 交通系 IC カード登録、マイナンバーカード連携、チケット登録の動的コンテンツの配信を行う。
CO005	サーバレスコンピューティングサービス	AWS Lambda	● マイナンバーカード情報の管理、実行を行う。
CO006	データベース	Amazon DynamoDB	● マイナンバーカード情報のデータを保存する。
CO007	仮想サーバサービス	Amazon EC2	● 交通系 IC カード、チケットの管理、実行を行う。
CO008	データベース	Amazon RDS	● 交通系 IC カード、チケット情報のデータの保存、管理を行う。
CO009	外部サービス	mytap(SaaS)	● マイナンバーカード情報を取得する。
CO010	外部サービス	Looker Studio(SaaS)	● さまざまなシステムで保有されている情報を一括集約し、外部連携可能とする。

2-1-3. ハードウェア (HW)
 2-1-3-a. ハードウェアアーキテクチャ



凡例	クラウド	PC	制御機器
	機能	機能	機能

図 2-19 ハードウェアアーキテクチャ

2-1-3-b. ハードウェア一覧

表 2-4 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	運賃箱 (車両用)	レシップ	LFZ-R-S0083	● 上毛電気鉄道利用者からの運賃収受を行うために設置される精算機能付運賃箱。本事業では、当該運賃箱に IC リーダライタを組込む。
HW002	地上機運賃箱 (駅設置用)	レシップ	NCR-42-80 ・ NCR-42-81	● 上毛電気鉄道利用者からの運賃収受を行うために設置される IC カード処理用地上機。本事業では、当該運賃箱に IC リーダライタを組み込む。
HW003	ICRW	レシップ	SUA-51 ・ SUB- 50	● 車両/有人駅に設置。交通系 IC カードを用いた乗車記録、運賃決済、及び認証処理を行うためのリーダライタ。

2-1-3-c. 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】 運賃箱：LFZ-R-S0083

- 概要
 - ICカードの普及に伴い軽量・小型金庫へリニューアルした即時計数機能付薄型運賃箱
- ベンダー
 - レシップ
- 仕様・スペック
 - CPU：i.MX RT、OS：FreeRTOS
 - 定格入力電圧：DC24V
 - 消費電力：最大 10A、通常 2A 以下
 - 質量：約 55 kg
 - 外形寸法：（横）560mm、（高さ）945mm、（幅）170mm
- イメージ図

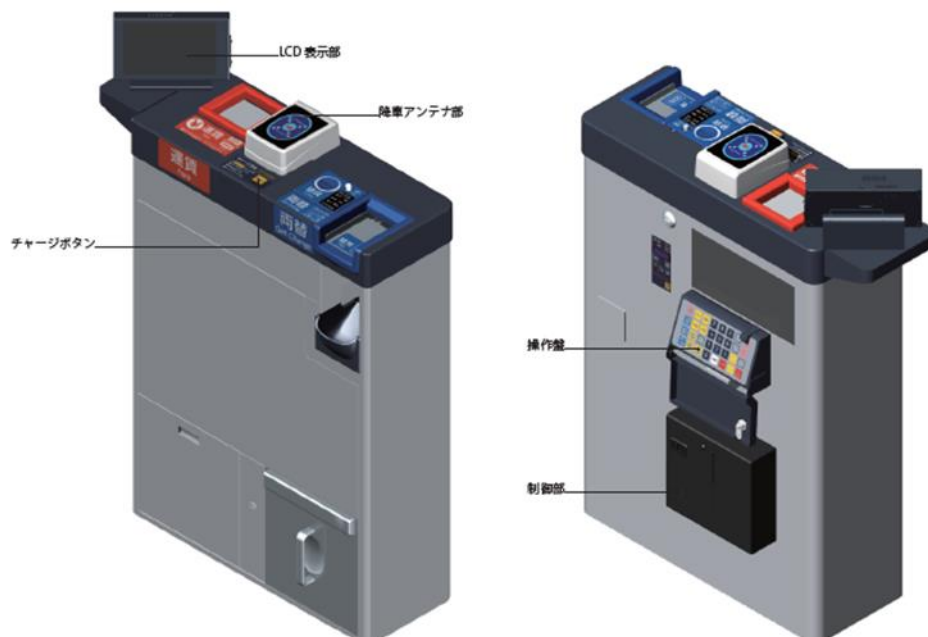


図 2-20 運賃箱：LFZ-R-S0083

【HW002】 地上機運賃箱：NCR-42-80（入場用）、NCR-42-81（出場用）

- 概要
 - ICカードの普及に伴い、駅側でICカード処理を行うための地上設置機器。上毛電気鉄道へ2025年に導入される予定。
- ベンダー
 - レシップ
- 仕様・スペック
 - CPU：SH7203、OS： μ ITRON
 - 定格入力電圧：AC100V
 - 消費電力：最大3A、通常1A以下
 - 質量：NCR-42-80（入場用）約10kg、NCR-42-81（出場用）約28kg
 - 外形寸法：NCR-42-80（入場用）：（横）213mm、（高さ）387mm、（幅）135mm
 - NCR-42-81（出場用）：（横）325mm、（高さ）650mm、（幅）323mm
- イメージ図

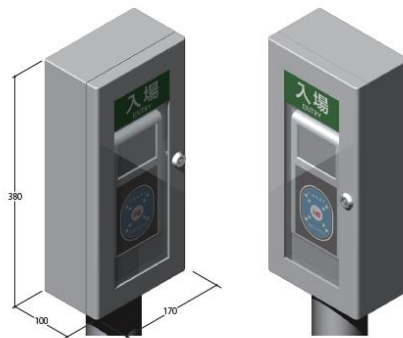


図 2-21 ICRW（有人駅設置用、NCR-42-80 入場用）

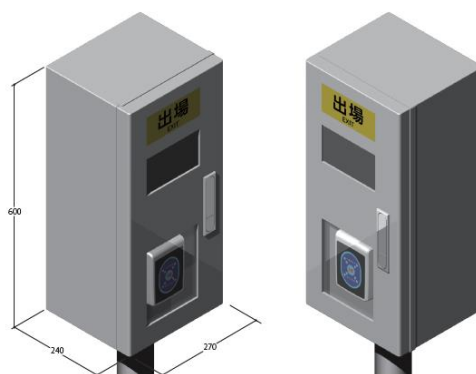


図 2-22 ICRW（有人駅設置用、NCR-42-81 出場用）

【HW003】ICRW (車両/有人駅設置用) : SUA-51 (乗車アンテナ)、SUB-50 (降車アンテナ)

- 概要
 - 交通系 IC カードを用いた乗車記録、運賃決済、及び認証処理を行うためのリーダー
- ベンダー
 - レシップ
- 仕様・スペック
 - CPU : SH7203、OS : μ ITRON
 - 定格入力電圧 : DC24V
 - 消費電力 : 最大 5A、通常 1A 以下
 - 質量 : SUA-51 (乗車アンテナ) 約 1.5 kg、SUB-50 (降車アンテナ) 約 0.7kg
 - 外形寸法 : SUA-51 (乗車アンテナ)、(横) 100mm、(高さ) 210mm、(幅) 62mm
 - SUB-50 (降車アンテナ)、(横) 100mm、(高さ) 135mm、(幅) 62mm
- イメージ



図 2-23 ICRW (車両/有人駅設置用 SUA-51 (乗車アンテナ))



図 2-24 ICRW (車両/有人駅設置用 SUB-50 (降車アンテナ))



図 2-25 Android

2-1-4. データインターフェース (IF)

2-1-4-a. データアーキテクチャ

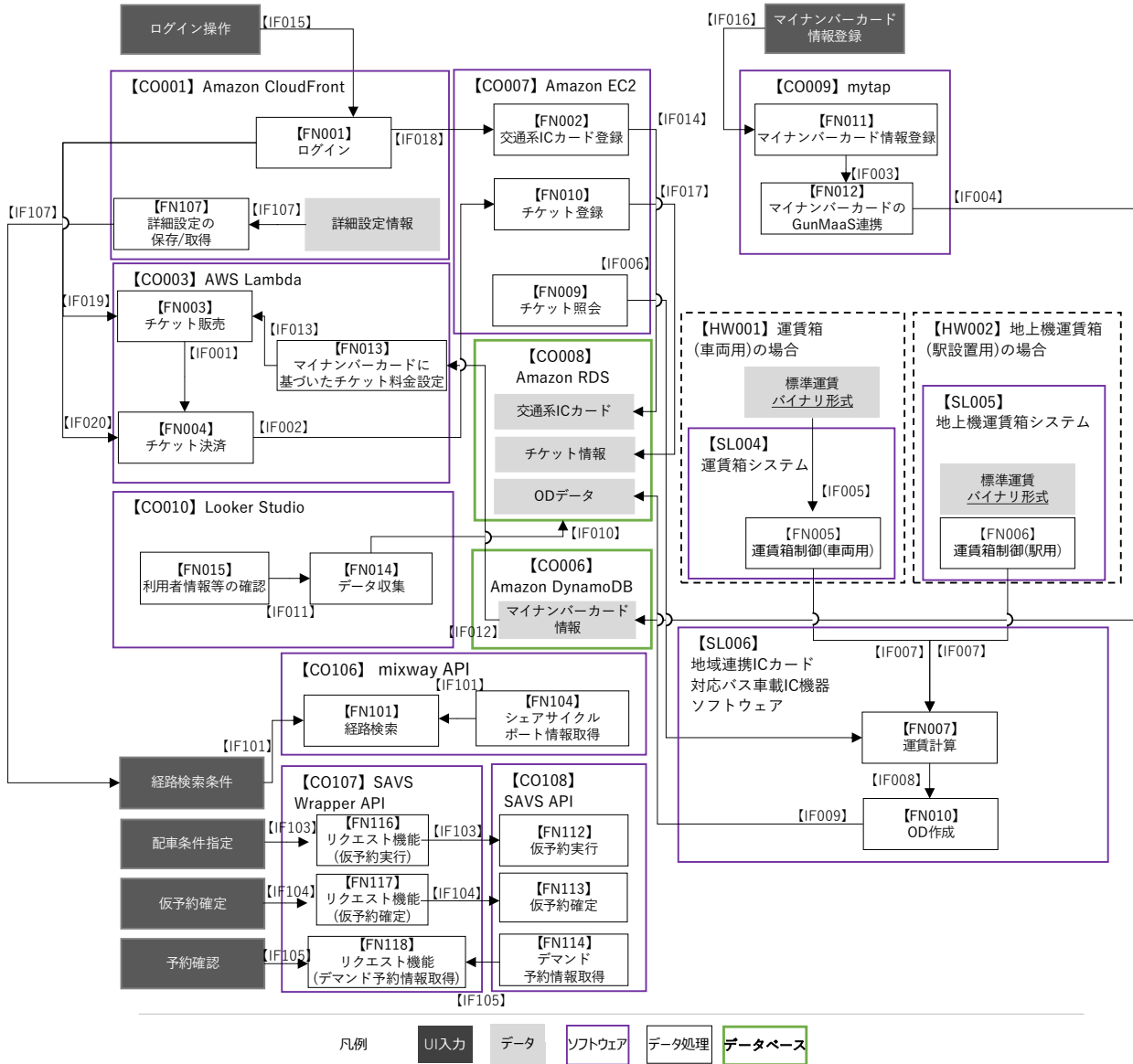


図 2-26 データアーキテクチャ

2-1-4-b. データインターフェース一覧

表 2-5 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	チケット販売 API	FN004	FN003
IF002	チケット登録 API	FN010	FN004
IF003	マイナンバーカード情報登録 API	FN012	FN011
IF004	マイナンバーカード登録情報連携 API	CO006	FN012
IF005	標準運賃取得	FN005	FN005
IF006	パス情報照会 API	FN007	FN007
IF007	操作部-車載 IC 機器間	FN007	FN005
			FN006
IF008	OD 作成処理要求	FN008	FN007
IF009	利用結果通知 API	FN008	FN008
IF010	データ収集 API	FN014	CO008
IF011	利用者情報等のダッシュボード連携 API	FN014	FN015
IF012	チケット料金設定用情報取得 API	FN013	CO006
IF013	チケット料金更新 API	FN003	FN013
IF014	交通系 IC カード情報登録要求 API	CO008	FN002
IF015	ログイン情報連携 API	FN001	FN001
IF016	マイナンバーカード情報取得 API	FN011	FN011
IF017	チケット情報登録要求 API	CO008	FN010
IF018	交通系 IC カード登録機能呼出 API	FN002	FN001
IF019	チケット販売機能呼出 API	FN003	FN001
IF020	チケット決済機能呼出 API	FN004	FN001

2-1-4-c. 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】 チケット販売 API

- 本インターフェースの概要
 - オンラインチケットを販売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
 - **【FN003】 チケット販売**
 - **【FN004】 チケット決済**
- 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
 - 共通リクエストパラメータ

項目	名称	説明	型	必須
-	-	-	-	-

- 共通レスポンスパラメータ

項目	名称	説明	値
時刻	timestamp	UNIX 時刻	整数

- レスポンスデータ
 - ◇ -。

- チケット発行
 - 概要
 - ◇ チケット発行 API
 - URI
 - ◇ <https://g3m.wallabeest.net/sp/g3m2/bridge/mlp/pay>
 - 使用例
 - ◇ <https://g3m.wallabeest.net/sp/g3m2/bridge/mlp/pay>
 - リクエストパラメータ

デマンド型交通リアルタイム経路検索システム システム設計書

項目	名称	説明	値	必須
決済 ID	mlpPaymentId	決済 ID	任意の文字列（上限 50 文字）	○
Mixway 予約 ID	mixway_reservation_id	mixway 予約を一意に識別する ID	任意の文字列（上限 50 文字）	-
パッケージ一覧	package_list	パッケージの一覧	配列	○
パッケージ ID	package_id	パッケージの ID	任意の文字列（上限 36 文字）	-
路線 ID	line_id	路線を識別する ID	任意の文字列（上限 50 文字）	-
出発 ID	place_id_from	出発 ID	任意の文字列（上限 36 文字）	-
到着 ID	place_id_to	到着 ID	任意の文字列（上限 36 文字）	-
価格の種類と数量の一覧	price_type_list	価格の種類と数量の一覧	配列	○
チケットタイプ識別 ID	type_id	チケットタイプを識別するための ID	任意の文字列（上限 36 文字）	-
チケット枚数	count	チケット枚数	整数	-

➤ レスポンスパラメータ

項目	名称	説明	値
時刻	timestamp	UNIX 時刻	整数
共通決済識別 ID	mlpPaymentId	共通決済を識別する ID	任意の文字列（上限 36 文字）
決済識別 ID	paymentId	決済を識別する ID	任意の文字列（上限 36 文字）
アカウント ID	accountId	ユーザアカウントを一意に識別する ID	整数
クレジットカード ID	creditCardId	クレジットカードを一意に識別する ID	整数
サービス名	serviceName	サービス名	任意の文字列（上限 7 文字）
商品カテゴリ	itemCategory	商品カテゴリ	任意の文字列（上限 3 文字）
支払金額	amount	支払金額 (>=1)	整数
通貨種類	currencyType	通貨種類 (JPY)	任意の文字列（上限 3 文字）

取引ステータス	status	取引ステータス	任意の文字列（上限 10 文字）
決済失敗理由	failureReason	決済が失敗した理由	任意の文字列（上限 36 文字）
支払日時	paidAt	支払日時 (date-time)	任意の文字列（上限 20 文字）
ロックバージョン	lockVersion	決済処理時に楽観的排他制御を行うためのレコードのロックバージョン	整数
支払詳細リスト	detailList	支払詳細リスト	配列
支払方法識別情報	methodCode	支払方法を識別するための情報	任意の文字列（上限 11 文字）

➤ レスポンスサンプル

```
{
  "timestamp": 1657769719,
  "mlpPaymentId": "A72E5296-9C96-X134-9GV8-D26355U03A9B",
  "paymentId": "payment1",
  "accountId": 0,
  "creditCardId": 1,
  "serviceName": "g3m",
  "itemCategory": "item category",
  "amount": 10000,
  "currencyType": "jpy",
  "status": "COMPLETED",
  "failureReason": "TIMEOUT",
  "paidAt": "2019-08-24T14:15:22Z",
  "lockVersion": 0,
  "detailList": [
    {
      "methodCode": "CREDIT",
      "amount": 10000
    }
  ]
}
```

【IF002】チケット登録 API

- 本インターフェースの概要
 - オンラインチケットを販売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
 - 【FN004】 チケット決済
 - 【FN010】 チケット登録
- 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
 - ◇ Shift_d
 - URI
 - ◇ -
 - 使用例

```
POST /-
HTTP/1.1
x-api-key : 1234567890123456789012345678901234567890
Host : host
Date : Wed, 5 Apr 2017 19 : 00 : 00 GMT
Content-Type : application/json
Content-Length :
{
  "ServiceId" : "123","AdminNo" : "12","TerminalId" : "A123456789012345","ReservationNumber" :
  "123456789012","TicketId" : "12345","PurchaseDate" : "12345678901234","UseStartDate" :
  12345678901234,"ConditionalFlag" : "1","Address" : "〇〇県","AgeType" : "1","LowerAge" :
  "1","UpperAge" : "1","PersonalidDuplication" : "1","ConditionalCardType" : "1","ConditionalCardId" : "
  1","CardType" : *1,"CardId" : "AB12345678901234"
}
```

➤ リクエストパラメータ

項目	名称	説明	値	必須
サービス ID	ServiceId	ID-PORT が提供するサービス毎に付与される一意な ID	0~999	○
行政 ID	AdminNo	ID-PORT から払い出される地方公共団体識別するための ID	0~255	○
端末 ID	TerminalId	ID-PORT から払い出される端末を一意に識別するための ID	1~24 文字の半角英数字	○
サービス付与 ID	GrantedId	興行主システム側で採番され	1~128 文字の半角英数字	※1

		る利用者を一意(サービス毎)に識別するための ID		
カード種別	CardType	カードの種別を指定するコード	0~999	※1
カード ID	CardId	利用者が使用するカードの ID	1~128 文字の半角英数字	※1
予約番号	ReservationNumber	チケット販売システム側で購入毎に採番される番号	1~32 文字の半角文字	-
バス用チケット ID	TicketId	バス用チケットを一意(サービス毎)に識別するための ID	1~99999	○
付与日時	PurchaseDate	当該チケットの付与日時	YYYYMMDDhhmmss の形式に合致する半角数字	○
利用開始日時	UseStartDate	当該チケットの利用開始日	YYYYMMDDhhmmss の形式に合致する半角数字	○
付与条件有無	ConditionalFlag	付与条件 (重複付与可否~カード ID 条件) の使用を判断するフラグ	0 : 使用しない 1 : 使用する	-
重複付与可否	DuplicateStatus	該当者に対する同一チケット ID の重複付与の可否	0 : 許可 1 : 不可	-
住所	Address	チケット付与条件となる住所	半角・全角文字 (文字数上限なし)	
年齢判定種別	AgeType	年齢 (下限)、年齢 (上限) で使用する年齢が、満年齢か年度末年齢か	1 : 満年齢 2 : 年度末年齢	※2
年齢 (下限)	LowerAge	チケット付与条件となる実年齢または年度末年齢の下限	0~999	※2
年齢 (上限)	UpperAge	チケット付与条件となる実年齢または年度末年齢の上限	0~999	※2
個人識別 ID 重複可否	PersonalIdDuplication	同一個人識別 ID に対する同一チケット ID の重複付与の可否	0 : 許可 1 : 不可	-
カード種別条件	ConditionalCardType	付与条件のカード種別	0~999	※3
カード ID 条件	ConditionalCardId	付与条件のカード ID	1~128 文字の半角英数字	※3

※1 カード種別、カード ID を指定しない場合、サービス付与 ID は必須。

サービス付与 ID を指定しない場合、カード種別、カード ID は必須。

サービス付与 ID およびカード種別、カード ID が両方指定された場合、サービス付与 ID が優先される。

※2 年齢 (下限) または年齢 (上限) を指定する場合、年齢判定種別は必須。

※3 カード ID 条件を指定する場合、カード種別条件は必須。

➤ レスポンスパラメータ

項目	名称	説明	値	必須
サービス付与 ID	GrantedId	リクエストに指定されたサービス付与 ID	1~128 文字の半角英数字	※1
バス用チケット ID	TicketId	リクエストに指定していない場合はカード情報に紐づくサービス付与 ID	1~99999	※1
予約番号	ReservationNumber	リクエストに指定されたバス用チケット ID	1~32 文字の半角文字	※1
カード使用状況	CardStatus	IC カードの使用状況	0 : 利用可 1 : 利用不可	※1
メッセージコード	MessageCode	エラー時のメッセージコード	11 文字の半角文字	※2
メッセージ	Message	エラー時のメッセージ	1~256 文字の半角・全角文字	※2

※1 正常終了時は必須、エラー発生時は不要

※2 エラー発生時は必須、正常終了時は不要

➤ レスポンスサンプル

```
{
  "GrantedId" : "1234567890","TicketId" : 12345,"ReservationNumber" : "12345","CardStatus" : 1
}
```

【IF003】 マイナンバーカード情報登録 API

- 本インターフェースの概要
 - mytap アプリを通じてマイナンバーカード情報を取得するインターフェース
- 本インターフェースを利用する機能
 - 【FN011】 マイナンバーカード情報登録
 - 【FN012】 マイナンバーカード情報の GunMaaS 連携
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -
 - レスポンスデータ形式
 - ◇ -
 - 文字コード
 - ◇ -
 - URI
 - ◇ -
 - リクエストパラメータ

- ◇ -
- レスポンスパラメータ
- ◇ -
- レスポンスサンプル
- ◇ -

【IF004】マイナンバーカード登録情報連携 API

- 本インターフェースの概要
 - ID-PORT にマイナンバーカード関連の情報を登録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
 - 【FN012】マイナンバーカード情報の GunMaaS 連携
- 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
 - URI
 - ◇ -
 - 使用例

```
POST /-
HTTP/1.1
Host : サーバ名
Accept : application/json
Date : Sun, 24 Apr 2011 06 : 12 : 18 GMT
x-api-key : 1234567890123456789012345678901234567890
Content-Length : 123

{
  "ServiceId" : 1234567890 ,
  "GrantedId" : 1234567890",
  "Municipalities" : 群馬県前橋市 “ ,
  "BirthMonth" : 12345678
}
```

- リクエストパラメータ

デマンド型交通リアルタイム経路検索システム システム設計書

項目	名称	説明	値	必須
サービス ID	Serviceld	サービスに割り当てられたサービス ID を指定する。	0~999	○
サービス付与 ID	GrantedId	住基情報と紐づけされたサービス付与 ID を指定する。	1~128 文字の半角英数字	○
証明書発行年月日	CertificateIssue	証明書の発行年月日を右の形式で指定する。	YYYYMMDD の形式に合致する 8 桁の数字	-
都道府県名	Prefectures	「都道府県」名を文字列で指定する。	1 文字以上の文字列 (上限なし)	-
市区町村名	Municipalities	「都道府県 + 市区町村」名を文字列で指定する。	1 文字以上の文字列 (上限なし)	-
性別	Sex	性別を以下の文字列で指定する	“Woman” “Man”	-
生年月	BirthMonth	生年月を右の形式で指定する。	YYYYMM の形式に合致する 6 桁の数字	-
生年月日	Birthday	生年月日を右の形式で指定する。	YYYYMMDD の形式に合致する 8 桁の数字	-
氏名	Name	氏名を文字列で指定する。	1 文字以上の文字列 (上限なし)	

➤ レスポンスパラメータ

項目	名称	説明	値
サービス ID	Serviceld	サービス毎に付与される一意な ID	0~999
サービス付与 ID	GrantedId	会員 ID	1~128 文字の半角英数字

➤ レスポンスサンプル

<p>HTTP 200 OK</p> <p>Date : Sun, 24 Apr 2011 06 : 12 : 18 GMT</p> <p>Content-Type : application/json</p> <p>Content-Length : 123</p> <pre>{ "Serviceld" : 1234567890 , "GrantedId" : "1234567890" }</pre>
--

【IF005】標準運賃取得

- 本インターフェースの概要
 - 使用する系統の標準運賃を運賃箱が取得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
 - 【FN005】運賃箱制御(車両用)
- 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
 - 内容
 - ◇ 運賃テーブルにより定められた運賃
 - 形式
 - ◇ バイナリ形
 - データ詳細
 - ◇ 入力

