

# 共同使用契約に基づくカーシェアリングシステム システム設計書

---



# 目次

---

---

1. 開発スコープ .....	- 1 -
1-1. 概要 .....	- 1 -
1-2. 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. 共同使用契約に基づくカーシェアリングシステム：機能要件（FN/SL/AL/CO/HW/IF/UI） .....	- 8 -
2-1. システム機能（FN） .....	- 8 -
2-1-1. システムアーキテクチャ .....	- 8 -
2-1-2. システム機能一覧 .....	- 9 -
2-1-3. システム機能の詳細 .....	- 12 -
2-1-4. ソフトウェア・ライブラリ（SL）の詳細 .....	- 60 -
2-1-5. 数理モデル・アルゴリズム（AL）の詳細 .....	- 67 -
2-2. システムコンポーネント（CO） .....	- 101 -
2-2-1. システムコンポーネント図 .....	- 101 -
2-2-2. システムコンポーネント一覧 .....	- 102 -
2-3. ハードウェア（HW） .....	- 113 -
2-3-1. ハードウェアアーキテクチャ .....	- 113 -
2-3-2. ハードウェア一覧 .....	- 114 -
2-3-3. ハードウェアの詳細 .....	- 115 -
2-4. データインターフェース（IF） .....	- 118 -
2-4-1. データアーキテクチャ .....	- 118 -
2-4-2. データインターフェース一覧 .....	- 119 -
2-4-3. データインターフェースの詳細 .....	- 121 -
2-5. ユーザーインターフェース（UI） .....	- 138 -
2-5-1. 画面遷移図 .....	- 138 -
2-5-2. ユーザーインターフェース一覧 .....	- 140 -
2-5-3. ユーザーインターフェースの詳細 .....	- 144 -
3. 共同使用契約に基づくカーシェアリングシステム：非機能要件（NF） .....	- 207 -
3-1. 非機能要件一覧 .....	- 207 -
3-2. 非機能要件の詳細 .....	- 208 -
4. 実証調査に利用するデータ（DT） .....	- 209 -
4-1. 実証調査に利用するデータ一覧 .....	- 209 -
4-2. 実証調査に利用するデータの詳細 .....	- 209 -
5. 用語集 .....	- 210 -

# 1. 開発スコープ

## 1-1. 概要

車両提供者（主に事業者や地方自治体）とユーザーの二者間で車両を共同で管理・使用するための共同使用契約の締結を媒介・支援し、車両利用のための予約の作成管理及び無人での車両鍵の受渡しができる等の予約、使用、精算までの一連のプロセスをスマートフォン等で完結できるカーシェアリングシステムを開発する。既存のシステムを改修・追加開発することで共同使用契約を通した車両の使用に必要な機能及び使用に当たって不都合のないUI・UXを構築する。

<ユーザー登録・車両提供者登録・車両登録>

本サービスを提供する上で必要になる下記リソースをサービス上にデータとして登録する。

- ユーザー登録
  - ▶ 本サービスを利用する者はユーザー・車両提供者問わず、ユーザーの登録が必要になる。ユーザーの登録は、電話番号・メールアドレスの連絡先、名前・住所・免許証等の個人情報を登録する。
- 車両提供者登録
  - ▶ 車両提供者登録は、事業者名・法人番号（法人の場合）・事業所住所・連絡先・口座番号を登録する。
- 車両登録
  - ▶ 車検証を基にした車両情報及び車両写真、共同使用契約の内容、傷の登録、ステーション、共同使用料、カーシェア専用保険の種別を選択し、登録する。

ユーザー・車両提供者共にユーザー登録は必須であり、車両提供者はこれに加えて、車両提供者登録・車両登録を行う。上記三つの登録内容は、サービス提供事業者の審査を経て登録完了となる。車両登録は、車両の初期費用、維持費用等を入力し、維持費の範囲内で受取可能な共同使用料の算出を行う。車両提供者は、登録された車両に対して、ユーザーが予約可能な時間帯をあらかじめ設定できる。車両審査完了後、スマートキーボックスを郵送で受取、TTLock アプリ又は管理画面上からワンタイムパスコードを発行してスマートキーボックスを開錠・車両鍵を格納した後、車両の後部ナンバー横に取り付け、車両を一般公開し予約を受け付ける。なお、既存スマートキーボックス保有者に向けては、本実証による新しいスマートキーボックスの配布を通知・連絡を行い、ユーザーが任意で交換を実施する

<予約の作成>

予約作成は以下の手順で行う。

1. 地図上から使いたい日時等の条件を指定し、車両を選択
  - A) 車両フィルタリング（車種・時間・乗車人数・駆動方式）を通して絞り込み、使用したい車両を選択する。
2. 詳細の確認、予約作成
  - A) 具体的に使用する日時に指定、共同使用契約の内容、車両設備等を確認し車両使用を予約する。
3. 車両のピックアップ
  - A) 使用開始時間に車両が配置されているステーションに移動する。
4. スマートキーボックスの解錠
  - A) 当該予約に割当てられたパスコードを確認し、車両後部に取り付けられているスマートキーボックスに入力して車両鍵を取り出す。
5. 利用開始報告

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- A) 車両写真・日常点検・傷の確認等を入力し、利用開始報告をシステム上で実施する。
- 6. 利用開始
  - A) 取り出した鍵で車両を解錠、エンジンを始動し、使用を開始する。
- 7. 利用終了・返却
  - A) 予約終了時間までにステーションに車両を駐車し、スマートキーボックスをパスコードで解錠、車両鍵を返却する。
  - B) 車両写真・日常点検・傷の確認等を入力し、利用終了報告をシステム上で実施する。
- 8. 支払
  - A) 共同使用料が確定し、クレジットカード又は電子決済による支払を実行する。
- 9. レビュー作成・公開
  - A) ユーザー・車両提供者がそれぞれ車両提供・利用に対する評価をレビューとして作成することができる。この評価内容はユーザー・車両提供者共に確認することができる。

予約作成を通して共同使用契約を締結したユーザー（以下共同使用者）は予約時に車両提供者が選択した必須加入のカーシェア専用保険の金額を確認でき、予約作成と同時に当該予約に対するカーシェア専用保険が適用される。当該車両の車両提供者に関する情報を予約画面からプロフィール画面に遷移することで確認でき、過去の予約レビューやその評価等が表示される。また、この予約は使用開始時間直前まで取り消すことができる。※キャンセル理由は問わない

車両提供者は予約されたユーザーのプロフィールを確認することができ、過去のレビュー・評価を確認することができる。その確認を通して、車両の提供を行いたくない場合、車両提供者側でその予約キャンセル及び当該共同使用者との共同使用契約を拒否することができる。ほかにも、車両提供者の都合で共同使用者の予約をキャンセルしたい場合、車両の管理画面からキャンセル実行することができる。また車両提供者は予約作成時、利用開始時、利用終了時に通知を受け取ることができ、それぞれの報告時の写真や総走行距離メーター、日常点検の確認結果等が確認できる。これらの予約データ（時間、利用中、利用終了等のステータス等）・レビューデータはデータベースに保管される。

### <共同使用料の振込>

共同使用された際に支払われた共同使用料は、車両提供者に対して月ごとサービス提供事業者から共同使用料として口座振込によって支払われる（口座振込手数料は車両提供者負担）が、累計受取共同使用料が初期費用、維持費用を超えないように車両ごとの受取金額をコントロールする。なお、共同使用料は、共同使用契約を通じた車両の維持管理に必要な実費を按分するために利用時間等に基づいて算出される金額である。この共同使用とは「同一の車両に対して複数人が主体的に共同の共同使用者として使用する」行為であり、当該車両の維持費用を上限として車両提供者が共同使用料を受け取ることができる。