項目	名称	説明	値	必須
系統番号	UnchinData.m_keito	系統番号（上り/下り）	0～99999999	○
運賃箱制御用	UnchinData.m_tei_tbl[koma].m_unko_mode	運賃箱制御用	0：多区運用 1：均一運用	○
駅連番、駅コード、エリアコード	UnchinData.m_tei_tbl[koma]	駅連番、駅コード、エリアコード等	0～63：駅連番 0～FFFF：駅コード 0～FFFF：エリアコード	○
標準運賃	UnchinData.m_tei_tbl[koma].m_un_tbl	標準運賃	(1～65535)×255 パターン：可変 所定運賃（IC 券種ごと） (最大 4081byte) ※サイズは当該の降車駅に対する乗車可能な駅数によって可変、1 駅～255 駅（最大）の範囲で受信可能	○

【IF006】パス情報照会 API

- 本インターフェースの概要
 - ID-PORT から GunMaaS 割引情報を URI 形式で取り込む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
 - 【FN007】運賃計算
- 本 API の共通仕様

デマンド型交通リアルタイム経路検索システム システム設計書

- 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
- URI
 - ◇ -
- 使用例
 - ◇ - /referencebusapi
 ?UserCD=" 0ABC"&SalesOfficeCD=" 1"&TerminalID=" 0ABC1234"&VehicleNo=1234
 &API_Num=2&TransactionID=12345&CardIDi=" 0FFF999FC79FFFFFF "
 &FacilityID=" 12345"&AreaCode4=" FFFD"&Fare_Adult=12345&LateNight=1
 &Board_BusStopID=" 0FFF1234"&BusStopID=" 0FFF1234"
- リクエストパラメータ

項目	名称	説明	値	必須
事業者 CD	UserCD	事業者を識別するコード情報	0FFF	○
営業所 CD	SalesOfficeCD	営業所を識別するコード情報	1～FF	○
車載 IC 機器 ID	TerminalID	車載 IC 機器を識別する ID	0FFF(UserCD)+0000 ～ 9999	○
車両番号	VehicleNo	車両番号	0000～9999	○
API 区分	API_Num	API 区分	2(固定、バス情報照会 API)	○
トランザクション ID	TransactionID	トランザクション ID	1～65535	○
カード IDi	CardIDi	交通系 IC カードを一意に識別する ID	0FFF999FC79FFFFFF	○
系統番号	FacilityID	系統番号	0～99999999	○
エリアコード	AreaCode4	エリアコード	FFFD(固定、停留所判定を示す)	○
基本運賃	Fare_Adult	基本運賃	0～65535	○
深夜区分	LateNight	深夜区分	0：昼間 1：深夜	○
乗車停留所 ID	Board_BusStopID	乗車停留所 ID	0FFF(UserCD)+0000 ～ FFFF	○
降車停留所 ID	BusStopID	降車停留所 ID	0FFF(UserCD)+0000 ～ FFFF	○

➤ レスポンスパラメータ

項目	名称	説明	値	必須
認証 NG 理由	Reason	認証が NG となった理由	0：認証 OK 1：パス無効化	○
カード IDi	CardIDi	交通系 IC カードを一意に識別する ID	0FFF999FC79FFFFFF (内部 IDi16 桁)	○
行政 ID	AdminNo	行政 ID	0～255	○
商品番号	MerchanNo	商品番号	0～65535	○
購入したパス終了日	PassExp	購入したパス終了日	20991231(YYYYMMDD)	○
トランザクション ID	TransactionID	トランザクション ID	1～65535	○
パス種類	PassType	パスの種類を識別する情報	0：無効化 (IDi 認証パスの無効化) 1：フリーパス 2：都度払い	○
回数制限機能有無	TicketFunction	回数制限機能の有無情報	0：機能なし 1：機能あり	○
利用残回数	RemainingUsage	利用の残回数情報	0～250	○
無効化判定時動作	InvalidAct	無効化の判定時動作情報	0：既存判定 1：エラー	○
IDi 認証パス名称	PassName	IDi を用いた認証パスの名称	Shift-JIS 全角 10 文字 (例)フリーパス： 8374838a815b8370：8358 00000000000000000000	○
引去区分	Category	引去区分	0：区分なし (無効化、フリーパス) 1：ワンコイン 2：割引額指定 3：割引率指定	○
深夜利用	MidNight	深夜利用判定情報	0：地域連携設定に準拠 1：IDi 認証パス独自 (深夜割増分を SF 引去りしない)	○
ワンコイン or 割引額指定	Amount	ワンコイン or 割引額指定	0～20000	○
割引率	Discount	割引率	0～100	○
残回数不足時の動作	RemainingAct	残回数不足時の動作	0：既存判定 (SF 精算) 1：エラー	○

時間制限機能有無	TicketFunction2	時間制限機能有無	0：機能なし 1：機能あり	○
購入したパス終了日時分	PassExp2	購入したパス終了日時分	209912312359 (YYYYMMDDhhmm)	○

➤ レスポンスサンプル

```

HTTP/1.1 200 OK
Content-Type : application/json
Date : 209912312359
... (略) ...
transfer-encoding : chunked
Connection : keep-alive

17a
{
  "Reason" : 1,
  "CardIDi" : " 0FFF999FC79FFFFFF ",
  "AdminNo" : 1,
  "MerchanNo" : 12345,
  "PassExp" : 1,
  "TransactionID" : 12345,
  "PassType" : 1,
  "TicketFunction" : 1,
  "RemainingUsage" : 123,
  "InvalidAct" : 1,
  "PassName" : " 123400000000000000000000",
  "Category" : 1,
  "MidNight" : 1,
  "Amount" : 12345,
  "Discount" : 1,
  "RemainingAct" : 1,
  "TicketFunction2" : 1,
  "PassExp2" : 123456789012,
}
    
```


【IF007】 操作部-車載 IC 機器間

- 本インターフェースの概要
 - 運賃箱から標準運賃をバイナリ形式で取り込む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
 - 【FN005】 運賃箱制御(車両用)
 - 【FN006】 運賃箱制御(駅用)
 - 【FN007】 運賃計算
- 本 API の共通仕様
 - プロトコル
 - ◇ RS-485
 - レスポンスデータ形式
 - ◇ 8bit バイナリ
 - 文字コード
 - ◇ UTF-8
 - リクエストパラメータ

項目	名称	説明	値	必須
系統番号	UnchinData.m_keito	系統番号 (上り/下り)	0~99999999	○
駅連番、駅コード、エリアコード等	UnchinData.m_tei_tbl[koma]	駅連番、駅コード、エリアコード等	34byte 0~63 : 駅連番 0~FFFF : 駅コード 0~FFFF : エリアコード	○
標準運賃	UnchinData.m_tei_tbl[koma].m_un_tbl	標準運賃 (大人、大割、小児、小割)	(1~65535) × 255 パターン : 可変 所定運賃 (IC 券種ごと) (最大 4081byte) ※サイズは当該の降車駅に対する乗車可能な駅数によって可変、1 駅~255 駅 (最大) の範囲で受信可能	○

➢ レスポンスパラメータ

項目	名称	説明	値	必須
機器状態	HostComData	機器状態	55byte Bit パターンにより機器状態を示す オフライン、オンライン等	○

受信系統番号	UnchinData.m_keito	受信系統番号	0~99999999	○
受信整理券番号	UnchinData.m_se	受信整理券番号	0~63	○
受信駅連番	UnchinData.m_koma	受信駅連番	0~63	○

【IF008】 D 作成処理要求

- 本インターフェース概要
 - 運賃計算で確定した処理内容を OD 作成処理で OD データとして生成するための内部処理
- 本インターフェースを利用する機能
 - 【FN007】 運賃計算
 - 【FN008】 OD 作成
- 本インターフェースを利用してやり取りを行うデータ詳細
 - 内容
 - ◇ OD 作成処理を IC リーダライターソフトウェアで実施する
 - 形式
 - ◇ IC リーダライターソフトウェアによる内部処理

【IF009】 利用結果通知 API

- 本インターフェースの概要
 - ID-PORT へ OD データを URI 形式で送信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
 - 【FN008】 OD 作成
- 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
 - URI
 - ◇ -
 - 使用例
 - ◇ -/notificationbusapi
 ?UserCD="0ABC"&SalesOfficeCD=" 1"&TerminalID=" 0ABC1234"&VehicleNo=1234
 &API_Num=3&TransactionID=12345&CardIDi=" 0FFF999FC79FFFFFF "
 &Cal_Num=1&UnFinished=1&ShortBalance=1&FacilityID=" 1234567"

&FacilityID_time=1&LateNight=1&BusStopID=" 0ABC12345"&AuthResult=1
 &CardID_Ver=123456789012&CardID_Diff_Ver=123456789012
 &Product_Ver=123456789012&Facility_Ver=123456789012
 &SeasonTicket_Ver=123456789012&Persons1=12&Persons2=12&Persons3=12
 &Persons4=12&AdminNo=12&MerchanNo=12&Fare_Adult=12&SFpresence=12
 &Outside_SFpresence=12345&Excess_SFpresence=12345&
 CashlessFlag=1&ChargeAmount=12345&Board_BusStopID=0ABC1234"

➤ リクエストパラメータ

項目	名称	説明	値	必須
事業者 CD	UserCD	事業者を識別するコード 情報	0FFF	○
営業所 CD	SalesOfficeCD	営業所を識別するコード 情報	1～FF	○
車載 IC 機器 ID	TerminalID	車載 IC 機器を識別する ID	0FFF(UserCD)+0000 ~ 9999	○
車両番号	VehicleNo	車両番号	0000～9999	○
API 区分	API_Num	利用実績データ区分(API 区分)	3(固定、利用結果通知 API)	○
トランザク ション ID	TransactionID	トランザクション ID	1～65535	○
カード IDi	CardIDi	交通系 IC カードを一意に 識別する ID	0FFF999FC79FFFFFF (内部 IDi16 桁)	○
利用日時	UseDateandtime	利用日時	209912312359 (YYYYMMDDhhmm)	○
精算区分	Cal_Num	精算区分	0：通常精算 1：区間設定 2：金額設定 3：全額引去 4：チャージ 5：一日乗車券購入・数日乗車 券購入 6：キャッシュレスチャージ・ オートチャージ 7：商品指定精算 8：停留所単位判定救済	○
書込未了フ ラグ	UnFinished	書込未了のフラグ情報	0：確定 1：書込未了発生時	○
残額不足フ	ShortBalance	残額不足のフラグ情報	0：残額不足なし	○

デマンド型交通リアルタイム経路検索システム システム設計書

ラグ			1：残額不足あり	
系統番号	FacilityID	系統番号の情報	1～99999999	○
系統設定時刻	FacilityID_time	系統設定時刻の情報	0000～2359(hhmm)	○
深夜区分	LateNight	深夜区分	0：昼間 1：深夜	○
降車停留所ID	BusStopID	降車停留所 ID	0FFF(UserCD)+ 0000 ～ FFFF	○
認証結果	AuthResult	認証結果	1：オンライン認証 OK 2：オフライン認証 OK 3：認証 NG	○
(一括)カード ID データバージョン	CardID_Ver	(一括)カード ID データバージョン	209912312359 (YYYYMMDDhhmm)	○
(差分)カード ID データバージョン	CardID_Diff_Ver	(差分)カード ID データバージョン	209912312359 (YYYYMMDDhhmm)	○
商品リストデータバージョン	Product_Ver	商品リストデータバージョン	209912312359 (YYYYMMDDhhmm)	○
系統番号リストデータバージョン	Facility_Ver	系統番号リストデータバージョン	209912312359 (YYYYMMDDhhmm)	○
バス定期券情報リストデータバージョン	SeasonTicket_Ver	バス定期券情報リストデータバージョン	209912312359 (YYYYMMDDhhmm)	○
利用人数 (大人)	Persons1	利用人数 (大人)	0～99	○
利用人数 (大人割引)	Persons2	利用人数 (大人割引)	0～99	○
利用人数 (小児)	Persons3	利用人数 (小児)	0～99	○
利用人数 (小児割引)	Persons4	利用人数 (小児割引)	0～99	○
行政 ID	AdminNo	行政 ID	0～255	○
商品番号	MerchanNo	商品番号	0～65535	○
基本運賃	Fare_Adult	基本運賃	0～65535	○
SF 引去額	SFpresence	引去額	0～20000	○

飛付 SF 引去額	Outside_SFpresence	飛付引去額	0~20000	○
乗越 SF 引去額	Excess_SFpresence	乗越引去額	0~20000	○
キャッシュレスチャージフラグ	CashlessFlag	オートチャージフラグ	0: オートチャージなし 1: オートチャージあり	○
チャージ額	ChargeAmount	チャージ額	0~20000	○
乗車停留所 ID	Board_BusStopID	乗車停留所 ID	0FFF(UserCD)+0000 ~ FFFF	○

➤ レスポンスパラメータ

項目	名称	説明	値	必須
カード IDi	CardIDi	交通系 IC カードを一意に識別する ID	0FFF999FC79FFFFFF	○
トランザクション ID	TransactionID	トランザクション ID	1~65535	○

➤ レスポンスサンプル

```

HTTP/1.1 200 OK
Content-Type : application/json
Date : 209912312359
... (略) ...
Content-Length : 123
Connection : keep-alive

{
  "CardIDi" : "0FFF999FC79FFFFFF ",
  "TransactionID" : 12345,
}
    
```

【IF010】 データ収集 API

- 概要
 - 各環境の RDS や API からデータを収集・統合し、分析や連携の中核となるデータハブ機能を提供する。
- 本インターフェースを利用する機能
 - 【FN014】 データ収集
- 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
- 共通リクエストパラメータ

項目	名称	説明	型	必須
-	-	-	-	-

- 共通レスポンスパラメータ

項目	名称	説明	値
データ件数	total_count	データ件数	整数
ページ情報	page_info	ページ情報 ・ 現在ページ番号 ・ 1 ページ件数 ・ 表示開始位置 (件数) ・ 表示終了位置 (件数) ・ 前ページ有無 ・ 次ページ有無	配列 (オブジェクト)
結果	results	実データ	配列 (オブジェクト)

- レスポンスサンプル

```

{
  "total_count": 138,
  "page_info": {
    "page_no": 1,
    "count": 4,
    "from": 1,
    "to": 4,
    "prev": false,
    "next": true
  },
  "results": [
    {
      "granted_id": "1234567890",
      "issue_at": "2025-03-28T16:19:56",
      "type": "demands_gunma001",
      "issued_id": "378",
    }
  ]
}
    
```

```
"ticket_id" : null,
"item_id" : null,
"package_name" : null,
"discount_type" : "regular",
"payment_type" : null,
"amount" : 0,
"status" : "FINISHED",
"count_01" : 0,
"count_02" : 0,
"user_cd" : null,
"user_name" : null,
"sales_office_cd" : null,
"sales_office_name" : null,
"terminal_id" : null,
"vehicle_no" : null,
"facility_no" : null,
"facility_name" : "",
"pick_up_position_id" : null,
"pick_up_position" : "12.345 : 123.45",
"pick_up_location_name" : "前橋市役所",
"pick_up_time" : "2025-03-28T07 : 03 : 00",
"drop_off_position_id" : null,
"drop_off_position" : "123.45 : 12.345",
"drop_off_location_name" : "群馬県庁",
"drop_off_time" : "2025-03-28T07 : 19 : 56",
"unique_id" : "ABC",
"user_type" : 1,
"user_pref" : "10",
"user_sex" : "2",
"user_birthyear" : "1991",
"user_info" : [
  {
    "card_type" : 1,
    "card_id" : " AB12345678901234",
    "masking_id" : " AB12345678901234",
    "available_flag" : true,
    "suica_last_registration_time" : "2024-03-02T18 : 58 : 18",
    "myna_last_registration_time" : "2024-02-29T21 : 14 : 43",
```

```

        "platform": "JaMid"
    }
}
    ]
}
}

```

● データ取得 API

- 概要
 - ◇ 入力を元にダッシュボード上で表示が必要なデータベースの情報を返却する。
- URI
 - ◇ <https://datamart-api-32uddcv9.an.gateway.dev/api/v1/datamart>
- 使用例
 - ◇ <https://datamart-api-32uddcv9.an.gateway.dev/api/v1/datamart>
- リクエストパラメータ

項目	名称	説明	値	必須
ページ番号	page	ページ番号	整数	○
データ件数	per_page	データ件数	整数	○
ソート対象	sort_field	ソート対象	配列	-
ソートタイプ	sort_type	ソートタイプ	配列	-
サービス付与 ID	granted_id	媒体 ID	配列	-
発行日時_FROM	issue_at_from	発行日時_FROM	YYYYMMDDhhmmss の形式に合致する半角数字	○
発行日時_TO	issue_at_to	発行日時_TO	YYYYMMDDhhmmss の形式に合致する半角数字	○
種別	type	種別	配列	-
ユーザ種別	user_type	ユーザ種別	数値	-
カードの種別	card_type	カードの種別	0~999	-

➤ レスポンスパラメータ

項目	名称	説明	値
サービス付与 ID	granted_id	媒体 ID	"0"~"9" の文字列
イベント日時	issue_at	イベント日時	YYYYMMDDhhmmss の形式に合致する半角数字
種別	type	種別	<ul style="list-style-type: none"> • bus_ride : バス乗降データ • demands_gunma001 : デマンド群馬 • demands_maebashi001 : デマンド前橋 • demands_shibukawa001 : デマンド渋川 • demands_shibukawa002 : デマンド渋川 • first_login : wallabee 会員登録 • purchase : wallabee チケット情報

デマンド型交通リアルタイム経路検索システム システム設計書

			・ taxi_ride : バス乗降データ (将来拡張あり)
イベント ID	issued_id	イベント ID	文字列
チケット ID	ticket_id	チケット ID	文字列
アイテム ID	item_id	アイテム ID	文字列
チケット名	package_name	チケット名	文字列
割引フラグ	discount_type	割引フラグ	文字列
決済方法	payment_type	決済方法	文字列
金額	amount	金額	整数
ステータス	status	ステータス	文字列
大人数量	count_01	大人数量	整数
小人数量	count_02	小人数量	整数
事業者コード	user_cd	事業者コード※ 1	文字列
事業者名	user_name	事業者名	文字列
営業所コード	sales_office_cd	営業所コード※ 2	整数
営業所名	sales_office_name	営業所名	文字列
事業者コード+車両番号	terminal_id	事業者コード+ 車両番号	文字列
車両番号	vehicle_no	車両番号	整数
系統番号	facility_no	系統番号	整数
系統名	facility_name	系統名	文字列
乗車位置 ID	pick_up_position_id	乗車位置 ID	文字列
乗車位置	pick_up_position	乗車位置	文字列
乗車位置名称	pick_up_location_name	乗車位置名称	文字列
乗車時間	pick_up_time	乗車時間	YYYYMMDDhhmmss の形式に合致する半角数字
降車位置 ID	drop_off_position_id	降車位置 ID	文字列
降車位置	drop_off_position	降車位置	文字列
降車位置名称	drop_off_location_name	降車位置名称	文字列

降車時間	drop_off_time	降車時間	YYYYMMDDhhmmss の形式に合致する半角数字
ユニーク ID	unique_id	ユニーク ID	文字列
ユーザ種別	user_type	ユーザ種別	整数
居住地	user_pref	居住地	文字列
性別	user_sex	性別	文字列
生年月日	user_birthday	生年月日	文字列
ユーザ情報	user_info	ユーザ情報	配列、下記項目を配列で保持
カード種別	card_type	カード種別	整数
カード識別子	card_id	カード識別子	文字
カード識別子	masking_id	カード識別子 (秘匿化)	文字
利用可否フラグ	available_flag	利用可否フラグ	論理値
Suica 連携日時	suica_last_registration_time	Suica 連携日時	YYYYMMDDhhmmss の形式に合致する半角数字
マイナンバーカード連携日時	myna_last_registration_time	マイナンバーカード連携日時	YYYYMMDDhhmmss の形式に合致する半角数字
プラットフォーム	platform	プラットフォーム	文字列

※1 事業者コードはセキュリティ担保のため非公開。

※2 コードリスト「営業所コード」を参照。

【IF011】利用者情報等のダッシュボード連携 API

- 本インターフェースの概要
 - 利用者情報等を確認するダッシュボードを提供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
 - 【FN014】データ収集
 - 【FN015】利用者情報等の確認
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -
 - レスポンスデータ形式
 - ◇ -
 - 文字コード
 - ◇ -
 - URI
 - ◇ -

- リクエストパラメータ
 - ◇ -
- レスポンスパラメータ
 - ◇ -
- レスポンスサンプル
 - ◇ -

【IF012】 チケット料金設定用情報取得 API

- 本インターフェースの概要
 - チケットの料金設定時に使用するユーザのマイナンバーカード情報を取得するインターフェース
- 本インターフェースを利用する機能
 - 【FN013】 マイナンバーカードに基づいたチケット料金設定
- 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
 - 共通リクエストパラメータ

項目	名称	説明	型	必須
サービス ID	ServiceId	サービスに割り当てられたサービス ID を指定する。	0~999	○
サービス付与 ID	GrantedId	住基情報と紐づけされたサービス付与 ID を指定する。	1 ~ 128 文字の 半角英 数字	○

- 共通レスポンスパラメータ

項目	名称	説明	値
時刻	timestamp	UNIX 時刻	整数

- レスポンスデータ
 - ◇ -

- マイナンバーカード情報年齢の取得・確認 (ID-PORT)
 - 概要
 - ◇ マイナンバーカード情報年齢の取得・確認
 - URI

◇ /bridge/idport/myn-v1/myncard/age

➤ 使用例

◇ /bridge/idport/myn-v1/myncard/age?ServiceId=102&GrantedId=100234

➤ リクエストパラメータ

項目	名称	説明	値	必須
サービス ID	ServiceId	サービスに割り当てられたサービス ID を指定する。	0~999	○
サービス付与 ID	GrantedId	住基情報と紐付されたサービス付与 ID を指定する。	1 ~ 128 文字の半角英数字	○
比較年齢	More	比較年齢	0~999	-
比較年齢	Less	比較年齢	0~999	-

➤ レスポンスパラメータ

項目	名称	説明	値
サービス ID	ServiceId	サービス ID	0~999
サービス付与 ID	GrantedId	サービス付与 ID	1~128 文字の半角英数字
満年齢	Age	満年齢	整数
年度末年齢	FiscalYearEndAge	年度末年齢	整数
年齢の確認結果	Confirm	年齢の確認結果	任意(Boolean)

➤ レスポンスサンプル

```
{
  "timestamp": 1657769719,
  "ServiceId": 102,
  "GrantedId": "1234567890",
  "Age": 21,
  "FiscalYearEndAge": 21,
  "Confirm": true
}
```

● マイナンバーカード情報 住居地の確認 (ID-PORT)

➤ 概要

◇ マイナンバーカード情報 住居地の確認

➤ URI

◇ /bridge/idport/myn-v1/myncard/residence

➤ 使用例

◇ /bridge/idport/myn-v1/myncard/residence?ServiceId=102&GrantedId=100234

➤ リクエストパラメータ

項目	名称	説明	値	必須
----	----	----	---	----

デマンド型交通リアルタイム経路検索システム システム設計書

サービス ID	Serviceld	サービスに割り当てられたサービス ID を指定する。	0~999	○
サービス付与 ID	Grantedld	住基情報と紐づけされたサービス付与 ID を指定する。	1~128 文字の半角英数字	○
比較年齢	Municipalities	比較年齢	0~999	-

➤ レスポンスパラメータ

項目	名称	説明	値
サービス ID	Serviceld	サービス ID	0~999
サービス付与 ID	Grantedld	サービス付与 ID	1~128 文字の半角英数字
住居地の確認結果	Resident	住居地の確認結果	必須 (Boolean)

➤ レスポンスサンプル

```
{
  "timestamp": 1657769719,
  "Serviceld": 102,
  "Grantedld": "1234567890",
  "Resident": true
}
```

【IF013】チケット料金更新 API

- 本インターフェースの概要
 - デジタルチケットの購入手続の際に、連携されたマイナンバーカード情報を元に割引料金に更新するインターフェース
- 本インターフェースを利用する機能
 - 【FN013】マイナンバーカードに基づいたチケット料金設定
- 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
 - 共通リクエストパラメータ