## 1-2. システムを利用する業務全体像とシステム利用フロー

### 1. 業務フロー

- ユーザー登録

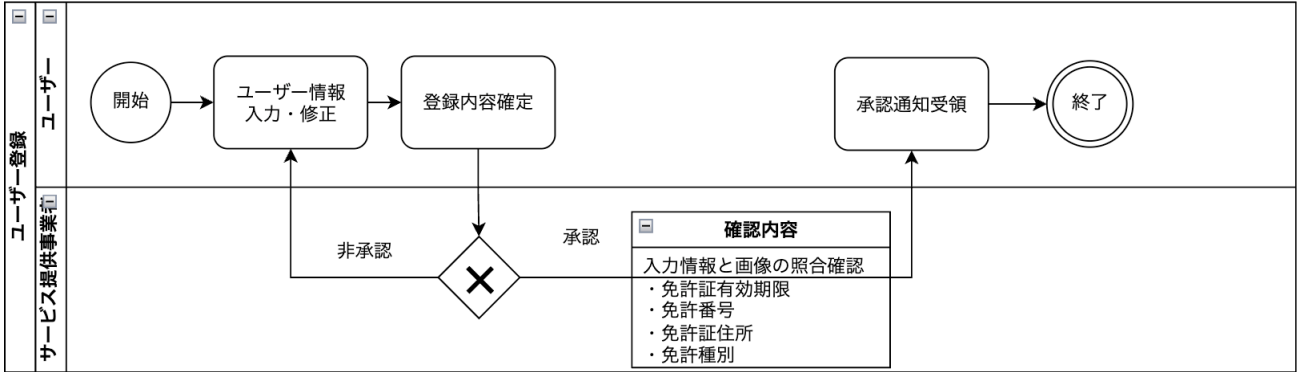


図 1-1 ユーザー登録業務フロー図

- 車両提供者登録

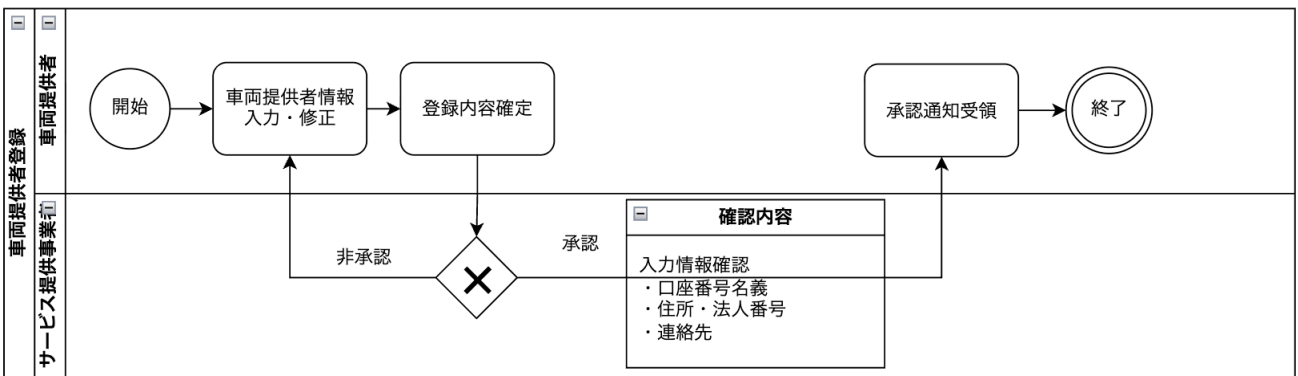


図 1-2 車両提供者登録業務フロー図

- 車両登録

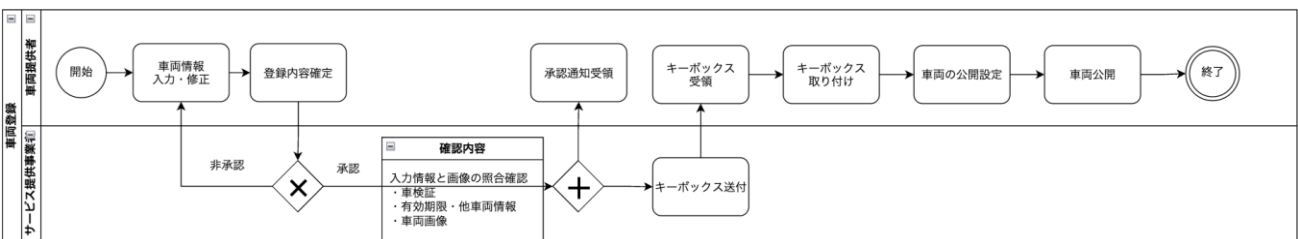


図 1-3 車両登録業務フロー図

● 予約～利用終了

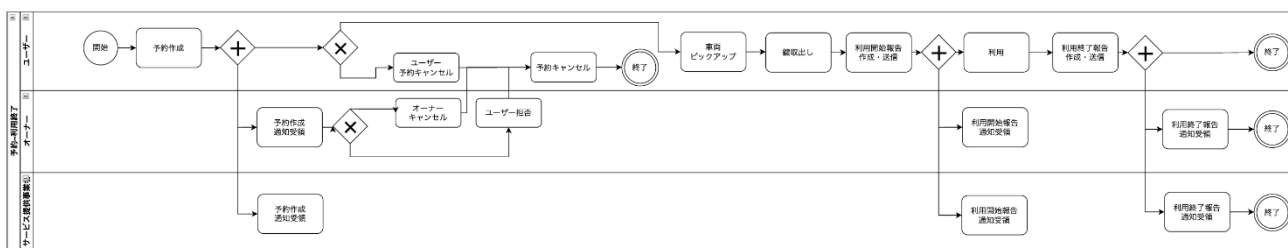


図 1-4 予約～利用終了業務フロー

● レビュー作成～公開

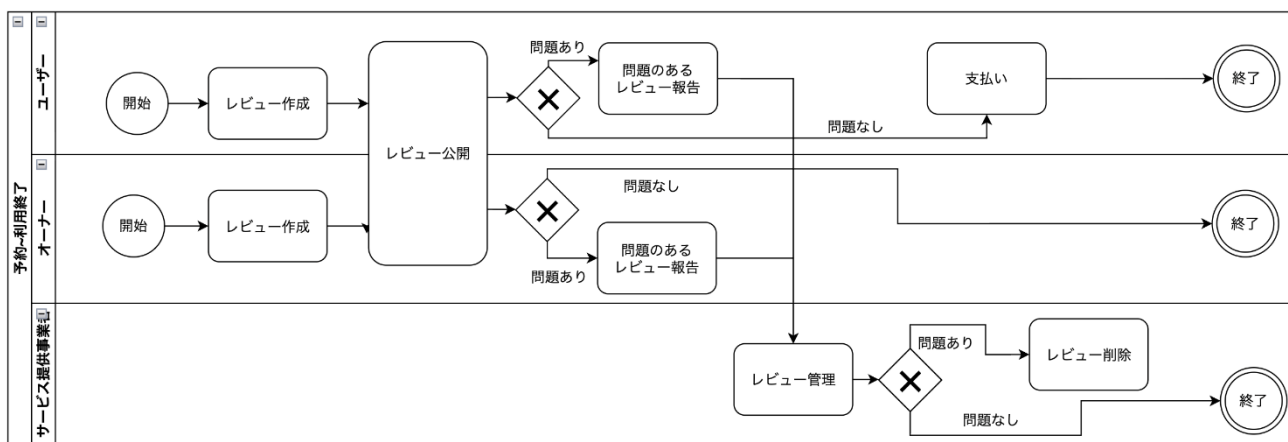


図 1-5 レビュー作成～公開業務フロー図

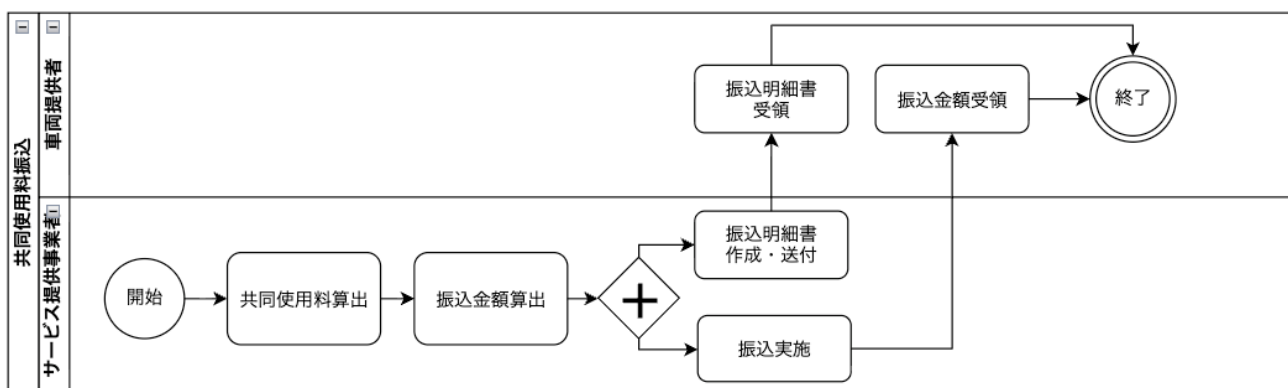


図 1-6 毎月の共同使用料振込業務フロー図

2. システムシーケンス図

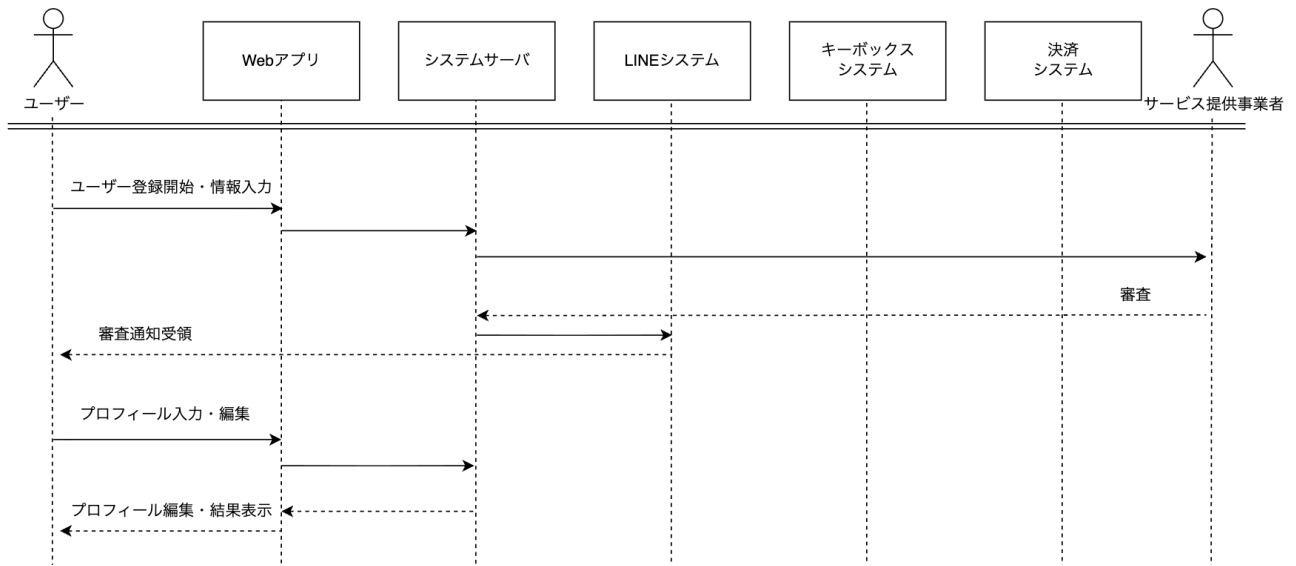


図 1-7 ユーザー登録処理

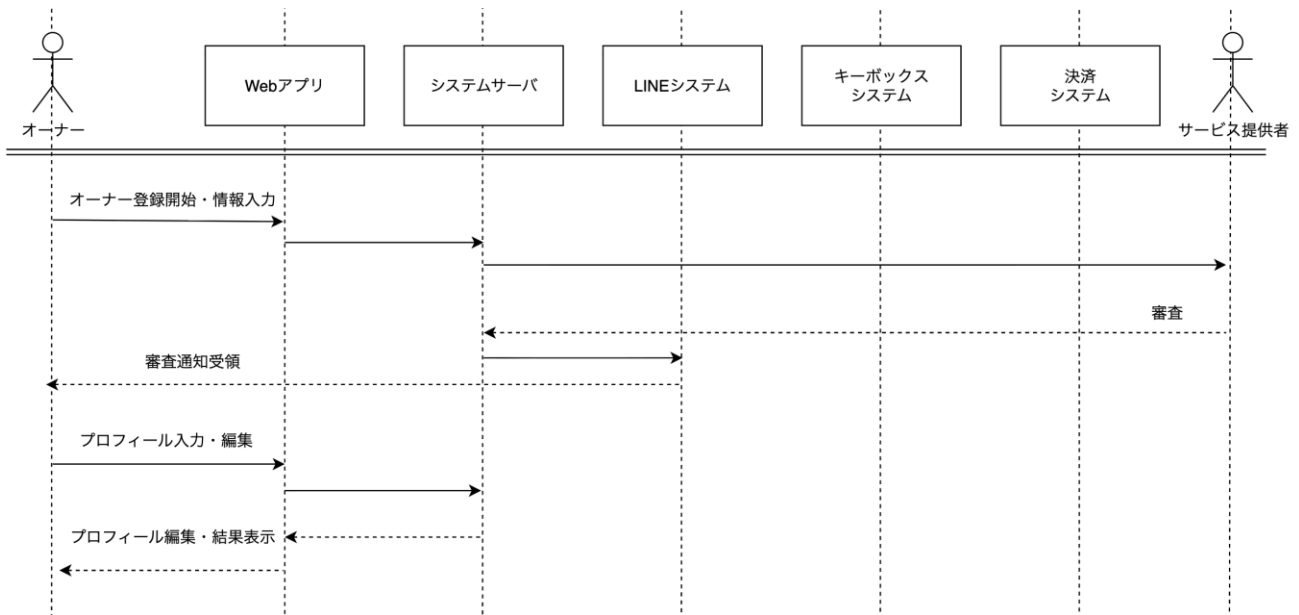


図 1-8 車両提供者登録処理

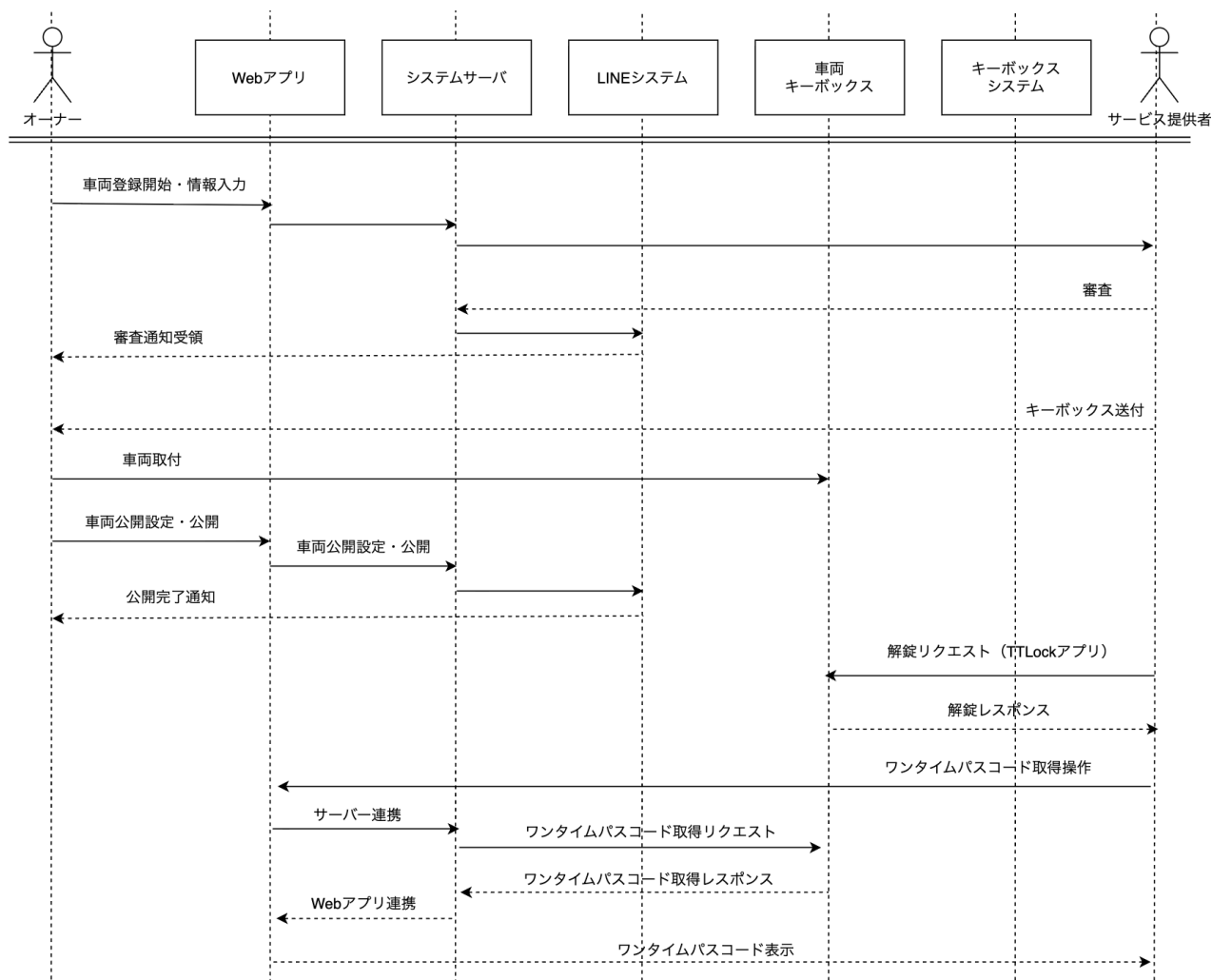


図 1-9 車両の登録の処理

共同使用契約に基づくカーシェアリングシステム システム設計書

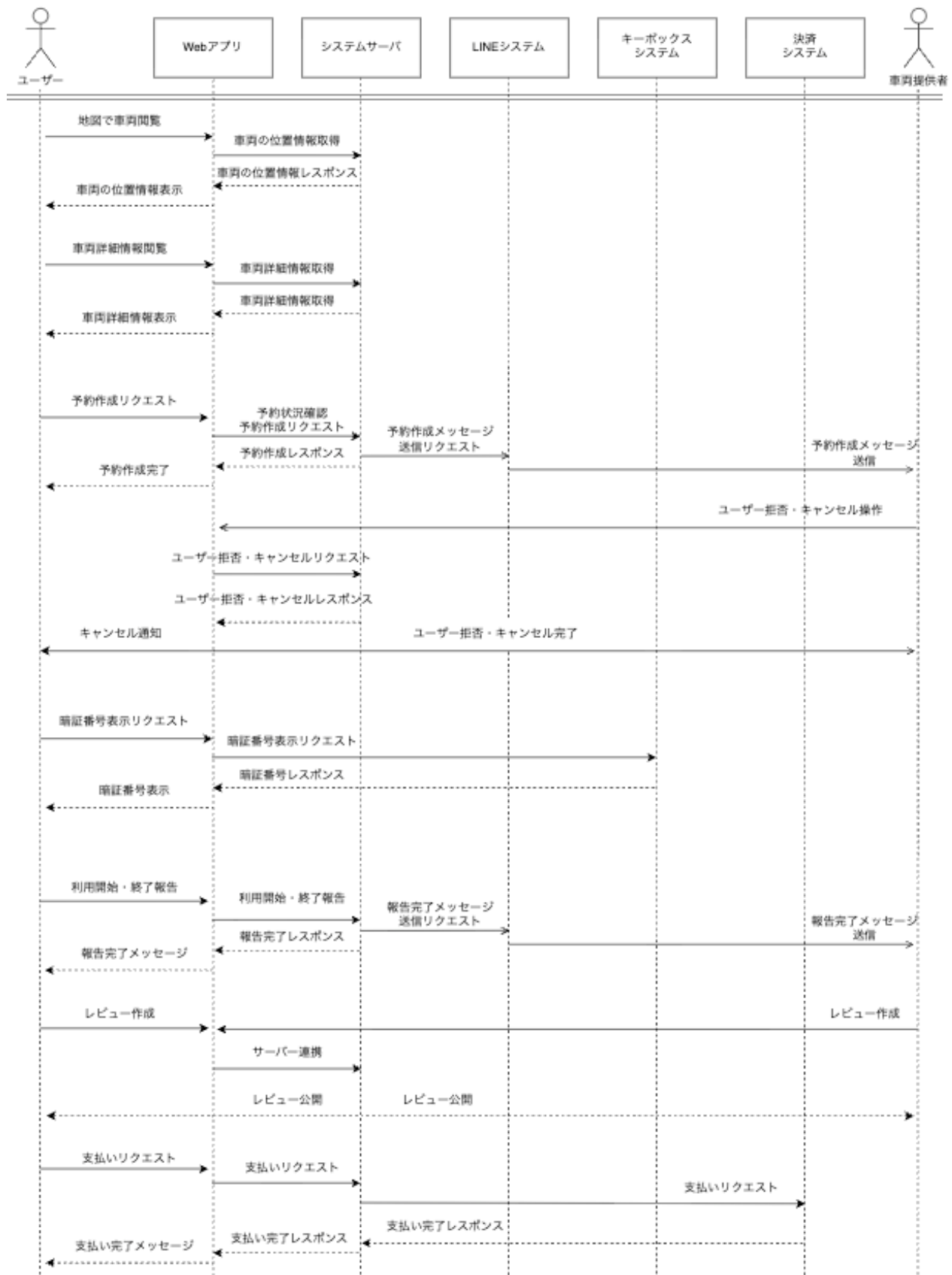


図 1-10 利用開始～利用終了の処理

## 2. 共同使用契約に基づくカーシェアリングシステム：機能要件 (FN/SL/AL/CO/HW/IF/UI)

### 2-1. システム機能 (FN) 2-1-1. システムアーキテクチャ

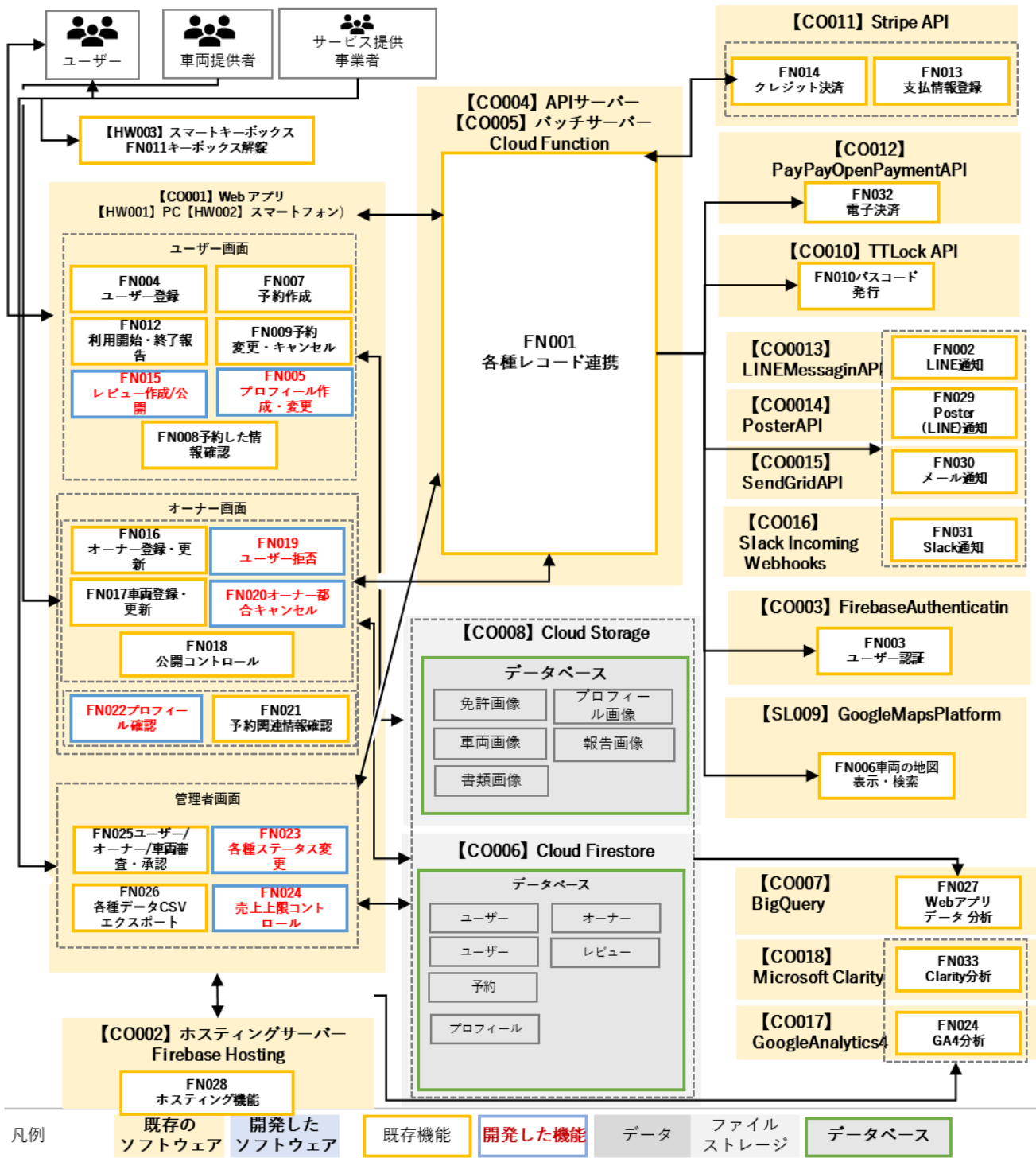


図 2-1 システムアーキテクチャ

## 2-1-2. システム機能一覧

表 2-1 【HW001・HW002】 PC・スマートフォン用機能一覧

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN001	各種レコード連携	<ul style="list-style-type: none"> <li>● ユーザーによる Web アプリの操作に応じ、ユーザー・車両提供者・車両・予約・レビューなどのデータをストレージに書き込み、また Web アプリの操作によってその他のシステム連携を行う。</li> </ul>
FN002	LINE 通知	<ul style="list-style-type: none"> <li>● ユーザー・車両提供者による Web アプリの操作やスケジュールに応じ、LINE Messaging API を通じた通知を行う。</li> </ul>
FN003	ユーザー認証	<ul style="list-style-type: none"> <li>● ユーザーによる Web アプリの操作に応じ、OIDC 認証 (LINE・Google) 又はパスワード認証+メールアドレス・SMS 認証によるユーザーのログイン機能を提供する。</li> </ul>
FN004	ユーザー登録	<ul style="list-style-type: none"> <li>● ユーザーによる Web アプリの操作に応じ、免許情報や電話番号等のデータを基にしたユーザーID を作成し、レコードをデータベースに作成する。</li> </ul>
FN005	プロフィール作成・変更	<ul style="list-style-type: none"> <li>● ユーザー・車両提供者のニックネーム等を含めたプロフィールデータの作成・変更を行い、レコードをデータベースに作成する。</li> </ul>
FN006	車両の地図表示・検索	<ul style="list-style-type: none"> <li>● 車両を地図上から車種や使用可能時間等でフィルタリングして検索できる。</li> </ul>
FN007	予約作成	<ul style="list-style-type: none"> <li>● ユーザーが使用したい時間帯・車両を選択し、予約が作成され、内容の基づいた共同使用料が算出される。</li> <li>● 更新レコードをデータベースに反映する。</li> </ul>
FN008	予約した情報確認	<ul style="list-style-type: none"> <li>● 予約に基づく各種情報 (予約時間・車両・保険種別等) を確認できる</li> </ul>
FN009	予約変更・キャンセル	<ul style="list-style-type: none"> <li>● ユーザーが車両予約に関する諸条件 (共同使用契約の内容等) を確認し、予約内容の変更、キャンセル、共同運転者の更新、削除の操作を実施し、レコードをデータベースに作成する。</li> </ul>
FN010	パスコード発行	<ul style="list-style-type: none"> <li>● ユーザーによる Web アプリの操作に応じ、予約時間内に利用可能なスマートキーボックスのパスコードを TTLock API を通じ、TTLock Open Platform から発行・取得する。</li> </ul>
FN011	キーボックス解錠	<ul style="list-style-type: none"> <li>● 車両にスマートキーボックスを取り付ける車両ID とひも付いたスマートキーボックス固有の番号があり、車両 ID に対して作成された予約時間のみ有効なパスコードをスマートキーボックスのインターフェースに入力すること</li> </ul>

		で解錠し、車両鍵を取り出せる。
FN012	利用開始・終了報告	<ul style="list-style-type: none"> <li>● 車両の利用終了時に車両状態及び総走行距離メーター数値・傷の有無・日常点検を確認・報告を作成し、車両提供者への通知を行う。</li> <li>● 利用終了時には走行距離に基づく距離料金が加算され決済と連携する。各報告にひも付く利用ステータスレコードを データベース に作成する。</li> </ul>
FN013	支払情報登録	<ul style="list-style-type: none"> <li>● 本サービスの決済時に利用するクレジットカード情報を登録する。</li> </ul>
FN014	クレジット決済	<ul style="list-style-type: none"> <li>● ユーザーによる Web アプリの操作に応じ、決済代行サービス (Stripe) を利用した決済を実行する。</li> </ul>
FN015	レビュー作成・公開	<ul style="list-style-type: none"> <li>● 利用及び支払が完了した予約にひも付く車両・ユーザーに対してのレビューをユーザー・車両提供者共に作成し、公開を行い、レコードを データベース に作成する。</li> </ul>
FN016	車両提供者登録・更新	<ul style="list-style-type: none"> <li>● 車両提供者による Web アプリの操作に応じ、ユーザーID にひも付く車両提供者アカウントの作成をし、レコードを データベース に作成する。</li> </ul>
FN017	車両登録・更新	<ul style="list-style-type: none"> <li>● 車両提供者 ID にひも付く車両データ及び傷、保険種別の登録データ作成をし、レコードを データベース に作成する。これ等の情報を確認更新できる。</li> </ul>
FN018	公開コントロール	<ul style="list-style-type: none"> <li>● スケジュールや一時的なメンテナンス、一時的な公開、現在時刻から設定時間の間は予約不可とする設定によって、車両を公開し予約可能状態か予約不可状態かをコントロールする。</li> </ul>
FN019	ユーザー拒否	<ul style="list-style-type: none"> <li>● 車両提供者が特定のユーザーに対して特定車両において共同使用契約締結を拒否し、更新レコードをデータベースに反映する。拒否後、当該ユーザーは当該車両における共同使用契約を締結することができなくなる。</li> </ul>
FN020	車両提供者都合キャンセル	<ul style="list-style-type: none"> <li>● ユーザー作成した予約に対して、車両提供者が予約のキャンセルを行い、更新レコードをデータベースに反映する。</li> </ul>
FN021	予約関連情報確認	<ul style="list-style-type: none"> <li>● 登録した車両に対して作成された予約、ユーザー情報を確認できる。</li> </ul>
FN022	プロフィール確認	<ul style="list-style-type: none"> <li>● ユーザー・車両提供者共に作成したプロフィール及び過去のレビュー、レビューに基づく評価を相互に確認できる。</li> </ul>
FN023	各種ステータス変更	<ul style="list-style-type: none"> <li>● サービス提供事業者がユーザー・車両提供者・車両・予約・レビューに関するステータス情報の変更を実施し、更新レコードをデータベースに反映する。</li> </ul>
FN024	売上上限コントロール	<ul style="list-style-type: none"> <li>● 車両 ID ごと受取金額を算出し、維持費用・初期費用の合計を超過しないように車両の公開状態をコントロール・</li> </ul>

共同使用契約に基づくカーシェアリングシステム システム設計書

		管理する。
FN025	ユーザー/車両提供者/車両 審査・承認	● ユーザー・車両提供者による登録データをサービス提供事業者が確認し、承認・拒否判断を行い、更新レコードをデータベースに反映する。
FN026	各種データ CSV エクスポート	● 利用データやユーザーデータ、車両データ等各種データがダウンロードできる。
FN027	Web アプリデータ分析	● 利用中画面の録画やヒートマップ、ページごとの遷移、利用データ全体の集計・分析を行うことができる。
FN028	ホスティング機能	● サーバー内で Web アプリケーションの実行、ドメインの管理を行う。
FN029	Poster (LINE) 通知	● LINE 管理ツールの「Poster」を介して通知をユーザー・車両提供者に送信する。
FN030	メール通知	● メールによる各種通知を送信する。
FN031	Slack 通知	● サービス提供事業者側に各種通知を送信する。
FN032	電子決済	● サービス利用に伴う電子決済 (PayPay) を実行する。
FN033	Clarity 分析	● クラリティーを使用したユーザー行動のデータ分析を実行する。
FN034	GA4 分析	● Google Analytics 4 を通したユーザー行動のデータ分析を実行する。

2-1-3. システム機能の詳細

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

【FN001】各種レコード連携

- 本システム機能の概要
  - 各インターフェースからの入力によって作成されたデータをデータベースに保存し、連携又は取り出しを行う
- 本システム機能の入力・処理・出力のフローチャート

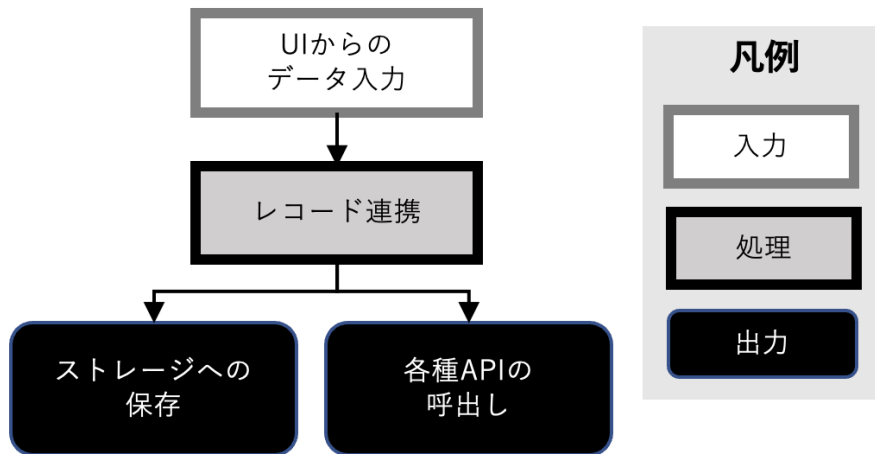


図 2-2 【FN001】のフローチャート

- 本システム機能の処理の詳細
  - データの呼出し
    - ◇ 処理内容
      - 画像データや予約時の文字データ等、違うストレージに保存されているデータを操作に応じて取り出し、表示できる
    - ◇ 利用するライブラリ
      - 【SL004】 Firebase Admin SDK (firebase-admin)
      - 【SL005】 Firebase (firebase)
    - ◇ 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
    - ◇ UIからのデータ入力
      - データの内容
        - ユーザーデータ・車両提供者データ・予約データ等 Web アプリを通じて作成されたデータ
      - データの形式
        - テキスト形式/JPEG
    - ◇ 利用するデータインターフェース
      - 【IF001】ユーザーデータの作成・取得 IF

- 【IF002】 キーボックスパスコード発行 API (TTLock API)
  - 【IF003】 車両の地図表示 API (Google Maps Platform API)
  - 【IF004】 クレジットカード決済 API (Stripe API)
  - 【IF005】 PayPay 決済 API (PayPay Open Payment API)
  - 【IF006】 LINE メッセージ送信 API (LINE Messaging API)
  - 【IF007】 メール送信 API (SendGrid API)
  - 【IF008】 ユーザー認証 IF (Firebase Authentication with Identity Platform)
  - 【IF009】 各種データのファイル出力 IF
  - 【IF010】 ページ取得 IF (Firebase Hosting)
  - 【IF012】 LINE ユーザー管理 API (Poster API)
  - 【IF015】 Slack メッセージ投稿 IF (Slack Incoming Webhooks)
- 出力
- ◇ ストレージへの保存
    - データの内容
      - 詳細はシステムアーキテクチャを参照
    - データの形式
      - リクエスト
    - 利用するデータインターフェース
      - 【IF001】 ユーザーデータの作成・取得 IF
      - 【IF009】 各種データのファイル出力 IF
      - 【IF010】 ページ取得 IF (Firebase Hosting)
  - ◇ 各種 API の呼出し
    - データの内容
      - Web アプリ操作をトリガーとした API とのつなぎ込み
        - ◇ メール/LINE 送付・Slack 送付等の各種通知
        - ◇ 決済処理の実行
    - データの形式
      - リクエスト
    - 利用するデータインターフェース
      - 【IF002】 キーボックスパスコード発行 API (TTLock API)
      - 【IF003】 車両の地図表示 API (Google Maps Platform API)
      - 【IF004】 クレジットカード決済 API (Stripe API)
      - 【IF005】 PayPay 決済 API (PayPay Open Payment API)
      - 【IF006】 LINE メッセージ送信 API (LINE Messaging API)
      - 【IF007】 メール送信 API (SendGrid API)
      - 【IF008】 ユーザー認証 IF (Firebase Authentication with Identity Platform)
      - 【IF012】 LINE ユーザー管理 API (Poster API)
      - 【IF015】 Slack メッセージ投稿 IF (Slack Incoming Webhooks)

### 【FN002】 LINE 通知

- 本システム機能の概要
  - 予約作成や、利用開始・終了報告等の通知を各ステークホルダーに対して LINE で送付する
- 本システム機能の入力・処理・出力のフローチャート

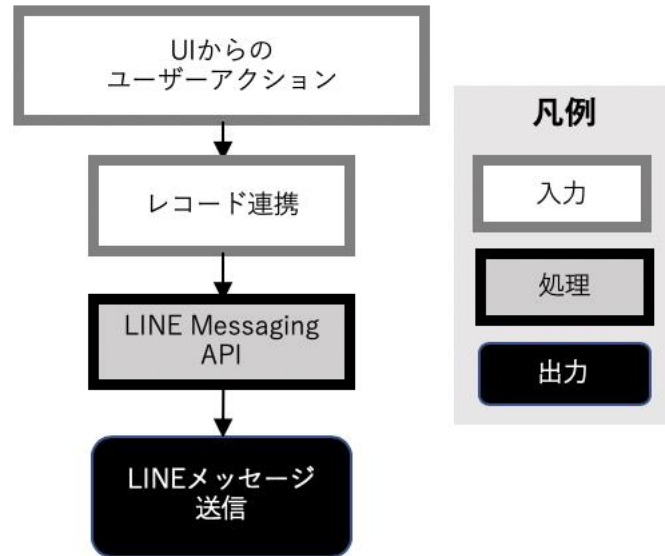


図 2-3 【FN002】のフローチャート

- 本システム機能の処理の詳細
  - LINE を通したメッセージ送信
    - ◇ 処理内容
      - アカウント認証が LINE の場合は LINE で下記のタイミングにおける通知を送信する
        - アカウント（ユーザー・車両提供者・車両）の審査時
        - 予約の作成時・キャンセル時・変更時・利用開始時・利用終了時
        - レビュー作成時、レビュー公開時
    - ◇ 利用するライブラリ
      - 【SL012】 LINE Messaging API SDK (@line/bot-sdk)
    - ◇ 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
    - ◇ レコード連携
      - データの内容
        - CloudFunction からの API 連携
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF006】 LINE メッセージ送信 API (LINE Messaging API)
  - 出力
    - ◇ LINE メッセージ送信

- データの内容
  - 各種 API から CloudFunction への連携
- データの形式
  - LINE メッセージ送信
- 利用するデータインターフェース
  - 【IF006】 LINE メッセージ送信 API (LINE Messaging API)

【FN003】 ユーザー認証

- 本システム機能の概要
  - ユーザーがサービス使用する際にログイン認証の方法を選択して適用できる
- 本システム機能の入力・処理・出力のフローチャート

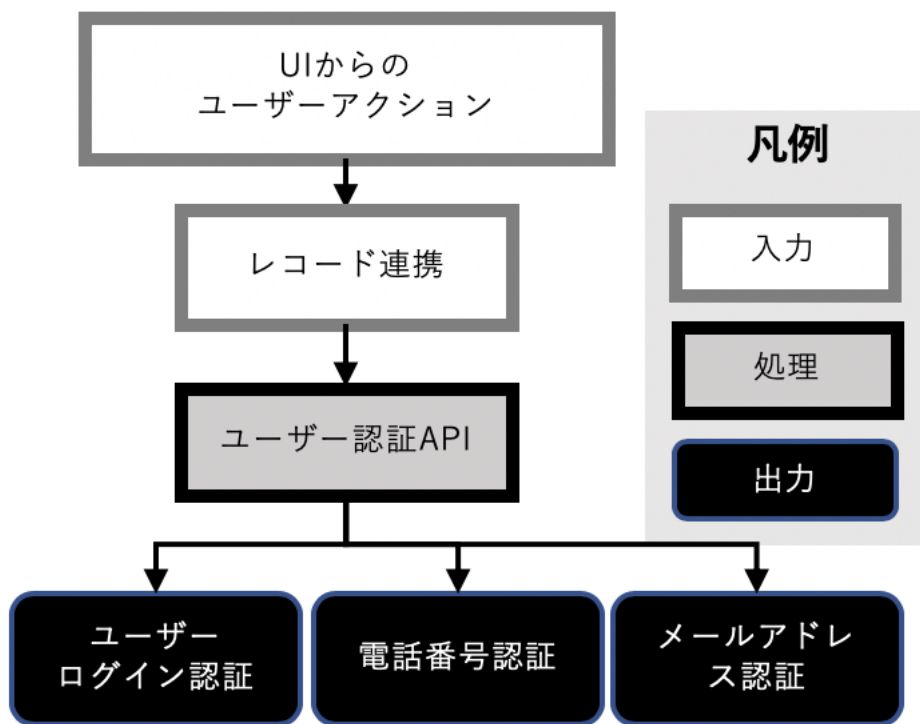


図 2-4 【FN003】のフローチャート

- 本システム機能の処理の詳細
  - 外部サービスを用いたログイン
    - ◇ 処理内容
      - OAuth 2.0 等を用いた LINE・Google アカウント・パスワード方式 (メールアドレス) を使用したログイン方法から選択できる
      - 登録電話番号への SMS を通じた暗証番号照合による本人確認を実施
      - 登録メールアドレスへの照合確認リンクによる本人確認を実施
    - ◇ 利用するライブラリ
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL005】 Firebase (firebase)

◇ 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

◇ レコード連携

- データの内容
  - アカウント連携方法の選択
- データの形式
  - リクエスト
- 利用するデータインターフェース
  - 【IF008】ユーザー認証 IF (Firebase Authentication with Identity Platform)

➤ 出力

◇ ユーザー認証

- データの内容
  - 選択されたアカウント連携の実行
  - CloudFunction への連携
- データの形式
  - リクエスト
- 利用するデータインターフェース
  - 【IF008】ユーザー認証 IF (Firebase Authentication with Identity Platform)

【FN004】ユーザー登録

● 本システム機能の概要

- ユーザーID とユーザーID にひも付く情報の登録を行いユーザーアカウントを作成する

● 本システム機能の入力・処理・出力のフローチャート

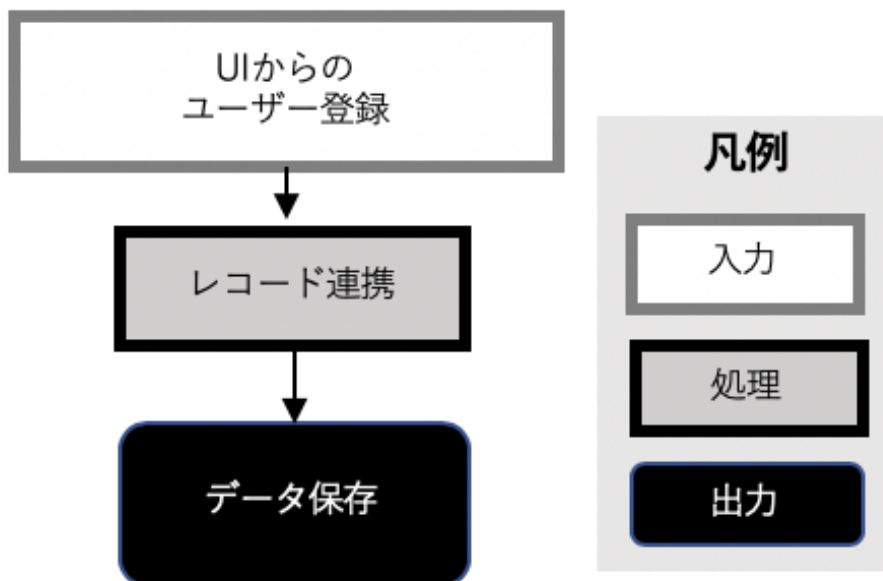


図 2-5 【FN004】のフローチャート

- 本システム機能の処理の詳細
    - ユーザー情報の登録
      - ◇ 処理内容
        - 名前、連絡先（メールアドレス・電話番号）、住所、生年月日、免許証の画像及び記載事項等を UI から入力し、ストレージに保存する
      - ◇ 利用するライブラリ
        - 【SL001】 TypeScript (typescript)
        - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
        - 【SL005】 Firebase (firebase)
        - 【SL006】 Next.js (next)
        - 【SL007】 React (react)
      - ◇ 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
      - ◇ UI からのユーザー登録
        - データの内容
          - 名前、住所等の本人確認のための個人情報の連携
        - データの形式
          - テキスト形式/JPEG
        - 利用するデータインターフェース
          - 【IF001】 ユーザーデータの作成・取得 IF
          - 【IF008】 ユーザー認証 IF (Firebase Authentication with Identity Platform)
          - 【IF010】 ページ取得 IF (Firebase Hosting)
    - 出力
      - ◇ データ保存
        - データの内容
          - 名前、住所等の本人確認のための個人情報の保存
        - データの形式
          - テキスト形式/JPEG
        - 利用するデータインターフェース
          - 【IF001】 ユーザーデータの作成・取得 IF
          - 【IF008】 ユーザー認証 IF (Firebase Authentication with Identity Platform)
          - 【IF010】 ページ取得 IF (Firebase Hosting)
- 【FN005】 プロフィール作成・変更 <新規開発>
- 本システム機能の概要
    - ユーザー・車両提供者の過去のレビューや総合評価、自己紹介や一言が確認できる
  - 本システム機能の入力・処理・出力のフローチャート

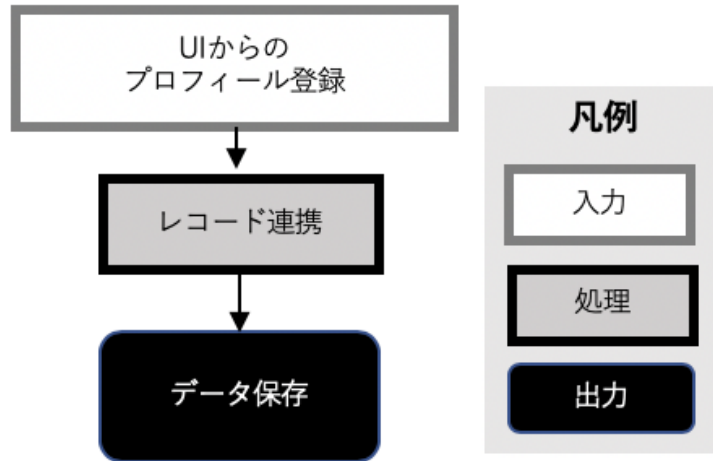


図 2-6 【FN005】のフローチャート

- 本システム機能の処理の詳細
  - プロフィールの作成
    - ◇ 処理内容
      - ユーザーID・車両提供者 ID にひも付き作成される、自身のプロフィール画面内の下記の項目を入力、編集することができる
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL005】 Firebase (firebase)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ UIからのプロフィール登録
      - データの内容
        - プロフィール情報の連携
      - データの形式
        - テキスト形式/JPEG
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
        - 【IF010】 ページ取得 IF (Firebase Hosting)
  - 出力
    - ◇ データ保存
      - データの内容
        - CloudFunctions への連携
        - プロフィールデータの保存

- データの形式
  - テキスト形式/JPEG
- 利用するデータインターフェース
  - 【IF001】ユーザーデータの作成・取得 IF
  - 【IF010】ページ取得 IF (Firebase Hosting)

【FN006】車両の地図表示・検索

- 本システム機能の概要
  - ユーザーが予約作成時に車両選択する際、地図上から条件を絞り込み、車両を選択できる
- 本システム機能の入力・処理・出力のフローチャート

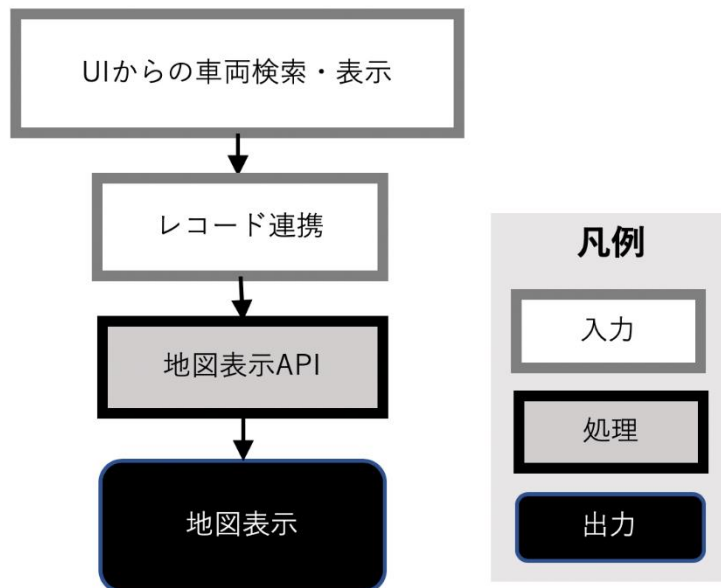


図 2-7 【FN006】のフローチャート

- 本システム機能の処理の詳細
  - ステーションを地図上に表示する
    - ◇ 処理内容
      - 車種や利用時間による絞り込みを行い、希望する条件に合致する車両を表示する
    - ◇ 利用するライブラリ
      - 【SL008】Google Maps JavaScript API React Wrapper (@googlemaps/react-wrapper)
    - ◇ 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
    - ◇ レコード連携
      - データの内容
        - ユーザーの地図画面の表示・範囲の変更操作
      - データの形式
        - リクエスト