項目	名称	説明	型	必須
-	-	-	-	-

- 共通レスポンスパラメータ

項目	名称	説明	値
時刻	timestamp	UNIX 時刻	整数

- レスポンスサンプル

```

{
  "timestamp": 1657769719,
  "package_list": [
    {
      "package_id": "PACKAGEID-1001",
      "package_code": "PA2203001",
      "package_name": "パッケージ名その1",
      "category_id": "fp",
      "area_list": [
        {
          "area_id": "102016"
        }
      ],
      "line_list": [
        {
          "line_id": "DFP0010",
    
```

```

        "line_name" : "有楽町線"
    }
],
"sale_date_from" : "2022-11-01T00 : 00 : 00+09 : 00",
"sale_date_to" : "2023-03-31T23 : 59 : 59+09 : 00",
"sale_time_from" : "00 : 00",
"sale_time_to" : "23 : 59",
"sale_flg" : "1",
"item_list" : [
    {
        "item_id" : "ITEMID-1001",
        "item_cd" : "A2203001",
        "item_name" : "アイテム名その1",
        "validity_type" : "4",
        "validity_date_from" : null,
        "validity_date_to" : null,
        "validity_hours" : 0,
        "validity_days" : 99,
        "validity_hhmm" : "23 : 59",
        "expiration_type" : "4",
        "expiration_months" : 6,
        "expiration_days" : 1,
        "expiration_hhmm" : "02 : 00",
        "facility_list" : [
            {
                "facility_id" : "FACILITYID-1001",
                "facility_name" : "施設その1"
            }
        ],
        "sort_order_number" : 1
    }
],
"fixed_price_list" : [
    {
        "type_id" : "01",
        "type_name" : "大人",
        "required_flg" : "1",
        "price" : 1000,
    }
]

```

```

        "price_unit_name" : "名",
        "cancel_fee_unit" : 300,
        "cancel_fee" : 220
    }
]
}
]
}

```

◇

● チケット種別一覧

➤ 概要

◇ リクエストに応じたチケット種別情報を返却する

➤ URI

◇ /ticketing/types

➤ 使用例

◇ /ticketing/types?package_id_list=1e49a0ec-6532-489b-b374-591fd5656184

➤ リクエストパラメータ

項目	名称	説明	値	必須
パッケージ ID リスト	package_id_list	パッケージ ID リスト	配列	○

➤ レスポンスパラメータ

項目	名称	説明	値
時刻	timestamp	UNIX 時刻	整数
パッケージ リスト	package_list	パッケージリスト	配列
パッケージ ID	package_list.package_id	パッケージ ID	任意の文字列(上限 36 文字)
パッケージ コード	package_list.package_code	パッケージコード	任意の文字列(上限 36 文字)
パッケージ 名	package_list.package_name	パッケージ名	任意の文字列 (上限 200 文字)
カ テ ゴ リ ID	package_list.category_id	カテゴリ ID (fp: フリーパス、rt: 特典チケット、op: 1 回券)	任意の文字列(上限 36 文字)
エリアリス ト	package_list.area_list	エリアリスト※1	配列
エリア ID	package_list.area_list.area_id	エリア ID	任意の文字列(上限 36 文字)

デマンド型交通リアルタイム経路検索システム システム設計書

路線リスト	package_list.line_list	路線リスト※2	配列
路線 ID	package_list.line_list.line_id	路線 ID	任意の文字列(上限 36 文字)
路線名	package_list.line_list.line_name	路線名	任意の文字列 (上限 200 文字)
販売開始日時	package_list.sale_date_from	販売開始日時 (YYYY-MM-DDThh : mm : ss+09 : 00 形式)	任意の文字列(上限 25 文字)
販売終了日時	package_list.sale_date_to	販売終了日時 (YYYY-MM-DDThh : mm : ss+09 : 00 形式)	任意の文字列(上限 25 文字)
販売開始時刻	package_list.sale_time_from	販売開始時刻	任意の文字列(上限 25 文字)
販売終了時刻	package_list.sale_time_to	販売終了時刻	任意の文字列(上限 25 文字)
販売可フラグ	package_list.sale_flg	販売可フラグ (0 : 販売不可、1 : 販売可)	任意の文字列 (上限 1 文字)
商品リスト	package_list.item_list	商品リスト	配列
商品 ID	package_list.item_list.item_id	商品 ID	任意の文字列(上限 36 文字)
商品コード	package_list.item_list.item_cd	商品コード※3	任意の文字列(上限 36 文字)
商品名	package_list.item_list.item_name	商品名	任意の文字列 (上限 200 文字)
使用期限区分	package_list.item_list.validity_type	使用期限区分 (1 : 期間、2 : 購入時点より○時間後、3 : 購入時点より○日後、4 : 購入時点より○日後□時△分、5 : 予約日、6 : 指定日(複数あり))	任意の文字列 (上限 1 文字)
使用開始日時	package_list.item_list.validity_date_from	使用開始日時 (使用期限区分 : 1)	任意の文字列(上限 25 文字)
使用終了日時	package_list.item_list.validity_date_to	使用終了日時 (使用期限区分 : 1)	任意の文字列(上限 25 文字)
使用期限時間数	package_list.item_list.validity_hours	使用期限時間数 (使用期限区分 : 2)	整数
使用期限日	package_list.item_list.validity_days	使用期限日数 (使用期限	整数

デマンド型交通リアルタイム経路検索システム システム設計書

数		区分：3、4)	
使用期限時刻	package_list.item_list.validity_hhmm	使用期限時刻（使用期限区分：4)	任意の文字列（上限 5 文字）
有効期限区分	package_list.item_list.expiration_type	有効期限区分（1：使い切り、2：アクティベートから○時間□分、3：アクティベートから○日間、4：アクティベートから○日後□時△分、5：アクティベートから○月後□時△分）	任意の文字列（上限 1 文字）
有効期限月数	package_list.item_list.expiration_months	有効期限月数	整数
有効期限日数	package_list.item_list.expiration_days	有効期限日数（有効期限区分：3、4)	整数
有効期限時刻	package_list.item_list.expiration_hhmm	有効期限時刻（有効期限区分：2、4)	任意の文字列（上限 5 文字）
関連施設	package_list.item_list.facility_list	関連施設	配列
施設 ID	package_list.item_list.facility_list.facility_id	施設 ID	任意の文字列(上限 36 文字)
施設名	package_list.item_list.facility_list.facility_name	施設名	任意の文字列（上限 200 文字）
表示順	package_list.item_list.sort_order_number	表示順	整数
固定価格リスト	package_list.fixed_price_list	固定価格リスト（価格区分：1)	配列
価格種別 ID	package_list.fixed_price_list.type_id	価格種別 ID	任意の文字列(上限 36 文字)
価格種別名	package_list.fixed_price_list.type_name	価格種別名	任意の文字列（上限 200 文字）
購入必須フラグ	package_list.fixed_price_list.required_flg	購入必須フラグ（0：任意、1：必須）	任意の文字列（上限 1 文字）
価格	package_list.fixed_price_list.price	価格	整数
価格単位名	package_list.fixed_price_list.price_unit_name	価格単位名	任意の文字列（上限 5 0 文字）
キャンセル単価	package_list.fixed_price_list.cancel_fee_unit	キャンセル単価	整数
キャンセル	package_list.fixed_price_list.cancel_fee	キャンセル手数料	整数

手数料			
-----	--	--	--

※1 コードリスト「エリアリスト」を参照。

※2 コードリスト「路線リスト」を参照。

※3 コードリスト「商品コード」を参照。

【IF014】交通系 IC カード情報登録要求 API

- 本インターフェースの概要
 - ID-PORT に交通系 IC カード情報を登録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
 - **【FN002】** 交通系 IC カード登録
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -
 - レスポンスデータ形式
 - ◇ -
 - 文字コード
 - ◇ -
 - URI
 - ◇ -
 - リクエストパラメータ
 - ◇ -
 - レスポンスパラメータ
 - ◇ -
 - レスポンスサンプル
 - ◇ -

【IF015】ログイン情報連携 API

- 本インターフェースの概要
 - ID-PORT に交通系 IC カード情報を登録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
 - **【FN001】** ログイン
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -

- レスポンスデータ形式
 - ◇ -
- 文字コード
 - ◇ -
- URI
 - ◇ -
- リクエストパラメータ
 - ◇ -
- レスポンスパラメータ
 - ◇ -
- レスポンスサンプル
 - ◇ -

【IF016】マイナンバーカード情報取得 API

- 本インターフェースの概要
 - マイナンバーカード情報の登録を行うための情報連携インターフェース
- 本インターフェースを利用する機能
 - **【FN011】マイナンバーカード情報登録**
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -
 - レスポンスデータ形式
 - ◇ -
 - 文字コード
 - ◇ -
 - URI
 - ◇ -
 - リクエストパラメータ
 - ◇ -
 - レスポンスパラメータ
 - ◇ -
 - レスポンスサンプル
 - ◇ -

【IF017】チケット情報登録要求 API

- 本インターフェースの概要

- ID-PORT にチケット情報を登録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
 - 【FN010】 チケット登録
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -
 - レスポンスデータ形式
 - ◇ -
 - 文字コード
 - ◇ -
 - URI
 - ◇ -
 - リクエストパラメータ
 - ◇ -
 - レスポンスパラメータ
 - ◇ -
 - レスポンスサンプル
 - ◇ -

【IF018】 交通系 IC カード登録機能呼出 API

- 本インターフェースの概要
 - GunMaaS のアクセストークンを利用し、ユーザが交通系 IC カードを登録する機能呼び出すインターフェース
- 本インターフェースを利用する機能
 - 【FN001】 ログイン
 - 【FN002】 交通系 IC カード登録
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -
 - レスポンスデータ形式
 - ◇ -
 - 文字コード
 - ◇ -
 - URI

- ◇ -
- リクエストパラメータ
- ◇ -
- レスポンスパラメータ
- ◇ -
- レスポンスサンプル
- ◇ -

【IF019】 チケット販売機能呼出 API

- 本インターフェースの概要
 - GunMaaS のアクセストークンを利用し、ユーザが交通系 IC カードを登録する機能呼び出すインターフェース
- 本インターフェースを利用する機能
 - **【FN001】 ログイン**
 - **【FN003】 チケット販売**
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -
 - レスポンスデータ形式
 - ◇ -
 - 文字コード
 - ◇ -
 - URI
 - ◇ -
 - リクエストパラメータ
 - ◇ -
 - レスポンスパラメータ
 - ◇ -
 - レスポンスサンプル
 - ◇ -

【IF020】 チケット決済機能呼出 API

- 本インターフェースの概要
 - GunMaaS のアクセストークンを利用し、ユーザが交通系 IC カードを登録する機能呼び出すインターフェース
- 本インターフェースを利用する機能

- 【FN001】 ログイン
- 【FN004】 チケット決済
- 本 API の共通仕様
 - プロトコル
 - ◇ -
 - メソッド
 - ◇ -
 - レスポンスデータ形式
 - ◇ -
 - 文字コード
 - ◇ -
 - URI
 - ◇ -
 - リクエストパラメータ
 - ◇ -
 - レスポンスパラメータ
 - ◇ -
 - レスポンスサンプル
 - ◇ -

2-1-5. ユーザインターフェース (UI)

2-1-5-a. 画面遷移図

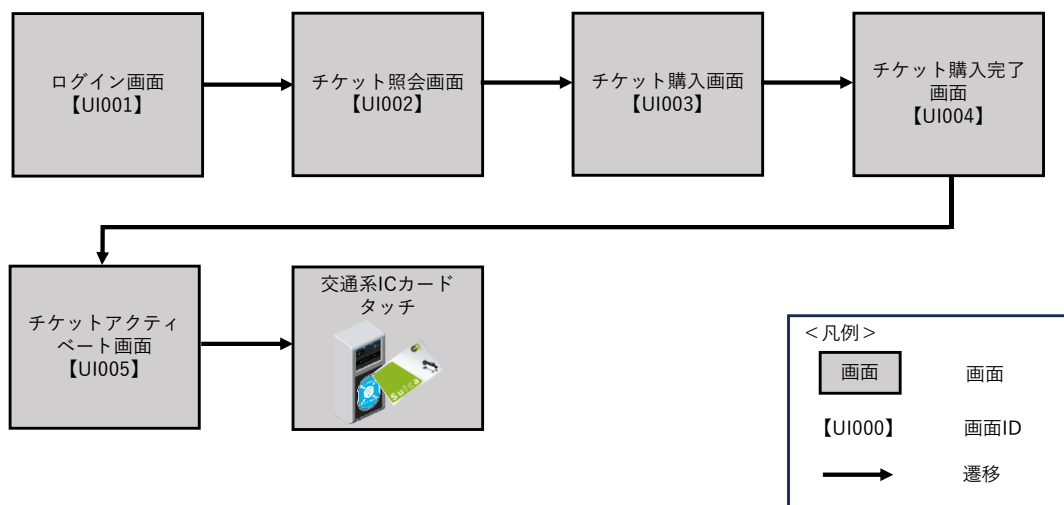


図 2-27 画面遷移図

2-1-5-b. ユーザーインターフェース一覧

表 2-6 上毛電気鉄道定期券購入時の GunMaaS Web サイト画面遷移一覧

※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/新規登録選択画面	● GunMaaS Web サイトにログインまたは新規会員登録をする場合に表示する画面	FN001
UI002	チケット照会画面	● GunMaaS Web サイトで販売中のチケットが一覧化されており、各チケットの概要を確認できる画面	FN003
UI003	チケット購入画面	● チケットの区間選択や人数選択、金額確認等、購入に伴う詳細情報を入力、確認できる画面	FN004
UI004	チケット購入完了画面	● チケット購入が確定した場合に購入完了画面を表示する画面	FN004
UI005	チケットアクティベート画面	● チケット名や有効期間、人数等を表示する画面 ● 使用開始ボタンを表示し、使用中とすることができる画面	FN005 FN006 FN007 FN008

2-1-5-c. ユーザインターフェースの詳細

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

【マルチモーダルな移動体験の実現】

【UI001】 ログイン/新規登録選択画面



図 2-28 ログイン画面

- 概要
 - 画面右上または下部より「ログイン/新規登録」ボタンをタップすると表示される
 - 既に会員登録済みの場合は「ログイン」、新規会員登録の場合は「新規登録」を選択する
- 本画面から利用する機能
 - **【FN001】 ログイン**

【UI002】チケット照会画面

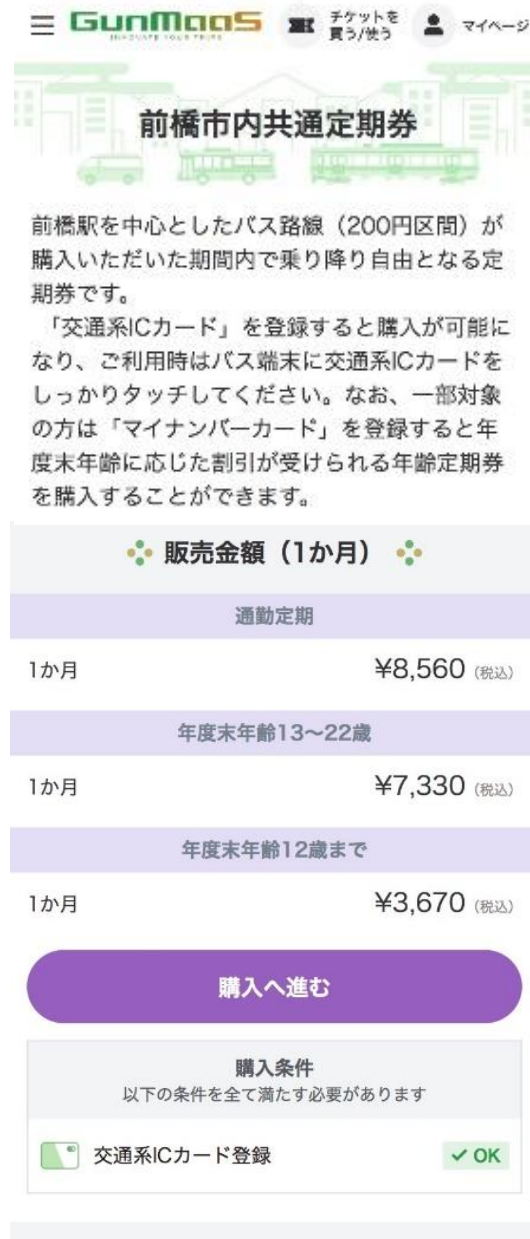


図 2-29 チケット照会画面

- 概要
 - チケットの利用条件や購入条件、金額等の詳細が確認できる
 - 「購入へ進む」をタップすると、【UI003】チケット購入画面へ遷移しチケット購入画面が表示される
- 本画面から利用する機能
 - 【FN003】チケット販売

【UI003】チケット購入画面



Gunmaas

チケットを
買う/使う

マイページ

購入

上毛電気鉄道定期券

乗降場所を指定してください

乗車場所

中央前橋

降車場所

西桐生

購入数を設定してください

※料金は税込価格です

大人 ¥8,560

※未就学児は無料です。
※一度に購入できるのは、5人分までです。

合計 (税込) ¥8,560

購入に関する注意事項

- チケット購入後、利用期間の変更や人数変更・オプションの変更・追加は出来ません。
- チケットの取り消しは、使用前に限り可能です。
- 購入完了後に領収書がメールにて送信されます。宛名の印字が必要な場合は、プロフィールより宛名を設定後、購入へお進みください。
- 有効範囲からの乗り越しされる場合、乗り越しされた区間の運賃の別途お支払いが必要となります。差額精算ではございません。なお、過剰額があっても払い戻しはいたしません。
- 購入日より100日間ご利用可能です。

戻る 確認へ進む

図 2-30 チケット購入画面

- 概要
 - 購入しようとするチケットの区間や枚数等の必要条件を選択でき、金額が確認できる
- 本画面から利用する機能
 - 【FN003】チケット販売

【UI004】チケット購入完了画面



図 2-31 チケット購入完了画面

- 概要
 - チケット購入に伴う決済が問題なくできた場合、チケット購入完了を確認することができる
- 本画面から利用する機能
 - 【FN004】チケット決済

【UI005】チケットアクティベート画面



図 2-32 チケットアクティベート画面

- 概要
 - チケットの有効期間や利用可能区間、人数、金額等を表示する
 - 画面下部で使用開始ボタンをタップすることでチケットを使用中にすることができる
- 本画面から利用する機能
 - 【FN005】 運賃箱制御(車両用)
 - 【FN006】 運賃箱制御(駅用)
 - 【FN007】 運賃計算
 - 【FN008】 OD 作成

2-2. 開発するシステム：非機能要件（NF）

2-2-1. 非機能要件一覧

表 2-7 非機能要件一覧

カテゴリ	ID	非機能項目	要件詳細
性能・拡張性	NF001	スケーラビリティと拡張性	<ul style="list-style-type: none"> システムは将来のユーザ数、データ量の増加によるCPU やメモリの増強や新機能の追加に対応できるように設計する。
運用・保守性	NF002	データのバックアップ	<ul style="list-style-type: none"> GunMaaS の社会的影響力を考慮し、適切な頻度および保存箇所にデータのバックアップの取得や運用保守作業を行う。

2-2-2. 非機能要件の詳細

【NF001】 スケーラビリティと拡張性

- 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
 - スパイクアクセス時に柔軟に対応できることと、単一のサーバに依存しないことからサーバダウン等の障害を回避できるため、サーバレスアーキテクチャである AWS Lambda を用いる。また AWS では、新機能を追加する際に、容易にリソースの追加や Amazon API Gateway 用いた境構築を行うことができる。

【NF002】 データのバックアップ

- 概要
 - 複数のアベイラビリティゾーンにデータを保存すること。また GunMaaS は社会的な影響が大きいシステムであることからデータのバックアップは手動ではなく日次より自動でバックアップを取得する。
- 設定理由
 - 特定のサーバが障害によって失われたとしても、複数のサーバへ保存することにより耐障害性を高めることができるため、複数のアベイラビリティゾーンにまたがってデータのバックアップを取得する。GunMaaS はデータのフルバックアップを行っており、障害発生時はバックアップデータから必要に応じて復旧作業を行う。

3. デマンド型交通リアルタイム経路検索システム

3-1. 機能要件 (FN/SL/AL/CO/HW/IF/UI)

3-1-1. システム機能 (FN)

3-1-1-a. システムアーキテクチャ

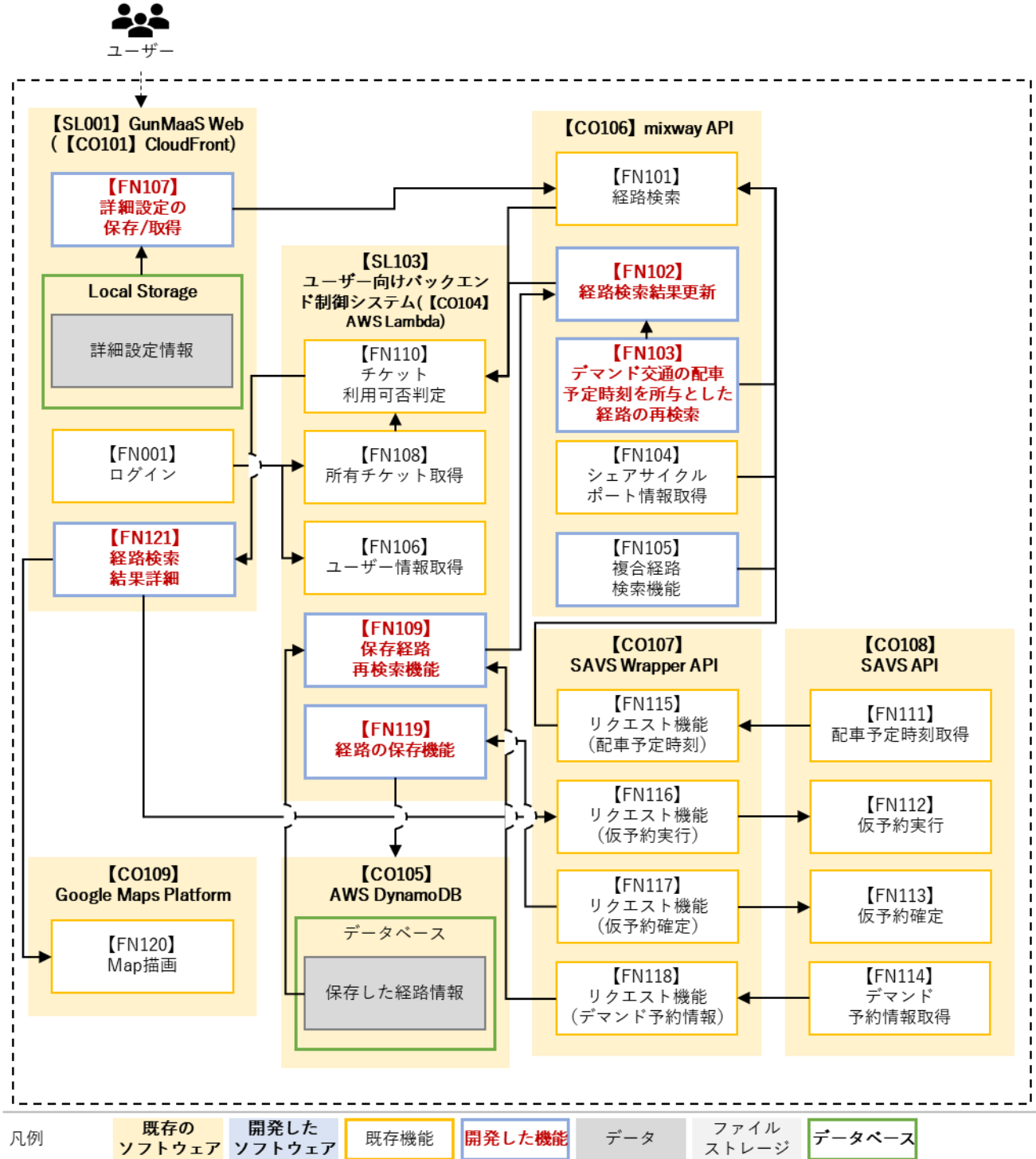


図 3-1 システムアーキテクチャ

3-1-1-b. システム機能一覧

表 3-1 機能一覧

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN101	経路検索	<ul style="list-style-type: none"> ● 出発地、目的地、日付を指定して、鉄道や路線バス、高速バス、空港連絡バス、飛行機、フェリーを利用した経路検索を行う機能
FN102	経路検索結果更新	<ul style="list-style-type: none"> ● 経路検索結果に含まれるデマンド交通区間のリアルタイム情報を外部システムから取得する機能 ● 【FN101】で出力された元経路の情報と、取得したデマンド交通区間の配車予定時刻を【FN103】に渡す機能
FN103	デマンド交通の配車予定時刻を所与とした経路の再検索	<ul style="list-style-type: none"> ● 指定した区間の情報を優先して、前後の区間を再検索する機能
FN104	シェアサイクルポート情報取得	<ul style="list-style-type: none"> ● 経路検索結果に含まれるシェアサイクルポートの貸出/返却可能台数を取得する機能
FN105	複合経路検索機能	<ul style="list-style-type: none"> ● シェアサイクルを利用した経路を案内する機能
FN106	ユーザ情報取得	<ul style="list-style-type: none"> ● 外部システムからユーザ情報を参照する機能
FN107	詳細設定の保存/取得	<ul style="list-style-type: none"> ● 経路検索の検索条件やデマンド交通の人数を保存する機能
FN108	所有チケット取得	<ul style="list-style-type: none"> ● 外部システムから所有するチケット情報を取得する機能
FN109	保存経路再検索機能	<ul style="list-style-type: none"> ● 保存した経路と、経路にデマンド交通が含まれている場合に【FN114】で取得した配車予定時刻とで、【FN102】を Call する機能
FN110	チケット利用可否判定	<ul style="list-style-type: none"> ● 所有するチケットが利用できる経路があるか判定する機能
FN111	配車予定時刻取得	<ul style="list-style-type: none"> ● 出発/到着地を渡してデマンド交通の現在の配車予定時刻を計算する機能
FN112	仮予約実行	<ul style="list-style-type: none"> ● 出発/到着地を渡してデマンド交通の仮予約を行う機能
FN113	仮予約確定	<ul style="list-style-type: none"> ● デマンド交通の仮予約を確定する機能
FN114	デマンド予約情報取得	<ul style="list-style-type: none"> ● デマンド交通の予約情報（配車予定時刻等）を返却する機能
FN115	リクエスト機能（配車予定時刻取得）	<ul style="list-style-type: none"> ● 経路検索結果の経路にデマンド交通区間が存在する場合、配車予定時刻を取得する機能
FN116	リクエスト機能（仮予約実行）	<ul style="list-style-type: none"> ● 経路検索結果の経路にデマンド交通区間が存在する場合、仮予約を実行する機能

デマンド型交通リアルタイム経路検索システム システム設計書

FN117	リクエスト機能（仮予約確定）	<ul style="list-style-type: none"> ● 仮予約 ID を渡して仮予約を実行する機能
FN118	リクエスト機能（デマンド予約情報取得）	<ul style="list-style-type: none"> ● 最新のデマンド交通の予約情報を取得する機能
FN119	経路の保存/削除	<ul style="list-style-type: none"> ● 経路の情報とデマンド交通の予約 ID を保存する機能 ● 経路の情報とデマンド交通の予約 ID を削除する機能
FN120	Map 描画	<ul style="list-style-type: none"> ● 経路を Map に描画する機能 ● シェアサイクルのポート情報を表示する機能

3-1-1-c. 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。

【FN101】 経路検索

- 本システム機能の概要
 - 出発地、目的地、日付を指定して、鉄道や路線バス、高速バス、空港連絡バス、飛行機、フェリーを利用した経路検索を行う機能
 - 経路検索結果にデマンド交通区間が含まれる場合、デマンド交通のリアルタイム情報を取得する機能。
 - 経路検索で出力された情報と、取得したデマンド交通の配車予定時刻で更新した経路を取得する機能。
- 本システム機能の入力・処理・出力のフローチャート

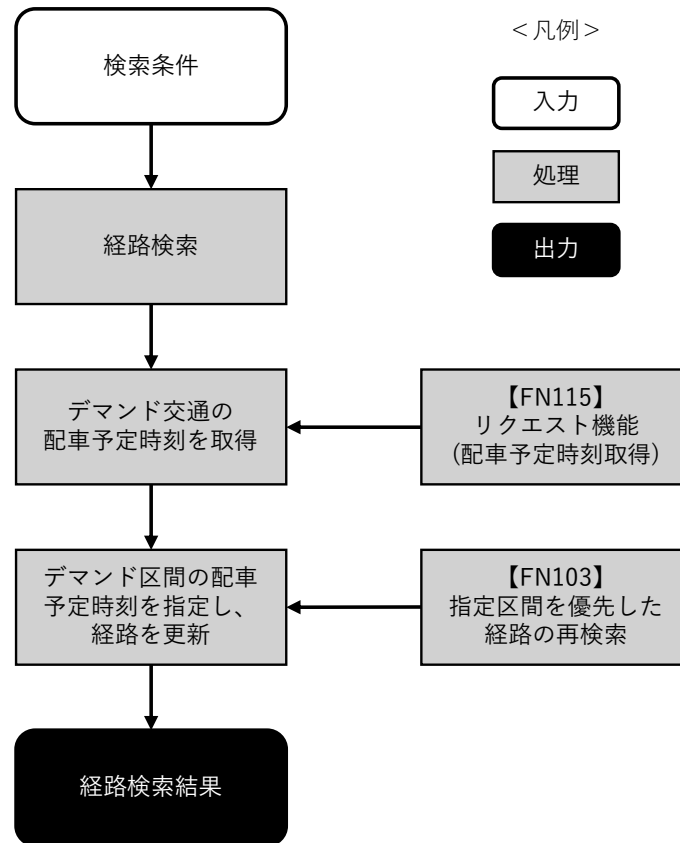


図 3-2 【FN101】 のフローチャート

- 本システム機能の処理の詳細
 - 経路検索
 - ◇ 処理内容
 - 検索条件（出発地/目的地/日付）を元に、最適な経路を検索する。

◇ 利用するライブラリ

- -

◇ 利用するアルゴリズム

- -

● 本システム機能の入出力データの仕様

➤ 入力

◇ 検索条件

- データの内容
 - 出発地/目的地/日付を指定
- データの形式
 - HTTP リクエスト
- 利用するデータインターフェース
 - 【IF101】 経路検索 API
- クエリパラメータ

名称	説明	型	必須
key	アクセスキー	string	○
viaList	出発地、到着地	string	○
date	探索日付	string (YYYYMMDD)	-
time	探索時刻	string (HHMM)	-
searchType	探索種別	string	-
sort	ソート種別	string	-
answerCount	最大回答数	integer	-
searchCount	最大探索数	integer	-
conditionDetail	詳細探索条件	string	-
mode	探索する経路に利用する交通機関	string	-
walkSpeed	徒歩速度。単位： km/h	float	-
maxWalkDistance	徒歩で移動する距離の上限	integer	-
maxWalkDistanceLocalBus	路線バスのバス停まで徒歩で移動する距離の上限	integer	-
applyMaxWalkDistanceStrict	出発地あるいは到着地から最寄駅までの距離制約を厳密に適用（徒歩）	boolean	-
carSpeed	タクシーでの移動速度。単位： km/h	float	-
minCarDistance	タクシーで移動する距離の下限	integer	-
maxCarDistance	タクシーで移動する距離の上限	integer	-
applyMaxCarDistanceStrict	出発地あるいは到着地から最寄駅までの距離制約を厳密に適用 ・ true: maxCarDistance に指定した距離以上に離れている駅は、最寄駅として採用されません	boolean	-

	・ false: 最も近い駅が maxCarDistance に指定した距離以上に離れている場合は、その駅を最寄駅として採用されます		
cycleSpeed	シェアサイクルでの移動速度。単位： km/h	float	-
minCycleDistance	シェアサイクルで移動する距離の下限	integer	-
maxCycleDistance	シェアサイクルで移動する距離の上限	integer	-
maxWalkDistanceCycle	シェアサイクルポートまで徒歩で移動する距離の上限	integer	-
cycleCorporationBind	シェアサイクル事業者絞り込み	string	-
maxWalkDistanceDemand	デマンド交通ポートまで徒歩で移動する距離の上限	integer	-
demandCorporationBind	デマンド交通事業者絞り込み	string	-
fromName	出発地の名称	string	-
toName	到着地の名称	string	-
demandAdultCount	デマンド交通予約時の大人の人数	integer	○
demandChildCount	デマンド交通予約時の小人の人数	integer	○
canGetDemandEstimations	デマンド交通における配車目安時間の取得可否	boolean	○

➤ 出力

◇ 経路検査結果

- データの内容
 - 経路検索結果の構造化データ（複数経路）
- データの形式
 - JSON
- レスポンスパラメータ

名称	説明	型
dataType	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain: 平均時刻に基づく探索	string
SerializeData	経路シリアルライズデータを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期券を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

【FN102】経路検索結果更新 <改修機能>

- 本システム機能の概要
 - 経路検索で出力された情報と、デマンド交通の配車予定時刻を「【FN103】デマンド交通の配車予定時刻を所与とした経路の再検索」に渡し、更新した経路を取得する機能。

既存機能の改修内容

従来の経路検索エンジンに内包されている静的な情報（時刻表）からの指定に加え、動的な情報（配車予定時刻）を経路検索エンジンに引数として渡すための改修を行う。

- 本システム機能の入力・処理・出力のフローチャート

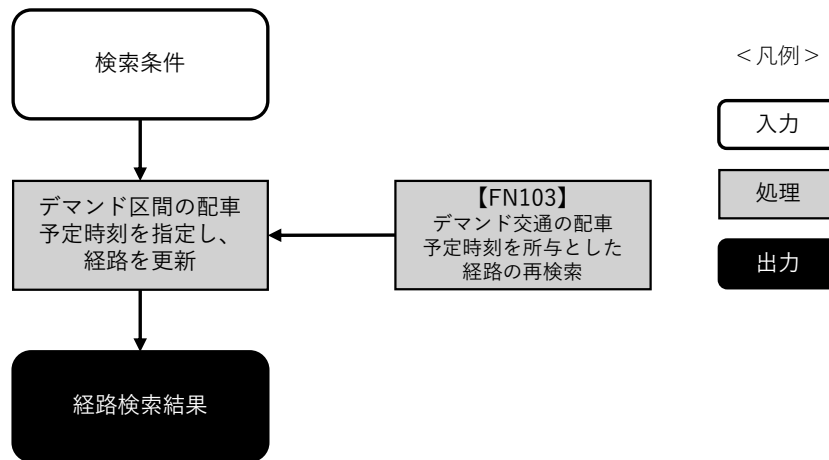


図 3-3 【FN102】のフローチャート（改修後）

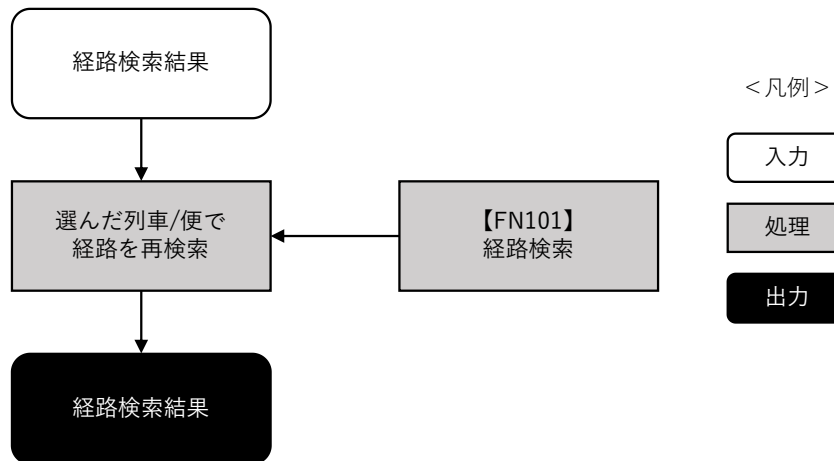


図 3-4 【FN102】のフローチャート（改修前）

- 本システム機能の処理の詳細
 - 経路検索結果更新
 - ◇ 処理内容
 - デマンド交通の配車予定時刻を取得する

デマンド型交通リアルタイム経路検索システム システム設計書

- 取得した配車予定時刻を所与して、経路を再検索する
- ◇ 利用するライブラリ
 - -
- ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 検索条件
 - データの内容
 - 経路シリアルライズデータの指定
 - デマンド交通区間の指定
 - デマンド交通区間の配車予定時刻/到着予定時刻
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF101】 経路検索 API
 - クエリパラメータ

名称	説明	型	必須
key	アクセスキー	string	○
serializeData	経路シリアルライズデータ	string	○
checkEngineVersion	エンジンバージョンの同一チェック	string	-
assignType	departure： 発時刻 arrival： 着時刻	string	
date	探索日付	string (YYYYMMDD)	
time	探索時刻	string (HHMM)	
sectionIndex	便を固定する区間を指し示す区間インデックス	integer	-
latestDepartureTime	便を固定する区間の最新配車予定時刻	string	-
latestArrivalTime	便を固定する区間の最新到着予定時刻	string	-

- 出力
 - ◇ デマンド交通区間のリアルタイム情報を考慮した経路検索結果
 - データの内容
 - 経路検索結果の構造化データ
 - データの形式
 - JSON
 - レスポンスパラメータ

名称	説明	型
dataType	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain : 平均時刻に基づく探索	string
SerializeData	経路シリアル化データを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期券を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

【FN103】 デマンド交通の配車予定時刻を所与とした経路の再検索 <改修機能>

- 本システム機能の概要
 - 「【FN102】 経路検索結果更新」から渡されたデマンド交通の配車予定時刻を所与として、経路を再検索する機能

既存機能の改修内容

従来は「経路」の最初（出発時刻）と最後（到着時刻）のどちらかを所与として、経路検索エンジン内の時刻表を元に経路を再検索しているが、本開発ではデマンド型交通区間の出発時刻/到着時刻を所与として再検索を行える様にするために、経路検索エンジンの外側から配車予定時刻を渡して再検索できる様に改修を行う。

- 本システム機能の入力・処理・出力のフローチャート

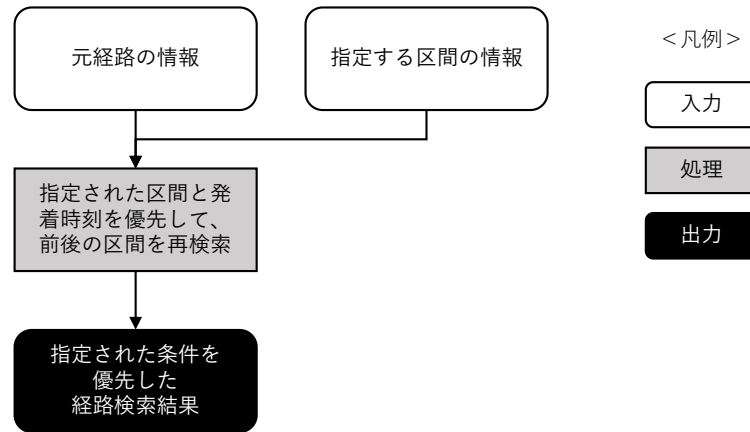


図 3-5 【FN103】 のフローチャート (改修後)

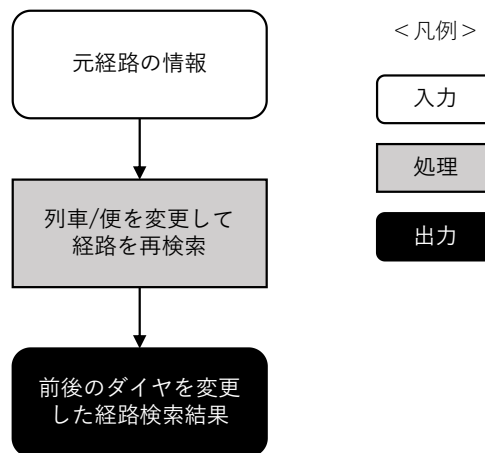


図 3-6 【FN103】 のフローチャート (改修前)

- 本システム機能の処理の詳細
 - 検索
 - ◇ 処理内容
 - 元経路に対し、指定した区間の出発/到着時刻を所与として、前後の区間の経路を再検索する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - ダイクストラ
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 検索条件
 - データの内容
 - 経路シリアルライズデータの指定
 - 所与とする区間の情報 (区間/出発時刻/到着時刻)
 - データの形式

- HTTP リクエスト
- 利用するデータインターフェース
 - 【IF101】 経路検索 API
- クエリパラメータ

名称	説明	型	必須
serializeData	経路シリアルライズデータ	string	○
sectionIndex	区間インデックス	integer	○
overrideDepartureTime	区間インデックスに指定した区間の便・列車を固定して、指定した出発時刻に上書き	string	○
overrideArrivalTime	区間インデックスに指定した区間の便・列車を固定して、指定した到着時刻に上書き	string	○

- 出力
 - ◇ 経路検索結果
 - データの内容
 - 経路検索結果の構造化データ
 - データの形式
 - JSON
 - レスポンスパラメータ

名称	説明	型
dataType	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain: 平均時刻に基づく探索	string
SerializeData	経路シリアルライズデータを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

【FN104】 シェアサイクルポート情報取得

- 本システム機能の概要
 - 経路検索結果に含まれるシェアサイクルポートの貸出/返却可能台数を取得する機能

● 本システム機能の入力・処理・出力のフローチャート

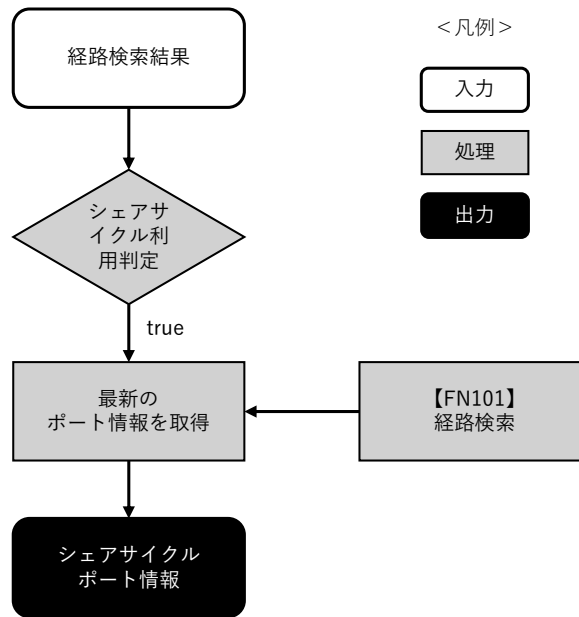


図 3-7 【FN104】のフローチャート

● 本システム機能の処理の詳細

➤ シェアサイクルポート情報取得

◇ 処理内容

- 経路検索結果にシェアサイクル利用区間が含まれているか判定する
- 含まれている場合、シェアサイクルの出発/到着ポートを抽出
- 抽出した出発/到着ポートを Key に「【FN101】経路検索」から周辺のポートと貸出/返却可能台数を取得する

◇ 利用するライブラリ

- -

◇ 利用するアルゴリズム

- -

● 本システム機能の入出力データの仕様

➤ 入力

◇ 検索条件

- データの内容
 - 出発/到着ポートの緯度経度
- データの形式
 - HTTP リクエスト
- 利用するデータインターフェース
 - 【IF101】経路検索 API
- クエリパラメータ

名称	説明	型	必須
----	----	---	----

geoPoint	任意の場所の座標情報	string	○
type	交通種別	string	-
addInformation	追加情報の返却	string	-

- 出力
 - ◇ シェアサイクルポート情報
 - データの内容
 - 指定した緯度経度周辺のポート情報
 - データの形式
 - JSON
 - レスポンスパラメータ

名称	説明	型
Point	ポートを表す要素	object(Array)

【FN105】複合経路検索機能

- 本システム機能の概要
 - 複合経路検索を実行する
 - ◇ 最寄駅までシェアサイクルを利用した経路を検索する機能
- 本システム機能の入力・処理・出力のフローチャート

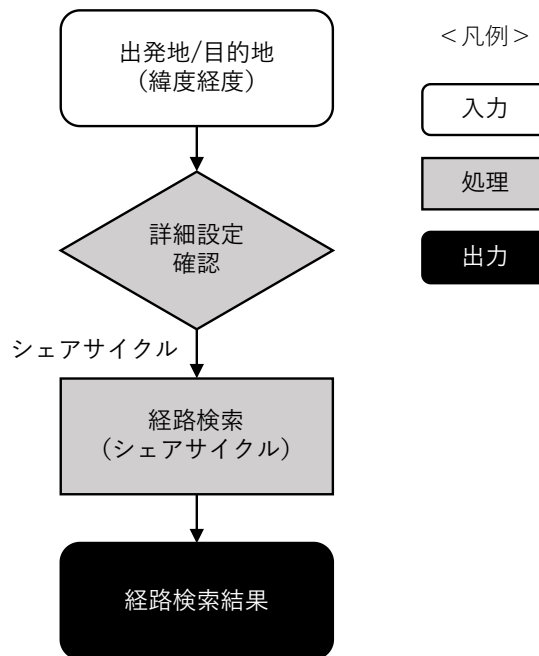


図 3-8 【FN105】のフローチャート

- 本システム機能の処理の詳細
 - 複合経路検索

デマンド型交通リアルタイム経路検索システム システム設計書

- ◇ 処理内容
 - シェアサイクル+公共交通機関を利用した経路検索
- ◇ 利用するライブラリ
 - -
- ◇ 利用するアルゴリズム
 - -

● 本システム機能の入出力データの仕様

- 入力
 - ◇ 検索条件
 - データの内容
 - 出発地/目的地/日付を指定
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF101】 経路検索 API
 - クエリパラメータ

名称	説明	型	必須
key	アクセスキー	string	○
viaList	出発地、到着地	string	○
date	探索日付	string (YYYYMMDD)	-
time	探索時刻	string (HHMM)	-
searchType	探索種別	string	-
sort	ソート種別	string	-
answerCount	最大回答数	integer	-
searchCount	最大探索数	integer	-
conditionDetail	詳細探索条件	string	-
mode	探索する経路に利用する交通機関	string	-
walkSpeed	徒歩速度。単位：km/h	float	-
maxWalkDistance	徒歩で移動する距離の上限	integer	-
maxWalkDistanceLocalBus	路線バスのバス停まで徒歩で移動する距離の上限	integer	-
applyMaxWalkDistanceStrict	出発地あるいは到着地から最寄駅までの距離制約を厳密に適用（徒歩）	boolean	-
carSpeed	タクシーでの移動速度。単位：km/h	float	-
minCarDistance	タクシーで移動する距離の下限	integer	-
maxCarDistance	タクシーで移動する距離の上限	integer	-

applyMaxCarDistanceStrict	出発地あるいは到着地から最寄駅までの距離制約を厳密に適用 ・ true: maxCarDistance に指定した距離以上に離れている駅は、最寄駅として採用されません ・ false: 最も近い駅が maxCarDistance に指定した距離以上に離れている場合は、その駅を最寄駅として採用されます	boolean	-
cycleSpeed	シェアサイクルでの移動速度。単位： km/h	float	-
minCycleDistance	シェアサイクルで移動する距離の下限	integer	-
maxCycleDistance	シェアサイクルで移動する距離の上限	integer	-
maxWalkDistanceCycle	シェアサイクルポートまで徒歩で移動する距離の上限	integer	-
cycleCorporationBind	シェアサイクル事業者絞り込み	string	-
maxWalkDistanceDemand	デマンド交通ポートまで徒歩で移動する距離の上限	integer	-
demandCorporationBind	デマンド交通事業者絞り込み	string	-
fromName	出発地の名称	string	-
toName	到着地の名称	string	-
demandAdultCount	デマンド交通予約時の大人の人数	integer	○
demandChildCount	デマンド交通予約時の小人の人数	integer	○
canGetDemandEstimations	デマンド交通における配車目安時間の取得可否	boolean	○

➤ 出力

◇ 経路検査結果

- データの内容
 - 経路検索結果の構造化データ（複数経路）
- データの形式
 - JSON
- レスポンスパラメータ

名称	説明	型
dataType	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain : 平均時刻に基づく探索	string
SerializeData	経路シリアル化データを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