- 利用するデータインターフェース
  - 【IF003】車両の地図表示 API (Google Maps Platform API)
- 出力
  - ◇ 地図表示
    - データの内容
      - 選択範囲での車両地図表示
    - データの形式
      - 画面表示
    - 利用するデータインターフェース
      - 【IF003】車両の地図表示 API (Google Maps Platform API)

### 【FN007】予約作成

- 本システム機能の概要
  - ユーザーの利用したい時間及び対象の車両に対して予約を作成する
- 本システム機能の入力・処理・出力のフローチャート

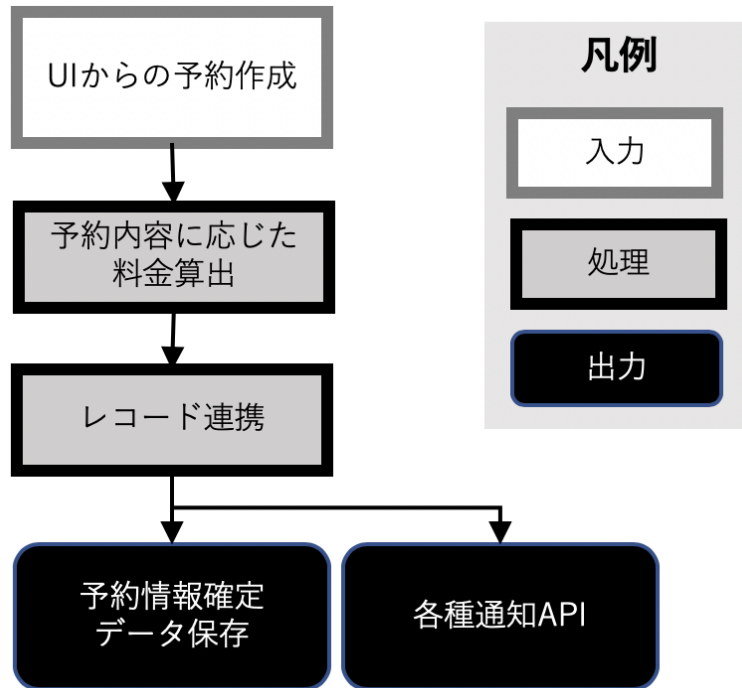


図 2-8 【FN007】のフローチャート

- 本システム機能の処理の詳細
  - 予約作成
    - ◇ 処理内容
      - 選択した車両の予約状況を確認し、利用希望の時間帯が予約可能であれば利用する時間、共同運転者、プロテクトプラン (NOC や免責金額に関する本サービス独自の保険プラン) を指定し、予約する
      - 予約が作成された時間帯は他のユーザーが予約作成することはできず、予約作成後、同期的に当該車両の利用状況を示すカレンダーに反映される
      - 【AL001】共同使用契約にのっとり車両上限金額のコントロールロジックに基づき、決

定された時間セットごとの単位料金を基に、作成された予約の利用開始・終了の時間に合わせた当該予約の時間料金が算出され、データ及び UI に反映される。

- ◇ 利用するライブラリ
  - 【SL001】 TypeScript (typescript)
  - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
  - 【SL006】 Next.js (next)
  - 【SL007】 React (react)
- ◇ 利用するアルゴリズム
  - 【AL001】 共同使用契約にのっとった車両上限金額のコントロールロジック
- 本システム機能の入出力データの仕様
  - 入力
    - ◇ UI からの予約作成
      - データの内容
        - UI からの予約作成操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
        - 【IF010】 ページ取得 IF (Firebase Hosting)
  - 出力
    - ◇ 予約情報確定データ保存
      - データの内容
        - 予約作成
      - データの形式
        - データ保存
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
        - 【IF010】 ページ取得 IF (Firebase Hosting)
    - ◇ 各種通知 API
      - データの内容
        - 各 API を通じて予約情報が確定したことを通知する
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF006】 LINE メッセージ送信 API (LINE Messaging API)
        - 【IF007】 メール送信 API (SendGrid API)
        - 【IF012】 LINE ユーザー管理 API (Poster API)
        - 【IF015】 Slack メッセージ投稿 IF (Slack Incoming Webhooks)

【FN008】 予約した情報確認

- 本システム機能の概要
  - ユーザーが作成済の予約の内容を確認できる
- 本システム機能の入力・処理・出力のフローチャート

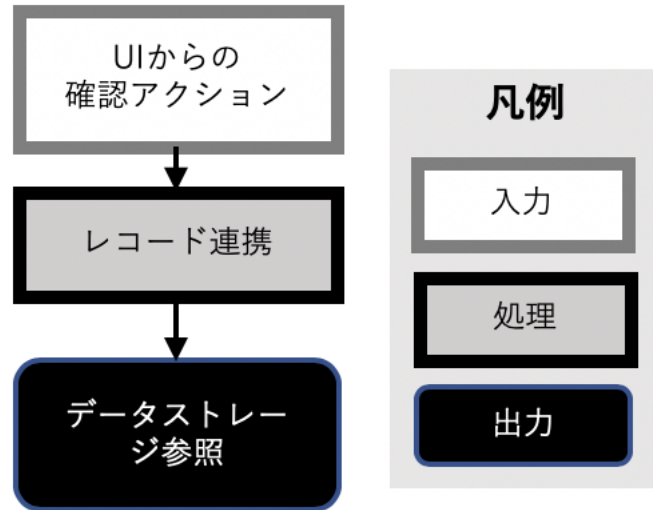


図 2-9 【FN008】 のフローチャート

- 本システム機能の処理の詳細
  - 予約内容確認
    - ◇ 処理内容
      - UI上からの予約内容確認のアクションによって、当該予約データを表示する
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ UIからの確認アクション
      - データの内容
        - UIからの予約確認操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
        - 【IF010】 ページ取得 IF (Firebase Hosting)
  - 出力
    - ◇ データストレージ参照

- データの内容
  - 予約データの表示
- データの形式
  - 画面表示
- 利用するデータインターフェース
  - 【IF001】 ユーザーデータの作成・取得 IF
  - 【IF010】 ページ取得 IF (Firebase Hosting)

**【FN009】 予約変更・キャンセル <新規開発>**

- 本システム機能の概要
  - 一度予約を作成した後で、その予約の時間、共同運転者の変更やキャンセルが実行可能
- 本システム機能の入力・処理・出力のフローチャート

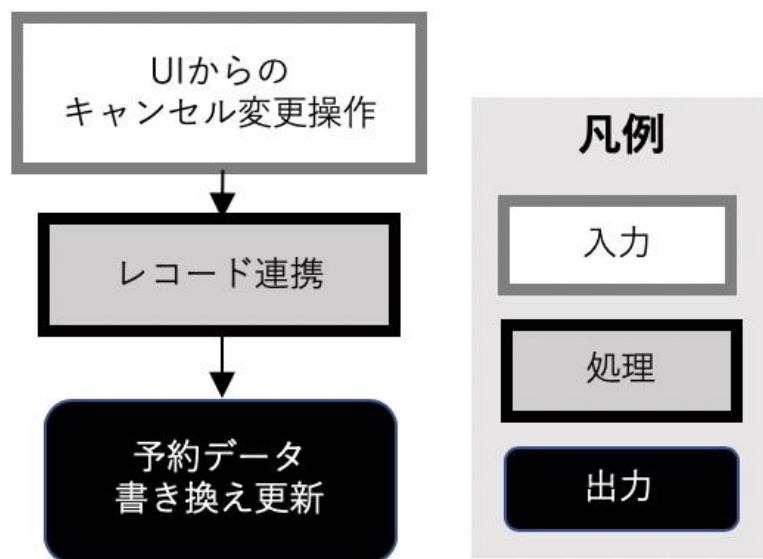


図 2-10 【FN009】のフローチャート

- 本システム機能の処理の詳細
  - 予約の変更・キャンセル
    - ◇ 処理内容
      - 当該予約の利用開始時間までであれば利用時間帯の変更が可能。変更に合わせて料金も再算出される
      - 当該予約に対する共同運転者の削除・追加ができる
      - 予約の変更・キャンセルの実行予約の利用時間までであれば予約をキャンセルすることができる
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ UIからのキャンセル変更操作
      - データの内容
        - UIからの予約変更操作・キャンセル操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF001】ユーザーデータの作成・取得 IF
        - 【IF010】ページ取得 IF (Firebase Hosting)
  - 出力
    - ◇ 予約データ書き換え更新
      - データの内容
        - 変更のデータ保存
        - 変更後の予約情報画面表示
      - データの形式
        - ストレージへのデータ保存
        - 画面表示
      - 利用するデータインターフェース
        - 【IF001】ユーザーデータの作成・取得 IF
        - 【IF010】ページ取得 IF (Firebase Hosting)

【FN010】パスコード発行

- 本システム機能の概要
  - 予約時間に合わせた時間制限付きのスマートキーパスコードが発行される
- 本システム機能の入力・処理・出力のフローチャート

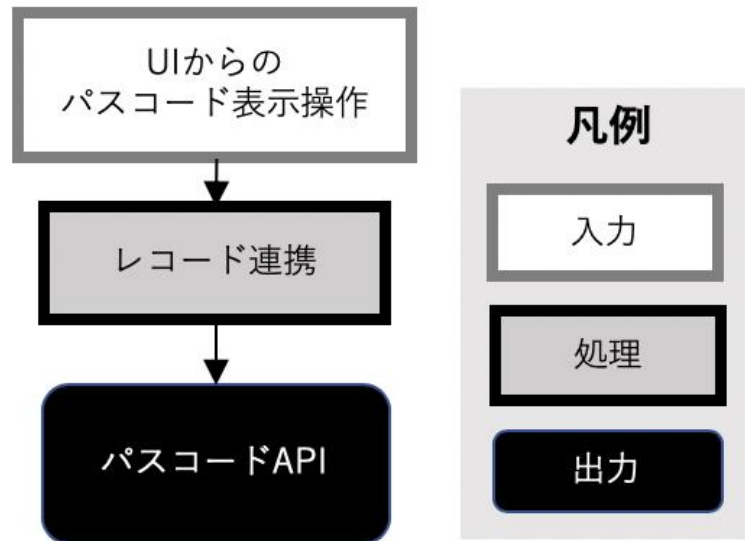


図 2-11 【FN010】のフローチャート

- 本システム機能の処理の詳細
  - パスワード表示
    - ◇ 処理内容
      - 予約ユーザーが利用開始時に「パスワードを発行」ボタンをクリックし、そのアクションをトリガーに、既存のオープンソフトウェアプラットフォームである TTLock Open Platform から、TTLock API を通じて、対象のスマートキーボックスに対して特定時間における唯一の数値（パスワード）を生成し、予約ユーザーが取得する。
    - ◇ 利用するライブラリ
      - 【SL003】 Firebase Cloud Functions SDK （firebase-functions）
    - ◇ 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
    - ◇ UIからのパスワード表示操作
      - データの内容
        - UIからのパスワード表示操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF002】 キーボックスパスワード発行 API（TTLock API）
  - 出力
    - ◇ パスワード API
      - データの内容
        - パスワードの取得、表示
      - データの形式
        - 画面表示

- 利用するデータインターフェース
  - 【IF002】キーボックスパスコード発行 API (TTLock API)

【FN011】キーボックス解錠<新規開発>

- 本システム機能の概要
  - ユーザーは予約時に表示されるパスコード(【FN010】)をスマートキーボックスに直接入力することによって車両鍵を取り出せる
  - 車両提供者は TTlock が提供するアプリでアプリからの解錠が可能
- 本システム機能の入力・処理・出力のフローチャート

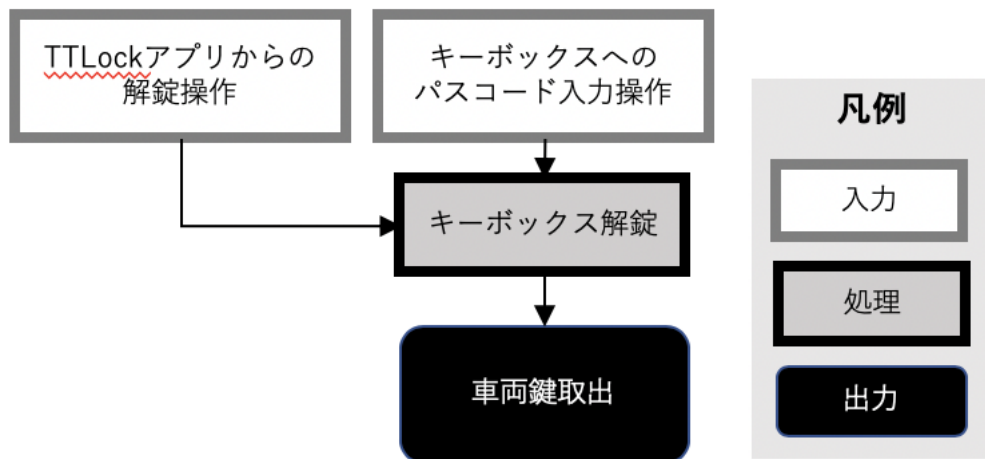


図 2-12 【FN011】のフローチャート

- 本システム機能の処理の詳細
  - スマートキーボックスの解錠
    - ◇ 処理内容
      - 車両提供者・ユーザー共に発行されたパスコードをスマートキーボックスに入力し解錠する
      - 当該スマートキーボックスに対して TTLock アプリ内の解錠ボタンを押下して解錠する (車両提供者のみ)
    - ◇ 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
    - ◇ スマートキーボックスへのパスコード入力操作
      - データの内容
        - パスコードの入力
      - データの形式
        - 物理的な数字番号の押下
      - 利用するデータインターフェース

- -
- ◇ TTLock アプリからの解錠
  - データの内容
    - アプリ内での解錠動作
  - データの形式
    - 近距離無線信号の発信
  - 利用するデータインターフェース
    - -
- 出力
  - ◇ 車両鍵取り出し
    - データの内容
      - パスコードが正の場合、スマートキーボックスを開錠する
      - 対象スマートキーボックスが一致した場合に開錠する
    - データの形式
      - モーターの回転、スマートキーボックスの開錠
    - 利用するデータインターフェース
      - -

【FN012】 利用開始・終了報告

- 本システム機能の概要
  - ユーザーの車両利用の開始時・終了時に写真等と主に車両の状態を報告し、日常点検を行う
- 本システム機能の入力・処理・出力のフローチャート

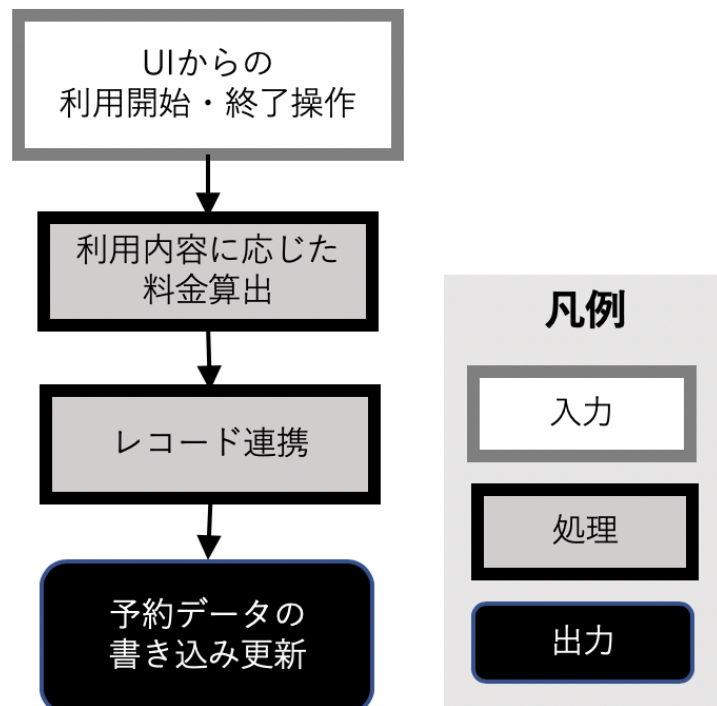


図 2-13 【FN012】のフローチャート

- 本システム機能の処理の詳細
  - 利用開始・利用終了の報告の実施
    - ◇ 処理内容
      - 利用開始、終了時の車両状態、日常点検の報告を行い、データをストレージに保管し、各種通知を関係者（ユーザー、車両提供者、サービス提供事業者）に送信する
      - 利用終了時に、報告日時に照らした実際の利用料金（利用時間と走行距離に応じた金額更新）を算出し、データ及びUIに反映する。
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ UIからの利用開始・終了操作
      - データの内容
        - 利用開始・終了報告操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
        - 【IF010】 ページ取得 IF (Firebase Hosting)
  - 出力
    - ◇ 予約データの書き込み更新
      - データの内容
        - データのストレージ保存
        - 報告の完了の画面表示
      - データの形式
        - データ保存
        - 画面表示
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
        - 【IF010】 ページ取得 IF (Firebase Hosting)

#### 【FN013】 支払情報登録

- 本システム機能の概要
  - 支払に使用するクレジットカードを事前に登録しておく
- 本システム機能の入力・処理・出力のフローチャート

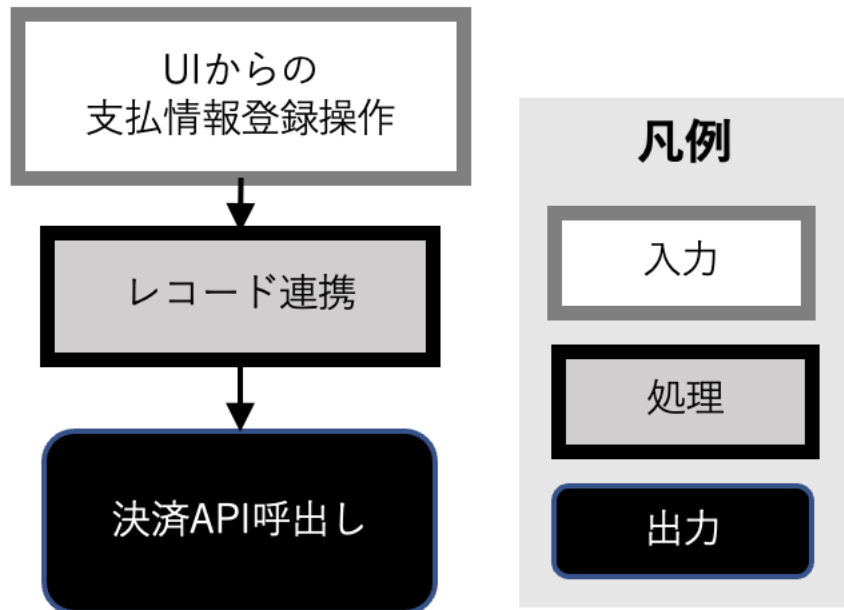


図 2-14 【FN013】のフローチャート

- 本システム機能の処理の詳細
  - クレジットカード情報のアプリ上で登録しておくことができる
    - ◇ 処理内容
      - UI 上に入力されたカード情報を、StripeAPI を通じて保存する
    - ◇ 利用するライブラリ
      - 【SL009】 Stripe Node.js Library (stripe)
    - ◇ 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
    - ◇ UI からの支払情報登録操作
      - データの内容
        - UI からのクレジットカード入力操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF004】 クレジットカード決済 API (Stripe API)
  - 出力
    - ◇ 決済 API 呼出し
      - データの内容
        - カード情報の保存
        - カード情報保存完了の表示
      - データの形式
        - データ保存

- 画面表示
- 利用するデータインターフェース
  - 【IF004】クレジットカード決済 API (Stripe API)

#### 【FN014】クレジット決済

- 本システム機能の概要
  - 予約終了時に算出される利用金額の支払をオンライン上で完結させる
- 本システム機能の入力・処理・出力のフローチャート

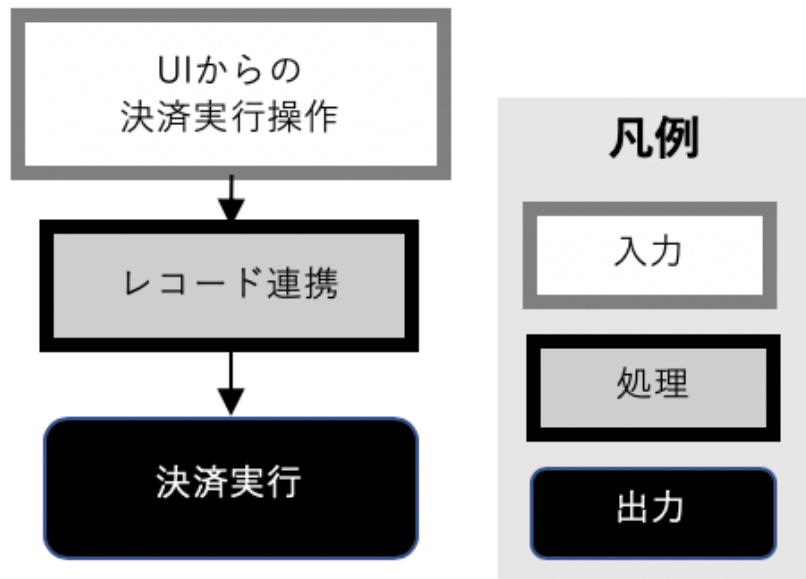


図 2-15 【FN014】のフローチャート

- 本システム機能の処理の詳細
  - クレジットカードによる決済
    - ◇ 処理内容
      - 登録されているクレジットカードで請求金額の支払を実行する
    - ◇ 利用するライブラリ
      - 【SL009】 Stripe Node.js Library (stripe)
    - ◇ 利用するアルゴリズム
      - -
  - クレジットカードでの支払
    - ◇ クレジットカード情報を Stripe を通じて、ユーザーアカウントにひも付いた状態で登録できる
    - ◇ 利用終了時に表示される料金を StripeAPI を通じて支払い、決済が完了する
- 本システム機能の入出力データの仕様
  - 入力
    - ◇ UI からの決済実行操作
      - データの内容
        - 支払実行操作
      - データの形式

- リクエスト
- 利用するデータインターフェース
  - 【IF004】クレジットカード決済 API (Stripe API)
- 出力
  - ◇ 決済実行
    - データの内容
      - 支払の実行
    - データの形式
      - API 実行
    - 利用するデータインターフェース
      - 【IF004】クレジットカード決済 API (Stripe API)

【FN015】レビュー作成・公開 <新規開発>

- 本システム機能の概要
  - ユーザー・車両提供者双方からのレビュー作成及び公開
- 本システム機能の入力・処理・出力のフローチャート

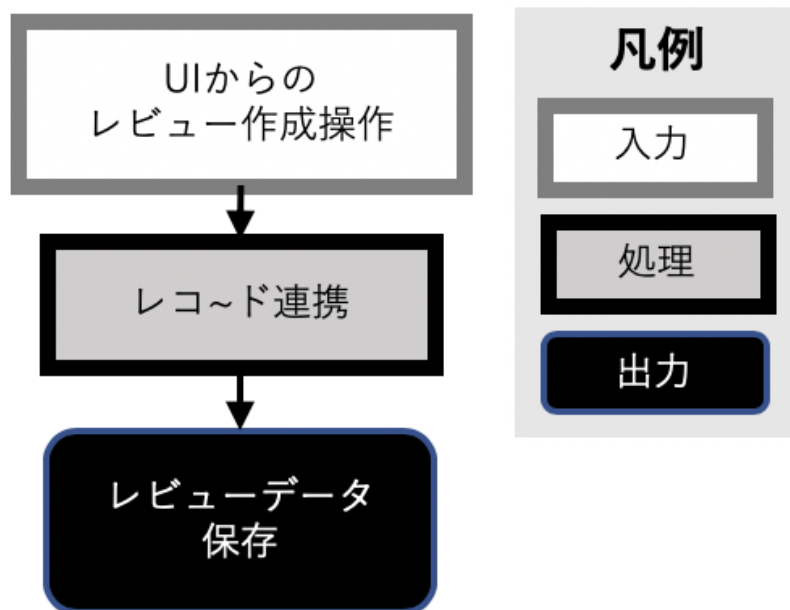


図 2-16 【FN015】のフローチャート

- 本システム機能の処理の詳細
  - レビュー作成・公開
    - ◇ 処理内容
      - 利用の終了が完了した利用に関してのみレビューの作成ができ、利用終了から 14 日間のみレビュー記載が可能
      - ユーザー・車両提供者の双方からレビューが作成された段階、又は 14 日経過後にレビュー作成の有無にかかわらずレビューは作成編集不可となり、公開される
      - 各レビューの集計から総合評価もプロフィール画面に表示され、これはユーザー・車両提

供者での編集はできない

- 作成するレビューは、ユーザー・車両提供者双方に対するもの及びサービス提供事業者のみが確認できるものの2種類存在し、サービス提供事業者側のレビューは外部に公開されない

◇ 利用するライブラリ

- 【SL001】 TypeScript (typescript)
- 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
- 【SL006】 Next.js (next)
- 【SL007】 React (react)

◇ 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

◇ UIからのレビュー作成操作

- データの内容
  - レビュー作成完了操作
- データの形式
  - データ連携
- 利用するデータインターフェース
  - 【IF001】ユーザーデータの作成・取得 IF
  - 【IF010】ページ取得 IF (Firebase Hosting)

➤ 出力

◇ レビューデータ保存

- データの内容
  - レビュー内容の保存・公開
- データの形式
  - データ保存
  - バッジ処理によるスクリプト実行
- 利用するデータインターフェース
  - 【IF001】ユーザーデータの作成・取得 IF
  - 【IF010】ページ取得 IF (Firebase Hosting)

#### 【FN016】車両提供者登録・更新

- 本システム機能の概要
  - 車両提供者 ID と車両提供者 ID にひも付く情報の登録を行い、車両提供者アカウントを作成する
- 本システム機能の入力・処理・出力のフローチャート

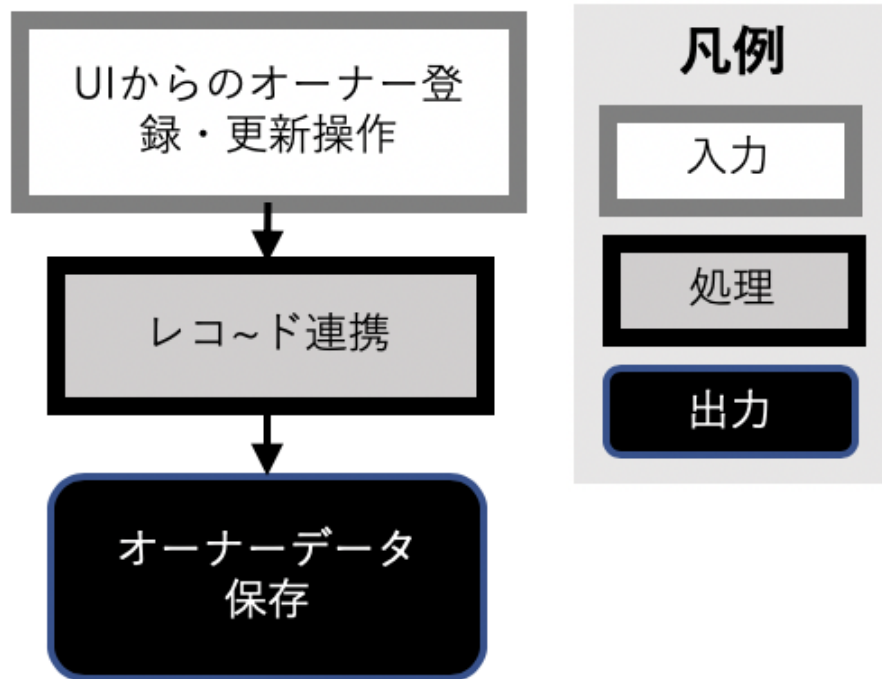


図 2-17 【FN016】のフローチャート

- 本システム機能の処理の詳細
  - 各必要情報の力を伴う車両提供者登録・更新処理
    - ◇ 処理内容
      - 車両提供者アカウントにひも付く人情報や口座情報を入力し、データストレージに一意の OwnerID と共に保存する
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ UIからの車両提供者登録・更新操作
      - データの内容
        - UIからの車両提供者情報登録
      - データの形式
        - データ連携
      - 利用するデータインターフェース
        - 【IF001】ユーザーデータの作成・取得 IF
        - 【IF010】ページ取得 IF (Firebase Hosting)
  - 出力

- ◇ 車両提供者データ保存
  - データの内容
    - データの保存
    - 完了の画面表示
  - データの形式
    - データ保存
    - 画面表示
  - 利用するデータインターフェース
    - 【IF001】ユーザーデータの作成・取得 IF
    - 【IF010】ページ取得 IF (Firebase Hosting)