【FN106】 ユーザ情報取得

- 本システム機能の概要
 - GunMaaS のユーザ情報を取得する機能
- 本システム機能の入力・処理・出力のフローチャート

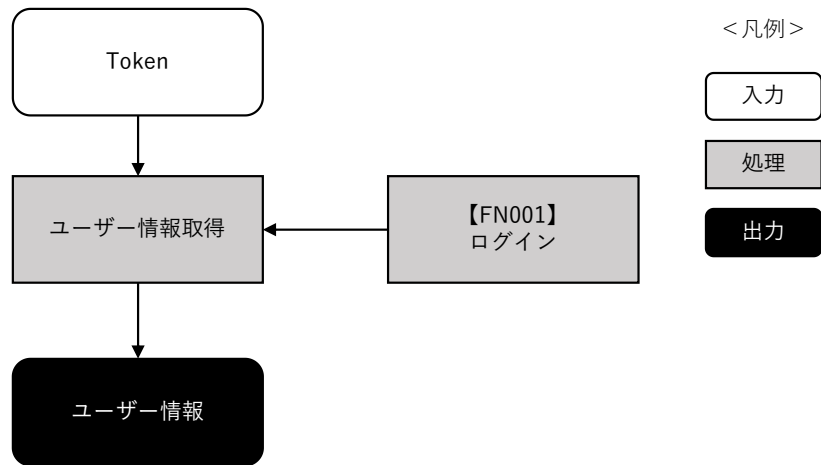


図 3-9 【FN106】 のフローチャート

- 本システム機能の処理の詳細
 - ユーザ情報取得
 - ◇ 処理内容
 - token を Key として「【FN001】 ログイン」から、認証情報を取得する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム

● -

- 本システム機能の入出力データの仕様
 - 入力
 - ◇ token
 - データの内容
 - 文字列
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF111】ログイン情報取得 API
 - 出力
 - ◇ 認証情報
 - データの内容
 - ユーザ情報の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF106】ユーザ情報取得 API

【FN107】 詳細設定の保存/取得 <改修機能>

- 本システム機能の概要
 - 経路検索の詳細設定を保存する機能
 - 経路検索の詳細設定を取得する機能
 - デマンド交通の人数を保存する機能
 - デマンド交通の人数を取得する機能

既存機能の改修内容

従来対応している経路検索の検索条件に加え、デマンド交通の人数を追加するための改修を行う。

- 本システム機能の入力・処理・出力のフローチャート

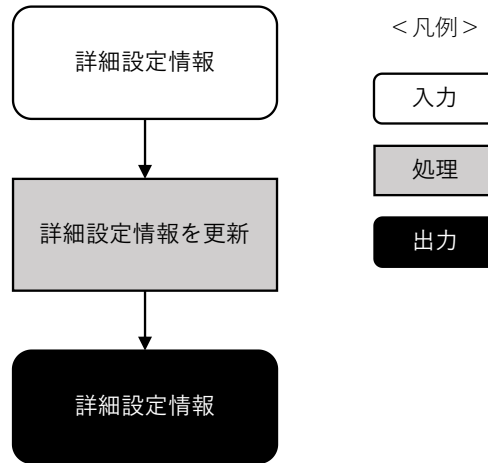


図 3-10 【FN107】のフローチャート

- 本システム機能の処理の詳細
 - 詳細設定の保存
 - ◇ 処理内容
 - 経路検索の詳細設定とデマンド交通の人数を Web ブラウザの Local Storage に保存する
 - ◇ 使用するライブラリ
 - -
 - ◇ 使用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 詳細設定情報
 - データの内容
 - 経路検索の詳細設定（歩く速さ/利用する交通機関/運賃表示/並び順/特急料金/リアルタイム経路検索の利用有無）の文字列
 - デマンド交通予約時の設定する人数の文字列
 - データの形式
 - String
 - 利用するデータインターフェース
 - 【IF107】詳細設定の保存/取得
 - 出力
 - ◇ 詳細設定情報
 - データの内容
 - ◇ 設定値の文字列
 - データの形式
 - ◇ String
 - 利用するデータインターフェース

◇ 【IF107】 詳細設定の保存/取得

【FN108】 所有チケット取得

- 本システム機能の概要
 - ユーザが所有するチケット情報を取得する機能
- 本システム機能の入力・処理・出力のフローチャート

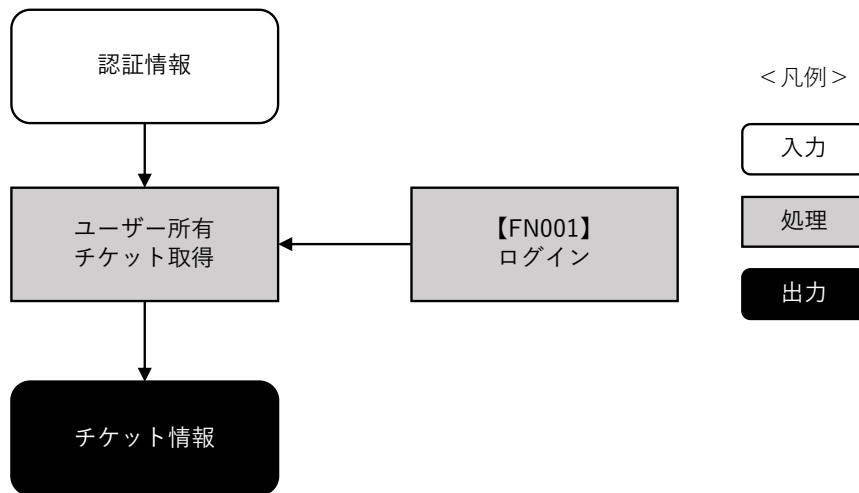


図 3-11 【FN108】 のフローチャート

- 本システム機能の処理の詳細
 - 所有チケット取得
 - ◇ 処理内容
 - 認証情報を Key として所有するチケット情報を取得する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 認証情報
 - データの内容
 - 文字列
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF111】 ログイン情報取得 API
 - 出力

- ◇ 所有するチケット
 - データの内容
 - チケット情報の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF108】所有チケット取得 API

【FN109】保存経路再検索機能 <改修機能>

- 本システム機能の概要
 - 保存した経路にデマンド交通の予約 ID が含まれている場合、経路の情報と、予約 ID を Key に外部から取得したデマンド交通の配車予定時刻とで、最新の経路を検索する機能。

既存機能の改修内容

従来の機能は保存した経路情報を、経路検索エンジンに内包されている静的な情報（時刻表）によって経路の検索（再現）をするものである。本開発ではデマンド交通のリアルタイム情報を反映するため、経路情報に加えて、デマンド交通のリアルタイム情報を取得し、それらを所与として経路を検索する様に改修を行う。

- 本システム機能の入力・処理・出力のフローチャート

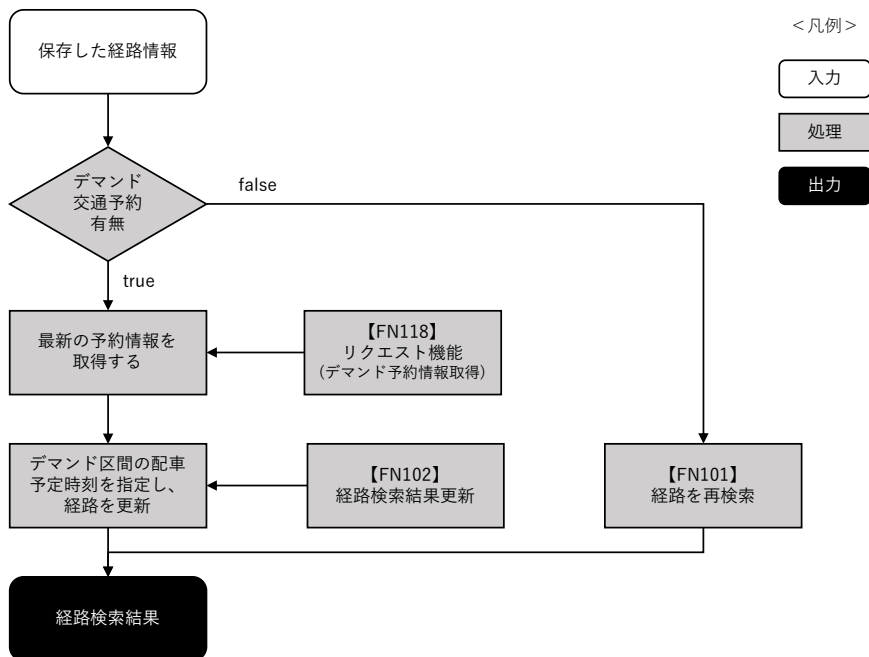


図 3-12 【FN109】のフローチャート（改修後）

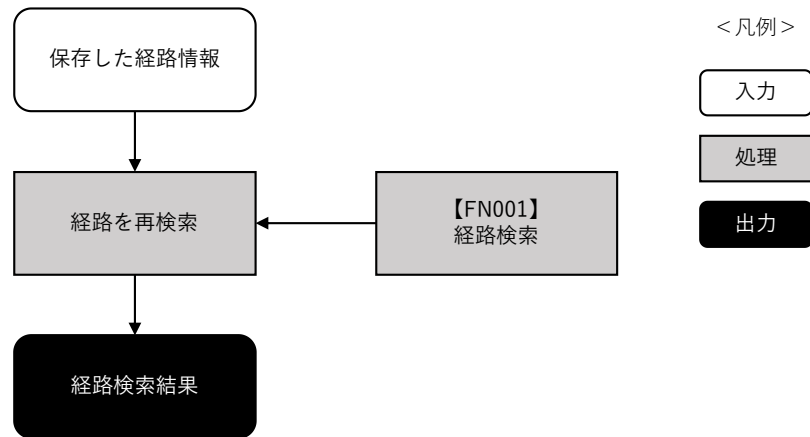


図 3-13 【FN109】のフローチャート（改修前）

- 本システム機能の処理の詳細
 - 保存経路再検索
 - ◇ 処理内容
 - 保存した経路情報にデマンド交通の予約 ID が含まれているか確認する
 - 含まれている場合、デマンド交通の予約 ID で「【FN118】リクエスト機能（デマンド予約情報取得）」を Call して、予約情報を取得する
 - デマンド区間を優先して、前後の区間を再検索する（【FN102】経路検索結果更新）
 - 含まれていない場合
 - 経路を再検索（【FN101】経路検索）
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 保存した経路情報
 - データの内容
 - 経路検索結果の構造化データ
 - デマンド交通の予約 ID
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF110】チケット利用可否判定 API
 - 出力
 - ◇ 経路検索結果
 - データの内容

- 経路検索結果の構造化データ
- データの形式
 - JSON
- レスポンスパラメータ

名称	説明	型
dataType	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain : 平均時刻に基づく探索	string
SerializeData	経路シリアルライズデータを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

【FN110】 チケット利用可否判定

- 本システム機能の概要
 - 所有するチケットが利用できる経路があるか判定する機能
- 本システム機能の入力・処理・出力のフローチャート

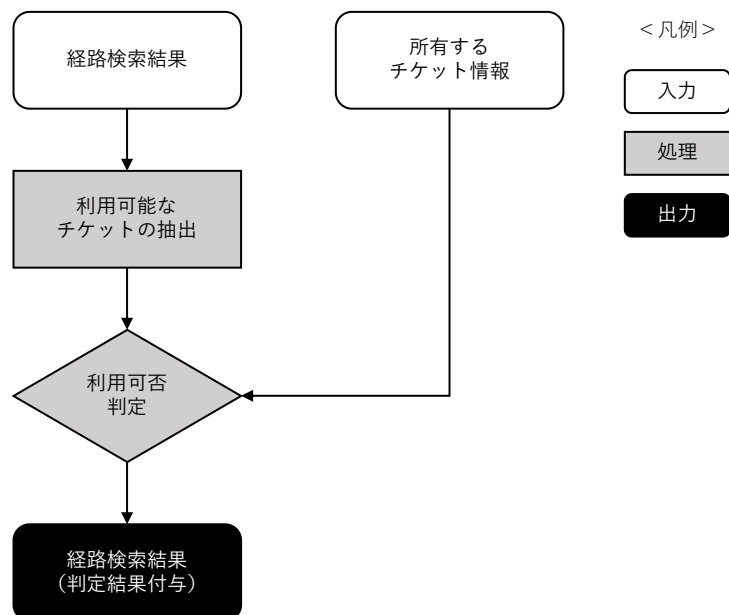


図 3-14 【FN110】 のフローチャート

- 本システム機能の処理の詳細
 - チケット利用判定
 - ◇ 処理内容
 - 経路に適用可能なチケット判定をする
 - 適用可能なチケットに所有するチケットが含まれるか判定をする
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 経路検索結果
 - データの内容
 - 経路検索結果の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース

名称	説明	型
dataType	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain: 平均時刻に基づく探索	string
SerializeData	経路シリアルライズデータを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

- ◇ 所有するチケット
 - データの内容
 - チケット情報の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF110】チケット利用可否判定

- 出力
 - ◇ 経路検索結果
 - データの内容
 - 経路検索結果の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF109】保存経路再検索機能 API

【FN111】配車予定時刻取得

- 本システム機能の概要
 - 出発/到着地/人数/希望配車時刻を渡して現在の配車予定時刻を計算する機能。
- 本システム機能の入力・処理・出力のフローチャート

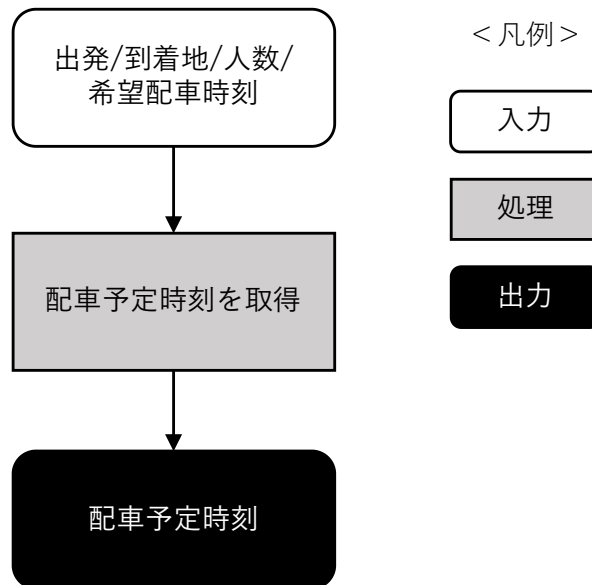


図 3-15 【FN111】のフローチャート

- 本システム機能の処理の詳細
 - 配車予定時刻
 - ◇ 処理内容
 - 出発/到着ポートと人数、希望出発時刻を引数にデマンド交通の配車予定時刻を計算する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様

- 入力
 - ◇ デマンド交通の情報
 - データの内容
 - 出発/到着ポートと人数、希望出発時刻
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF102】配車予定時刻取得 API
- 出力
 - ◇ 配車予定時刻
 - データの内容
 - 配車予定時刻の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF102】配車予定時刻取得 API

【FN112】仮予約実行

- 本システム機能の概要
 - 出発/到着地を渡してデマンド交通の仮予約を行う機能。
- 本システム機能の入力・処理・出力のフローチャート

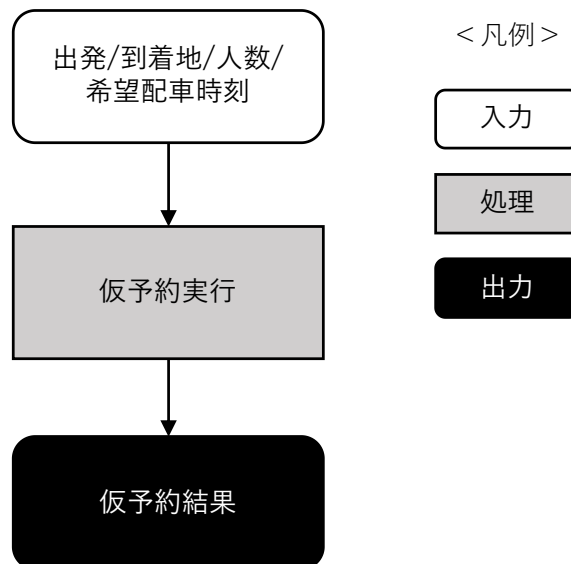


図 3-16 【FN112】のフローチャート

- 本システム機能の処理の詳細
 - 仮予約実行
 - ◇ 処理内容

- 出発/到着ポートと人数、希望出発時刻を引数にデマンド交通の仮予約を行う。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
 - 本システム機能の入出力データの仕様
 - 入力
 - ◇ デマンド交通区間の情報
 - データの内容
 - 出発/到着ポートと人数、希望出発時刻
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF103】 仮予約 API
 - 出力
 - ◇ 仮予約結果
 - データの内容
 - 仮予約結果の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF103】 仮予約 API
- 【FN113】 仮予約確定
- 本システム機能の概要
 - デマンド交通の仮予約を確定する機能。
 - 本システム機能の入力・処理・出力のフローチャート

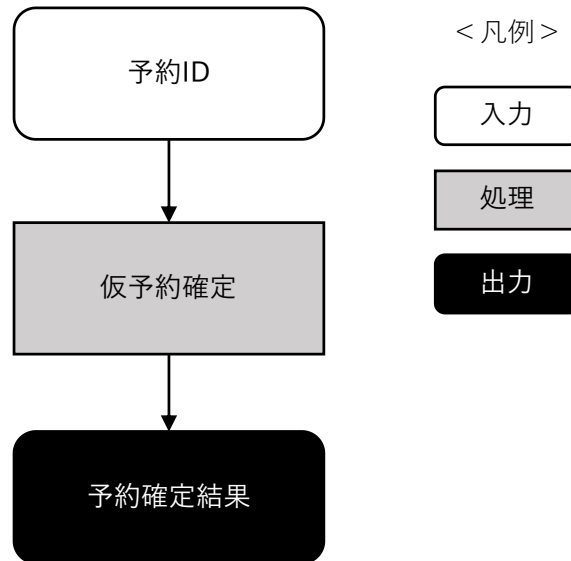


図 3-17 【FN113】のフローチャート

- 本システム機能の処理の詳細
 - 仮予約確定
 - ◇ 処理内容
 - 予約 ID を引数にデマンド交通の仮予約を確定する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 予約 ID
 - データの内容
 - 予約 ID
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF104】仮予約確定 API
 - 出力
 - ◇ 予約確定結果
 - データの内容
 - 仮予約確定の実行結果の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース

➤ 【IF104】仮予約確定 API

【FN114】デマンド予約情報取得

- 本システム機能の概要
 - 予約 ID を Key に最新のデマンド予約情報を返却する機能。
- 本システム機能の入力・処理・出力のフローチャート

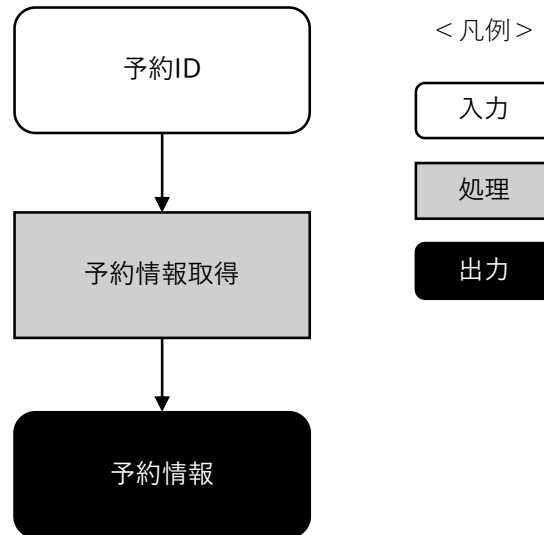


図 3-18 【FN114】のフローチャート

- 本システム機能の処理の詳細
 - 予約情報取得
 - ◇ 処理内容
 - 予約 ID を引数にデマンド交通の予約情報を取得する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 予約 ID
 - データの内容
 - 予約 ID
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF105】デマンド予約情報取得 API
 - 出力

- ◇ 予約情報
 - データの内容
 - 予約情報の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF105】 デマンド予約情報取得 API

【FN115】 リクエスト機能（配車予定時刻取得）

- 本システム機能の概要
 - 経路検索結果の経路にデマンド交通区間が存在する場合、SAVS から前述のデマンド交通区間の配車予定時刻を取得する機能。
- 本システム機能の入力・処理・出力のフローチャート

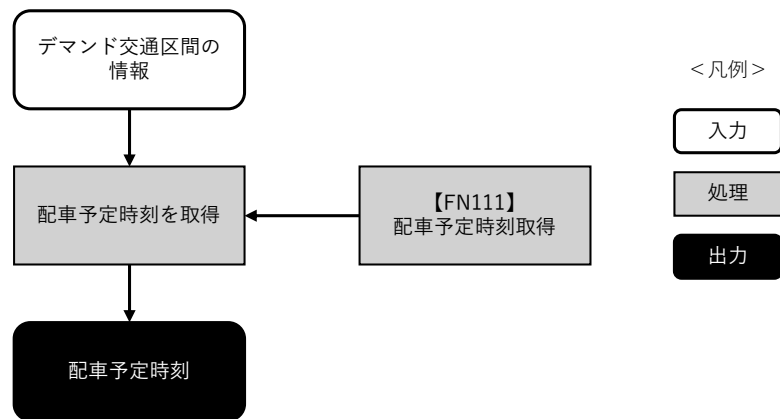


図 3-19 【FN115】 のフローチャート

- 本システム機能の処理の詳細
 - 配車予定時刻
 - ◇ 処理内容
 - 経路検索結果にデマンド交通区間があるか判定
 - 存在する場合、デマンド区間の出発/到着ポートと出発時刻を抽出
 - 出発/到着ポートと人数、希望出発時刻を引数に、「【FN111】配車予定時刻取得」からデマンド交通の配車予定時刻を取得する。
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ デマンド交通区間の情報

- データの内容
 - 経路検索結果の中のデマンド交通区間のデータから、配車予定時刻の取得に必要な項目を抜き出した構造化データ
- データの形式
 - JSON
- 利用するデータインターフェース
 - 【IF102】配車予定時刻取得 API
- 出力
 - ◇ 配車予定時刻
 - データの内容
 - 配車予定時刻の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF102】配車予定時刻取得 API

【FN116】リクエスト機能（仮予約実行）

- 本システム機能の概要
 - 経路検索結果の経路にデマンド交通区間が存在する場合、SAVS に対して前述のデマンド交通区間の仮予約を実行する機能。
- 本システム機能の入力・処理・出力のフローチャート

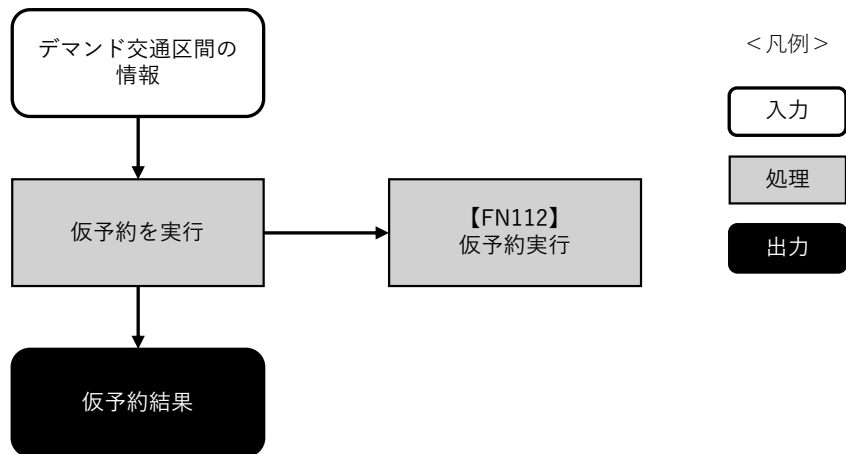


図 3-20 【FN116】のフローチャート

- 本システム機能の処理の詳細
 - 仮予約実行
 - ◇ 処理内容
 - 経路検索結果にデマンド交通区間があるか判定
 - 存在する場合、デマンド区間の出発/到着ポートと出発時刻を抽出

- 「【FN112】 仮予約実行 」を Call して、デマンド交通の仮予約を実行する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ デマンド交通区間の情報
 - データの内容
 - 経路検索結果の中のデマンド交通区間のデータから、仮予約に必要な項目を抜き出した構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF103】 仮予約 API
 - ◇ 仮予約結果
 - データの内容
 - 仮予約結果の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF103】 仮予約 API

【FN117】 リクエスト機能（仮予約確定）

- 本システム機能の概要
 - ユーザが UI にて予約確定ボタンをタップした場合、SAVS に対して予約 ID を Key に仮予約を確定する機能。
- 本システム機能の入力・処理・出力のフローチャート