**【FN017】車両登録・更新<改修機能>**

- 本システム機能の概要
  - 提供する車両情報を登録・更新する
  - 登録する車両 ID にひも付くスマートキーボックスを車両に取り付ける

既存機能の改修内容

新しく開発する共同使用料の計算方法に合わせて、必要な情報として本体取得関連費用・維持関連費用・年間想定走行距離等を追加する。

- 本システム機能の入力・処理・出力のフローチャート

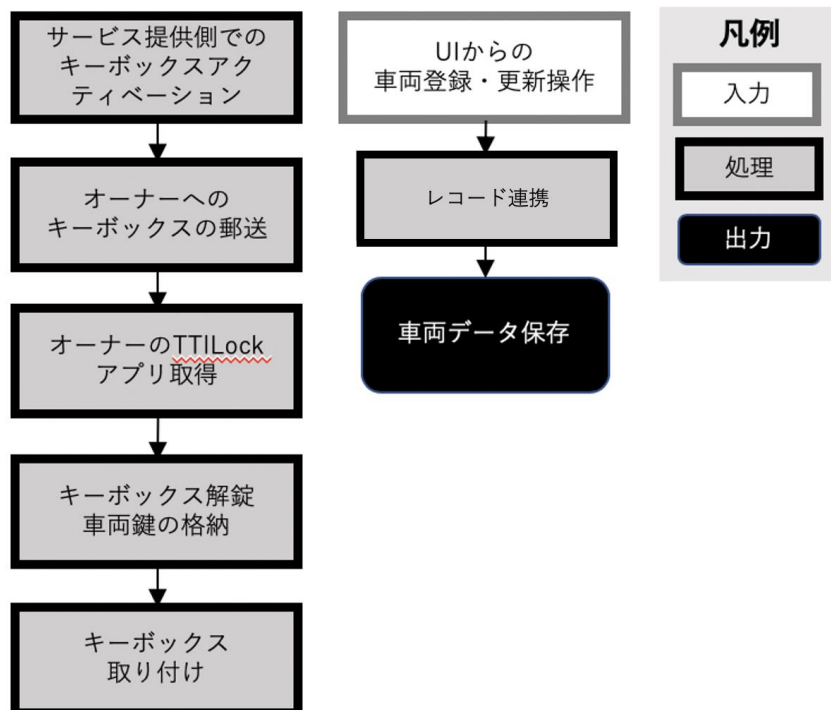


図 2-18 【FN017】のフローチャート (改修後)

- 本システム機能の処理の詳細
  - 車両登録・更新
    - ◇ 処理内容
      - 車検情報等の情報入力によって車両登録行う
      - 車両 ID ごとのスマートキーボックスの割当て及び取り付けを行う
      - 入力項目に合わせて画面を複数に分ける
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 利用するアルゴリズム
      - -
  - 車両 ID ごとのスマートキーボックスの割当て及び取り付け
    - ◇ 処理内容
      - 車両の登録時に車両 ID と各スマートキーボックスに固有の LockID を連携する(サービス側)
      - サービス提供側でスマートキーボックスを下記の手順でアクティベーションを実施した後、車両提供者に郵送する
        - スマートキーボックスを通電状態にし、サービス提供側の TTLock アカウントでログインした状態の TTLock アプリで当該スマートキーボックスと通信し、スマートキーボックスの車両提供者権限を取得する
        - サービス提供側の TTLock アカウントから車両提供者のメールアドレス又は携帯番号に対して、当該車両に取り付けるスマートキーボックスの権限を付与する
      - 車両提供者が TTLock アプリを取得し、スマートキーボックスの権限があることを確認する
      - 車両提供者は TTLock アプリを通じて当該スマートキーボックスを解錠し車両鍵を格納し、車両の後方に取り付ける(当該スマートキーボックスに対して車両提供者はいつでも解錠することが可能)
      - 定期的にスマートキーボックスの電池残量を確認し、電池の交換を実施する
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ UI からの車両登録・更新操作
      - データの内容

- UIからの車両情報の入力
- データの形式
  - データ連携
- 利用するデータインターフェース
  - 【IF020】車両データの作成・取得 IF
- 出力
  - ◇ 車両データ保存
    - データの内容
      - データストレージへの保存
      - 完了のアプリ表示
    - データの形式
      - データの保存
      - 画面表示
    - 利用するデータインターフェース
      - 【IF016】Cloud Storage イベントトリガー
      - 【IF017】Cloud Functions プログラムアクセス

【FN018】公開コントロール

本システム機能の概要車両 ID と車両 ID にひも付く情報の登録を行い、車両データを作成する

- 本システム機能の入力・処理・出力のフローチャート

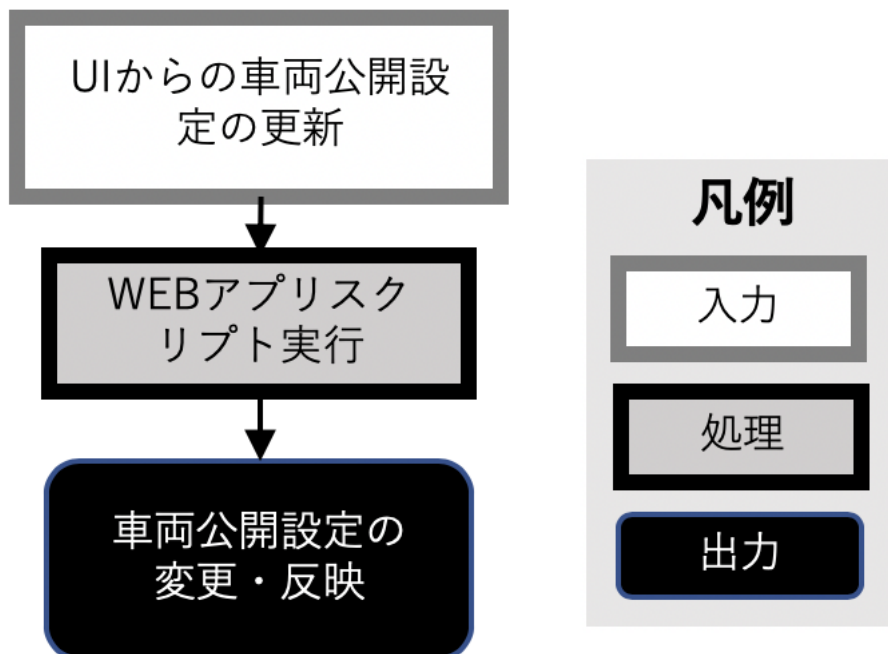


図 2-19 【FN018】のフローチャート

- 本システム機能の処理の詳細
  - 車両の公開状況をスケジュールや繰り返し設定でコントロールできる
    - ◇ 処理内容
      - 任意の公開スケジュール設定を入力し、その設定をデータベースに反映、車両の予約状況

UI に反映する

- ◇ 利用するライブラリ
  - 【SL001】 TypeScript (typescript)
  - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
  - 【SL006】 Next.js (next)
  - 【SL007】 React (react)
- ◇ 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
    - ◇ UI からの車両公開設定の更新
      - データの内容
        - UI からの公開スケジュール変更
      - データの形式
        - データ連携
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
        - 【IF010】 ページ取得 IF (Firebase Hosting)
  - 出力
    - ◇ 車両公開設定の変更・反映
      - データの内容
        - データの保存・設定反映
      - データの形式
        - データ保存・設定反映
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
        - 【IF010】 ページ取得 IF (Firebase Hosting)

**【FN019】 ユーザー拒否 <新規開発>**

- 本システム機能の概要
  - 車両提供者が自らの裁量で、過去のレビューや評価を確認し判断することで、特定のユーザーに車両を貸し出さないことができる
- 本システム機能の入力・処理・出力のフローチャート

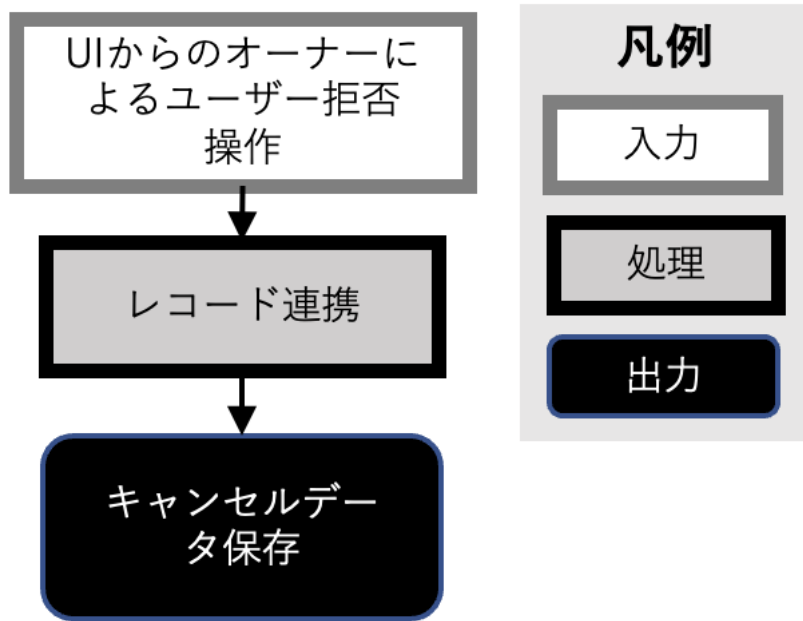


図 2-20 【FN019】のフローチャート

- 本システム機能の処理の詳細
  - ユーザー拒否
    - ◇ 処理内容
      - 予約作成後、車両提供者がユーザーのプロフィール画面等でレビュー・評価を確認する
      - 車両提供者が自らの裁量でこのユーザーとの共同使用契約締結を拒否する(車両提供者予約キャンセル)と、当該ユーザーの予約はキャンセルされ(利用開始時刻前の場合のみ)、今後このユーザーは当該車両に対しての予約作成ができなくなる
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ UIからの車両提供者によるユーザー拒否操作
      - データの内容
        - UIからのユーザー拒否操作
      - データの形式
        - データ連携
      - 利用するデータインターフェース
        - 【IF001】ユーザーデータの作成・取得 IF
  - 出力
    - ◇ キャンセルデータ保存

- データの内容
  - データの保存
  - UI への設定内容の反映
- データの形式
  - データ保存
  - UI への設定内容の反映
- 利用するデータインターフェース
  - 【IF001】 ユーザーデータの作成・取得 IF

【FN020】 車両提供者都合キャンセル <新規開発>

- 本システム機能の概要
  - 取消し、ユーザーが利用できないようにすることができる
- 本システム機能の入力・処理・出力のフローチャート

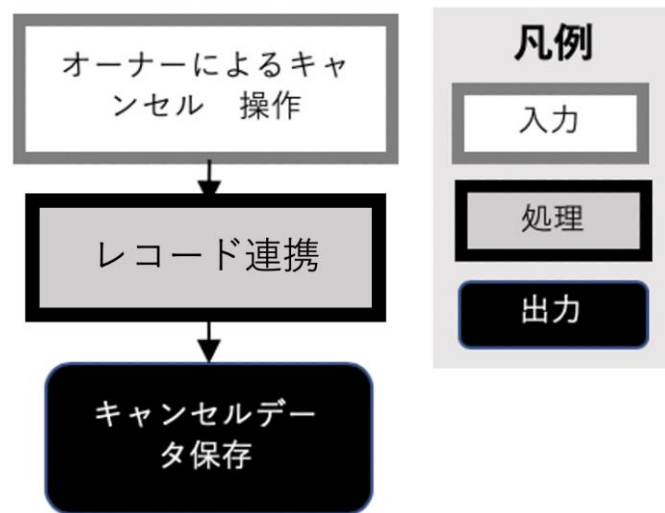


図 2-21 【FN020】のフローチャート

- 本システム機能の処理の詳細
  - キャンセル処理をデータベースに連携する
    - ◇ 処理内容
      - ユーザーが予約作成後、当該車両の車両提供者はこの予約に対して取り消すことができる
      - 取消しの際ユーザーへ事由と共にメッセージを送付し、ユーザーはこれを受け取る
      - ユーザーの許可等は必要なく、車両提供者が独自に実行できるため、急な業務での車両使用等に対応できる
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
  - ◇ 車両提供者によるキャンセル操作
    - データの内容
      - 当該予約に対してのキャンセル操作
    - データの形式
      - データ連携
    - 利用するデータインターフェース
      - 【IF001】ユーザーデータの作成・取得 IF
- 出力
  - ◇ キャンセルデータ保存
    - データの内容
      - データベースへのキャンセル操作反映
    - データの形式
      - データ保存
      - UI 画面表示でのキャンセル
    - 利用するデータインターフェース
      - 【IF001】ユーザーデータの作成・取得 IF

【FN021】予約関連情報確認

- 本システム機能の概要
  - 登録した車両に対して作成された予約に関連する、ユーザープロフィール、予約時間等の情報が確認できる
- 本システム機能の入力・処理・出力のフローチャート

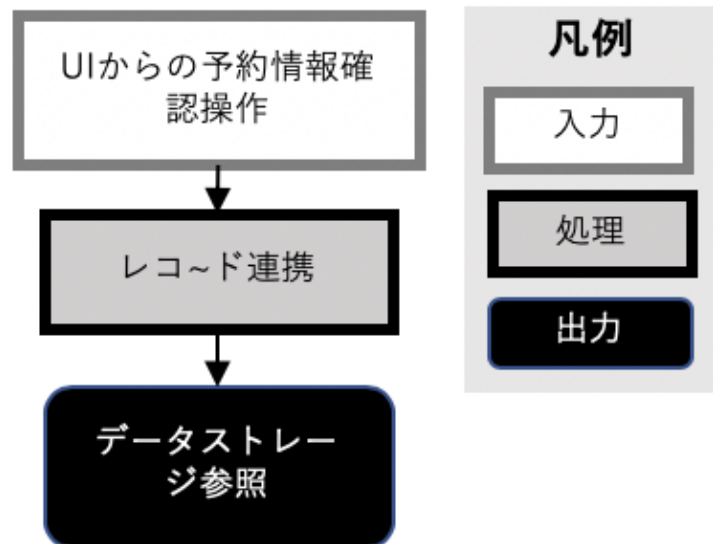


図 2-22 【FN021】のフローチャート

- 本システム機能の処理の詳細
  - 予約作成時の通知から Web アプリ上で、予約を作成したユーザーのプロフィールが確認できる
    - ◇ 処理内容

- 作成された予約データを確認することができる
- ◇ 利用するライブラリ
  - 【SL001】 TypeScript (typescript)
  - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
  - 【SL006】 Next.js (next)
  - 【SL007】 React (react)
- ◇ 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
    - ◇ 予約情報確認操作
      - データの内容
        - UIからのデータ閲覧操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF001】ユーザーデータの作成・取得 IF
  - 出力
    - ◇ データストレージ参照
      - データの内容
        - 予約データの表示
      - データの形式
        - 画面表示
      - 利用するデータインターフェース
        - 【IF001】ユーザーデータの作成・取得 IF
- 【FN022】プロフィール確認
- 本システム機能の概要
  - ユーザー・車両提供者共に相互にプロフィールを確認できる
- 本システム機能の入力・処理・出力のフローチャート

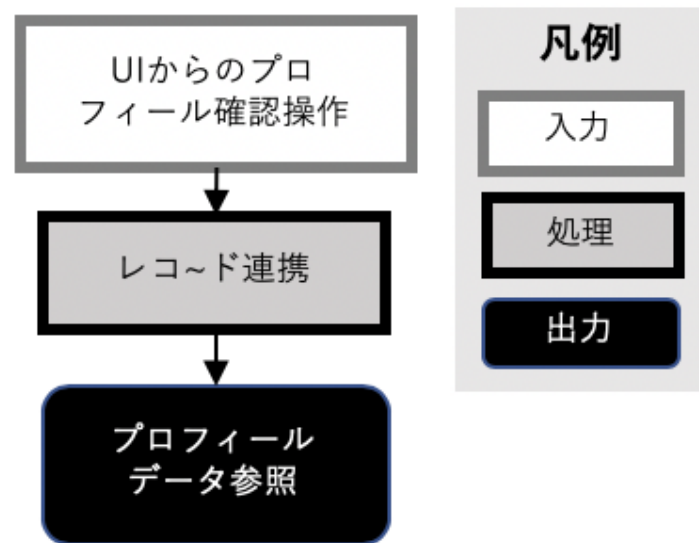


図 2-23 【FN022】のフローチャート

- 本システム機能の処理の詳細
  - プロフィール画面の表示
    - ◇ 処理内容
      - 車両検索時や予約作成時、利用終了後も関連するユーザー及び車両提供者は相互にプロフィールを閲覧することができる
      - プロフィールには過去のレビューコメントや、総合評価が確認できる
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ プロフィール確認操作
      - データの内容
        - UI画面からのプロフィール閲覧操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF001】ユーザーデータの作成・取得 IF
  - 出力
    - ◇ プロフィールデータ参照
      - データの内容
        - UIへのプロフィール画面の表示
      - データの形式

- 画面表示
- 利用するデータインターフェース
  - 【IF001】ユーザーデータの作成・取得 IF

**【FN023】各種ステータス変更 <新規開発>**

- 本システム機能の概要
  - サービス提供事業者がユーザー・車両提供者・車両・予約・レビューに関するステータスを変更できる
- 本システム機能の入力・処理・出力のフローチャート

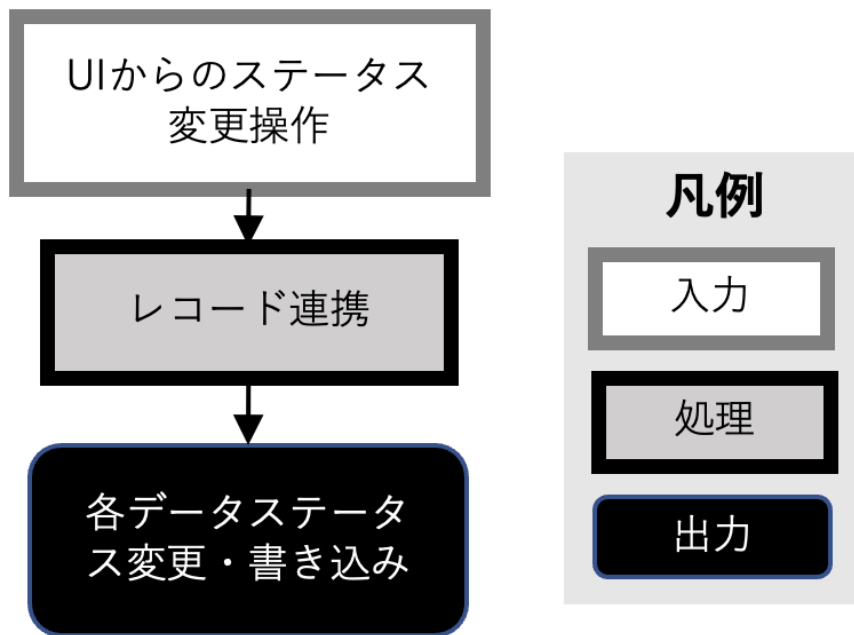


図 2-24 【FN023】のフローチャート

- 本システム機能の処理の詳細
  - 各リソースのステータス変更
    - ◇ 処理内容
      - ユーザーに対し、下記のステータスの中から変更することができる
        - approved (承認済み) /in review (審査中) /stop (一時停止) /deleted (削除)
      - 車両提供者に対し、下記のステータスの中から変更することができる
        - approved (承認済み) /in review (審査中) /stop (一時停止) /deleted (削除)
      - 車両に対し、下記のステータスの中から変更することができる
        - approved (承認済み) /in review (審査中) /stop (一時停止) /deleted (削除)
      - 予約に対し、下記のステータスの中から変更することができる
        - reserved (利用前) /issued (パスコード発行済み) /using (利用中) /invoiced (支払待ち) /paid (支払済み)
      - レビューに対し、下記のステータスの中から変更することができる

- Inreview (レビュー作成中) /reviewed (レビュー作成済み) /published (公開済み) /expired (レビュー期限切れ)
  - 具体のステータス名は変更の可能性あり
- ◇ 利用するライブラリ
  - 【SL001】 TypeScript (typescript)
  - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
  - 【SL006】 Next.js (next)
  - 【SL007】 React (react)
- ◇ 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
    - ◇ ステータス変更操作
      - データの内容
        - 各リソースに対するステータス変更操作 (取消し/キャンセル/停止等)
      - データの形式
        - データ連携
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
  - 出力
    - ◇ 各データステータス変更・書き込み
      - データの内容
        - 各リソースのステータス変更のデータ保存
        - UI 画面への反映
      - データの形式
        - ストレージへのデータ保存
        - UI 画面表示
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF

**【FN024】 売上上限コントロール <新規開発>**

- 本システム機能の概要
  - 共同使用契約に基づく共同使用料受取上限を自動でコントロールする
- 本システム機能の入力・処理・出力のフローチャート

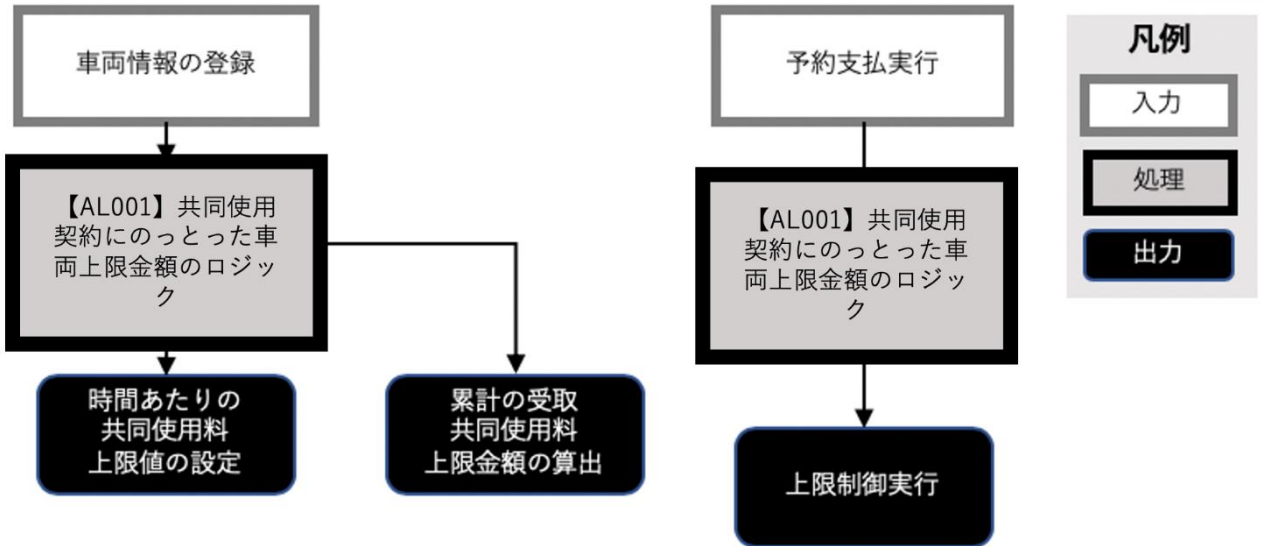


図 2-25 【FN024】のフローチャート

- 本システム機能の処理の詳細
  - 1台当たりの上限金額コントロール
    - ◇ 処理内容
      - **【AL001】共同使用契約にのった車両上限金額のコントロールロジック**に従って共同使用料の受取をコントロールし、車両の公開・非公開制御する
      - 1台当たりの受取使用料の上限を車両登録時の情報を基に決定する
      - 予約に対する支払が発生する都度、上限との照合確認を行い、制御を実行する
    - ◇ 利用するライブラリ
      - **【SL001】TypeScript** (typescript)
      - **【SL003】Firebase Cloud Functions SDK** (firebase-functions)
      - **【SL006】Next.js** (next)
      - **【SL007】React** (react)
    - ◇ 利用するアルゴリズム
      - **【AL001】共同使用契約にのった車両上限金額のコントロールロジックのアルゴリズム**を参照
- 本システム機能の入出力データの仕様
  - 入力
    - ◇ データストレージに予約データの書き込み
      - データの内容
        - 予約利用終了時のデータ連携
      - データの形式
        - データ連携

- 利用するデータインターフェース
  - 【IF001】ユーザーデータの作成・取得 IF
- 出力
  - ◇ 車両の非公開化
    - データの内容
      - 当該車両の累計受取共同使用料の更新
      - 上限ルール規制の場合車両の非公開化
    - データの形式
      - ストレージへのデータ保存
      - 当該車両の累計受取共同使用料の更新
    - 利用するデータインターフェース
      - 【IF001】ユーザーデータの作成・取得 IF

【FN025】ユーザー/車両提供者/車両審査・承認

- 本システム機能の概要
  - 登録されたアカウントデータの内容を確認し、審査する
- 本システム機能の入力・処理・出力のフローチャート

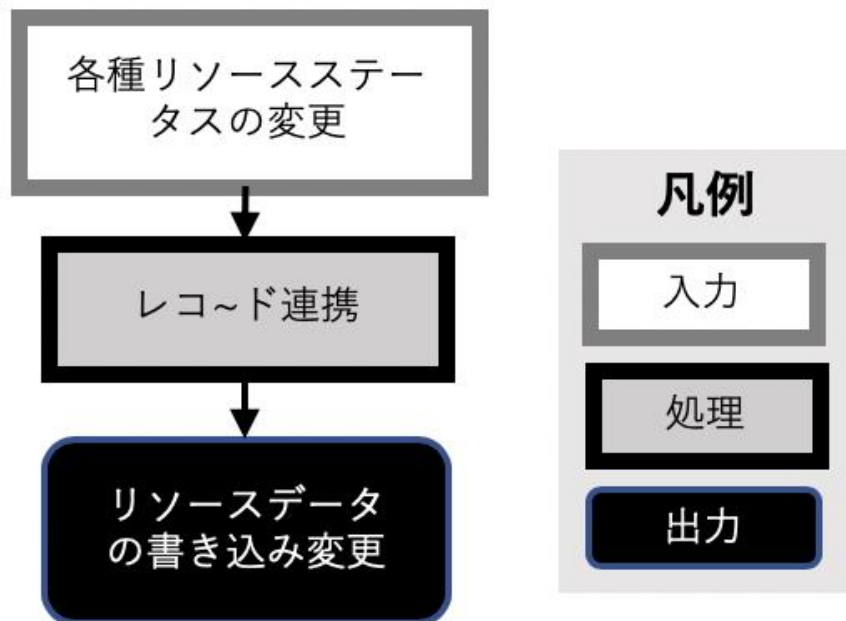


図 2-26 【FN025】のフローチャート

- 本システム機能の処理の詳細
  - 各リソースの審査
    - ◇ 処理内容
      - ユーザー審査の場合、免許画像とそれにひも付く記載事項の内容の正誤を確認し、承認・拒否を判断する
      - 車両提供者審査の場合、登録された連絡先・住所及び口座番号と名義の一致を確認し、承認・拒否を判断する

- 車両審査の場合、登録された車両情報・ステーション情報等の一致を確認し、承認拒否を判断する
  - 過去に登録されている車両の車体番号を参照し、重複がある場合は本申請の車両提供者と過去登録されていた車両提供者が同一か否かを確認し、1 車両に対して同一車両提供者が複数回登録できないように、バリデーションを実施する
    - 同一の場合は承認拒否を実施
  - ◇ 利用するライブラリ
    - 【SL001】 TypeScript (typescript)
    - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
    - 【SL006】 Next.js (next)
    - 【SL007】 React (react)
  - ◇ 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
      - ◇ 各種リソースステータスの変更
        - データの内容
          - UI が面からの審査要求
        - データの形式
          - データ連携
        - 利用するデータインターフェース
          - 【IF001】 ユーザーデータの作成・取得 IF
    - 出力
      - ◇ リソースデータの書き込み変更
        - データの内容
          - 当該リソースを指定ステータスに変更
        - データの形式
          - ストレージへのデータ保存
          - UI 画面への反映
        - 利用するデータインターフェース
          - 【IF001】 ユーザーデータの作成・取得 IF
- 【FN026】 各種データ CSV エクスポート
- 本システム機能の概要
    - ユーザーデータ、車両データ等の各種データを CSV 形式でエクスポートできる
  - 本システム機能の入力・処理・出力のフローチャート

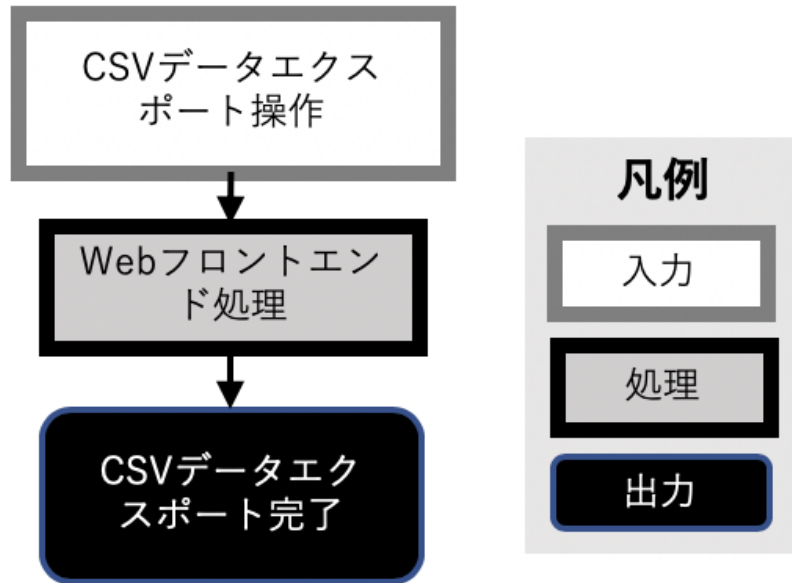


図 2-27 【FN026】のフローチャート

- 本システム機能の処理の詳細
  - Web アプリに保存している各種データの CSV エクスポート
    - ◇ 処理内容
      - 各種データを CSV 形式にしてエクスポートする
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)
    - ◇ 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
    - ◇ CSV データエクスポート操作
      - データの内容
        - CSV エクスポートの操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF001】 ユーザーデータの作成・取得 IF
  - 出力
    - ◇ CSV データエクスポート
      - データの内容
        - エクスポートの処理実施
      - データの形式

- CSV データ作成
- 利用するデータインターフェース
  - 【IF001】 ユーザーデータの作成・取得 IF

【FN027】 Web アプリデータ分析

- 本システム機能の概要
  - Web アプリのアクティビティログをデータとして取得し分析することができる
- 本システム機能の入力・処理・出力のフローチャート

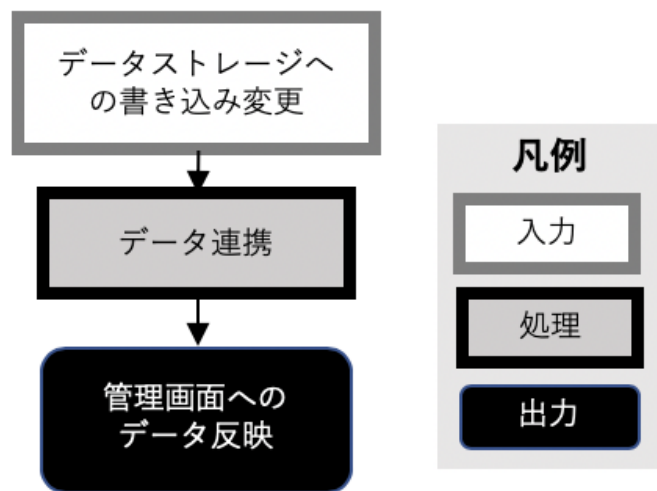


図 2-28 【FN027】のフローチャート

- 本システム機能の処理の詳細
  - BigQuery を用いた Web アプリのアクション分析
    - ◇ 処理内容
      - Web アプリ上で発生したイベントデータを、BigQuery を通じて取得し、分析する
    - ◇ 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
    - ◇ データストレージへの書き込み変更
      - データの内容
        - Web アプリ上でのイベント発生
      - データの形式
        - データ連携
      - 利用するデータインターフェース

- 【IF011】 Cloud Firestore と BigQuery の連携 IF
- 出力
  - ◇ 管理画面へのデータ反映
    - データの内容
      - イベントの計測
      - Bigquery 画面上へのイベント反映
    - データの形式
      - 画面表示
    - 利用するデータインターフェース
      - 【IF011】 Cloud Firestore と BigQuery の連携 IF

【FN028】ホスティング機能

- 本システム機能の概要
  - サーバー内で Web アプリケーションの実行、ドメインの管理を行う
- 本システム機能の入力・処理・出力のフローチャート

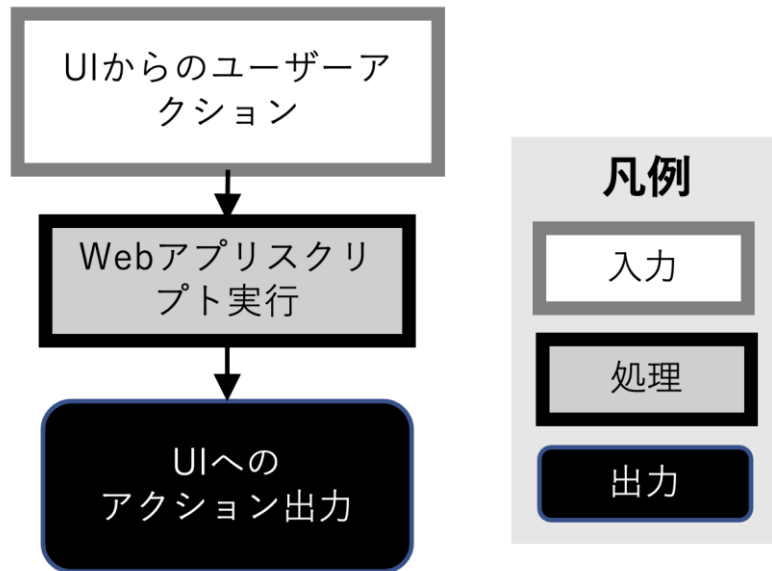


図 2-29 【FN028】のフローチャート

- 本システム機能の処理の詳細
  - Web アプリスクリプト実行
    - ◇ 処理内容
      - ファイルの保存と要求に対するファイル等の配信
      - サーバーサイドスクリプトの実行
      - ドメインのひも付けや SSL/TLS 照明書の適用
    - ◇ 利用するライブラリ
      - 【SL001】 TypeScript (typescript)
      - 【SL003】 Firebase Cloud Functions SDK (firebase-functions)
      - 【SL006】 Next.js (next)
      - 【SL007】 React (react)

◇ 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

◇ ユーザーアクション

- データの内容
  - Web アプリスクリプトの実行要求
- データの形式
  - リクエスト
- 利用するデータインターフェース
  - 【IF001】ユーザーデータの作成・取得 IF

➤ 出力

◇ アクション出力

- データの内容
  - Web アプリスクリプト実行
- データの形式
  - スクリプト実行
- 利用するデータインターフェース
  - 【IF001】ユーザーデータの作成・取得 IF

【FN029】Poster 通知

- 本システム機能の概要
  - LINE 管理ツールの「Poster」を通じて通知をユーザー・車両提供者に送信する
- 本システム機能の入力・処理・出力のフローチャート

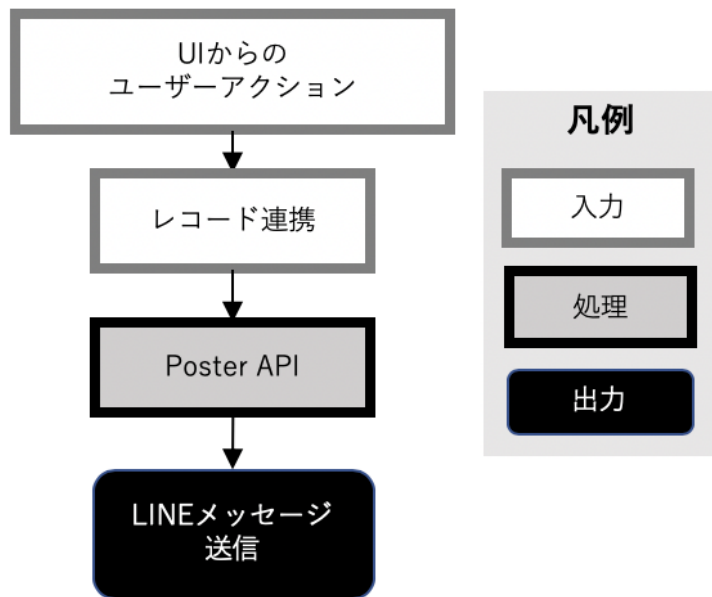


図 2-30 【FN029】のフローチャート

- 本システム機能の処理の詳細

➤ Poster API

◇ 処理内容

- アカウント認証が LINE の場合は LINE で、下記タイミングにおける通知を送信する
  - アカウント（ユーザー・車両提供者・車両）の審査時
  - 予約の作成時・キャンセル時・変更時・利用開始時・利用終了時
  - レビュー作成時、レビュー公開時

◇ 利用するライブラリ

- 【SL012】 LINE Messaging API SDK (@line/bot-sdk)

◇ 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

◇ レコード連携

- データの内容
  - CloudFunction からの API 連携
- データの形式
  - リクエスト
- 利用するデータインターフェース
  - 【IF012】 LINE ユーザー管理 API (Poster API)

➤ 出力

◇ LINE メッセージ送信

- データの内容
  - Poster API から CloudFunction への連携
- データの形式
  - LINE メッセージ送信
- 利用するデータインターフェース
  - 【IF012】 LINE ユーザー管理 API (Poster API)

【FN030】 メール通知

- 本システム機能の概要
  - メールによる各種通知を送信する。
- 本システム機能の入力・処理・出力のフローチャート

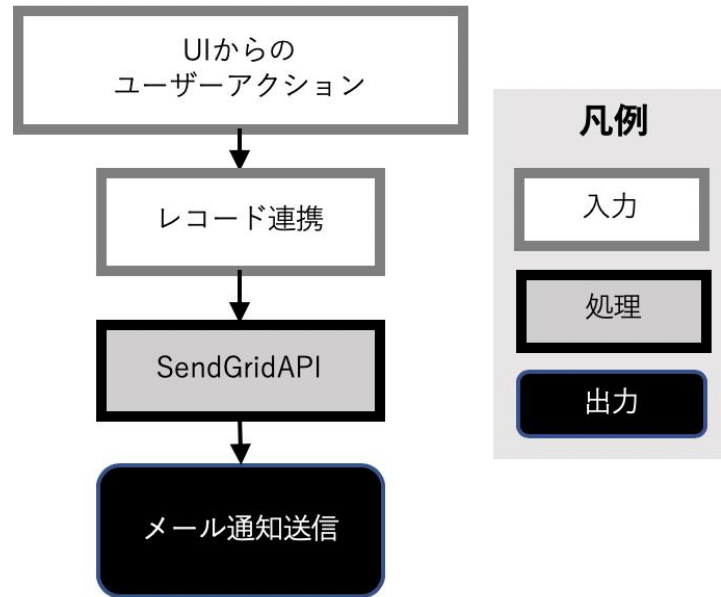


図 2-31 【FN030】のフローチャート

- 本システム機能の処理の詳細
  - SendGridAPI
    - ◇ 処理内容
      - アカウント認証が Google、パスワード方式（メールアドレス）の場合はメールで、下記のタイミングにおける通知を送信する
        - アカウント（ユーザー・車両提供者・車両）の審査時
        - 予約の作成時・キャンセル時・変更時・利用開始時・利用終了時
        - レビュー作成時、レビュー公開時
    - ◇ 利用するライブラリ
      - 【SL013】 SendGrid Mail SDK (@sendgrid/mail)
    - ◇ 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
    - ◇ レコード連携
      - データの内容
        - CloudFunction からの API 連携
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF007】メール送信 API（SendGrid API）
  - 出力
    - ◇ メール通知送信
      - データの内容

- SendGridAPI から CloudFunction への連携
- データの形式
  - メール送信
- 利用するデータインターフェース
  - 【IF007】メール送信 API (SendGrid API)

【FN031】 Slack 通知

- 本システム機能の概要
  - サービス提供事業者側に各種通知を送信する。
- 本システム機能の入力・処理・出力のフローチャート

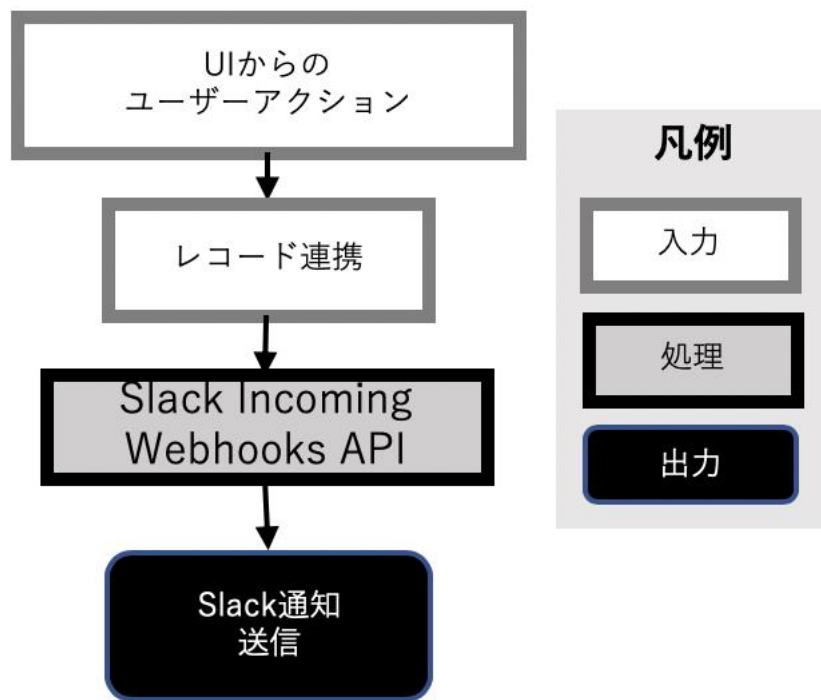


図 2-32 【FN031】のフローチャート

- 本システム機能の処理の詳細
  - Slack 送付による通知を行う
    - ◇ 処理内容
      - 下記イベント発生タイミングにおける通知を Slack で送信する
        - アカウント（ユーザー・車両提供者・車両）の審査時
        - 予約の作成時・キャンセル時・変更時・利用開始時・利用終了時
        - レビュー作成時、レビュー公開時
    - ◇ 利用するライブラリソフトウェア・ライブラリを参照
      - 【SL014】 Slack Incoming Webhooks SDK (@slack/Webhook)
    - ◇ 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
  - ◇ レコード連携
    - データの内容
      - CloudFunction からの API 連携
    - データの形式
      - リクエスト
    - 利用するデータインターフェース
      - 【IF015】 Slack メッセージ投稿 IF (Slack Incoming Webhooks)
- 出力
  - ◇ Slack 通知送信
    - データの内容
      - SlackIncoming Webhooks API から CloudFunction への連携
    - データの形式
      - Slack 通知送信
    - 利用するデータインターフェース
      - 【IF015】 Slack メッセージ投稿 IF (Slack Incoming Webhooks)

【FN032】 電子決済

- 本システム機能の概要
  - サービス利用に伴う電子決済 (PayPay) を実行する
- 本システム機能の入力・処理・出力のフローチャート

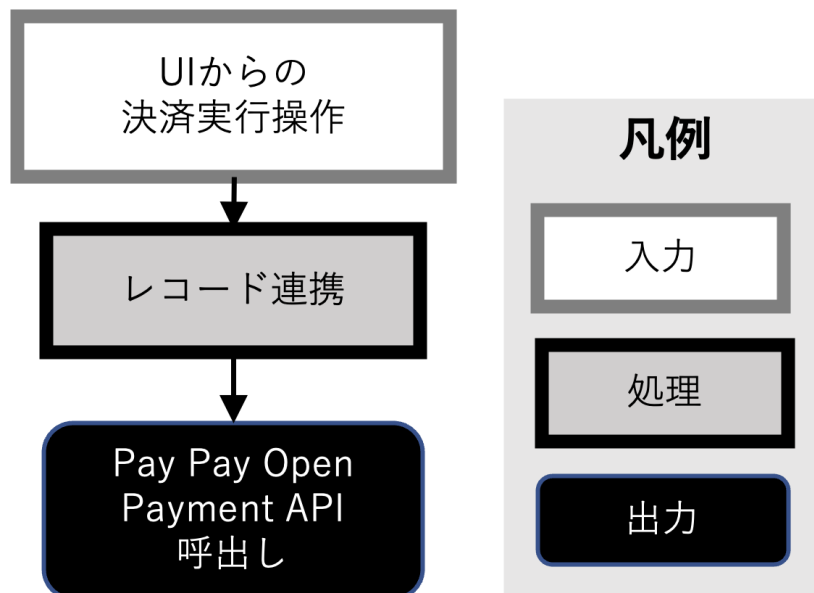


図 2-33 【FN032】 のフローチャート

- 本システム機能の処理の詳細
  - 電子決済
    - ◇ 処理内容

- 電子決済で請求金額の支払を実行する
  - 決済方法を選択した際に表示される QR コードを電子決済アプリで読み込む
  - 金額の入力を手動で行い、決済をし、決済 ID をチャットアプリを通じて送付する
  - サービス提供事業者でそれを手動で確認し、「支払待ち」から「支払の完了」ステータスに変更する
- ◇ 利用するライブラリ
  - 【SL011】 PayPay OPA SDK
- ◇ 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
    - ◇ 決済実行操作
      - データの内容
        - 支払実行操作
      - データの形式
        - リクエスト
      - 利用するデータインターフェース
        - 【IF005】 PayPay 決済 API (PayPay Open Payment API)
  - 出力
    - ◇ PayPay Open Payment API 呼出し
      - データの内容
        - 支払の実行
      - データの形式
        - API 実行
      - 利用するデータインターフェース
        - 【IF005】 PayPay 決済 API (PayPay Open Payment API)
- 【FN033】 Clarity 分析
  - 本システム機能の概要
    - Microsoft Clarity を使用したユーザー行動のデータ分析
  - 本システム機能の入力・処理・出力のフローチャート

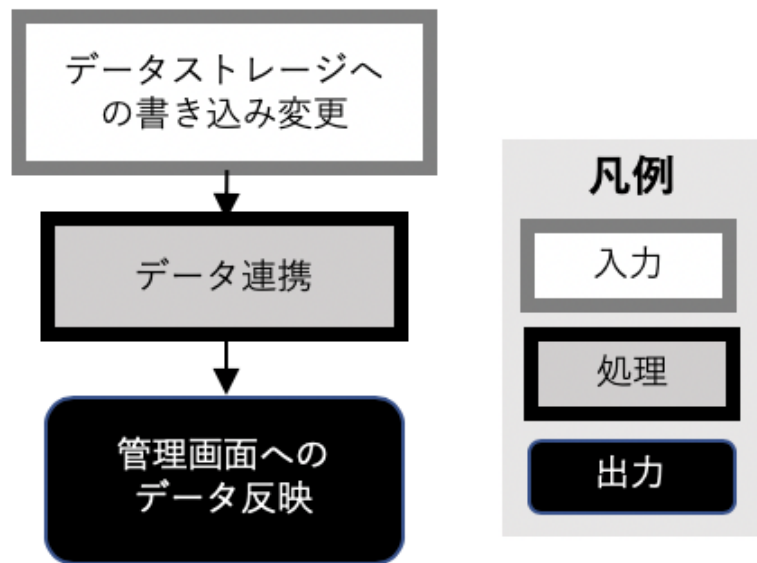


図 2-34 【FN033】のフローチャート

- 本システム機能の処理の詳細
  - Clarity を用いたユーザーの操作画面観察、ヒートマップ等のデータ取得、分析
    - ◇ 処理内容
      - Web アプリ上で発生したユーザーの操作画面観察、ヒートマップ等のデータ取得分・析
    - ◇ 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
    - ◇ データストレージへの書き込み変更
      - データの内容
        - Web アプリ上でのイベント発生
      - データの形式
        - データ連携
      - 利用するデータインターフェース
        - 【IF014】行動分析ツール連携 IF (Microsoft Clarity)
  - 出力
    - ◇ 管理画面へのデータ反映
      - データの内容
        - イベントの計測
        - Clarity 画面上へのイベント反映
      - データの形式
        - 画面表示

- 利用するデータインターフェース
  - 【IF014】 行動分析ツール連携 IF (Microsoft Clarity)

【FN034】 GA4 分析

- 本システム機能の概要
  - Google Analytics 4 を通したユーザー行動のデータ分析
- 本システム機能の入力・処理・出力のフローチャート

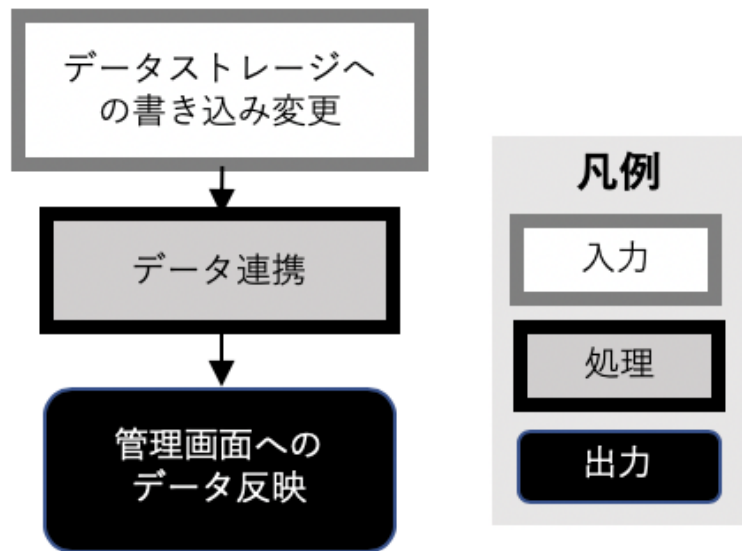


図 2-35 【FN034】 のフローチャート

- 本システム機能の処理の詳細
  - GoogleAnalytics を用いた Web アプリのアクション分析
    - ◇ 処理内容
      - Web アプリ上で発生したイベント、各ページにおける滞在時間や発生イベントのデータ取得、分析
    - ◇ 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
    - ◇ データストレージへの書き込み変更
      - データの内容
        - Web アプリ上でのイベント発生
      - データの形式
        - データ連携
      - 利用するデータインターフェース
        - 【IF013】 ウェブ解析ツール連携 IF (Google Analytics 4)
  - 出力

- ◇ 管理画面へのデータ反映
  - データの内容
    - イベントの計測
    - GoogleAnalytics 画面上へのイベント反映
  - データの形式
    - 画面表示
  - 利用するデータインターフェース
    - 【IF013】ウェブ解析ツール連携 IF (Google Analytics 4)

## 2-1-4. ソフトウェア・ライブラリ (SL) の詳細

表 2-2 ソフトウェア・ライブラリー一覧

※朱文字：新規開発・既存改修

ID	名称	バージョン	内容
SL001	TypeScript (typescript)	5.8.3	Web アプリのフロントエンドとバックエンドのビジネスロジックを実装する言語。
SL002	Firebase CLI (firebase-tools)	14.5.1	Firebase のコマンドラインインターフェース。
SL003	Firebase Cloud Functions SDK (firebase-functions) (firebase-functions)	5.0.1	Web アプリのバックエンドを実装するための SDK。
SL004	Firebase Admin SDK (firebase-admin)	12.3.0	Web アプリのバックエンドから Firebase のサービスにアクセスするための SDK。
SL005	Firebase (firebase)	9.23.0	Web アプリのフロントエンドから Firebase のサービスにアクセスするための SDK。
SL006	Next.js (next)	12.3.4	Web アプリのフロントエンドを実装するためのフレームワーク。
SL007	React (react)	18.2.0	Web アプリのフロントエンドの UI を実装するためのライブラリ。
SL008	Google Maps JavaScript API React Wrapper (@googlemaps/react-wrapper)	1.2.0	Web アプリのフロントエンドで Google マップの表示をするためのライブラリ。
SL009	Stripe Node.js Library (stripe)	16.6.0	Web アプリのバックエンドでクレジットカードによる請求を発行するための SDK。
SL010	Stripe.js (stripe-js)	4.10.0	Web アプリのフロントエンドでクレジットカードによる請求に対する支払を完了させる SDK。

SL011	PayPay OPA SDK (@PayPayopa/PayPayopa -sdk-node)	2.1.0	Web アプリのバックエンドで PayPay による請求を発行するための SDK。
SL012	LINE Messaging API SDK (@line/bot-sdk)	7.7.0	Web アプリのバックエンドから LINE メッセージを送信するための SDK。
SL013	SendGrid Mail SDK (@sendgrid/mail)	8.1.4	Web アプリのバックエンドからメールを送信するための SDK。
SL014	Slack Incoming Webhooks SDK (@slack/Webhook)	7.0.2	Web アプリのバックエンドから Slack にメッセージを投稿するための SDK。

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）を行うソフトウェア・ライブラリを**朱文字**で示す。

#### 【SL001】 TypeScript (typescript)

- ベンダー
  - Microsoft
- 公式サイト
  - <https://github.com/microsoft/TypeScript> (typescript)
- 本ソフトウェア・ライブラリの概要
  - Microsoft によって開発されたオープンソースのプログラミング言語で、JavaScript に静的型付けとクラスベースのオブジェクト指向機能を追加したスーパーセットである。TypeScript (typescript) で書かれたコードは JavaScript にコンパイルされるため、既存の JavaScript コードとの互換性が高く、大規模なアプリケーションの開発に向いている。
- 開発するシステムにおいて利用する機能の詳細
  - 静的型付け：変数や関数の引数・戻り値に型を明示できるため、実行前に型エラーを検出でき、開発時のバグを減らすことができる。
  - 最新の JavaScript 機能のサポート：ECMAScript の将来の機能を事前に使用でき、TypeScript (typescript) がそれを JavaScript にトランスパイルすることで、古い環境でも動作させることができる。
  - 強力な IDE サポート：型情報に基づいて補完、リファクタリング、ナビゲーションなどの高度な機能を提供し、Visual Studio Code などのエディタと非常に親和性が高い。

#### 【SL002】 Firebase CLI (firebase-tools)

- ベンダー
  - Google
- 公式サイト
  - <https://github.com/firebase/firebase-tools>

- 本ソフトウェア・ライブラリの概要
  - Firebase プロジェクトの管理と操作をコマンドラインから実行できる CLI ツール。開発者は Firebase の各種サービスをローカルでエミュレート・テストし、デプロイすることが可能になる。
- 開発するシステムにおいて利用する機能の詳細
  - プロジェクトの設定：プロジェクトの初期化、各種サービスの有効化、firebase.json などの設定ファイルの生成を支援する。
  - プロジェクトへのデプロイ：各種サービスについて、アプリケーションや設定のデプロイを行うことができる。
  - エミュレーターの実行：ローカル環境でエミュレーターを実行し、クラウドを使用せずに開発・テストを行うことができる。

#### 【SL003】 Firebase Cloud Functions SDK (firebase-functions)

- ベンダー
  - Google
- 公式サイト
  - <https://github.com/firebase/firebase-functions>
- 本ソフトウェア・ライブラリの概要
  - Firebase の Cloud Functions for Firebase 機能を Node.js ベースで記述するためのライブラリ。開発者はこのライブラリを使って、HTTP リクエスト、Firebase Authentication、Cloud Firestore、Cloud Storage for Firebase などのイベントに対応するサーバーレス関数を記述し、Google Cloud 上で実行することができる。
- 開発するシステムにおいて利用する機能の詳細
  - イベント駆動型関数の作成：Firebase Authentication、Cloud Firestore、Cloud Storage for Firebase などの Firebase サービスに連動する関数を定義できる。
  - HTTP 関数の作成：REST API のように HTTP リクエストに応答する関数を定義できる。
  - 環境設定と構成：関数内で使用する環境変数の管理や、リージョン・メモリ・タイムアウトなどのリソース設定を行える。

#### 【SL004】 Firebase Admin SDK (firebase-admin)

- ベンダー
  - Google
- 公式サイト
  - <https://github.com/firebase/firebase-admin-node>
- 本ソフトウェア・ライブラリの概要
  - Firebase Admin SDK の Node.js 向け実装で、サーバーサイド環境から Firebase の各種サービスを管理・操作できる SDK。管理者権限でアクセス可能なため、アプリケーションのバックエンド処理や管理タスクに利用される。
- 開発するシステムにおいて利用する機能の詳細
  - 管理者権限での認証操作：ユーザーの作成・削除、カスタムトークンの発行など、Firebase Authentication を管理できる。
  - 管理者権限での Cloud Firestore 操作：高度な読み書き操作や管理機能を提供する。

- 管理者権限での Cloud Storage for Firebase 操作： アップロードされたファイルの読み書き、メタデータの取得・更新、ファイルの削除などの管理機能を提供する。

#### 【SL005】 Firebase (firebase)

- ベンダー
  - Google
- 公式サイト
  - <https://github.com/firebase/firebase-js-sdk>
- 本ソフトウェア・ライブラリの概要
  - Firebase の公式 JavaScript SDK で、フロントエンド環境で Firebase の各種サービスを利用するためのライブラリ。クライアントサイドで Firebase の各種サービスを簡単に統合・操作できる。
- 開発するシステムにおいて利用する機能の詳細
  - 認証機能： パスワード認証、Google、LINE などの OAuth プロバイダーによる認証など多彩な認証方法をサポート。
  - Cloud Firestore 操作： 単純な読み書き操作をモバイルアプリから直接実行可能。
  - Cloud Storage for Firebase 操作： ファイルのアップロード・ダウンロードをブラウザやモバイルアプリから直接実行可能。