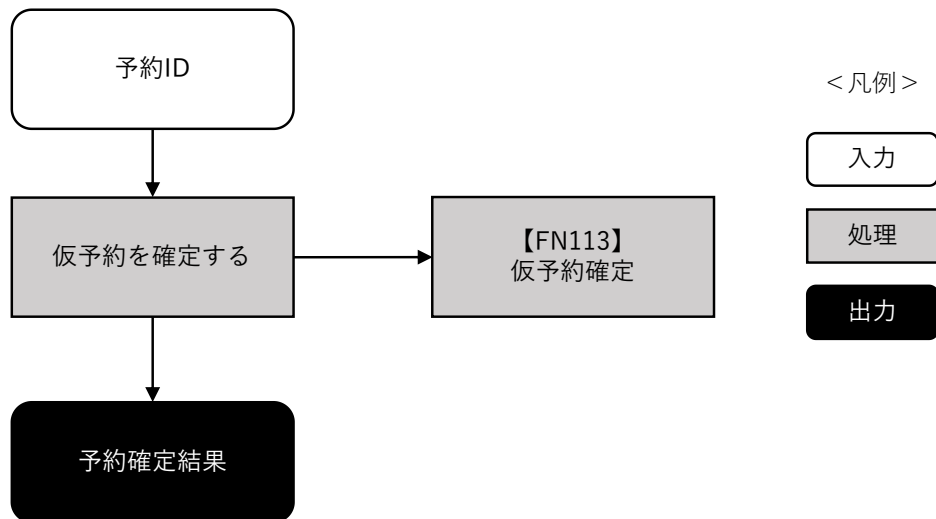


図 3-21 【FN117】のフローチャート

- 本システム機能の処理の詳細
 - 仮予約確定
 - ◇ 処理内容
 - 「【FN113】仮予約確定」を Call して、デマンド交通の仮予約を確定する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 予約 ID
 - データの内容
 - 予約 ID
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF104】仮予約確定 API
 - 出力
 - ◇ 予約確定結果
 - データの内容
 - 仮予約確定の実行結果の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF104】仮予約確定 API

【FN118】リクエスト機能（デマンド予約情報取得）

- 本システム機能の概要
 - 保存した経路の情報に予約済みのデマンド交通区間が含まれる場合、保存した経路の情報にある予約 ID を Key に、最新のデマンド交通の予約情報を SAVS から取得する機能。
- 本システム機能の入力・処理・出力のフローチャート

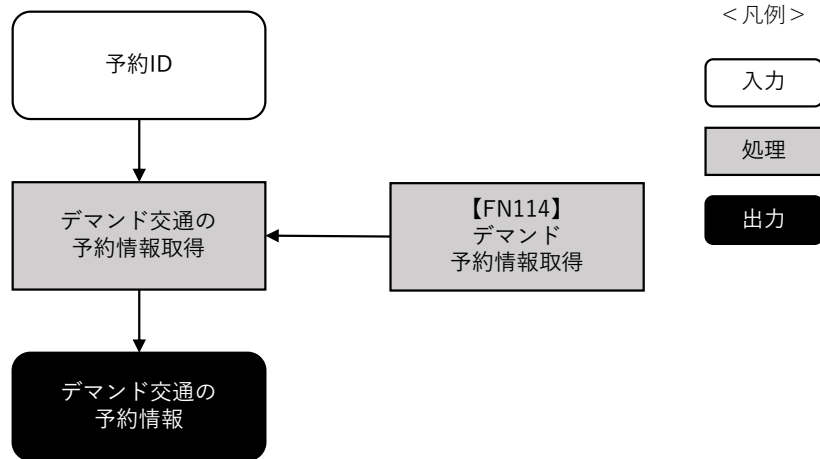


図 3-22 【FN118】のフローチャート

- 本システム機能の処理の詳細
 - 予約情報取得
 - ◇ 処理内容
 - 「【FN114】デマンド予約情報取得」を Call して、デマンド交通の予約情報を取得する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 予約 ID
 - データの内容
 - 予約 ID
 - データの形式
 - HTTP リクエスト
 - 利用するデータインターフェース
 - 【IF105】デマンド予約情報取得 API
 - 出力
 - ◇ 予約情報
 - データの内容
 - 予約情報の構造化データ

- データの形式
 - JSON
- 利用するデータインターフェース
 - 【IF105】 デマンド予約情報取得 API

【FN119】 経路の保存/削除

- 本システム機能の概要
 - ユーザが UI にて予約を確定させた際に、経路の情報とデマンド交通の予約 ID を保存する機能。
 - 保存した経路の情報とデマンド交通の予約 ID を削除する機能。
- 本システム機能の入力・処理・出力のフローチャート

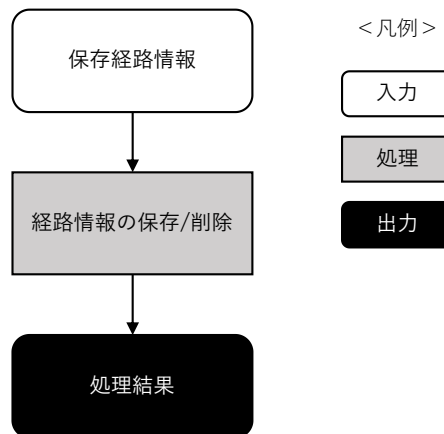


図 3-23 【FN119】 のフローチャート

- 本システム機能の処理の詳細
 - 経路の保存機能
 - ◇ 処理内容
 - 予約確定から遷移された場合
 - 経路の情報を保存
 - デマンド交通の予約 ID を保存する
 - 予約確定をしていない場合
 - 経路の情報のみを保存
 - 保存経路の削除
 - 保存した経路を削除する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 保存経路情報
 - データの内容

- 保存経路を一意にする識別子
- 経路検索結果の構造化データ
- 経路検索情報を持つ構造化データ
- デマンド交通の予約 ID
- データの形式
 - JSON
- 利用するデータインターフェース
 - 【IF112】経路情報保存/削除 API
- 出力
 - ◇ 処理結果
 - データの内容
 - 保存経路の識別子
 - 経路情報取得成功フラグ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF112】経路情報保存/削除 API

【FN120】Map 描画

- 本システム機能の概要
 - 経路検索結果の徒歩・自動車区間は Map 側、公共交通区間は経路検索側から描画するルートを取得し、Map 上に描画する機能。
 - シェアサイクルのポート情報を表示する機能。
- 本システム機能の入力・処理・出力のフローチャート

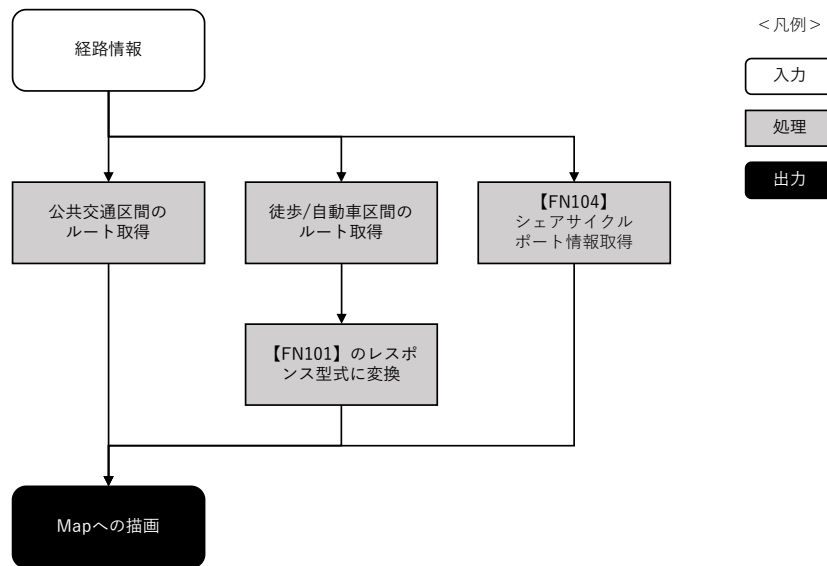


図 3-24 【FN120】のフローチャート

- 本システム機能の処理の詳細

➤ Map 描画

◇ 処理内容

- 経路検索結果の徒歩/自動車区間のルートを取得し、【FN101】経路検索のレスポンス型式に変換する
- 公共交通区間のルートを取得する
- シェアサイクルのポート情報を取得する
- Map への描画

◇ 利用するライブラリ

- -

◇ 利用するアルゴリズム

- -

● 本システム機能の入出力データの仕様

◇ 経路情報

- データの内容
 - 経路検索結果の構造化データ
- データの形式
 - JSON
- 利用するデータインターフェース

名称	説明	型
datatype	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain : 平均時刻に基づく探索	string
SerializeData	経路シリアル化データを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

➤ 出力

◇ Map への描画情報

- データの内容
 - ルートの構造化データ
 - シェアサイクルのポート情報
- データの形式

- JSON
- 利用するデータインターフェース
 - 【IF113】 Map 描画情報

【FN121】 経路検索結果詳細 <改修機能>

- 本システム機能の概要
 - 経路検索結果にデマンド交通区間が含まれる場合、デマンド交通の仮予約を実行する機能
 - 経路検索で出力された情報と、仮予約したデマンド交通の配車予定時刻で更新した経路を取得する機能。

既存機能の改修内容

従来は「経路検索結果」をそのまま UI に表示していたが、本開発では「経路検索結果」にデマンド型交通区間が含まれる場合、デマンド型交通の「仮予約」を実行する改修を行う。

仮予約を実行した後、仮予約の結果である動的な情報（配車予定時刻）を経路検索 API に引数として渡して経路を更新する改修を行う。

- 本システム機能の入力・処理・出力のフローチャート

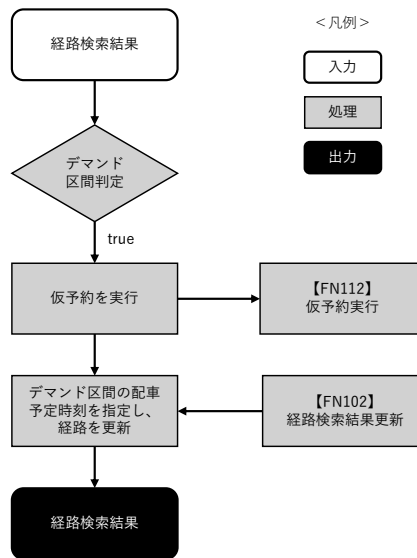


図 3-25 【FN121】のフローチャート（改修後）

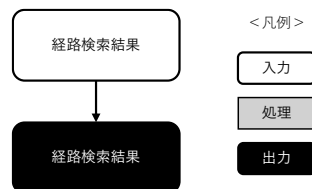


図 3-26 【FN121】のフローチャート（改修前）

- 本システム機能の処理の詳細
 - 経路検索結果詳細

- ◇ 処理内容
 - デマンド交通の仮予約を実行する
 - 仮予約した配車予定時刻を所与として、経路を再検索する
- ◇ 利用するライブラリ
 - -
- ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 経路情報
 - データの内容
 - 経路検索結果の構造化データ
 - データの形式
 - JSON
 - 利用するデータインターフェース
 - 【IF101】 経路検索 API
 - データ詳細

名称	説明	型
datatype	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain : 平均時刻に基づく探索	string
SerializeData	経路シリアルライズデータを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

- 出力
 - ◇ 経路情報
 - データの内容
 - 経路検索結果の構造化データ (デマンド予約実行後)
 - データの形式
 - JSON
 - データ詳細

デマンド型交通リアルタイム経路検索システム システム設計書

名称	説明	型
datatype	時刻を元にした探索結果かどうか ・ onTimetable: 時刻に基づく探索 ・ plain : 平均時刻に基づく探索	string
SerializeData	経路シリアル化データを表す要素	string
PassStatus	定期券の状態を表す要素	object
Price	金額を表す要素	object(Array)
Teiki	定期を表す要素	object
Relation	関係を表す要素	object
OperationLinePattern	経路の区間に紐づく運行路線パターンを表す要素	object(Array)
Route	経路探索結果の経路のみを表す要素	object
AssignStatus	割り当てステータスを表す要素	object
searchType	探索種別	string

3-1-1-d. ソフトウェア・ライブラリ (SL) の詳細

表 3-2 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
SL101	Node.js	20	● AWS Lambda 上で動作するランタイム
SL102	Vue.js	3	● スマートフォン上に経路結果を表示するライブラリ
SL103	ユーザ向けバックエンド制御システム	-	● ユーザ向けのサービス画面と、外部のさまざまなシステム（ユーザ情報管理、詳細設定、外部サービス API など）をつなぐ機能群

【SL101】 Node.js

- ベンダー
 - OpenJS Foundation
- 公式サイト
 - <https://nodejs.org/ja/>
- 概要
 - V8 JavaScript エンジン上に構築された JavaScript 実行環境の 1 つである。イベント化された入出力を扱うサーバサイド JavaScript 環境であり、Web サーバなどのスケーラブルなネットワークプログラムの記述を意図している。ライアン・ダールによって 2009 年に作成され、ダールを雇用している Joyent の支援により成長している。
- 主な機能
 - 静的コンテンツの配信： 経路検索結果の出力やデマンド予約管理等

【SL102】 Vue.js

- ベンダー
 - オープンソース
- 公式サイト
 - <https://vuejs.org/>
- 概要
 - ウェブアプリケーションにおけるユーザインタフェースを構築するための、オープンソースの JavaScript フレームワークである。他の JavaScript ライブラリを使用するプロジェクトへの導入において、容易になるように設計されている。一方で高機能なシングルページアプリケーション (SPA) を構築することも可能である。
- 主な機能

- スマートフォン上の表示： 経路検索結果の表示やデマンド予約情報の表示等

【SL103】 ユーザ向けバックエンド制御システム

- ベンダー
 - ヴァル研究所
- 公式サイト
 - <https://www.val.co.jp/>
- 概要
 - ユーザ向けのサービス画面と、外部のさまざまなシステム（ユーザ情報管理、詳細設定、外部サービス API など）をつなぐ機能群。このシステムが、ユーザの指示（例：検索ボタンを押す、設定を保存する）を受け取ると、裏側にある複数のシステムに適切な指示を出し、その結果を一つにまとめてユーザの画面に表示する。これにより、ユーザは一つのシンプルな操作で複雑な処理を実行でき、サービスを快適に利用することが可能になる。
- 主な機能
 - ユーザ情報の取得
 - 所有チケットの取得と経路検索結果に対しての所有チケットの利用可否判定
 - 経路の保存
 - 保存した経路の再検索機能

3-1-1-e. 数理モデル・アルゴリズム（AL）の詳細

表 3-3 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
AL001	デマンド交通リアルタイム反映アルゴリズム	デマンド交通配車システムの API を活用し、動的な配車予約状況を取得した上で、静的な公共交通データと統合するリアルタイム経路再構成アルゴリズム	FN102 FN103

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。

【AL001】デマンド交通リアルタイム反映アルゴリズム<新規開発>

- 本アルゴリズムの概要
 - 従来の経路検索エンジンは、ライブラリ化された静的データのみを参照するため、予約状況により変動するデマンド交通の情報を直接扱うことができない。本アルゴリズムでは、まず仮のダイヤを用いて抽出した経路から「利用者が希望する時間帯」を特定し、その時間帯に合致する実際の運行便を外部システムから取得する。この動的情報を起点として経路全体を再構成することにより、エンジンの基本構造を維持したまま、パーソナライズされたリアルタイム経路提示を実現している。
- 本アルゴリズムを利用した機能
 - **【FN102】経路検索結果更新**
 - **【FN103】デマンド交通の配車予定時刻を所与とした経路の再検索**
- 本アルゴリズムの計算・処理の詳細
 - ステップ1
 - ◇ 既存の経路検索エンジンを用い、デマンド交通区間に便宜的な仮設ダイヤを適用して、出発地から目的地までの経路を仮出力する。
 - ステップ2
 - ◇ 仮出力により得られたデマンド交通の乗降時刻に基づき、外部のデマンド配車システムへリアルタイム運行状況を照会し、最も条件に近い実際の運行便を特定する。
 - ステップ3
 - ◇ 特定された運行便の時刻を「所与（確定条件）」として固定する。この確定した乗降時刻に基づき、その前後区間（出発地からデマンド乗車地点、および降車地点から最終目的地まで）の経路を、静的データを用いて再検索する。

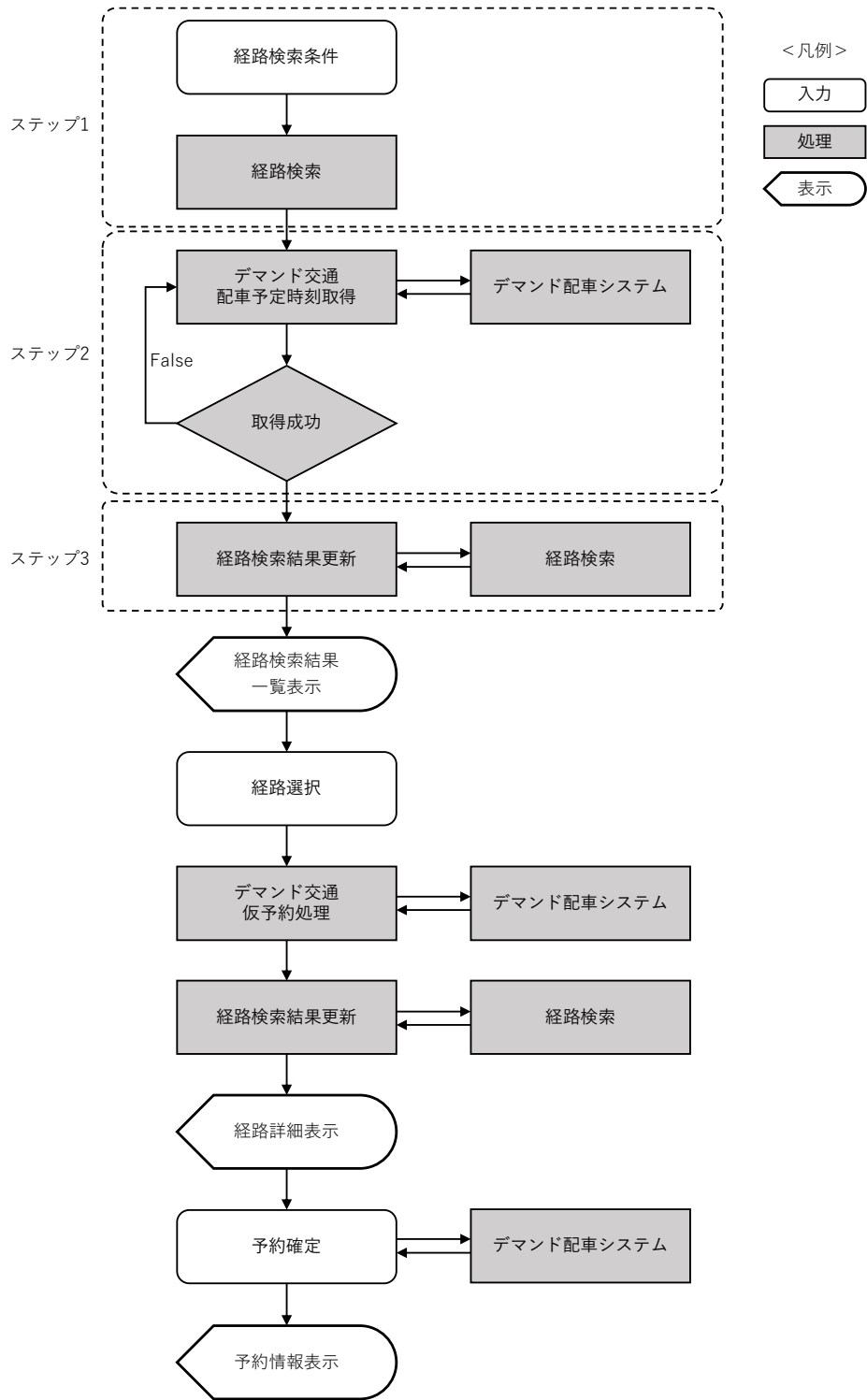


図 3-27 デマンド交通リアルタイム反映アルゴリズム

3-1-2. システムコンポーネント (CO)

3-1-2-a. システムコンポーネント図

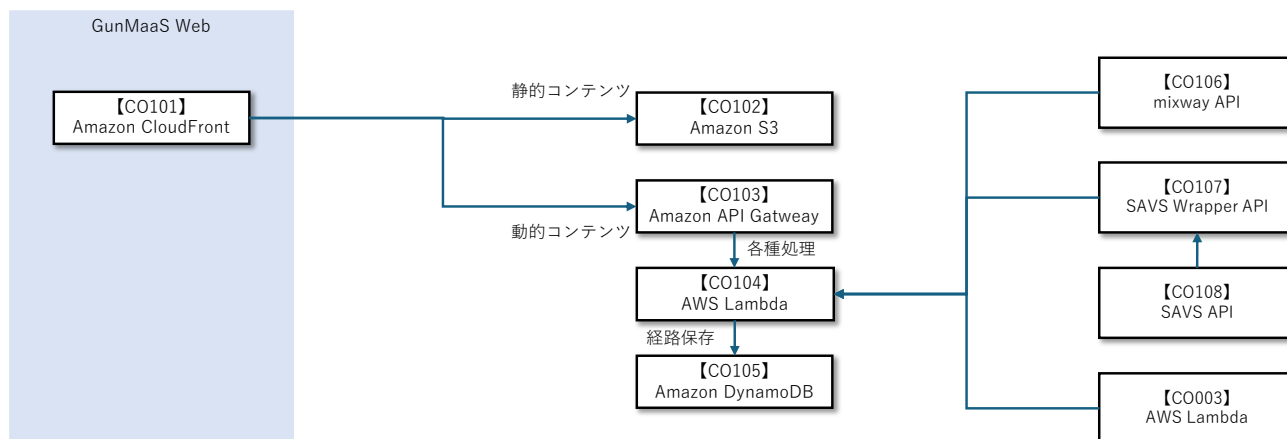


図 3-28 システムコンポーネント図

3-1-2-b. 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
CO101	CDN	Amazon CloudFront	<ul style="list-style-type: none"> ● Web ブラウザ上で動作する、(HTML・CSS・JavaScript による) フロントエンドアプリケーション。API サーバ・その他各種サーバとの通信を行い、経路の表示やデマンド予約、チケットリンク、シェアサイクルのポート表示等を行う
CO102	ファイルサーバ	Amazon S3	<ul style="list-style-type: none"> ● Web サーバのファイル送信を行う。 ● HTML 等のコンテンツを保存する
CO103	API サーバ	Amazon API Gateway	<ul style="list-style-type: none"> ● リクエストのルーティングと変換を行う
CO104	サーバレスコンピューティングサービス	AWS Lambda	<ul style="list-style-type: none"> ● 経路の表示やデマンド予約、チケットリンク、シェアサイクルのポート表示等、動的コンテンツの配信を行う
CO105	データベース	Amazon DynamoDB	<ul style="list-style-type: none"> ● 経路検索、デマンドの予約管理、チケット情報の取得、シェアサイクルのポート情報取得等の動的コンテンツの管理、実行等を行う
CO106	外部サービス	ヴァル研究所 mixway API (SaaS)	<ul style="list-style-type: none"> ● 経路の保存、デマンドの予約管理等を行う
CO107	外部サービス	ヴァル研究所 SAVS Wrapper (SaaS)	<ul style="list-style-type: none"> ● 未来シェア社「SAVS」API を Call する Wrapper API ● デマンド交通の配車予定時刻の取得や、仮予約、仮予約確定を実行する
CO108	外部サービス	未来シェア SAVS API (SaaS)	<ul style="list-style-type: none"> ● 未来シェア社「SAVS」API ● デマンド交通の配車管理等を行う。デマンド交通の配車予定時刻の取得や、仮予約、仮予約確定を実行する
CO109	外部サービス	Google Maps Platform (SaaS)	<ul style="list-style-type: none"> ● Map への描画、道路のルート描画を行う

3-1-3. ハードウェア (HW)