#### 【SL006】 Next.js (next)

- ベンダー
  - Vercel
- 公式サイト
  - <https://github.com/vercel/next.js>
- 本ソフトウェア・ライブラリの概要
  - React ベースのフレームワークで、サーバーサイドレンダリング (SSR) や静的サイト生成 (SSG)、API ルートの構築などを簡単に行える。パフォーマンス最適化や SEO 対策が組み込まれており、モダンな Web アプリケーション開発に広く利用されている。
- 開発するシステムにおいて利用する機能の詳細
  - サーバーサイドレンダリング (SSR)： リクエスト時にサーバーで HTML を生成し、SEO や初期表示速度を向上。
  - 静的サイト生成 (SSG)： ビルド時に HTML を生成し、高速なページ配信を実現。
  - API ルート： API エンドポイントを簡単に作成でき、バックエンドレスな構成が可能。

#### 【SL007】 React (react)

- ベンダー
  - Meta
- 公式サイト
  - <https://github.com/facebook/react>
- 本ソフトウェア・ライブラリの概要

- ユーザーインターフェース構築のための JavaScript ライブラリで、コンポーネントベースの設計を特徴とする。仮想 DOM を使って効率的に UI を更新し、シングルページアプリケーション (SPA) やモバイルアプリの開発に広く利用されている。
- 開発するシステムにおいて利用する機能の詳細
  - コンポーネントベース開発： UI を再利用可能なコンポーネントとして設計でき、ソースコードの保守性と拡張性を高める。
  - 仮想 DOM： 実際の DOM 操作を最小限に抑え、パフォーマンスを最適化。
  - 宣言的 UI： 状態に応じて UI の見た目を宣言的に記述でき、ソースコードが直感的かつ管理しやすくなる。

#### 【SL008】 Google Maps JavaScript API React Wrapper (@googlemaps/react-wrapper)

- ベンダー
  - Google
- 公式サイト
  - <https://github.com/googlemaps/react-wrapper>
- 本ソフトウェア・ライブラリの概要
  - Google Maps JavaScript API を React アプリケーションに統合するための軽量なラッパーコンポーネント。React のコンポーネントとして Google Maps を扱えるようにし、マップの読み込み状態管理や API の初期化を簡素化する。
- 開発するシステムにおいて利用する機能の詳細
  - ラッパーコンポーネント： Google Maps API の初期化やマップ表示を React コンポーネントとして簡単に利用可能。
  - ローディングステータス管理： マップの読み込み状態 (loading、success、failure) を管理し、UI の状態に応じた表示が可能。
  - 柔軟な拡張性： 子コンポーネントで Google Maps の各種 API (マーカー、ポリゴン、イベントリスナーなど) を操作できる。

#### 【SL009】 Stripe Node.js Library (stripe)

- ベンダー
  - Stripe, Inc.
- 公式サイト
  - <https://github.com/stripe/stripe-node>
- 本ソフトウェア・ライブラリの概要
  - Stripe の公式 Node.js 用 SDK で、サーバーサイドから安全に決済処理や顧客管理、請求管理などの Stripe API を操作できるライブラリ。クレジットカード決済やサブスクリプション管理、チャージバック対応など、オンライン決済の実装に広く利用されている。
- 開発するシステムにおいて利用する機能の詳細
  - 支払処理： クレジットカードや Apple Pay、Google Pay など多様な決済方法に対応した支払の作成・管理。
  - 顧客管理： 顧客情報の作成・更新、支払方法の保存、サブスクリプションの管理などをサポート。
  - 請求書・請求管理： 請求書の作成や送付、支払の追跡、払戻し処理などを行うことが可能。

【SL010】 Stripe.js (stripe-js)

- ベンダー
  - Stripe Inc.
- 公式サイト
  - <https://github.com/stripe/stripe-js>
- 本ソフトウェア・ライブラリの概要
  - Stripe の公式 JavaScript クライアントライブラリで、主にフロントエンド（ブラウザ環境）での安全な決済フォームの構築をサポートする。カード情報を直接サーバーに送信せず、PCI 準拠のトークン化を実現するために使われる。
- 開発するシステムにおいて利用する機能の詳細
  - 安全な決済 UI の提供： Stripe Elements を使ったカード入力フォームの作成とカスタマイズが容易。
  - トークン化： ユーザーのカード情報をトークン化し、サーバーに安全に送信できる。
  - 支払フローの管理： 支払認証（3D セキュアなど）や決済Intentの確認をブラウザ上でサポート。

【SL011】 PayPay OPA SDK (@PayPayopa/PayPayopa-sdk-node)

- ベンダー
  - PayPay 株式会社
- 公式サイト
  - <https://github.com/PayPay/PayPayopa-sdk-node>
- 本ソフトウェア・ライブラリの概要
  - Node.js 環境向けの PayPay 公式 SDK であり、QR コード決済をはじめとした PayPay 決済機能をサーバーサイドで実装可能にするものである。加盟店向けの決済フローを簡潔に構築できる。
- 開発するシステムにおいて利用する機能の詳細
  - QR コード決済の生成： 支払用 QR コードを生成し、ユーザーがスマートフォンでスキャンして決済を行えるようにする。
  - 支払状況の確認： 決済ステータスをリアルタイムで取得し、支払の完了状況を管理できる。
  - JWT (JSON WEB TOKEN) 認証： API 通信のセキュリティを確保するために JWT (JSON WEB TOKEN) による認証を行う。

【SL012】 LINE Messaging API SDK (@line/bot-sdk) (@line/bot-sdk)

- ベンダー
  - LINE ヤフー株式会社
- 公式サイト
  - <https://github.com/line/line-bot-sdk-nodejs>
- 本ソフトウェア・ライブラリの概要
  - LINE Messaging API を利用したボット開発を支援する Node.js 向けの公式 SDK である。LINE プラットフォームとの連携を容易にし、メッセージの送受信やユーザー情報の取得、Webhook の署名検証など、ボット開発に必要な機能を提供する。
- 開発するシステムにおいて利用する機能の詳細

- メッセージ送信：テキスト、画像、動画、音声、スタンプ、位置情報など、多様なメッセージをユーザーに送信できる。
- ユーザー情報の取得：ユーザーのプロフィール情報を取得し、個別に対応ができる。
- Webhook 署名の検証：受信したリクエストが LINE プラットフォームからのものであるかを検証する機能を備えている。

#### 【SL013】 SendGrid Mail SDK (@sendgrid/mail)

- ベンダー
  - Twilio SendGrid, Inc.
- 公式サイト
  - <https://github.com/sendgrid/sendgrid-nodejs>
- 本ソフトウェア・ライブラリの概要
  - Node.js 環境で SendGrid の Web API v3 を利用してメールを送信するための公式 SDK である。シンプルな API 設計により、開発者は迅速にメール送信機能を実装できる。
- 開発するシステムにおいて利用する機能の詳細
  - シンプルなメール送信：テキスト及び HTML 形式のメールを簡単に送信できる。
  - 複数宛先への送信：複数の受信者に対して一度の API 呼出しでメールを送信できる。
  - 動的テンプレートの利用：SendGrid で作成した動的テンプレートを使用し、パラメータを埋め込んだメールを送信できる。

#### 【SL014】 Slack Incoming Webhooks SDK (@slack/Webhook)

- ベンダー
  - Slack Technologies, Inc.
- 公式サイト
  - <https://github.com/slackapi/node-slack-sdk/tree/main/packages/Webhook>
- 本ソフトウェア・ライブラリの概要
  - Slack の Incoming Webhooks を利用してチャットチャンネルへ通知を送信するための公式 Node.js クライアントライブラリである。シンプルな API を提供し、アプリケーションや CI/CD パイプライン等から容易にメッセージを投稿できる。
- 開発するシステムにおいて利用する機能の詳細
  - メッセージ送信：テキスト、添付ファイル、Block Kit 形式のリッチメッセージを送信できる。
  - デフォルトパラメータ設定：アイコン絵文字やユーザー名など、Webhook 利用時に再利用可能なデフォルトオプションを指定可能である。
  - プロキシ利用対応：HTTPS エージェント指定により、企業ネットワーク内のプロキシ経由での送信に対応できる。

## 2-1-5. 数理モデル・アルゴリズム（AL）の詳細

表 2-3 数理モデル・アルゴリズム一覧

※朱文字：新規開発・既存改修

ID	名称	説明	アルゴリズムを利用した機能
AL001	共同使用契約にのっとり車両上限金額のコントロールロジック	● 受け取る共同使用料が車両の初期費用、維持費用の合計を超過しないようにするためのロジック。	FN024

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】共同使用契約にのっとり車両上限金額のコントロールロジック

- 本アルゴリズムの概要
  - 本サービスを含む CtoC カーシェアリング事業においては、共同使用の考え方に基づくサービス設計・運用が必要である
    - ◇ 自家用自動車を「業として有償で貸し渡」す場合（道路運送法 80 条 1 項。いわゆるレンタカー事業）には、国土交通大臣の許可を受けることが必要となる。
    - ◇ もっとも、いわゆる CtoC カーシェアリング事業において、車両提供者である個人又は法人が自ら許可を取得することは現実的ではなく、多くの CtoC カーシェアリング事業が、車両提供者及びユーザーにおける自家用自動車の共同使用を行うためのサービスとして設計され、運営されている。
    - ◇ このような共同使用に関する考え方については、過去、法令適用事前確認手続（いわゆるノーアクションレター）を通じて、国土交通省自動車局旅客課長から回答がなされているところ（平成 26 年 12 月 19 日付け法令適用事前確認手続 回答書）を通じ、本サービスは、同回答が示す共同使用の考え方に従い設計・運用されるものであり、本サービスにおける自動車の利用は「業として有償で貸し渡」すものには該当せず、道路運送法 80 条 1 項の適用を受けないものである。
  - 共同使用の考え方について
    - ◇ 過去のノーアクションレター（平成 26 年 12 月 19 日付け法令適用事前確認手続 回答書；令和 3 年 6 月 7 日付け法令適用事前確認手続 回答書；令和 4 年 10 月 26 日付け法令適用事前確認手続 回答書）において、「自家用自動車の共同使用」とは、同一の自動車を二以上の者のそれぞれが、自己の欲求充足のために主体的な立場において使用することをいうとされている。「主体的な立場において使用する」ということの要件として、共同で使用する者のそれぞれが自動車の使用及び管理に関する実質的な権限と責任を有することが必要であるとされる。また、「自動車を使用する」とは、自動車の管理も合わせて行うことを常態とするものであり、日常

点検整備だけでなく、定期点検整備等も行うものとされる。

◇ 同回答では、その上で、以下のような考え方の具体例が示されている。

【共同使用の考え方の具体例】

(平成26年12月19日付け法令適用事前確認手続 回答書；令和3年6月7日付け法令適用事前確認手続 回答書；令和4年10月26日付け法令適用事前確認手続 回答書)

- 同一の自動車を二以上の者それぞれが、自己の欲求充足のために主体的な立場において使用すること
- 共同使用者が具体的に特定され、自動車の使用及び管理に関してあらかじめの合意が存在していること
- 共同使用者それぞれが自動車の使用及び管理に関する権限と責任を有すること
- 共同使用料は営利を目的とせず、自動車の維持費の範囲内であること

➤ 前提となる本サービスの概要

◇ 前提として、本サービスは、地域における生活者を中心としたサービスを提供する目的で、車両提供者兼共同使用者（以下「車両提供者」という。）及び当該自家用自動車の共同使用を希望する者（以下「ユーザー」という。）間の共同使用に係る契約（以下「共同使用契約」という。）を媒介・支援するためのサービスである。

◇ なお、本サービスを利用するに当たって、共同使用契約の締結を希望する者は、当社（株式会社ディー・エヌ・エー）の定める約款（以下「利用規約」という。）に同意し、当社との契約関係に入ることを要するものとする。

- 過去事例として、以下が存在している。

➤ ① 株式会社ディー・エヌ・エーによる法令適用事前確認手続の照会書（<https://www.mlit.go.jp/common/001262675.pdf>）に対する、2014年12月19日付国土交通省自動車局旅客課長回答（<https://www.mlit.go.jp/common/001262677.pdf>）（以下「過去事例①」という。）

## 法令適用事前確認手続（照会書）

2014年10月31日

国土交通省 自動車局旅客課長 殿

照会者名：株式会社ディー・エヌ・エー  
代表取締役 守安 功  
住所：東京都渋谷区渋谷二丁目 21 番 1 号

下記について、照会いたします。

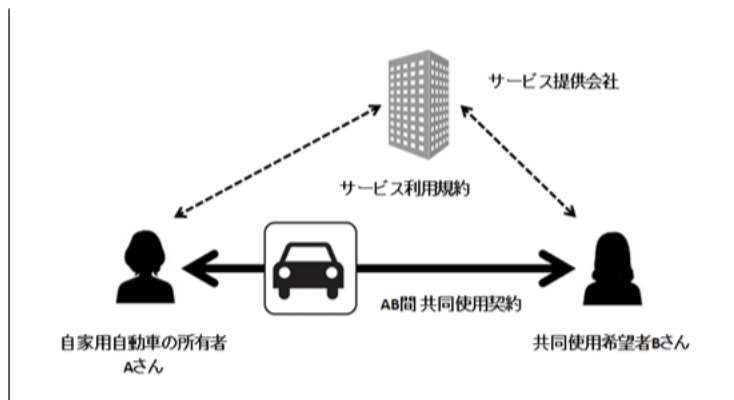
照会及び回答内容が、第4項の期間の後、公表されることに同意いたします。なお、期間の経過にかかわらず、照会者名（商号、代表者名、住所）、下記第4項（のうち延期の期間及び延期の理由のみ）及び第5項については非公表の維持をお願い申し上げます。

### 記

#### 1. 法令名および条項

道路運送法第80条第1項

#### 2. 将来自らが行おうとする行為に係る個別具体的な事実



図：本サービスを利用した場合の関係図の一例

#### (1) 本サービスの概要

当社（上図参照、サービス提供会社に該当する。）は、地域における生活者を中心としたサービスを提供する目的で、車輛の所有者兼使用者 A（上図参照、以下、「A（保有者）」という。）及び当該自家用自動車の共同使用を希望する B（上図参照、以下「B（共同使用者）」という。実際には、C、D・・・と複数人になることが想定される。）間の共同使用に係る契約（以下、「共同使用契約」という。）を媒介・支援するためのインターネットサービス（以下、「本サービス」という。）の提供を検討している。

なお、本サービスを利用するにあたって、共同使用契約の締結を希望する者（上図においては、A（所有者）及びB（共同使用者））は、当社の定める約款（以下、「サービス利用規約」という。）に同意し、当社との契約関係に入ることを要するものとする。

## (2)本サービスの構成要素

本サービスを構成する主な事実は、下記の通りである。

### ①「長期継続的な契約関係・利用関係」を確保するための施策

- ・ A（所有者）及び B（共同使用者）が、両者間で「共同使用契約」を締結することをサービス利用規約上義務付ける。
- ・ 本サービスにおいて、人的な信頼関係を構築することを目的として、A（所有者）及び B（共同使用者）間の「共同使用契約」締結に際して、相互の属性を閲覧できる機能を実装する。
- ・ A（所有者）及び B（共同使用者）間の共同使用に関する情報を交換するためのコミュニケーション機能を実装する。
- ・ 共同使用契約の契約期間は、一定期間以上（例えば、6ヶ月以上）とすることを、サービス利用規約上義務付ける。

### ②「共同で車両の整備・管理」を行うことの義務付け

- ・ B（共同使用者）が車両の共同使用開始前に、その都度、対象となる車両の点検を行うことを、サービス利用規約上義務付ける。
- ・ 上記点検において適切な使用を妨げる問題を B（共同使用者）が認識した場合、A（所有者）と連携して問題を解消することを、サービス利用規約上義務付ける。

### ③A（所有者）が、道義に反するようリスクを負担又は利益を享受しないための義務付け

#### (保険加入)

- ・ B（共同使用者）が保険に加入することを、サービス利用規約上義務付ける。

#### (点検等)

- ・ 車両不備による事故発生を防ぐため、必要な書類を当社へ提出することを、サービス利用規約上義務付ける。

#### (盗難等)

- ・ 車両の盗難や置き捨て防止に関する措置を講じる（例えば、ユーザーが使用時のモラル等について相互に相手を評価・反映できるポイント制度）。
- ・ 共同使用者（B）が所有者（A）に不法な損害（反則金の未納付などの事案を含む。）を与えた場合に当該損害を補償する内容を共同使用契約に定めることを、サービス利用規約上義務付ける。

#### (利用料)

- ・ B（共同使用者）から A（所有者）へ支払う「共同使用料」については、A（所有者）が定める。但し、サービス提供会社は、A（所有者）が設定において参酌することを目

的とした共同使用料に関するガイドラインを提供する。「共同使用料」は、貸渡の対価を含むものではなく、ガソリン代、オイル交換代、自動車税、12ヶ月点検に係る費用、自賠責の保険料、車検費用、タイヤ費用、駐車場代、車両本体の取得費（経年劣化に係る部品交換費用を含む。）、洗車代・清掃・車内外の美観維持に要する費用等の維持費（車両の維持に必要とされる実費）を基礎に按分された額をいう。

・サービス提供会社は、(i) 共同使用者（上図では B（共同使用者））から、共同使用料及び本サービスの運営に係る費用等を取得し、(ii) 本サービスの運営に係る費用等を控除した上で、(iii) 保有者（上図では A（保有者））へ支払いを行う。但し、共同使用の機会毎に精算するものではなく、毎月月末締めを行うなど一定の期間分についての精算を行うこととする。

・また、サービス提供会社は、(iv) 保有者（上図では A（保有者））から、本サービスの運営に係る費用等を取得する。

#### ④その他

・一定の車種（例えば、積載量 5 トン以上のトラック、マイクロバス）について、本サービスの対象外とすることを、サービス利用規約上定める。

・車両の転貸が第三者との契約上禁止されている場合（例えば、リースにおいて契約上禁止されている場合）について、本サービスの対象外であることをサービス利用規約上定める。

### 3. 当該法令の条項の適用に関する照会者の見解及びその根拠

#### (1) 見解

本サービスは同法第 80 条第 1 項の適用を受けない。

#### (2) 根拠

（上記態様において）共同使用契約に基づいて発生する共同使用料は、貸渡の対価を含むものではなく、維持費（車両の維持に必要とされる実費）を按分したものであり、「有償で貸し渡（す）」（道路運送法第 80 条第 1 項）に該当する態様ではない。したがって、同法第 80 条第 1 項本文の許可を要しないと解される。

### 4. 公表の延期の希望

希望する

延期の期間：回答の日から 7 ヶ月

延期の理由：

・弊社が現時点で見積もっている「システム開発」に要する期間は最低 7 ヶ月（早くても 2015 年 4 月）であり、本サービスのリリース前までは、特に高度な営業秘密に属します。

・本件照会及び回答が、本サービスのリリース前に公表された場合には、外資系を含む他のインターネット事業者などが容易に模倣し参入する可能性があります（既存のシステム流用等により、弊社より先に類似サービスをリリースする事態も十分に想定

されます)。したがって、リリース前の公表は、公正な競争の面でも、不測の損害を生じさせるおそれがございます。

・以上の理由から、回答の日から7ヶ月の延期をお願いする次第です。

#### 5. 連絡先

〒150-8510

東京都渋谷区渋谷二丁目21番1号 渋谷ヒカリエ

株式会社ディー・エヌ・エー

経営企画本部 法務部

(本件担当 : 渡部 友一郎)

(以上)

図 2-36 株式会社ディー・エヌ・エーによる法令適用事前確認手続の照会書

第 2 号様式

法令適用事前確認手続 回答書

平成 26 年 12 月 19 日

株式会社ディー・エヌ・エー  
代表取締役 守安 功 殿

国土交通省自動車局旅客課長

2014 年 10 月 31 日付けをもって照会のあった件について、下記のとおり回答します。

なお、本回答は、照会に係る法令の条項を所管する立場から、照会者から提示された事実のみを前提に、照会対象法令の条項との関係のみについて、現時点における見解を示すものであり、もとより、捜査機関の判断や罰則の適用を含めた司法判断を拘束するものではありません。

記

1 回答

- (1) 照会のあった事例（以下「照会事例」という。）について、貴社が提供しようとするサービスは照会法令の適用対象とならない。
- (2) 他方、サービスを利用する者（この場合、照会事例にある「車両の保有者」とする。）が行う行為が照会法令の適用対象となるかについては、個々の態様により個別に判断することとなる。

2 照会事例における照会法令の適用に関する見解及び根拠

道路運送法第 80 条第 1 項において「自家用自動車は、国土交通大臣の許可を受けなければ、業として有償で貸し渡してはならない。ただし、その借受人が当該自家用自動車の使用者である場合は、この限りでない。」と規定している。

- (1) 照会事例によれば、貴社が貴社の自家用自動車を有償で貸し渡すものではないと見受けられることから、貴社が提供しようとする「共同使用に係る契約を媒介・支援するためのインターネットサービス」は、照会法令の適用対象とはならない。
- (2) 照会事例によれば、「共同使用料」については車両の維持に必要とされる実費を基礎に按分された額として、車両の保有者が定めるものとしているが、その額は、どの時点を基準として算出するのか、また、共同使用者の数は固定されるのか増減するのかによって変動するものと考えられる。

このため、例えば、「共同使用料」についての設定が形骸化したり、

日・時間単位だけで使用料の設定を行うような場合には、「自家用自動車の共同使用」の適用範囲を超え、自家用自動車有償貸渡業との差はないものと考えられるため、車両の保有者は照会法令の許可を要する場合もあると考える。

なお、「自家用自動車の共同使用」とは、同一の自動車を2以上の者のそれぞれが、自己の欲求充足のために主体的な立場において使用することであり、主体的な立場において使用するための要件としては、共同で使用する者のそれぞれが自動車の使用及び管理に関する実質的な権限と責任を有することが必要であると考えられる。また、「自動車を使用する」とは、自動車の管理も合わせて行うことを常態とするものであり、日常点検整備だけでなく、定期点検整備等も行うものである。

よって、例えば共同使用者が複数の場合に、共同使用者は使用料を支払い自動車を使用するのみで、保有者と共同使用者間に自動車の管理についての合意が無い場合又は形骸化している場合に、保有者が常態的かつ単独で管理を行うのであれば、自家用自動車の共同使用とは言い難い場合もある。

図 2-37 株式会社ディー・エヌ・エーによる法令適用事前確認手続の回答書

- ②株式会社しえあくによる法令適用事前確認手続の照会書 (<https://www.mlit.go.jp/appli/content/001407847.pdf>) に対する、2021年6月7日付国土交通省自動車局旅客課長回答 (<https://www.mlit.go.jp/appli/content/001407849.pdf>)

法令適用事前確認手続(照会書)

2021年5月6日

国土交通省 自動車局旅客課長 殿

照会者名: 株式会社しえあくる 代表取締役 後藤 善午

住 所: 愛知県名古屋市中区大須4丁目13-46

下記について、照会いたします。

なお、照会及び回答内容(照会に係る法令の条項の性質上照会者名を公にすることが回答にあたって必要とされる場合にあつては、照会及び回答内容並びに照会者名)が公表されることに同意いたします。

記

1. 法令名および条項

道路運送法第80条第1項

2. 将来自らが行おうとする行為に係る個別具体的な事実

(1) 本サービスの概要

ア 当社の既存のサービスについて

当社は、現在、「しえあくる Me」「しえあくる We」という名称で、車輛の所有者兼使用者である法人及び当該自家用自動車の共同使用を希望する法人従業員(実際には、保有者の他の従業員複数人が共同使用をすることが想定される。)間の共同使用に係る契約(以下、「共同使用契約」という。)を媒介・支援するためのSaaSサービス(以下、「当社既存サービス」という。)の提供をしている。

当社既存サービスにつき、貴省貴課に対し、令和2年7月7日に法令適用事前確認手続(照会書)(以下「前回照会書」という。<https://www.mlit.go.jp/appli/content/001356712.pdf>)を提出し、同月31日に「照会のあった事例(以下「照会事例」という。)について、貴社が提供しようとするサービスは照会法令の適用対象とならない」旨の回答(以下「前回回答」という。<https://www.mlit.go.jp/appli/content/001356713.pdf>)を受けている。

イ 本サービスについて

当社は、今般、当社既存サービスの基本的な枠組である上記照会事例、すなわち、「保有

者と共同使用者が法人とその法人の従業員であり、両者が自家用自動車の共同使用契約を締結し、両者間の金銭の授受が、貸渡の対価を含まない共同使用料の範囲で行われ、また、規約上共同で車両の整備・管理を行うことを義務付ける」仕組みを維持したまま、照会事例とは自動車の保有者が逆の事例、すなわち、法人の従業員が車両の保有者であり、法人が当該車両の共同使用を希望する場合（実際には法人の他の従業員が当該車両を法人業務のために利用することも想定される）を媒介・支援するためのSaaSサービス(以下、「本サービス」という。)の提供を開始しようとしている。

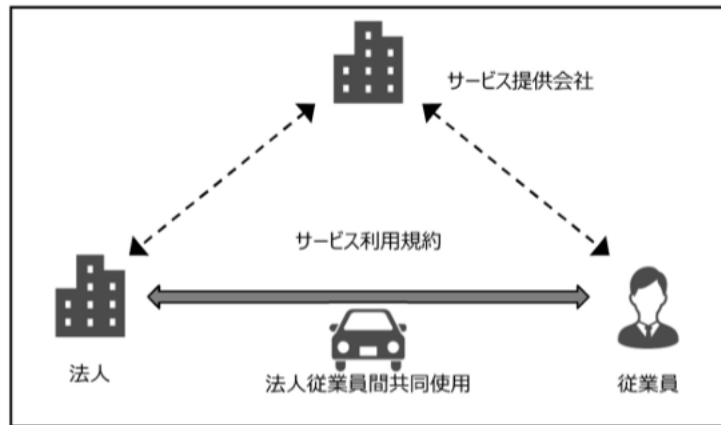


図1 本サービスを利用した場合の関係図の一例

なお、本サービスを利用するにあたって、共同使用契約の締結を希望する者(上図においては、法人及び従業員)は、当社の定める約款(以下、「サービス利用規約」という。)に同意し、当社との契約関係に入ることを要することは、当社既存サービスと同様である。

## (2)本サービスの構成要素

本サービスの構成要素は、前回照会書におけるものと、保有者が法人から法人の従業員に変わったこと以外、なんら変更をもたらすものではない。具体的には以下のとおりである(前回照会事例との変更部分に下線部を付す)。

### ア.「長期継続的な契約関係・利用関係」を確保するための施策

- ・ 保有者及び共同使用者が、両者間で共同使用契約を締結することをサービス利用規約上義務付ける。
- ・ 保有者は、共同使用者たる法人の従業員(正規であれ、派遣であれ、共同使用契約の継続期間において継続して保有者の労働契約上の指揮監督下にある者をいう。以下同じ。)であることが本サービスを利用することの前提条件である。
- ・ 共同使用契約の契約期間は、一定期間以上(例えば、6ヶ月以上)とすることを、サ

サービス利用規約上義務付ける。また、保有者が共同使用者たる法人を退職した際には、共同使用契約は終了する。

イ. 共同使用契約上に定めることについてのサービス利用規約上の義務

共同使用契約はあくまで保有者及び共同使用者の合意であり、契約自由の原則が適用されるが、本サービスを利用するために、サービス利用規約上、保有者及び共同使用者に対し、以下の点を義務づける。

(ア) 「共同で車両の整備・管理」を行うことの義務付け

- ・ 共同使用者が車両の共同使用開始前に、その都度、対象となる車両の点検を行うことを共同使用契約に定めることを、サービス利用規約上義務付ける。
- ・ 上記点検において適切な使用を妨げる問題を共同使用者が認識した場合、保有者と連携して問題を解消することを共同使用契約に定めることを、サービス利用規約上義務付ける。
- ・ 共同使用者から保有者へ、維持管理コストとしての「共同使用料」が支払われる。「共同使用料」は、貸渡の対価を含むものではなく、ガソリン代、オイル交換代、自動車税、12ヶ月点検に係る費用、自賠責の保険料、車検費用、タイヤ費用、駐車場代、車両本体の取得費(経年劣化に係る部品交換費用を含む。)、その他車両の維持に必要とされる実費を基礎に按分された額をいう。
- ・ 上記共同使用料については、保有者と共同使用者の合意に基づき共同使用契約により定める。但し、サービス提供会社は、両当事者が設定において参酌することを目的とした共同使用料に関するガイドラインを提供する。

(イ) 自動車保険、その当事者の行為による損害発生対応方法に関する合意

- ・ 保有者及び共同使用者間で保険の加入必要性の有無、事故時にどちらの保険を適用するかについて共同使用契約により合意することを、サービス利用規約上義務付ける。
- ・ 共同使用者が保有者に不法な損害(反則金の未納付などの事案を含む。)を与えた場合に当該損害を補償する内容を共同使用契約に定めることを、サービス利用規約上義務付ける。

ウ. サービス提供会社の徴収する報酬

- ・ サービス提供会社は、(i)共同使用者から、共同使用料及び本サービスの運営に係る費用等を取得し、(ii)本サービスの運営に係る費用等を控除した上で、(iii)保有者へ支払いを行う。但し、共同使用の機会毎に精算するものではなく、毎月月末締めを行うなど一定の期間分についての精算を行うこととする。
- ・ また、サービス提供会社は、(iv)法人から、本サービスの運営に係る費用等を取得

する。

3. 当該法令の条項の適用に関する照会者の見解及びその根拠

(1) 見解

本サービスは同法第 80 条第 1 項の適用を受けない。

(2) 根拠

上記態様において共同使用契約に基づいて発生する共同使用料は、貸渡の対価を含むものではなく、維持費(車両の維持に必要とされる実費)を按分したものであり、かつ、本サービスは法人とその従業員との継続的な雇用関係を前提としており、「業として有償で貸し渡(す)」(道路運送法第 80 条第 1 項)に該当する態様ではないことは前回照会書における照会事例と異なるところはない。したがって、同法第 80 条第 1 項本文の許可を要しないと解される。

4. 連絡先

〒460-0011

愛知県名古屋市中区大須4丁目13-46 (本件担当 : 稲垣)

TEL:052-211-7626 FAX:052-251-1877

以上

図 2-38 株式会社しえあくるによる法令適用事前確認手続の照会書

第2号様式

法令適用事前確認手続 回答書

令和3年6月7日

株式会社しえあくる 代表取締役 後藤 善午 殿

国土交通省自動車局旅客課長

令和3年5月6日付けをもって照会のあった件について、下記のとおり回答します。

なお、本回答は、照会に係る法令の条項を所管する立場から、照会者から提示された事実のみを前提に、照会対象法令の条項との関係のみについて、現時点における見解を示すものであり、もとより、捜査機関の判断や罰則の適用を含めた司法判断を拘束するものではありません。

記

1. 回答

- (1) 照会のあった事例（以下「照会事例」という。）について、貴社が提供しようとするサービスは照会法令の適用対象とならない。
- (2) 他方、サービスを利用する者（この場合、照会事例にある「車両の所有者兼使用者である法人従業員」又は当該「法人」とする。）が行う行為が照会法令の適用対象となるかについては、個々の態様により個別に判断することとなる。

2. 当該事実が照会法令の適用対象とならないことに関する見解及び根拠

道路運送法第80条第1項において「自家用自動車は、国土交通大臣の許可を受けなければ、業として有償で貸し渡してはならない。ただし、その借受人が当該自家用自動車の使用者である場合は、この限りでない。」と規定している。

- (1) 照会事例によれば、貴社が貴社の自家用自動車を有償で貸し渡すものではないと見受けられることから、貴社が提供しようとする「共同使用に係る契約を媒介・支援するためのインターネットサービス」は、照会法令の適用対象とはならない。
- (2) 照会事例によれば、「共同使用料」については車両の維持管理コストとして、ガソリン代その他車両の維持に必要とされる実費を基礎に按分された額をいい、その額については車両の所有者と共同使用者の合意に基づき共同使用契約により定めるものとしている。当該共同使用料の額は、どの時点を基準として算出するのか、また、共同使用者の数は固定されるのか増減するのかによって変動するものと考えられる。

このため、例えば、「共同使用料」についての設定が形骸化したり、日・時間単位だけで使用料の設定を行うような場合には「自家用自動車の共同使用」の適用範囲を超え、自家用自動車有償貸渡業との差はないものと考えられるため、車両の保有者は照会法令の許可を要する場合もあると考える。

なお、「自家用自動車の共同使用」とは、同一の自動車を2以上の者のそれぞれが、自己の欲求充足のために主体的な立場において使用することであり、主体的な立場において使

用するための要件としては、共同で使用する者のそれぞれが自動車の使用及び管理に関する実質的な権限と責任を有することが必要であると考えられる。また、「自動車を使用する」とは、自動車の管理も合せて行うことを常態とするものであり、日常点検整備だけでなく、定期点検整備等も行うものである。

【共同使用の考え方の具体例】

- ・ 同一の自動車を2以上の者それぞれが、自己の欲求充足のために主体的な立場において使用すること
- ・ 使用者が具体的に特定され、自動車の使用及び管理に関して予めの合意が存在していること
- ・ 使用者それぞれが自動車の使用及び管理に関する権限と責任を有すること
- ・ 共同使用料は営利を目的とせず、自動車の維持費の範囲内であること

照会事例は、車両の所有者である法人従業員と当該従業員を雇用する法人が自家用自動車の共同使用契約を締結し、貸渡の対価を含まない共同使用料の範囲内で行われるものであり、また、規約上共同で車両の整備・管理を行うことを義務づけることとしていることから、その限りにおいて、道路運送法第80条第1項の許可は要しないものと思料されるが、その実態において、

- ・ 自家用自動車有償貸渡業の営業に類似している
- ・ 共同使用者が複数の場合に、共同使用者は車両の所有者に使用料を支払い自動車を使用するのみで、所有者と共同使用者間に自動車の管理についての合意が無い又は合意があっても形骸化しており、所有者が常態的かつ単独で管理を行っている。

などの事象があれば、「自家用自動車の共同使用」とは言いがたい場合もある。

また、車両の所有者である法人従業員と当該法人が自家用自動車の共同使用契約を締結し、当該法人が使用する場合に、車の使用が法人の業務範囲と異なり、法人の従業員の私的使用を認めるなど実質的に法人の従業員の自由裁量に属している場合は「自家用自動車の共同使用」とは言いがたい。

図 2-39 株式会社しえあくるによる法令適用事前確認手続の回答書

- ③株式会社ビジョナグループによる法令適用事前確認手続の照会書 (<https://www.mlit.go.jp/jidosha/content/001572222.pdf>) に対する、2022年10月26日付国土交通省自動車局旅客課長回答 (<https://www.mlit.go.jp/jidosha/content/001572223.pdf>)

法令適用事前確認手続(照会書)

2022年9月26日

国土交通省 自動車局旅客課長 殿

照会者名: 株式会社ビジョナグループ 代表取締役 鷲尾 潤二

住 所: 東京都立川市高松町2-39-20

下記について、照会いたします。

なお、照会及び回答内容(照会に係る法令の条項の性質上照会者名を公にすることが回答にあたって必要とされる場合にあつては、照会及び回答内容並びに照会者名)が公表されることに同意いたします。

記

1. 法令名および条項

道路運送法第80条第1項

2. 将来自らが行おうとする行為に係る個別具体的な事実

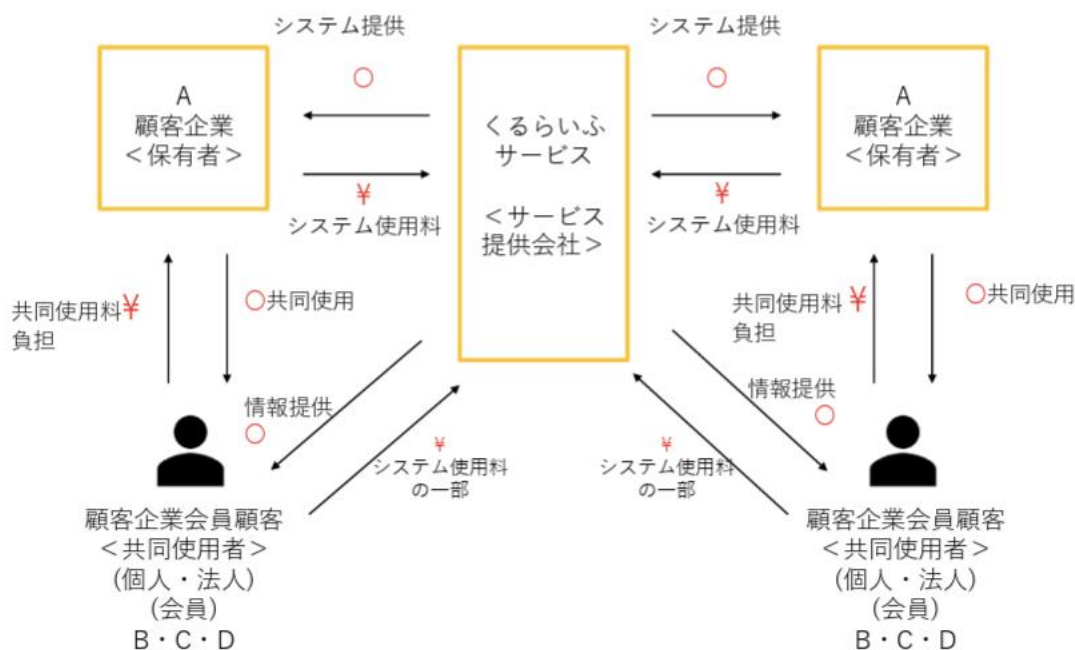


図:本サービスを利用した場合の関係図の一例

(1) 本サービスの概要

当社(上図参照、サービス提供会社に該当する。)は、車輛の所有者兼使用者である自動車販売関連法人(レンタカー会社、カーシェアリング会社を除く自動車小売業及び自動車整備業、上図参照、以下、「A(顧客企業)」という。)及びAが保有する車両の共同使用を希望するA(顧客企業)の会員顧客(上図参照、以下「B(会員)」という。実際には、A(顧客企業)の会員C、D・・・と複数人が共同使用をすることが想定される。)間の共同使用に係る契約(以下、「共同使用契約」という。)を媒介・支援するためのインターネットサービス(以下、「本サービス」という。)の提供を開始しようとしている。

なお、本サービスを利用するにあたって、共同使用契約の締結を希望する者(上図におけるA(顧客企業)及びB(会員))は、当社の定める約款(以下、「サービス利用規約」という。)に同意し、当社との契約関係に入ることを要するものとする。

(2) 本サービスの構成要素

本サービスを構成する主な事実は、下記の通りである。

ア. 「長期継続的な契約関係・利用関係」を確保するための施策

- ・ A(顧客企業)及びB(会員)が、両者間で共同使用契約を締結することをサービス利用規約上義務付ける。
- ・ B(会員)は A(顧客企業)の会員顧客(共同使用契約の継続期間において継続してA(保有者)の会員顧客にある者をいう。)であることが本サービスを利用することの前提条件である。
- ・ 共同使用契約の契約期間は、一定期間以上(例えば、6ヶ月以上)とすることを、サービス利用規約上義務付ける。また、B(会員)がA(顧客企業)の会員を脱会した際には、共同使用契約は終了する。

イ. 共同使用契約上に定めることについてのサービス利用規約上の義務

共同使用契約はあくまでA(顧客企業)及びB(会員)の合意であり、契約自由の原則が適用されるが、本サービスを利用するために、サービス利用規約上、A(顧客企業)及びB(会員)に対し、以下の点を義務づける。

(ア) 「共同で車両の整備・管理、保険の加入」を行うことの義務付け

- ・ B(会員)が車両の共同使用開始前に、その都度、対象となる車両の点検、自動車保険加入を行うことを共同使用契約に定めることを、サービス利用規約上義務付ける。
- ・ 上記点検において適切な使用を妨げる問題を B(会員)が認識した場合、A(顧客企業)と連携して問題を解消することを共同使用契約に定めることを、サービス利用規約上義務付ける。
- ・ B(会員)から A(顧客企業)へ、維持管理コストとしての「共同使用料」が支払われる。「共同使用料」は、貸渡の対価を含むものではなく、ガソリン代、オイル交換代、自動車税、12ヶ月点検に係る費用、自賠責の保険料、車検費用、タイヤ費用、駐車場代、車両本体の取得費(経年劣化に係る部品交換費用を含む。)、その他車両の維持に必要とされる実費を基礎に按分された額をいう。
- ・ 上記共同使用料については、A(顧客企業)とB(会員)の合意に基づき共同使用契約により定める。但し、サービス提供会社は、両当事者が設定において参酌することを目的とした共同使用料に関するガイドラインを提供する。

(イ)自動車保険、その当事者の行為による損害発生対応方法に関する合意

- ・ A(顧客企業)及びB(会員)間で保険の加入必要性の有無、事故時にどちらの保険を適用するかについて共同使用契約により合意することを、サービス利用規約上義務付ける。
- ・ 会員(B)が顧客企業(A)に不法な損害(反則金の未納付などの事案を含む。)を与えた場合に当該損害を補償する内容を共同使用契約に定めることを、サービス利用規約上義務付ける。

ウ. サービス提供会社の徴収する報酬

- ・ サービス提供会社は、(i)共同使用者(上図ではB(会員顧客))から、共同使用料及び本サービスの運営に係る費用等を取得し、(ii)本サービスの運営に係る費用等を控除した上で、(iii)保有者(上図ではA(顧客企業))へ支払いを行う。但し、共同使用の機会毎に精算するものではなく、毎月月末締めを行うなど一定の期間分についての精算を行うこととする。
- ・ また、サービス提供会社は、(iv)A(顧客企業)から、本サービスの運営に係る費用等を取得する。

3. 当該法令の条項の適用に関する照会者の見解及びその根拠

(1) 見解

本サービスは同法第 80 条第 1 項の適用を受けない。

(2) 根拠

(上記態様において)共同使用契約に基づいて発生する共同使用料は、貸渡の対価を含むものではなく、維持費(車両の維持に必要とされる実費)を按分したものであり、また保険加入は原則会員であり、かつ法人と会員顧客との継続的な関係を前提としており、「業として有償で貸し渡(す)」(道路運送法第 80 条第 1 項)に該当する態様ではない。したがって、同法第 80 条第 1 項本文の許可を要しないと解される。

4. 連絡先

〒190-0011

東京都立川市高松町 2-39-20 (本件担当 : 鷲尾)

TEL:042-511-2612

以上

図 2-40 株式会社ビジョナグループによる法令適用事前確認手続の照会書

第2号様式

法令適用事前確認手続 回答書

令和4年10月26日

株式会社ビジョナグループ 代表取締役 鷺尾 潤二 殿

国土交通省自動車局旅客課長

令和4年9月26日付けをもって照会のあった件について、下記のとおり回答します。

なお、本回答は、照会に係る法令の条項を所管する立場から、照会者から提示された事実のみを前提に、照会対象法令の条項との関係のみについて、現時点における見解を示すものであり、もとより、捜査機関の判断や罰則の適用を含めた司法判断を拘束するものではありません。

## 記

### 1. 回答

- (1) 照会のあった事例（以下「照会事例」という。）について、貴社が提供しようとするサービスは照会法令の適用対象とならない。
- (2) 他方、サービスを利用する者（この場合、照会事例にあるA（顧客企業）とする。）が行う行為が照会法令の適用対象となるかについては、個々の態様により個別に判断することとなる。

### 2. 当該事実が照会法令の適用対象とならないことに関する見解及び根拠

道路運送法第80条第1項において「自家用自動車は、国土交通大臣の許可を受けなければ、業として有償で貸し渡してはならない。ただし、その借受人が当該自家用自動車の使用者である場合は、この限りでない。」と規定している。

- (1) 照会事例によれば、貴社が貴社の自家用自動車を有償で貸し渡すものではないと見受けられることから、貴社が提供しようとする「共同使用に係る契約を媒介・支援するためのインターネットサービス」は、照会法令の適用対象とはならない。
- (2) 照会事例によれば、「共同使用料」については車両の維持管理コストとして、ガソリン代その他車両の維持に必要とされる実費を基礎に按分された額をいい、その額については車両の所有者と共同使用者の合意に基づき共同使用契約により定めるものとしている。当該共同使用料の額は、どの時点を基準として算出するのか、また、共同使用者の数は固定されるのか増減するのかによって変動するものと考えられる。

このため、例えば、「共同使用料」についての設定が形骸化したり、日・時間単位だけで使用料の設定を行うような場合には「自家用自動車の共同使用」の適用範囲を超え、自家用自動車有償貸渡業との差はないものと考えられるため、車両の保有者は照会法令の許可を要する場合もあると考える。

なお、「自家用自動車の共同使用」とは、同一の自動車を2以上の者のそれぞれが、自己の欲求充足のために主体的な立場において使用することであり、主体的な立場において使

用するための要件としては、共同で使用する者のそれぞれが自動車の使用及び管理に関する実質的な権限と責任を有することが必要であると考えられる。また、「自動車を使用する」とは、自動車の管理も合せて行うことを常態とするものであり、日常点検整備だけでなく、定期点検整備等も行うものである。

【共同使用の考え方の具体例】

- ・ 同一の自動車を2以上の者それぞれが、自己の欲求充足のために主体的な立場において使用すること
- ・ 使用者が具体的に特定され、自動車の使用及び管理に関して予めの合意が存在していること
- ・ 使用者それぞれが自動車の使用及び管理に関する権限と責任を有すること
- ・ 共同使用料は営利を目的とせず、自動車の維持費の範囲内であること

照会事例は、車両の所有者である A（顧客企業）と B（会員）が自家用自動車の共同使用契約を締結し、貸渡の対価を含まない共同使用料の範囲内で行われるものであり、また、規約上共同で車両の整備・管理を行うことを義務づけることとしていることから、その限りにおいて、道路運送法第80条第1項の許可は要しないものと思料されるが、その実態において、

- ・ 自家用自動車有償貸渡業の営業に類似している
- ・ 共同使用者が複数の場合に、共同使用者は車両の所有者に使用料を支払い自動車を使用するのみで、所有者と共同使用者間に自動車の管理についての合意が無い又は合意があっても形骸化しており、所有者が常態的かつ単独で管理を行っている。

などの事象があれば、「自家用自動車の共同使用」とは言いがたい場合もある。

図 2-41 株式会社ビジョナグループによる法令適用事前確認手続の回答書

## 具体的検討

- 本サービスは、前述の各法令適用事前確認手続の回答書に記載されている共同使用の考え方に基づき設計・運用されているサービスである。以下、共同使用の考え方に基づくサービスの具体的な仕様」を記載する。

### 1. 「同一の自動車を二以上の者それぞれが、自己の欲求充足のために主体的な立場において使用すること」

- この点に関連し、例えば、過去事例①では、共同使用としての長期継続的な契約関係・利用関係を確保すべく、以下の内容を掲げている。
  - ①共同使用契約の締結の義務付け
  - ②契約締結に際して相互に属性の閲覧が可能な機能の実装
  - ③情報交換のためのコミュニケーション機能の実装
  - ④共同使用契約の契約期間を一定期間以上（例えば、6 か月以上）とするものの義務付け
- 本サービスにおいても、以下のとおり、同様に長期継続的な契約関係・利用関係を確保するための施策を講じている。
  - 車両提供者及びユーザーにおいて、共同使用契約を締結することを、本サービスの共同使用規約内で義務付けている（3.1 条）（①共同使用契約の締結の義務付け）。
  - 本サービスの仕様上、ユーザーは、自動車の使用予約作成段階（共同使用契約の締結段階）において、事前に設定された車両提供者の属性（実名/電話番号/住所/メールアドレス等）に係る情報を閲覧することができる。また、同様に、車両提供者は、当該予約作成を行ったユーザーの属性（実名/電話番号/住所/メールアドレス等）に係る情報を閲覧することができる（②契約締結に際して相互に属性の閲覧が可能な機能の実装）。
  - 車両提供者及びユーザーにおいて、共同使用契約に基づいて、必要に応じ相互に連絡を取り合うことを本サービスの共同使用規約内で規定しており（3.7 条）、本サービスの仕様上、車両提供者及びユーザーは、設定された連絡先情報を通じて、情報交換のためにコミュニケーションを取ることが可能である（③情報交換のためのコミュニケーション機能の実装）。
  - 共同使用契約は、少なくとも 6 か月以上の有効期間を設定することを本サービスの共同使用規約内で義務付けている（3.2 条 1 項）。本サービスの仕様上、6 か月未満の期間での有効期間の設定はできない（④共同使用契約の契約期間を一定期間以上（例えば、6 か月以上）とするものの義務付け）。
- したがって、本サービスは、車両提供者及びユーザーにおける共同使用としての長期継続的な契約関係・利用関係が確保しており、「同一の自動車を二以上の者それぞれが、自己の欲求充足のために主体的な立場において使用すること」という考え方に適合するものである。

### 2. 「共同使用者が具体的に特定され、自動車の使用及び管理に関してあらかじめの合意が存在していること」

- この点に関し、共同使用契約の締結の前提として、相手方の情報が明示され、双方にとって締結する相手方が具体的に特定されていることが必要であると考えられる。
- 本サービスにおいては、前述のとおり、車両提供者及びユーザーに対して、共同使用者それぞれが自動車の使用及び管理に関する権限と責任を有することを規定するための共同使用契約を締結することを利用

規約上義務付けている。また、同様に前述のとおり、車両提供者及びユーザーは、共同使用契約の締結に当たり、相互に属性に係る情報を閲覧することが可能であり、双方にとって共同使用者たる相手方が具体的に特定されている。

- したがって、本サービスは、具体的に特定された相手方との間で、共同使用契約を締結させるものであり、「共同使用者が具体的に特定され、自動車の使用及び管理に関してあらかじめの合意が存在していること」という考え方に適合するものである。
  - なお、共同使用契約の内容である「自動車の仕様及び管理に関してあらかじめの合意」については、後述の3.のとおりである。

### 3. 「共同使用者それぞれが自動車の使用及び管理に関する権限と責任を有すること」

- この点に関連し、例えば、過去事例②においては、共同使用契約に基づき(1)「共同で車両の整備・管理」を行うことを義務付けている。
  - 具体的には、規約において(イ)ユーザーにおける利用開始前の点検、(ロ)問題を認識した場合における解消のための連携を義務付けている。
- また、同じく過去事例②においては、共同使用契約として(2)「自動車保険、その当事者の行為による損害発生対応方法に関する合意」をすることを義務付けている。
  - 具体的には、規約において(ハ)保有者及びユーザー間で保険の加入必要性の有無、事故時にどちらの保険を適用するかについて共同使用契約により合意すること、(ニ)ユーザーが車両提供者に不法な損害を与えた場合の当該損害を補償する内容を共同使用契約で定めることを義務付けている。
  - なお、(ハ)につき、過去事例①においては、単に、ユーザーが保険に加入することを、本サービスの共同使用規約内で義務付ける形としている。
- 本サービスにおいても、以下のとおり(イ)及び(ロ)に関する施策を講じている。
  - 「共同で車両の整備・管理」を行うことの義務付け(1)
    - ◇ 車両提供者及びユーザーにおいて、共同使用契約に基づき、登録自動車の使用前及び共同使用契約の有効期間中適切に登録自動車の点検を行い、不備があった場合は直ちに指摘する等、共同使用者として共同して登録自動車の管理を行うことを本サービスの共同使用規約内で義務付けている(3.8条)(イ)。
      - ✓ なお、本サービスの仕様上、ユーザーは、利用開始時間前に日常点検を行い(3.8条)、車両の外傷、ランプ類の点灯等の一定の項目に従った確認を行い(ユーザーのスマートフォンにおいてチェックボックスにチェックを入れる方法により)利用開始報告を行うこととなっている。
    - ◇ ユーザーにおいて、上記の点検の結果、登録自動車の適切な利用を妨げる問題を認識した場合、登録自動車の使用を停止しなければならないものとし、車両提供者と連携して速やかに問題を解決することを本サービスの共同使用規約内で義務付けている(3.3条3項)(ロ)。
  - 自動車保険、その当事者の行為による損害発生対応方法に関する合意(2)
    - ◇ ユーザーにおいて、登録自動車を使用する場合、その使用の都度、その使用に先立ち、自己の責任で、登録自動車の使用に伴い発生し得る対人賠償責任、対物賠償責任、搭乗者の傷害及び登録自動車の復旧費用を十分に補償する損害保険に係る契約の締結を利用規約上義務付け(2.5

条)、これに基づく本サービスの仕様として、ユーザーは、登録自動車を使用する場合、カーシェア専用保険へ自動加入をすることとなっている（ハ）。

◇ ユーザーにおいて、車両提供者に不法な損害（反則金の未納付などの事案を含む。）を与えた場合に、当該損害を補償する内容を共同使用契約に定めることを、利用規約上義務付けている（2.4条4項）（ニ）。

- したがって、本サービスは、車両提供者及びユーザーに対して、共同で車両の整備・管理を行い、また、自動車保険、その当事者の行為による損害発生対応方法に関する合意をすることを利用規約上義務付けることによって「共同使用者それぞれが自動車の使用及び管理に関する権限と責任を有すること」という考え方に適合するものである。

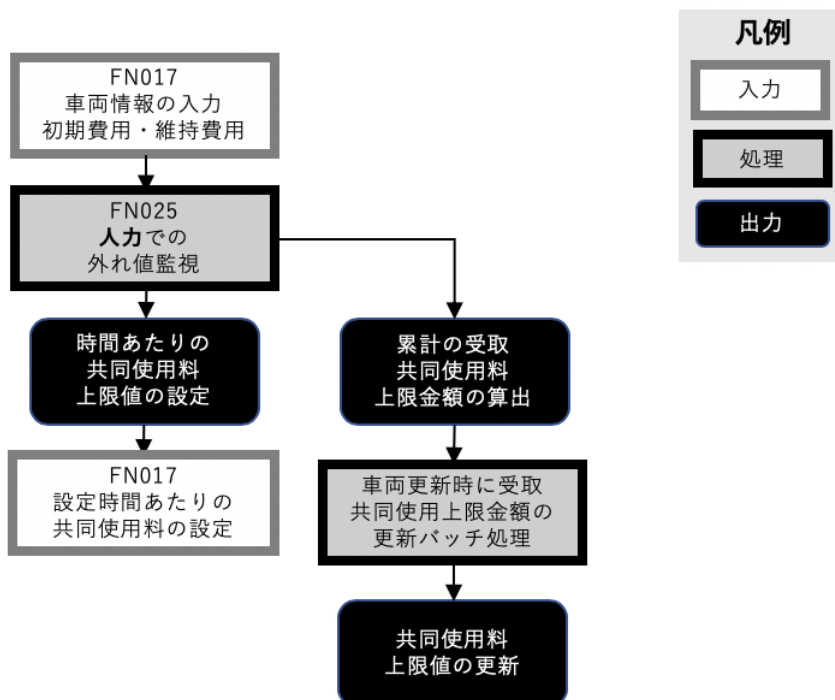
#### 4. 「共同使用料は営利を目的とせず、自動車の維持費の範囲内であること」

- この点に関し、本サービスにおいて、車両提供者は、登録した車両についての共同使用料をユーザーから受け取るが、自家用自動車の有償での貸渡りに該当しないよう、共同使用料を自動車の維持費の範囲内に収める必要があると考えられる。
- 本サービスにおいては、過去事例における車両の維持費用に係る考え方を基に、以下の措置を講じている。
  - 本サービスの仕様上、車両提供者による共同使用料の設定・受取に関し、①利用時間（30分/6時間/12時間/24時間で設定）当りの上限額を超える設定をできないようにし、また、②車両提供者が累計受取共同使用料に係る上限額を超える金額を受け取ることがないように、制御を行っている。
  - 車両提供者において、本サービス上設定された上限を上回る共同使用料の設定及び請求を利用規約上禁止している（2.3条3項）。
- 上記①及び②の上限額の算定ロジックについては、以下のとおりである。
  - まず、いずれの過去事例においても、車両維持費用の算定に当たり、車両本体の取得費用を算定基礎に含めることが許容されることを前提に回答がなされていることから、以下では、車両維持費用を大きく二つに分類して整理している。
    - ◇ 車両本体の取得費用（以下「取得関連費用」という。）：車両本体の取得費（経年劣化に係る部品交換費用を含む。）
    - ◇ 車両の維持費用（以下「維持関連費用」という。）：ガソリン代、オイル交換代、自動車税、12か月点検に係る費用、自賠責の保険料、車検費用、タイヤ費用、駐車場代、洗車代・清掃・車内外の美観維持に要する費用等の維持費
  - ①について、取得関連費用及び維持関連費用のそれぞれの償却金額を計算し、その合計額を利用時間当りの上限金額として算定する。これにより、理屈上、車両提供者は維持管理費用を超える収益を得られることがなくなる。
    - ◇ 取得関連費用：自動車の走行限界距離を19万kmとし、そこから登録時の累計走行距離を控除した距離を走行可能距離として捉え、設定された取得関連費用を走行可能距離で除して1km当たりの償却単価を算定する。この償却単価に、単位時間（利用時間）ごとの想定走行距離を乗じることで、単位時間当たりの維持管理費用を算定することができる。
    - ✓ なお、取得関連費用は、車両提供者の申告により設定するが、市場価格から外れたものは、サービス運営側で非承認とするなどの対応を行う。