3-1-3-a. ハードウェアアーキテクチャ

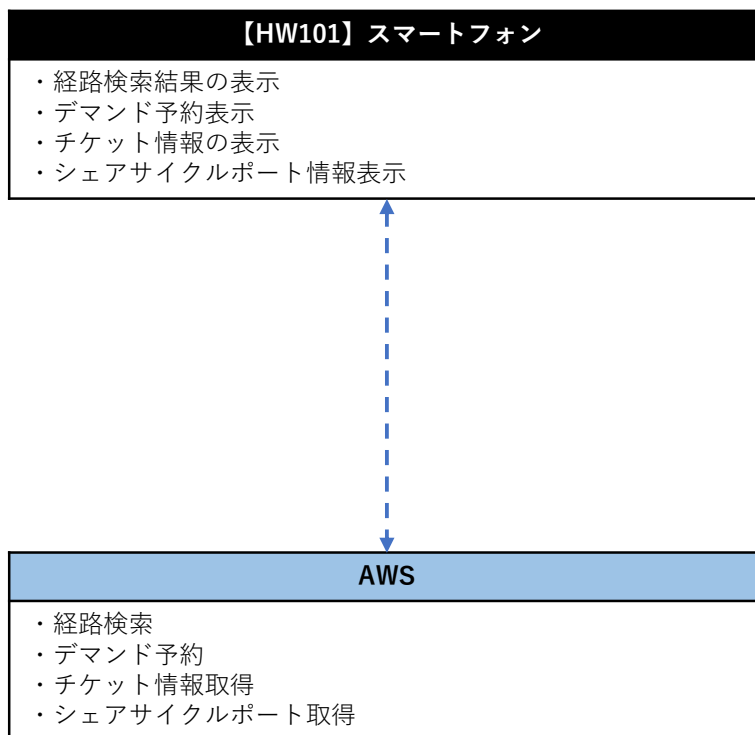


図 3-29 ハードウェアアーキテクチャ

3-1-3-b. ハードウェア一覧

表 3-4 ハードウェア一覧

※朱文字：新規開発・既存改修

ID	種別	対応ブラウザ	用途
HW101	スマートフォン	<ul style="list-style-type: none">● Safari 17 (最新版)● Google Chrome 136 (最新版)	<ul style="list-style-type: none">● GunMaaS の各サービスの利用

3-1-3-c. 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。

【HW101】スマートフォン

- 概要
 - iOS 又は Android を搭載したスマートフォン
 - OS の標準ブラウザによる動作
- ベンダー
 - iOS : Apple
 - Android : Google 他
- 対応ブラウザ
 - Safari 17 (最新版)
 - Google Chrome 136 (最新版)
- イメージ



図 3-30 iPhone

3-1-4. データインターフェース (IF)

3-1-4-a. データアーキテクチャ

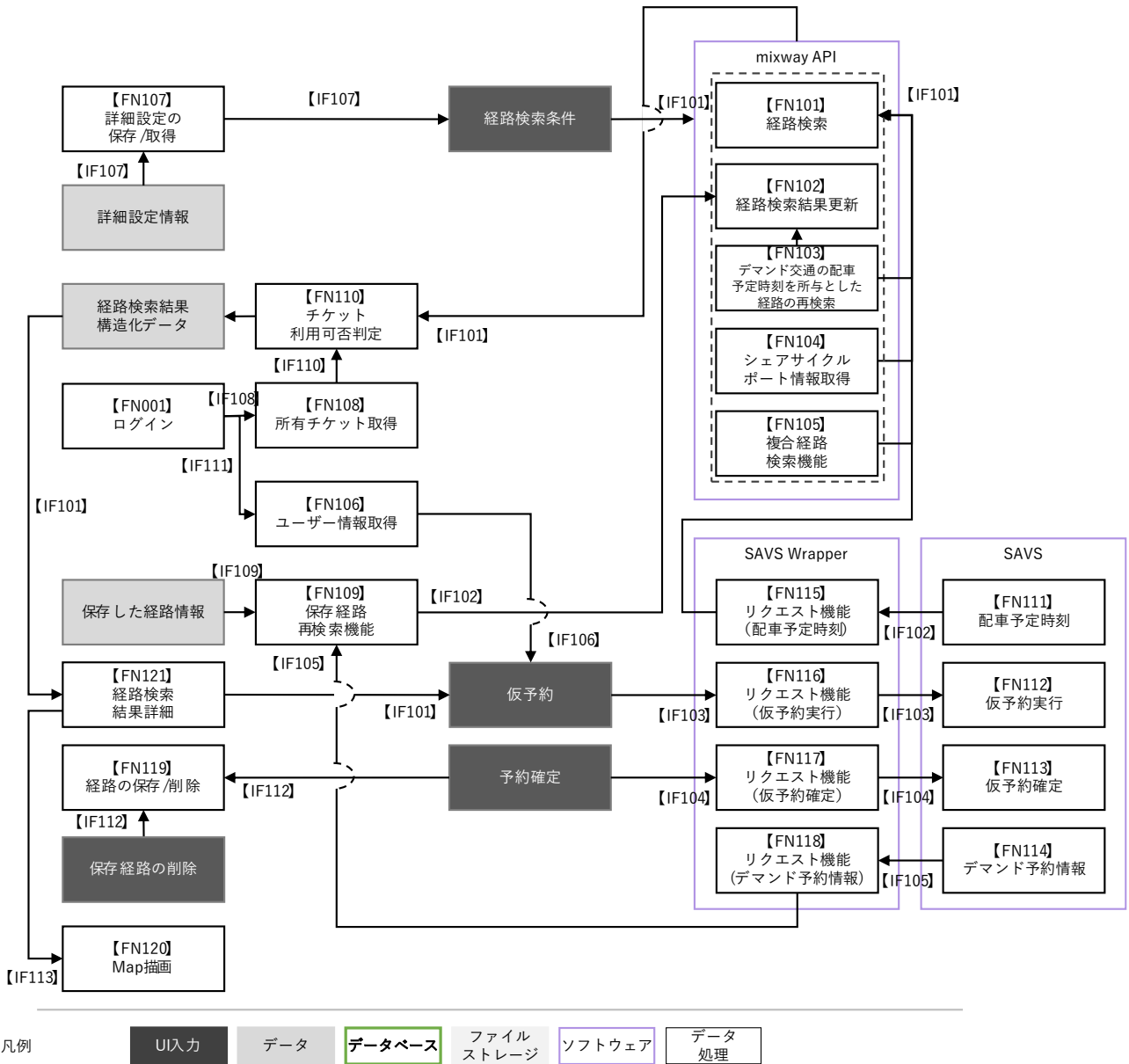


図 3-31 データアーキテクチャ

3-1-4-b. データインターフェース一覧

表 3-5 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
IF101	経路検索 API	FN110	FN101
		FN101	FN102
		FN102	FN103
		FN101	FN104
		FN101	FN105
		FN102	FN109
		FN116	FN110
IF102	配車予定時刻取得 API	FN115	FN111
		FN102	FN115
IF103	仮予約 API	FN112	FN116
IF104	仮予約確定 API	FN113	FN117
IF105	デマンド予約情報取得 API	FN118	FN114
IF106	ユーザ情報取得 API	FN106	FN001
IF107	詳細設定の保存/取得	FN101	FN107
IF108	所有チケット取得 API	FN108	FN001
IF109	保存経路再検索機能 API	FN102	FN109
IF110	チケット利用可否判定	FN110	FN108
IF111	ログイン情報取得 API	FN106	FN001
IF112	経路情報保存/削除 API	FN119	FN119
IF113	Map 描画情報	FN120	FN110

3-1-4-c. 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。

【IF101】経路検索 API

- 概要
 - ヴェアル研究所が提供する複合経路検索 API 「mixway API」
 - 出発地、目的地、日付を指定して、鉄道や路線バス、高速バス、空港連絡バス、飛行機、フェリーを利用した経路検索ができる API
 - 駅名を入力すると、それぞれの候補が返される
 - 公共交通機関のリアルタイム情報（遅延情報）を考慮した経路検索ができる API
 - シェアサイクルポートの情報を検索/取得できる API
 - デマンド交通の配車予定時刻を反映した経路検索ができる API
- 本インターフェースを利用する機能
 - **【FN101】経路検索**
 - **【FN102】経路検索結果更新**
 - **【FN103】デマンド交通の配車予定時刻を所与とした経路の再検索**
 - **【FN104】シェアサイクルポート情報取得**
 - **【FN105】複合経路検索機能**
 - **【FN109】保存経路再検索機能**
 - **【FN121】経路検索結果詳細**
- 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
- 経路検索
 - 概要
 - ◇ 指定した条件で経路を検索する。
 - URI
 - ◇ <https://mixway.ekispert.jp/v1/json/realtime/search/course/extreme>
 - 使用例
 - ◇ https :

//mixway.ekispert.jp/v1/json/realtime/search/course/extreme?key=xxx&viaList=xxxxxxx

- データ詳細
 - ◇ 以下のリファレンスを参照
 - <https://docs-mixway.ekispert.jp/api#tag/realtime-search>

● 経路検索結果更新

- 概要
 - ◇ 指定した条件で経路を更新する。
- URI
 - ◇ <https://mixway.ekispert.jp/v1/json/realtime/course/edit>
- 使用例
 - ◇ <https://mixway.ekispert.jp/v1/json/realtime/course/edit?key=xxx&serializeData=xxxxxxxxxxxx>
- データ詳細
 - ◇ 以下のリファレンスを参照
 - <https://docs-mixway.ekispert.jp/api#tag/realtime-course-edit>

【IF102】配車予定時刻取得 API

- 概要
 - ヴァル研究所が提供する、未来シェア社「SAVS API」を利用するための中間層
 - デマンド交通の配車予定時刻を取得する API
- 本インターフェースを利用する機能
 - 【FN111】配車予定時刻
 - 【FN115】リクエスト機能（配車予定時刻）
- 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

項目	名称	説明	型	必須
アクセスキー	x-val-booking-key	アクセスキー	string	○
言語指定	lang	言語指定 (ja,en)	string	-

● 配車予定時刻取得

- 概要
 - ◇ 指定した内容で配車予定時刻を取得する。
- URI
 - ◇ <https://demand-bff.ekispert.jp/xxx/demand-estimations>
- 使用例
 - ◇ <https://demand-bff.ekispert.jp/xxx/demand-estimations>
- リクエストパラメータ

項目	名称	説明	型	必須
モビリティ ID	mobilityId	モビリティ ID	string	○
乗車地点の緯度 経度	pickUpPosition	乗車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string	○
降車地点の緯度 経度	dropOffPosition	降車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string	○
乗車人数(大人)	seatCountForAdult	乗車人数(大人)	integer	○
乗車人数(小人)	seatCountForChild	乗車人数(小人)	integer	○
乗車/降車希望 日時	datetime	乗車/降車希望日時	string(\$date-time)	○
乗降車指定	datetimeFor	乗降車指定 (希望日時を乗車時刻 or 降車時刻どちらで指定するか)	string	○
相乗り可否	shareable	相乗り可否	boolean	-

- レスポンスパラメータ

項目	名称	説明	型
モビリティ ID	mobilityId	モビリティ ID	string
推定乗車日時	estimatePickUpTime	推定乗車日時	string
推定降車日時	estimateDropOffTime	推定降車日時	string
車両 ID	savId	車両 ID	string
乗車地点の緯度 経度	pickUpPosition	乗車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string
降車地点の緯度 経度	dropOffPosition	降車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string
乗車人数	seatCountForAdult	乗車人数(大人)	integer
	seatCountForChild	乗車人数(小人)	integer

相乗り可否	shareable	相乗り可否	boolean
-------	-----------	-------	---------

➤ レスポンスサンプル

```
{
  "mobilityId": "maebashi001",
  "estimatePickUpTime": "2025-10-01T15:10:00+09:00",
  "estimateDropOffTime": "2025-10-01T15:45:00+09:00",
  "savId": 1,
  "pickUpPosition": "36.412086:139.160391",
  "dropOffPosition": "36.413474:139.158737",
  "seatCountForAdult": 1,
  "seatCountForChild": 0,
  "shareable": true
}
```

【IF103】 仮予約 API

- 概要
 - ヴァル研究所が提供する、未来シェア社「SAVS API」を利用するための中間層
 - デマンド交通の仮予約を行う API
- 本インターフェースを利用する機能
 - 【FN112】 仮予約実行
 - 【FN116】 リクエスト機能（仮予約実行）
- 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

項目	名称	説明	型	必須
アクセスキー	x-val-booking-key	アクセスキー	string	○

- 仮予約
 - 概要
 - ◇ 指定した内容で配車の仮予約を行う。

- URI
 - ◇ <https://demand-bff.ekispert.jp/xxx/demands>
- 使用例
 - ◇ <https://demand-bff.ekispert.jp/xxx/demands>
- リクエストパラメータ

項目	名称	説明	型	必須
乗車地点の名称	pickUpLocationName	乗車地点の名称	string	-
乗車地点の緯度経度	pickUpPosition	乗車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string	-
降車地点の名称	dropOffLocationName	降車地点の名称	string	-
降車地点の緯度経度	dropOffPosition	降車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string	-
乗車人数	seatCountForAdult	乗車人数(大人)	integer	○
	seatCountForChild	乗車人数(小人)	integer	○
乗車/降車希望日時	datetime	乗車/降車希望日時	string(\$date-time)	○
乗降車指定	datetimeFor	乗降車指定 (希望日時を乗車時刻 or 降車時刻どちらで指定するか)	string	○

- レスポンスパラメータ

項目	名称	説明	型
予約 ID	id	予約 ID(配車システムにて採番された ID)	Integer
予約状態	status	予約状態(仮予約済み、配車中、乗車中、輸送完了、仮予約キャンセル、確定後キャンセル)	string
乗車地点の名称	pickUpLocationName	乗車地点の名称	string
乗車地点の緯度経度	pickUpPosition	乗車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string
降車地点の名称	dropOffLocationName	降車地点の名称	string
降車地点の緯度経度	dropOffPosition	降車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string
乗車人数	seatCountForAdult	乗車人数(大人)	integer

	seatCountForChild	乗車人数(小人)	integer
支払予定金額	price	支払予定金額	integer
推定乗車日時	latestEstimatePickUpTime	推定乗車日時	string
推定降車日時	latestEstimateDropOffTime	推定降車日時	string
車両 ID	savId	車両 ID	string
車両名称	savName	車両名称	string
車両の現在位置の緯度 経度	savPosition	車両の現在位置の緯度経度 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string

➤ レスポンスサンプル

```
{
  "id": 1,
  "status": "ASSIGNED",
  "pickUpLocationName": "(大胡 001) ふれあい農園",
  "pickUpPosition": "36.412086 : 139.160391",
  "dropOffLocationName": "(大胡 002) 消防 16 分団 1 部",
  "dropOffPosition": "36.413474 : 139.158737",
  "seatCountForAdult": 1,
  "seatCountForChild": 0,
  "price": 300,
  "latestEstimatePickUpTime": "2025-10-01T15 : 10 : 00+09 : 00",
  "latestEstimateDropOffTime": "2025-10-01T15 : 45 : 00+09 : 00",
  "savId": 1,
  "savName": "1 号車",
  "savPosition": "36.413072 : 139.156728"
}
```

【IF104】仮予約確定 API

- 概要
 - ヴェアル研究所が提供する、未来シェア社「SAVS API」を利用するための中間層
 - デマンド交通の仮予約を確定する API
- 本インターフェースを利用する機能
 - 【FN113】仮予約確定
 - 【FN117】リクエスト機能（仮予約確定）仮予約確定
- 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

項目	名称	説明	型	必須
アクセスキー	x-val-booking-key	アクセスキー	string	○
予約 ID	demandId	予約 ID	Integer	○

- 仮予約確定
 - 概要
 - ◇ 指定した予約の確定を行う。
 - URI
 - ◇ <https://demand-bff.ekispert.jp/xxx/demands/{demandid}/approve>
 - 使用例
 - ◇ <https://demand-bff.ekispert.jp/xxx/demands/1/approve>
 - リクエストパラメータ

項目	名称	説明	型	必須
モビリティ ID	mobilityId	モビリティ ID	string	○
ユーザ ID	userId	ユーザ ID	string	○

- レスポンスパラメータ

項目	名称	説明	型
予約 ID	id	予約 ID(配車システムにて採番された ID)	Integer
予約状態	status	予約状態(仮予約済み、配車中、乗車中、輸送完了、仮予約キャンセル、確定後キャンセル)	string
乗車地点の名称	pickUpLocationName	乗車地点の名称	string
乗車地点の緯度経度	pickUpPosition	乗車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string
降車地点の名称	dropOffLocationName	降車地点の名称	string
降車地点の緯度経度	dropOffPosition	降車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string

乗車人数	seatCountForAdult	乗車人数(大人)	integer
	seatCountForChild	乗車人数(小人)	integer
支払予定金額	price	支払予定金額	integer
推定乗車日時	latestEstimatePickUpTime	推定乗車日時	string
推定降車日時	latestEstimateDropOffTime	推定降車日時	string
車両 ID	savId	車両 ID	string
車両名称	savName	車両名称	string
車両の現在位置の緯度 経度	savPosition	車両の現在位置の緯度経度 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string

➤ レスポンスサンプル

```
{
  "id": 1,
  "status": "PICKING_UP",
  "pickUpLocationName": "(大胡 001) ふれあい農園",
  "pickUpPosition": "36.412086 : 139.160391",
  "dropOffLocationName": "(大胡 002) 消防 16 分団 1 部",
  "dropOffPosition": "36.413474 : 139.158737",
  "seatCountForAdult": 1,
  "seatCountForChild": 0,
  "price": 300,
  "latestEstimatePickUpTime": "2025-10-01T15 : 10 : 00+09 : 00",
  "latestEstimateDropOffTime": "2025-10-01T15 : 45 : 00+09 : 00",
  "savId": 1,
  "savName": "1 号車",
  "savPosition": "36.413072 : 139.156728"
}
```

【IF105】デマンド予約情報取得 API

- 概要
 - ヴェル研究所が提供する、未来シェア社「SAVS API」を利用するための中間層
 - デマンド交通の予約情報を取得する API
- 本インターフェースを利用する機能
 - 【FN109】保存経路再検索機能
 - 【FN118】リクエスト機能 (デマンド予約情報取得)
- 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
- 共通リクエストパラメータ

項目	名称	説明	型	必須
アクセスキー	x-val-booking-key	アクセスキー	string	○
モビリティ ID	mobilityid	モビリティ ID	string	○
ユーザ ID	userId	ユーザ ID	string	-

- デマンド予約情報取得
 - 概要
 - ◇ 指定した予約の情報を取得する。
 - URI
 - ◇ <https://demand-bff.ekispert.jp/xxx/demands/{id}>
 - 使用例
 - ◇ <https://demand-bff.ekispert.jp/xxx/demands/1>
 - リクエストパラメータ

項目	名称	説明	型	必須
予約 ID	id	予約 ID(配車システムにて採番された ID)	Integer	○

- レスポンスパラメータ

項目	名称	説明	型
予約 ID	id	予約 ID(配車システムにて採番された ID)	Integer
予約状態	status	予約状態(仮予約済み、配車中、乗車中、輸送完了、仮予約キャンセル、確定後キャンセル)	string
乗車地点の名称	pickUpLocationName	乗車地点の名称	string
乗車地点の緯度経度	pickUpPosition	乗車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string
降車地点の名称	dropOffLocationName	降車地点の名称	string
降車地点の緯度経度	dropOffPosition	降車地点の緯度経度。 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string

乗車人数	seatCountForAdult	乗車人数(大人)	integer
	seatCountForChild	乗車人数(小人)	integer
支払予定金額	price	支払予定金額	integer
推定乗車日時	latestEstimatePickUpTime	推定乗車日時	string
推定降車日時	latestEstimateDropOffTime	推定降車日時	string
車両 ID	savId	車両 ID	string
車両名称	savName	車両名称	string
車両の現在位置の緯度 経度	savPosition	車両の現在位置の緯度経度 (フォーマット: "{緯度}:{経度}") 座標は世界測地系	string

➤ レスポンスサンプル

```
{
  "id": 1,
  "status": "PICKING_UP",
  "pickUpLocationName": "(大胡 001) ふれあい農園",
  "pickUpPosition": "36.412086 : 139.160391",
  "dropOffLocationName": "(大胡 002) 消防 16 分団 1 部",
  "dropOffPosition": "36.413474 : 139.158737",
  "seatCountForAdult": 1,
  "seatCountForChild": 0,
  "price": 300,
  "latestEstimatePickUpTime": "2025-10-01T15 : 10 : 00+09 : 00",
  "latestEstimateDropOffTime": "2025-10-01T15 : 45 : 00+09 : 00",
  "savId": 1,
  "savName": "1 号車",
  "savPosition": "36.413072 : 139.156728"
}
```

【IF106】 ユーザ情報取得 API

- 概要
 - ユーザ情報を取得する
- 本インターフェースを利用する機能
 - 【FN106】 ユーザ情報取得
- 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
- 共通リクエストパラメータ

項目	名称	説明	型	必須
GunMaaS のトークン	authorization	GunMaaS のトークン	string	○

- ユーザ情報取得
 - 概要
 - ◇ トークンを元にユーザの情報を取得する
 - URI
 - ◇ <https://xxx.wallabeest.net/sp/g3m2/>
 - 使用例
 - ◇ <https://xxx.wallabeest.net/sp/g3m2/bridge/xxx/account>
 - レスポンスパラメータ

項目	名称	説明	型
メールアドレス	mailAddress	メールアドレス	string
姓	lastName	姓	string
名	firstName	名	string
電話番号	phoneNumberInfo	電話番号	string

- レスポンスサンプル

```
{
  "timestamp": 1657769719,
  "mailAddress": "test@ringopass.co.jp",
  "lastName": "yamada",
  "firstName": "taro",
  "phoneNumberInfo": {
    "phoneNumber": "9012340000",
    "phoneCountryCode": "81"
  }
}
```

【IF107】 詳細設定の保存/取得

- 概要
 - 探索条件を記録して再利用可能にする
- 本インターフェースを利用する機能

➤ 【FN107】 詳細設定の保存/取得

● データ詳細

名称	説明	型
rtWebCondition	詳細設定の文字列	string
demandAdultCount	デマンド交通予約時の設定する人数の文字列(大人)	string
demandChildCount	デマンド交通予約時の設定する人数の文字列(小人)	string

【IF108】 所有チケット取得 API

● 概要

- ユーザが持っているチケット情報を取得する

● 本インターフェースを利用する機能

- 【FN108】 所有チケット取得

● 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
- 共通リクエストパラメータ

項目	名称	説明	型	必須
GunMaaS のトークン	authorization	GunMaaS のトークン	string	○

● 所有チケット取得

- 概要
 - ◇ 所有しているチケット一覧を返却する
- URI
 - ◇ <https://xxx.wallabeest.net/sp/xxx/tickets>
- 使用例
 - ◇ https://xxx.wallabeest.net/sp/xxx/tickets?status_list=1
- リクエストパラメータ

項目	名称	説明	型	必須
チケットの状態	status_list	チケットの状態	string	○

- レスポンスパラメータ

項目	名称	説明	型
チケットの ID	id	チケットの ID	string
チケットの名称	name	チケットの名称	string

➤ レスポンスサンプル

```
{
  "success": true,
  "ticket": [
    {
      "id": "PACKAGEID-1001",
      "name": "パッケージ名その 1"
    }
  ]
}
```

【IF109】 保存経路再検索機能 API

- 概要
 - 保存した経路を利用して経路を再検索する
- 本インターフェースを利用する機能
 - 【FN109】 保存経路再検索機能
- 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
- 保存経路情報取得
 - 概要
 - ◇ 保存した経路文字列を出力する
 - ◇ デマンドの予約 ID を出力する
 - URI
 - ◇ <https://xxx.com/api/routeHistory>
 - 使用例
 - ◇ <https://xxx.com/api/routeHistory?serviceld=MaeMaaS&token=88f69421-5cba-4e86-933f-bff85bb4a14f&id=e6e1ce70-6b8d-4fe6-a2b3-6b8b1690e21a>
 - リクエストパラメータ

デマンド型交通リアルタイム経路検索システム システム設計書

項目	名称	説明	型	必須
ユーザトークン	token	ユーザトークン	string	○
GunMaaS からのアクセスであることを示す識別子	serviceId	GunMaaS からのアクセスであることを示す識別子	string	○
保存した経路の識別子	id	保存した経路の識別子	string	○

➤ レスポンスパラメータ

項目	名称	説明	型
経路検索文字列	params	経路検索文字列 ※経路検索文字列の詳細はドキュメントを参照 (https://docs-mixway.ekispert.jp/api#tag/search/operation/search/course/extreme)	string
モビリティ ID	mobilityId	モビリティ ID	string
デマンドの予約 ID	id	デマンドの予約 ID	string

➤ レスポンスサンプル

```
{
  "success": true,
  "items": [
    {
      "params": {
        "date": "20241002",
        "maxWalkDistanceCycle": 800,
        "maxWalkDistanceLocalBus": 1500,
        "minCarDistance": 0,
        "searchType": "departure",
        "maxCarDistance": 0,
        "minCycleDistance": 0,
        "addStop": true,
        "maxWalkDistanceDemand": 1000,
        "viaList": "36.49691319999999,139.0176693 : 36.468193,139.0388488",
        "routeSectionMode": "operationLine",
        "maxWalkDistance": 1500,
        "answerCount": 10,
        "toName": "セブン-イレブン 渋川北橋町真壁店",
        "conditionDetail": "T1111111111319 : F2321121120000 : A23121141 : ",
      }
    }
  ]
}
```

```

        "maxCycleDistance" : 0,
        "fromName" : "八崎口",
        "time" : "1149",
        "demandCorporationBind" : "MAERUNRUNBUS : MAEFURUSATOBUS :
MAEJONANAOZORABUS : SHIBUKAWAHOKKITSU : SHIBUKAWAONOGAMI",
        "addTimeTransfer" : true,
        "cycleCorporationBind" : "DOCOMO : HELLOCYCLING : COGBE",
        "key" : "UUGGcEKWWDAnGnSW",
        "cycleAllowCorporationList" : "HELLOCYCLING : COGBE"
    },
    "reservations" : [
        {
            "id" : 3403,
            "mobilityId" : "maebashi001"
        }
    ],
    "serviceld" : "MaeMaaS"
}
]
}

```

【IF110】 チケット利用可否判定

- 概要
 - ユーザが持っているチケット情報が経路に適用できるか判定する
- 本インターフェースを利用する機能
 - 【FN110】 チケット利用可否判定
- データ詳細

名称	説明	型
id	チケットの ID	string
name	チケットの名称	string

【IF111】 ログイン情報取得 API

- 概要
 - GunMaaS のログイン情報を取得できるようにする
- 本インターフェースを利用する機能
 - 【FN106】 ユーザ情報取得
- 共通仕様
 - プロトコル

- ◇ HTTPS
 - メソッド
 - ◇ PUT
 - レスポンスデータ形式
 - ◇ JSON
 - 文字コード
 - ◇ UTF-8
- ログイン情報取得
 - 概要
 - ◇ GunMaaS のユーザ ID とアクセストークンで、ユーザトークンを取得する
 - URI
 - ◇ `https://xxx.jre-maas.com/api/user/management`
 - 使用例
 - ◇ `https://xxx.jre-maas.com/api/user/management`
 - リクエストパラメータ

項目	名称	説明	型	必須
GunMaaS のユーザ ID	id	GunMaaS のユーザ ID	string	○
GunMaaS のトークン	token	GunMaaS のトークン	string	○

- レスポンスパラメータ

項目	名称	説明	型
ユーザトークン	token	ユーザトークン	string

- レスポンスサンプル

```
{
  "success": true,
  "token": "88f69421-5cba-4e86-933f-bff85bb4a14f"
}
```

【IF112】経路情報保存/削除 API

- 概要
 - 再検索に必要な経路の情報を保存する。
- 本インターフェースを利用する機能
 - **【FN119】経路の保存/削除**
- 共通仕様
 - プロトコル

- ◇ HTTPS
- メソッド
 - ◇ PUT
- レスポンスデータ形式
 - ◇ JSON
- 文字コード
 - ◇ UTF-8
- 経路情報保存/削除
 - 概要
 - ◇ 下記を保存する
 - 保存する経路を一意にする識別子
 - 経路検索文字列
 - デマンド交通の予約 ID
 - URI
 - ◇ <https://xxx.com/api/routeHistory>
 - 使用例
 - ◇ <https://xxx.com/api/routeHistory>
 - リクエストパラメータ

項目	名称	説明	型	必須
ユーザトークン	token	ユーザトークン	string	○
GunMaaS からのアクセスであることを示す識別子	serviceld	GunMaaS からのアクセスであることを示す識別子	string	○
保存した経路の識別子	id	保存した経路の識別子 ※削除時のみ指定	string	-
経路検索文字列	params	経路検索文字列 ※保存時のみ ※経路検索文字列の詳細はドキュメントを参照 (https://docs-mixway.ekispert.jp/api#tag/search/operation/search/course/extreme)	string	-
モビリティ ID	mobilityId	モビリティ ID ※保存時のみ	string	-
デマンドの予約 ID	demandId	デマンドの予約 ID ※保存時のみ	string	-

- レスポンスパラメータ

項目	名称	説明	型
保存した経路の識別子	id	保存した経路の識別子 ※保存時のみ	string

➤ レスポンスサンプル

```
{
  "success": true,
  "id": " e6e1ce70-6b8d-4fe6-a2b3-6b8b1690e21a "
}
```

【IF113】 Map 描画情報

- 概要
 - 経路検索結果の徒歩・自動車区間は Map 側、公共交通区間は経路検索側から描画するルートを取得し、Map に渡す。
 - 経路検索に含まれるシェアサイクルポートと、その周辺のシェアサイクルポートを Map に渡す。
- 本インターフェースを利用する機能
 - 【FN120】 Map 描画
- データ詳細
 - 概要
 - ◇ 経路検索結果のルート情報
 - ◇ シェアサイクルのポート情報
 - データ詳細
 - ◇ 以下のリファレンスを参照
 - <https://docs-mixway.ekispert.jp/api#tag/shape/operation/course/shape>
 - <https://docs-mixway.ekispert.jp/api#tag/port/operation/geo/point>

3-1-5. ユーザインターフェース (UI)

3-1-5-a. 画面遷移図

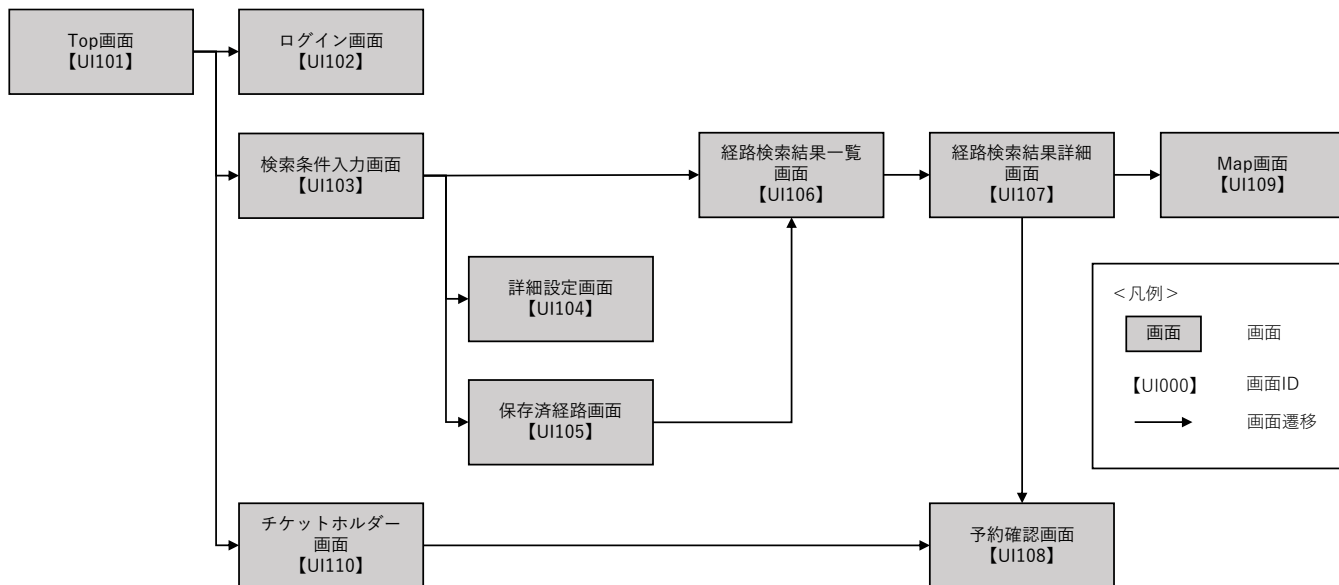


図 3-32 画面遷移図

3-1-5-b. ユーザーインターフェース一覧

表 3-6 リアルタイム経路検索の GunMaaS Web サイト画面遷移一覧

※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	Top 画面	<ul style="list-style-type: none"> ● システムの機能が一覧化されており、各機能呼び出すことができる画面 	-
UI102	ログイン画面	<ul style="list-style-type: none"> ● ID/パスワードを入力することでログインする画面 	-
UI103	検索条件入力画面	<ul style="list-style-type: none"> ● 出発地/目的地、日時を指定する画面 	FN101
UI104	詳細設定画面	<ul style="list-style-type: none"> ● 経路検索の検索条件やデマンド交通の人数を設定する画面 	FN107
UI105	保存済み経路画面	<ul style="list-style-type: none"> ● 保存した経路の確認、保存した経路からの再検索を行う画面 	FN109
UI106	経路検索結果一覧画面	<ul style="list-style-type: none"> ● 経路検索した結果を表示する画面 ● 所有する利用可能なチケットの有無を表示する画面 	FN101 FN102 FN104 FN105 FN106 FN108 FN110
UI107	経路検索結果詳細画面	<ul style="list-style-type: none"> ● 選択した経路の詳細を表示する画面 ● 経路にデマンド交通が含まれる場合、仮予約を実行する画面 ● 仮予約実行後の結果を反映した経路を表示する画面 ● 保存済み経路の場合、予約済みのデマンド交通の最新情報を反映した経路を表示する画面 	FN102 FN109 FN116 FN118 FN121
UI108	予約確認画面	<ul style="list-style-type: none"> ● デマンドモビリティの予約情報を表示する 	FN118
UI109	Map 画面	<ul style="list-style-type: none"> ● 経路を表示する画面 ● 出発地/目的地周辺のシェアサイクルポートと貸出/返却可能台数を表示する画面 	FN104 FN120
UI110	チケットホルダー画面	<ul style="list-style-type: none"> ● 所有チケットを表示する画面 ● デマンド交通の予約を表示する画面 	-

【UI101】 Top 画面



図 3-33 Top 画面

- 概要
 - 既存システムのメニューを表示する
 - ログインしていない場合は【UI001】ログイン/新規登録選択画面に遷移する
- 本画面から利用する機能
 - -

【UI102】 ログイン画面

JRE ID

「GunMaaS」からアクセスしています。

ログイン

パスキーでログインする

JRE ID 必須

パスワード 必須

ログインする

[パスワードを忘れた ㊦](#)

[JRE IDを忘れた ㊦](#)

[ログインできない ㊦](#)

図 3-34 ログイン画面

- 概要
 - JRE ID のログイン画面
- 本画面から利用する機能
 - -

【UI103】 検索条件入力画面

The screenshot shows the search condition input screen for Gunmaas. At the top left is the Gunmaas logo with the tagline 'INNOVATE YOUR TRIPS'. At the top right are the language options 'JP | EN'. The main form has three input fields: '出発地' (Departure) with '現在地' (Current Location) selected, '目的地' (Destination) with '目的地' (Destination) entered, and '日時' (Date/Time) with 'いますぐ' (Now) selected. There is a '経由地を追加' (Add Stopover) button between the destination and date/time fields. Below the input fields are four buttons: '出発' (Departure), '到着' (Arrival), '始発' (Start), and '終電' (Last Train). Below these are two more buttons: '詳細設定' (Detailed Settings) and '保存済み経路' (Saved Routes). At the bottom is a large orange '検索' (Search) button. At the very bottom, it says 'Powered by JR東日本'.

図 3-35 検索条件入力画面

- 概要
 - 出発地/目的地、日時を指定する
 - 詳細設定に遷移する
 - 保存済み経路に遷移する
- 本画面から利用する機能
 - 【FN101】 経路検索

【UI104】 詳細設定画面



図 3-36 詳細設定画面

- 概要
 - 詳細な経路検索条件を設定する
 - デマンド交通の人数を設定する
- 本画面から利用する機能
 - 【FN107】 詳細設定の保存/取得

【UI105】保存済み経路画面



図 3-37 保存済み経路画面

- 概要
 - 保存した経路を表示する
 - 表示された経路を選択した場合、【UI106】経路検索結果一覧画面に遷移する
- 本画面から利用する機能
 - 【FN109】保存経路再検索機能

【UI106】経路検索結果一覧画面

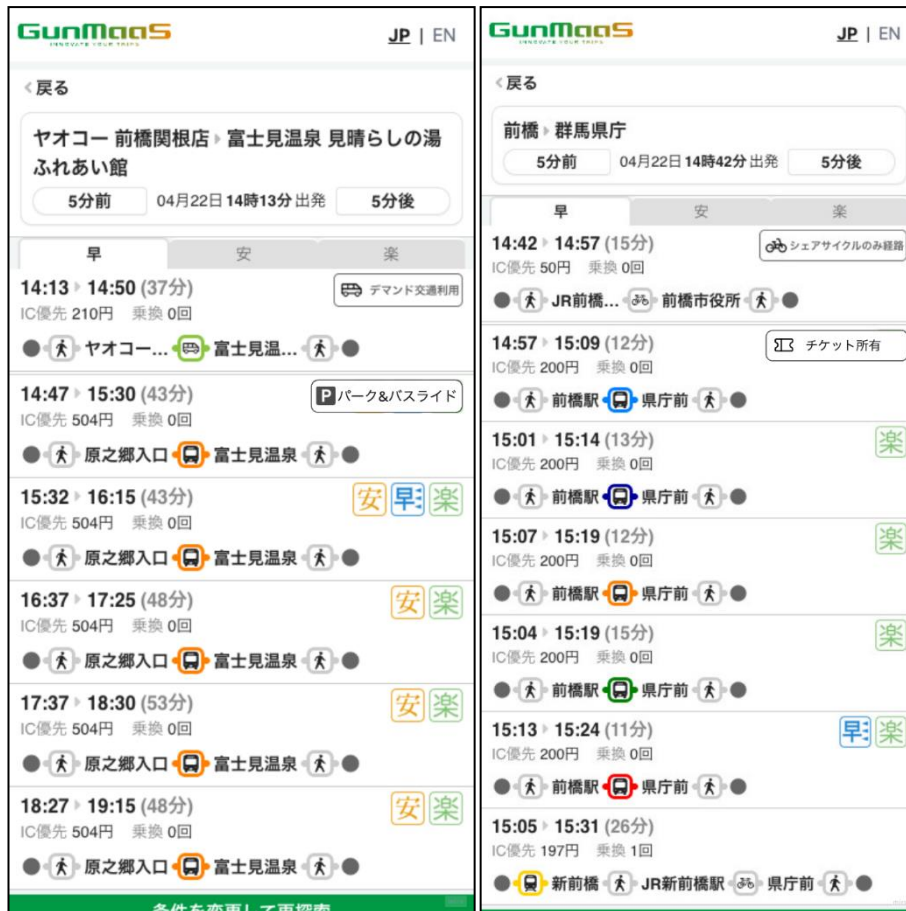


図 3-38 経路検索結果一覧画面

- 概要
 - 経路検索した結果を一覧表示する
 - デマンド交通の配車予定時刻を反映した経路を表示する
 - 所有するチケットの利用可否を表示する
- 本画面から利用する機能
 - 【FN101】経路検索
 - 【FN102】経路検索結果更新
 - 【FN104】シェアサイクルポート情報取得
 - 【FN105】複合経路検索機能
 - 【FN106】ユーザ情報取得
 - 【FN108】所有チケット取得
 - 【FN110】チケット利用可否判定

【UI107】 経路検索結果詳細画面

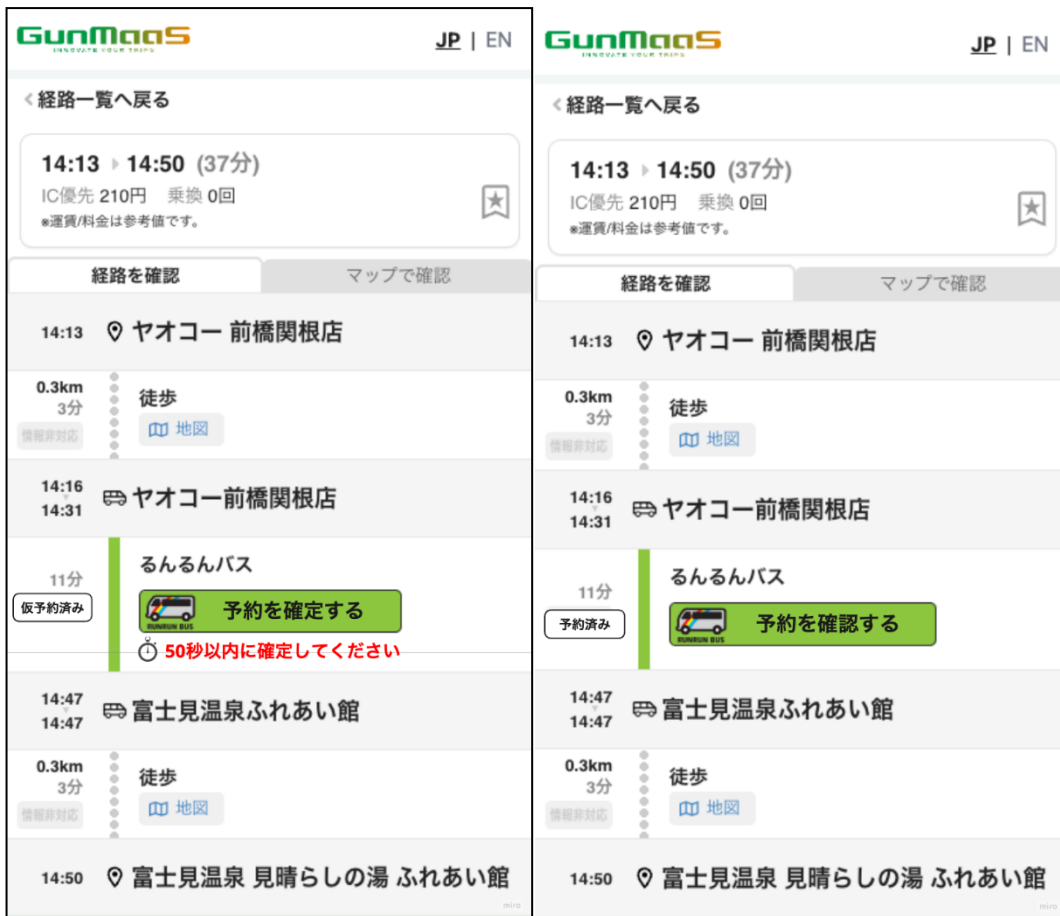


図 3-39 経路検索結果詳細画面（左図：予約を「確定」するボタン（予約確定前） 右図：予約を「確認」するボタン（予約確定後））

- 概要
 - 【UI106】 経路検索結果一覧画面から選択した経路の詳細を表示する
 - 経路にデマンド交通が含まれる場合、仮予約を実行する
 - 仮予約実行後の結果を反映した経路を表示する
 - 保存済み経路の場合、予約済みのデマンド交通の最新情報を反映した経路を表示する
 - 経路にシェアサイクルが含まれる場合、貸出/返却可能台数を表示する
- 本画面から利用する機能
 - 【FN102】 経路検索結果更新
 - 【FN109】 保存経路再検索機能
 - 【FN116】 リクエスト機能（仮予約実行）
 - 【FN118】 リクエスト機能（デマンド予約情報取得）
 - 【FN119】 経路の保存/削除

【UI108】 予約確認画面



図 3-40 予約確認画面

- 概要
 - デマンド交通の予約情報を表示する
- 本画面から利用する機能
 - 【FN118】 リクエスト機能（デマンド予約情報取得）

【UI109】 Map 画面

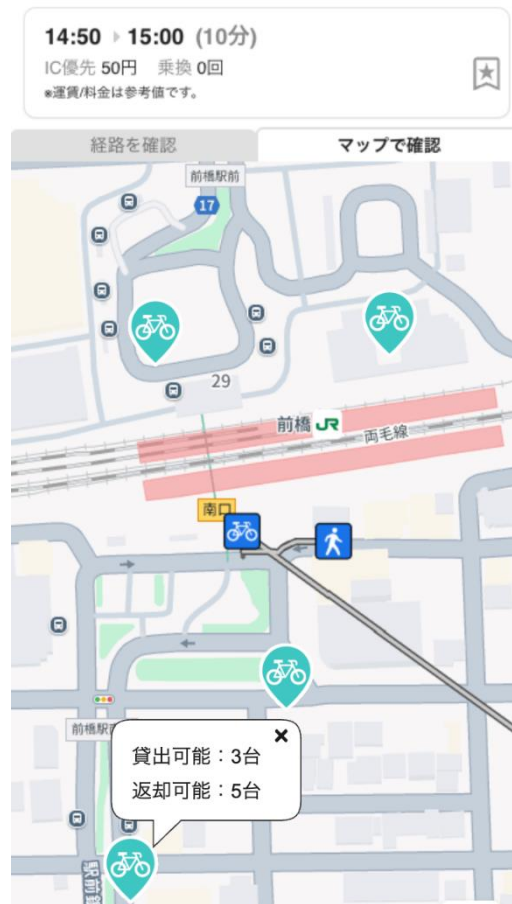


図 3-41 Map 画面

- 概要
 - 経路を表示する
 - 出発地/目的地周辺のシェアサイクルポートと貸出/返却可能台数を表示する
- 本画面から利用する機能
 - 【FN104】シェアサイクルポート情報取得

【UI110】 チケットホルダー画面



図 3-42 チケットホルダー画面

- 概要
 - 所有チケットを表示する
 - デマンド交通の予約を表示する
 - 予約確認画面に遷移する
- 本画面から利用する機能
 - -

3-2. 開発するシステム：非機能要件（NF）

3-2-1. 非機能要件一覧

表 3-7 非機能要件一覧

カテゴリ	ID	非機能項目	要件詳細
性能・拡張性	NF001	スケーラビリティと拡張性	<ul style="list-style-type: none"> システムは将来のユーザ数、データ量の増加によるCPU やメモリの増強や新機能の追加に対応できるように設計する。
運用・保守性	NF002	データのバックアップ	<ul style="list-style-type: none"> GunMaaS の社会的影響力を考慮し、適切な頻度および保存箇所にデータのバックアップの取得や運用保守作業を行う。

3-2-2. 非機能要件の詳細

【NF001】 スケーラビリティと拡張性

- 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
 - スパイクアクセス時に柔軟に対応できることと、単一のサーバに依存しないことからサーバダウン等の障害を回避できるため、サーバレスアーキテクチャである AWS Lambda を用いる。また AWS では、新機能を追加する際に、容易にリソースの追加や Amazon API Gateway 用いた境構築を行うことができる。

【NF002】 データのバックアップ

- 概要
 - 複数のアベイラビリティゾーンにデータを保存すること。また GunMaaS は社会的な影響が大きいシステムであることからデータのバックアップは手動ではなく日次より自動でバックアップを取得する。
- 設定理由
 - 特定のサーバが障害によって失われたとしても、複数のサーバへ保存することにより耐障害性を高めることができるため、複数のアベイラビリティゾーンにまたがってデータのバックアップを取得する。GunMaaS はデータのフルバックアップを行っており、障害発生時はバックアップデータから必要に応じて復旧作業を行う。

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	デマンド交通ポート情報	JSON	前橋市（未来シェア）	FN101

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・作成行うデータを**朱文字**で示す。

【DT001】 デマンド交通ポート情報

- 概要
 - 前橋市で運行しているデマンド交通（「ふるさとバス」「るんるんバス」「城南あおぞら号」）のポート情報
- データ定義

表 4-2 デマンド交通ポート情報のサンプル

No	項目名	内容	データ型
1	landmarkId	ポート ID	integer
2	Names	ポート名の配列	array
3	names.realName	ポート名	string
4	names.language	ポート名言語	string
5	Position	ポート位置	string
6	position.lat	ポート緯度	number(\$double)
7	position.lng	ポート経度	number(\$double)
8	categoryIds	カテゴリ ID（エリアを区別する ID）	integer

- データ形式
 - JSON
- 出所
 - 未来シェア社 SAVS より API で取得
- 変換方法
 - 処理内容・手順
 - ◇ 取得したデータを Excel 化し、手入力データ編集
 - 利用したソフトウェア
 - ◇ Excel

5. 用語集

用語	定義・説明
ICRW	「IC リーダライタ」の略称で、IC カードのデータを読み書きするための装置。
ABT	「Account Based Ticketing」の略称で、アカウント単位でチケットを管理する仕組み。
GunMaaS	群馬県新モビリティサービス推進協議会が運営する MaaS の Web サービス。



**デマンド型交通リアルタイム経路検索システム 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: 東日本旅客鉄道株式会社
株式会社ヴァル研究所
株式会社ケー・シー・エス