- ◇ 維持関連費用：設定された年間想定走行距離に対し、年間維持関連費用として設定した金額で除して1 km当たりの償却単価を算定する。この償却単価に、単位時間（利用時間）ごとの想定走行距離を乗じることで、単位時間当たりの維持管理費用を算定することができる。
  - ✓ なお、年間想定走行距離について、車両提供者の申告により設定するが、車両登録以降の車検更新時には、登録時からの走行距離実績を参考に、必要に応じて車両提供者の申告により年間想定走行距離を更新する。
- ②について、利用時間当たりの上限金額のほか、取得関連費用と維持関連費用の合計金額である維持管理費用を、車両提供者が当該登録自動車につき累計で受け取ることができる上限の金額として算定する。これにより、これを超えて共同使用料を受け取ることとなる場合には、強制的に車両登録を非公開にして予約を受け付けられないようにするなど、車両提供者が維持管理費用を超える収益を得ることがないようにし、累計受取共同使用料の観点からも、自動車の維持費の範囲内に収めることとしている。
- ◇ なお、初回の車両登録時における維持関連費用は、年間の維持関連費用に次回車検更新の時点までの期間を乗じた金額として算定され、これと取得関連費用を合計した金額が、初回登録時点における累計受取共同使用料の上限として設定される。
- ◇ 初回登録時の累計受取共同使用料の上限は、車検更新の時点で更新される。
  - ✓ 初回登録時に維持管理費用として算定された金額は（当該車検更新時点までに車両提供者が実際に受け取った共同使用料の金額にかかわらず）、リセットされる。
  - ✓ その上で、改めて次回車検更新の時点までの維持関連費用を算定し、同金額を加算したものを累計受取共同使用料の上限とする。
  - ✓ なお、実際に車両提供者が受け取る共同使用料は（車検更新ごとにリセットされる）維持関連費用の部分に充当され、累計受取共同使用料が維持関連費用を超えた場合に、取得関連費用の部分に充当される。
- したがって、本サービスは、その仕様として自動車の維持管理費用を超える共同使用料を受け取ることができず、また、利用規約上もそのような維持管理費用を超える共同使用料の受取を禁止していることによって、「共同使用料は営利を目的とせず、自動車の維持費の範囲内であること」という考え方に適合するものである。

- 本システム機能の入力・処理・出力のフローチャート

設定時間あたりの共同使用料上限値  
及び  
受取共同使用料上限の算出



受取共同使用料上限金額計制御

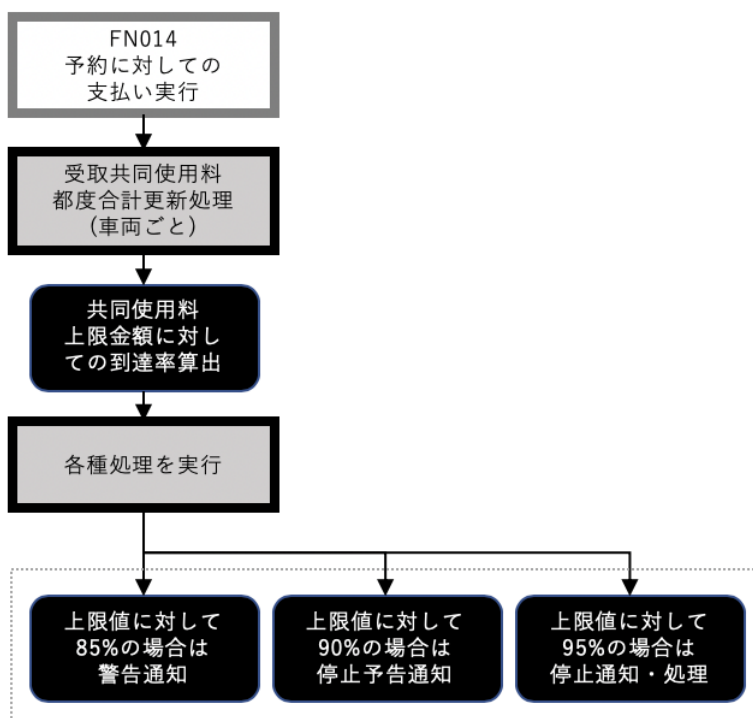


図 2-42 上限金額ロジックイメージ図

- 本アルゴリズムを利用した機能
  - 【FN024】売上上限コントロール
- アルゴリズムの詳細
  - 受取共同使用料上限金額の算出基礎の対象となる車両コスト
    - ◇ 過去事例①の法令適用事前確認手続の照会書にある下記の記載内容より、費用種別に記載の大きく二つの費用種別に分類できる。
      - ガソリン代、オイル交換代、自動車税、12 か月点検に係る費用、自賠責の保険料、車検費用、タイヤ費用、駐車場代、車両本体の取得費（経年劣化に係る部品交換費用を含む。）、洗車代・清掃・車内外の美観維持に要する費用等の維持費（車両の維持に必要とされる実費）を基礎に按分された額
    - ◇ 費用種別
      - ①車両本体の取得費用（以下車両本体の取得関連費用）：車両本体の取得費（経年劣化に係る部品交換費用を含む。）
      - ②車両の維持費用（以下車両の維持関連費用）：ガソリン代、オイル交換代、自動車税、12 か月点検に係る費用、自賠責の保険料、車検費用、タイヤ費用、駐車場代、洗車代・清掃・車内外の美観維持に要する費用等の維持費
  - 受取共同使用料上限金額計制御
    - ◇ 受取共同使用料上限金額算出：受取①車両本体の取得関連費用と②車両の維持関連費用の合計金額を車両提供者が受取可能な共同使用料上限金額とする（以下、受取共同使用料上限金額）。
      - 車両本体の取得関連費用と維持関連費用は、車両提供者が車両登録をする際に入力、登録する下記項目の数値と定義する。
        - ①車両登録時の本体取得関連費用の入力項目
          - ◇ 入力項目 1：購入時価格（販売代理店等の手数料含む）
          - ◇ 入力項目 2：その他（部品交換等）
        - ②車両登録時の維持関連費用の入力項目
          - ◇ 入力項目 3：年間維持費用
      - 市場価格から外れているものはサービス運営側で車両審査時に非承認とする。価格の外れ値の確認方法としては、以下のとおりとする。
        - 初年度登録（車検証で確認）と購入年月の差分（車両登録時に入力）から何年落ちの状態（以後、経過年数）で購入したかを把握する。
          - ◇ ※購入年月は任意項目とし、入力がない場合は初年度登録年を対象とする（経過年数 0）
        - 現在市場の中で現時点から同様の経過年数の同じ車種の市場価格幅を確認する。この価格幅内に収まっているかを車両審査維持に確認する。
        - 【本スコープ対象外】登録車両データが増加した際はデータによって判別可能にする。
    - ◇ 現状の受取共同使用料算出：車両提供者がサービス手数料引き後に受け取る共同使用料(以下、

「受取共同使用料」という) を、共同使用者からの支払が発生するたびに車両ごとに算出し、データを保持する。

- 累計受取共同使用料算出式

$$● \quad Fta = Ft0 * (1-R) + Ft1 * (1-R) + \dots + Ftn * (1-R)$$

- ◇ 受取共同使用料の照合確認・制御：この対象期間中の累計受取共同使用料が受取共同使用料上限金額に対して超過していないかを当該車両に対する共同使用者からの共同使用料支払が完了するたびに下記計算式による確認・措置を実施する。

- 上限超過制御計算式

$$Fta \leq Flimit$$

- 現状の累計受取共同使用料が受取共同使用料上限金額に対して一定の割合に達した段階で各措置を実行する。
  - 85%の段階で警告
  - 90%停止予告
  - 95%車両公開の停止
- ◇ 受取共同使用料上限金額の更新：受取共同使用料上限金額は車検更新時と同時に更新されるものとする。車両の維持費用のうち、車検のタイミングに合わせて、車両の修理なども行われるため、ユーザビリティを考慮し、車検更新時に上限金額を更新する。

- 受取共同使用料上限金額の更新処理式 (Fta が年間維持費用以上の場合)

$$Flimit = (Ci + Ti / 365 \text{ 日} * Cm) - Fta + T'i / 365 \text{ 日} * Cm$$

- 受取共同使用料上限金額の更新処理式 (Fta が年間維持費用以下の場合)
- ※サービスに車両が登録された時点での受取共同使用料上限金額も同様

$$Flimit = Ci + Ti / 365 \text{ 日} * Cm$$

- ◇ 数値定義

- Fta：(算出変数) 前回の車検更新時 (又は車両登録時) から車検更新時までの累計受取共同使用料
- Ftn：(算出変数) 1回使用で発生する共同使用料。n が 0,1,2,... と変化する。例えば Ft0 であればサービス登録以降、初回利用の共同使用料を指す。
- R：(サービス側で定める定数) サービス手数料 (30%)
- Flimit：(定数) 受取可能な共同使用料上限金額
- Ci：(車両提供者入力項目) 自動車本体関連費用 (償却上限金額)
  - ◇ 車両本体の取得費 (経年劣化に係る 部品交換 費用を含む。) を指す。
  - ◇ 車両登録時に車両提供者が登録する
  - ◇ 運営側で外れ値がないか車両審査時にモニタリングを実施
- Ti：(算出変数) 車両登録又は前回車検時から次回車検までの日数 (T'i は前回の車検更新

時までの日数)

- Cm：(車両提供者入力項目) 年間の自動車維持関連費用
  - 車両の年間維持費用想定金額を指す。車両登録時・車検更新時に車両提供者が本数値を登録する。
  - 本年度は初回登録時の年間維持費用に対してランダムチェックを行い必要に応じてヒアリングを実施する。
  - 車両データが蓄積され次第、車両・エリア等の平均値基にしたシステムチェックにおいて対応想定(本年度のスコープ範囲外) 運営側で外れ値がないか車両審査時にモニタリングを実施
- 1 使用当たりの共同使用料の金額設定
  - ◇ 1 使用当たりの共同使用料上限金額を①自動車本体関連費用からくる償却金額と②自動車維持関連費用からくる償却金額を最終的に足しあわせ、設定時間(30分/6時間/12時間/24時間)ごとに定める。最終的には、当該金額以下で車両提供者が任意に設定するものとする。
  - ◇ ①自動車本体関連費用からくる利用時間に応じた償却金額の算出
    - 車両の登録時走行距離と車両本体の価値が0になる距離(以降「限界距離：D0」と呼称する)に達するまでの走行可能距離から本体関連費用から、1km当たりの償却金額を算出する。
    - 1km当たりの初期費用償却金額算出式
$$Cikm=Ci/(Do-Dr)$$
    - 次に、1km償却単価を利用時間に応じた想定走行距離で掛け合わせ、利用に応じた本体関連費用の償却金額を決定する。
    - 利用時間に応じた初期費用償却金額算出式
$$Ciu=Cikm*Du$$
  - ◇ 数値定義
    - Cikm：(定数) 1km当たりの初期費用償却金額
    - Do：(定数) 限界距離(車両の価値が0となる総走行距離)
      - 車両の価値が0になる累計走行距離。この距離は19万kmと定める。
        - ◇ なお、既に19万キロを走行しているが取得に100万円の経費を要した場合には、取得時金額100万円は共同使用料として償却できない。
      - JAAI(自動車査定協会)が出している中古車査定基準の上限距離を参照
        - ◇ <http://www.jaai.or.jp/images/jjssiten/sateikijun240327/kijun-1.pdf>

表1 乗用車、ワゴン、オフロードタイプ

特C・B・A・Iクラス		標準走行キロ																										
走行 km (千)	使用経過月数																				走行 km (千)							
	0 5 4	5 5 8	9 5 12	13 5 16	17 5 20	21 5 24	25 5 28	29 5 32	33 5 36	37 5 40	41 5 44	45 5 48	49 5 54	55 5 60	61 5 66	67 5 72	73 5 78	79 5 84	85 5 90	91 5 96		97 5 102	103 5 108	109 5 114	115 5 120	121 5 (上限)		
5			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	5		
10	-3	-2			2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	10		
15	-5	-4	-3				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	15		
20	-7	-6	-5	-3				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
25	-9	-8	-7	-5	-3	-2						3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	25	
30	-11	-10	-9	-7	-5	-4	-3						3	4	5	6	7	8	9	10	11	12	13	14	15	30		
35	-13	-12	-11	-9	-7	-6	-5	-4	-3	-2					4	5	6	7	8	9	10	11	12	13	14	35		
40	-15	-14	-13	-11	-9	-8	-7	-6	-5	-4	-3	-2				4	5	6	7	8	9	10	11	12	13	40		
45	-17	-16	-15	-13	-11	-10	-9	-8	-7	-6	-5	-4	-3					5	6	7	8	9	10	11	12	45		
50	-19	-18	-17	-15	-13	-12	-11	-10	-9	-8	-7	-6	-5	-3					5	6	7	8	9	10	11	50		
55	-21	-20	-19	-17	-15	-14	-13	-12	-11	-10	-9	-8	-7	-5	-3	-2				5	6	7	8	9	10	55		
60	-23	-22	-21	-19	-17	-16	-15	-14	-13	-12	-11	-10	-9	-7	-5	-4	-3							7	8	60		
65	-25	-24	-23	-21	-19	-18	-17	-16	-15	-14	-13	-12	-11	-9	-7	-6	-5	-3								65		
70	-27	-26	-25	-23	-21	-20	-19	-18	-17	-16	-15	-14	-13	-11	-9	-8	-7	-5	-3							70		
75	-29	-28	-27	-25	-23	-22	-21	-20	-19	-18	-17	-16	-15	-13	-11	-10	-9	-7	-5	-3						75		
80	-31	-30	-29	-27	-25	-24	-23	-22	-21	-20	-19	-18	-17	-15	-13	-12	-11	-9	-7	-5	-4	-3	-2			80		
90	-34	-33	-32	-30	-28	-27	-26	-25	-24	-23	-22	-21	-20	-18	-16	-15	-14	-12	-10	-8	-7	-6	-5	-3	-2	90		
100	-37	-36	-35	-33	-31	-30	-29	-28	-27	-26	-25	-24	-23	-21	-19	-18	-17	-15	-13	-11	-10	-9	-8	-6	-5	100		
110	-42	-41	-40	-38	-36	-35	-34	-33	-32	-31	-30	-29	-28	-26	-24	-23	-22	-20	-18	-16	-15	-14	-13	-11	-10	110		
120	-47	-46	-45	-43	-41	-40	-39	-38	-37	-36	-35	-34	-33	-31	-29	-28	-27	-25	-23	-21	-20	-19	-18	-16	-15	120		
130	-52	-51	-50	-48	-46	-45	-44	-43	-42	-41	-40	-39	-38	-36	-34	-33	-32	-30	-28	-26	-25	-24	-23	-21	-20	130		
140	-57	-56	-55	-53	-51	-50	-49	-48	-47	-46	-45	-44	-43	-41	-39	-38	-37	-35	-33	-31	-30	-29	-28	-26	-25	140		
150	-60	-60	-60	-58	-56	-55	-54	-53	-52	-51	-50	-49	-48	-46	-44	-43	-42	-40	-38	-36	-35	-34	-33	-31	-30	150		
160	↑	↑	↑	-60	-60	-60	-59	-58	-57	-56	-55	-54	-53	-51	-49	-48	-47	-45	-43	-41	-40	-39	-38	-36	-35	160		
170	↑	↑	↑	↑	↑	↑	-60	-60	-60	-60	-60	-59	-58	-56	-54	-53	-52	-50	-48	-46	-45	-44	-43	-41	-40	170		
180	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	-60	-60	-60	-59	-58	-57	-55	-53	-51	-50	-49	-48	-46	-45	180
190	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	190	
ISO超 (上限)	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	ISO超 (上限)	

基本価格に対する加減率 (%)

(注) 1. 走行キロ数の適用は、以内とする。

(注) 2. 端数の処理は、小数点以下第一位を四捨五入する。

図 2-43 中古車査定基準

- Dr : (定数) 車両登録時に入力する累積走行距離 (総走行距離メーター数値)
- Ciu : (変数) 利用に応じて受け取った初期費用償却分の共同使用料
- Du : (変数) 利用時間に応じた想定走行距離。

表 2-4 各利用時間に応じた平均走行距離 (例)

利用時間	平均走行距離(Du)
30min	7.75km
6 時間	82.7km
12 時間	156.4km
24 時間	164.8km

- ②自動車維持関連費用からくる利用時間に応じた償却金額の算出
  - ◇ まず、年間想定走行距離と年間維持費用から 1km 当たりの維持関連費用の償却金額を定める。
    - 1km 当たりの維持費用償却金額算出式
 
$$Cmkm = Cm / Dy$$
  - ◇ 次に、利用時間に応じた走行距離と 1km 当たりの償却金額から利用時間に応じた維持関連費用の償却金額を定める。
    - 利用時間に応じた維持費用償却金額算出式
 
$$Cmu = Cmkm * Du$$
  - ◇ 数値定義
    - Cmkm：(定数) 1km 当たりの維持費用償却金額
    - Cm：(車両提供者入力項目) 年間維持費用
    - Dy：(車両提供者入力項目) 年間想定走行距離
      - 初回の車両登録時に車両提供者が年間の想定走行距離を記入する。
      - 以降の車検更新時には当該時点での総走行距離を入力し、車両登録時からの走行距離を把握する。
      - これを基に次回車検時までの想定される走行距離を算出・提示し、その数値を基に車両提供者想定年間走行距離を入力する。
    - Cmu：(変数) 利用に応じた維持費用償却分の共同使用料
    - Du：利用時間に応じた想定走行距離
      - 自動車取得費用内「Du：利用時間に応じた想定走行距離」参照
- 利用時間に応じた 1 使用当たりの共同使用料上限金額の算出
  - ◇ ①自動車本体取得関連費用及び②自動車維持関連費用で算出した Cmu 及び Ciu の合計金額が設定時間に対する共同使用料の上限金額とする。
    - 使用時間に応じた共同使用料上限金額算出式
 
$$Culimit = (1 + R) * (Cmu + Ciu)$$
  - ◇ この金額を上限とし、車両登録時に車両提供者が任意の金額を各時間に対して設定できるものとする。
 
$$Cu \leq Culimit$$
  - ◇ 最終的に共同使用者に請求される共同使用料は、各設定時間に応じて定められた Cu を実際の使用時間に応じて足し合わせた金額とする。また、使用時間が予約した時間よりも短かった場

合、実際の使用時間に応じた最も低い金額が算出・請求される。

- 詳細は具体ユースケースシュミレーション参照

◇ 数値定義

- Culimit：(変数) 使用時間に応じた 1 使用当たりの共同使用料上限金額
- Cu：(定数) 車両提供者が任意に定める共同使用料

## ■具体ユースケースシュミレーション

具体的な例を挙げて本サービスにおける共同使用料の算出及び請求・受取の流れを整理する。

### ①車両登録

下記条件の車両を車両提供者が登録する。

表 2-5 各利用時間に応じた平均走行距離(例)

項目	数値
自動車取得関連費用（購入時費用）	¥1,000,000
登録時走行距離	90,000km
年間想定走行距離	10,000km
自動車維持関連費用（年間維持費用）	¥200,000
車両登録日	2025 年 1 月 1 日
サービス手数料	30%

各時間の共同使用料上限の算出

自動車取得関連費用に関して、限界距離 19 万 km に対して 9 万 km 走行している状況であるため、残りの 10 万 km を基にして取得費用 100 万円の償却を行う。この場合の 1km 当たりの償却金額は ¥10/km となる。

自動車維持関連費用に関して、年間想定走行距離 1 万 km に対して、年間の維持費用が 20 万円の償却を行う。この場合の 1km 当たりの償却金額は ¥20/km となる。

### ②車両審査

登録された情報（初年度登録、型式、居住エリア等）から取得関連費用・維持関連費用の妥当性を確認する。異常値の場合、車両の審査を拒否した上で登録者に状況のヒアリングを行う。

### ③運用公開

車両の審査承認後、一般公開された車両に対する累計受取共同使用料の金額が受取共同使用料の上限金額を超過しないように、モニタリングを行う。

### ④車検更新時の受取共同使用料上限金額の更新

登録～サービス登録後の初回車検更新時まで

2025年1月1日に車両を登録し、2026年3月15日に車検有効期限が切れる場合を考える。

①取得関連費用 100万円

②維持関連費用 20万円を 365日 で割り、日額換算した 548円 を有効期間間での期間 1年 2か月 15日 (440日) で掛け合わせた 24万 1,120円

①+②=124万 1,120円が 2026年 3月 15日 までの受け取れる共同共同使用料の上限になる。

初回車検更新以降

ここでは 2025年 1月 1日～2026年 3月 15日 までに受け取った金額について、下記の A、B のパターンに分けて次年度以降の上限金額の算出を検討する。2026年 3月 15日 に実施した車検の有効期限が 2028年 3月 15日 であったと設定する。

A. 20万円

B. 30万円

A.の場合

2026年 3月 15日 までの上限金額の精算を行う。受け取った 20万円 は維持関連費用以下のため、上限金額から維持関連費用を引く。

$$124万 1,120円 - 24万 1,120円 = 100万$$

ここに 2028年 3月 15日 までの維持関連費用分 40万 40円 (2年 × 365日 × 548円) を足し合わせる。

$$100万円 + 40万 40円 = 140万 40円$$

この金額が 2026年 3月 15日～2028年 3月 15日 までの期間の受取共同使用料上限金額になる。

B.の場合

下記の計算式によって 2026年 3月 15日 までの上限金額の精算を行う。

$$124万 1,120円 - 30万円 = 94万 1,120円$$

ここに 2028年 3月 15日 までの維持関連費用分 40万 40円 (2年 × 365日 × 548円) を足し合わせる。

$$94万 1,120円 + 40万 40円 = 134万 1,160円$$

これが 2026年 3月 15日～2028年 3月 15日 までの期間の共同使用料受取上限金額になる。

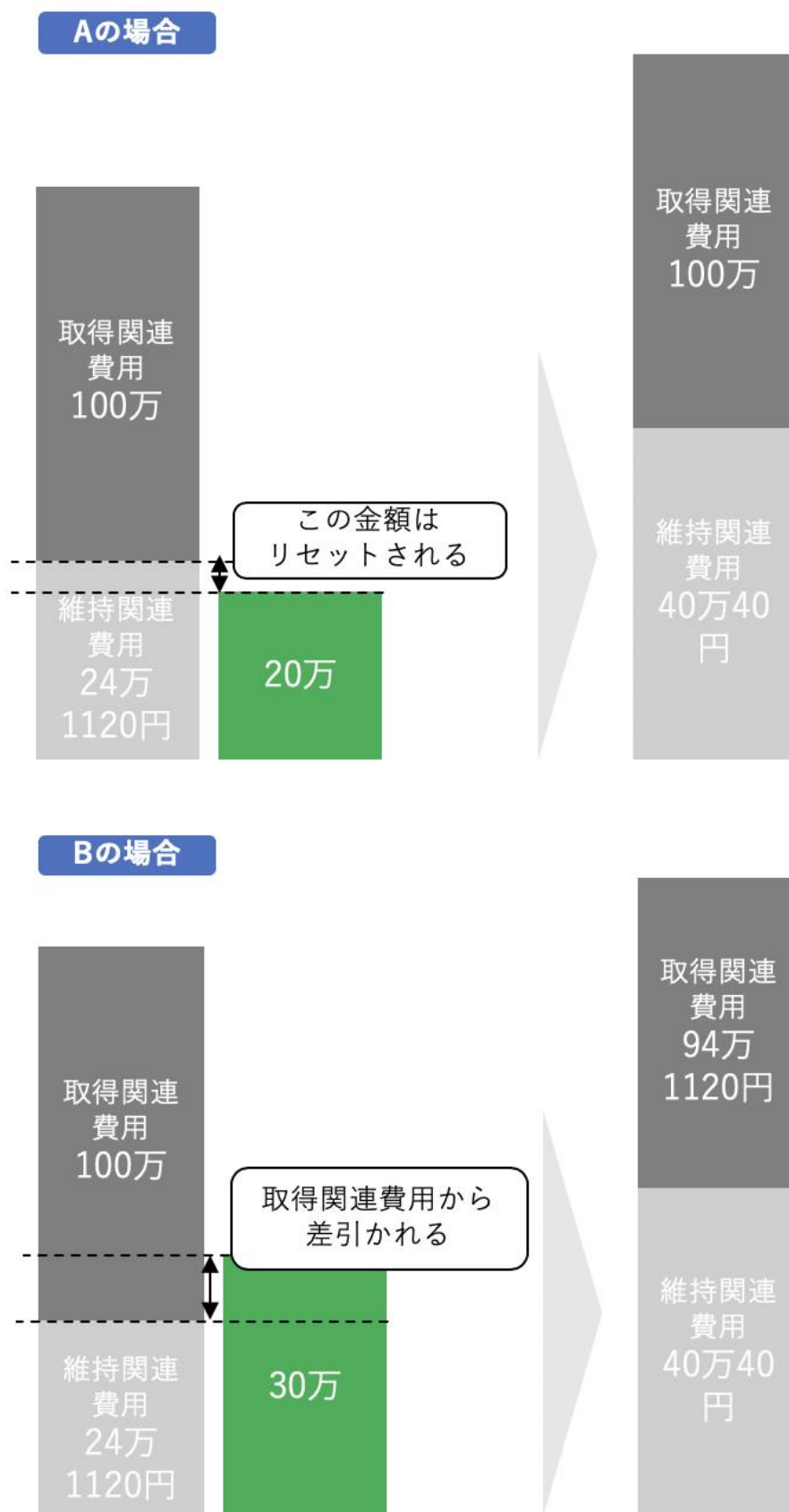


図 2-44 上限金額ロジックイメージ図

⑤共同使用料支払発生時の累計受取共同使用料と受取共同使用料上限金額との照合確認と処理

上記計算ロジックによって定められた受取共同使用料上限金額に対して、共同使用料の支払が発生するたびに当該車両の累計受取共同使用料を計算・更新する。

受取共同使用料上限金額が 124 万 1,120 円の車両の場合

利用 1：5,000 円の共同使用料が発生

サービス手数料 30%を引いた 3,500 円が受け取った累計受取共同使用料に算入される

利用 2：8,000 円の共同使用料が発生

サービス手数料 30%を引いた 5,600 円が累計受取共同使用料に算入される

利用 1+利用 2 を合わせた 9,100 円が累計受取共同使用料となる。

受取共同使用料上限金額の 124 万 1,120 に対して、この累計受取共同使用料の到達段階に応じて各処理を実施する。

85% (105 万 4,952 円) に達した場合：警告の通知

90% (111 万 7,008 円) に達した場合：停止予告の通知

95% (117 万 9,064 円) に達した場合：公開停止処理を実行し、新規予約が作成できないようにする。

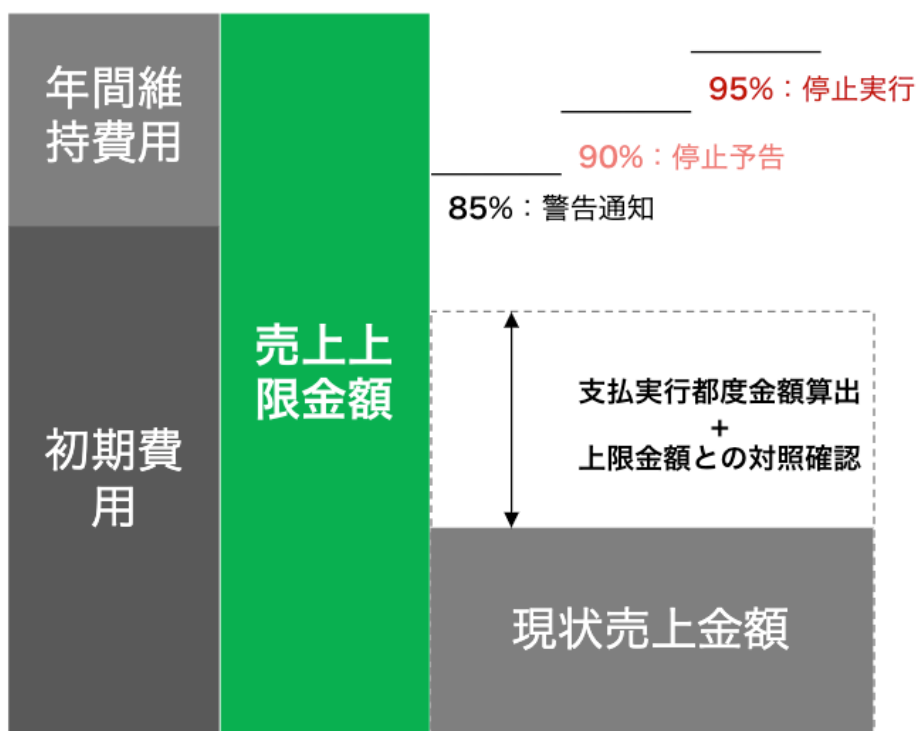


図 2-45 上限金額ロジックイメージ図

## 2-2. システムコンポーネント (CO)

### 2-2-1. システムコンポーネント図

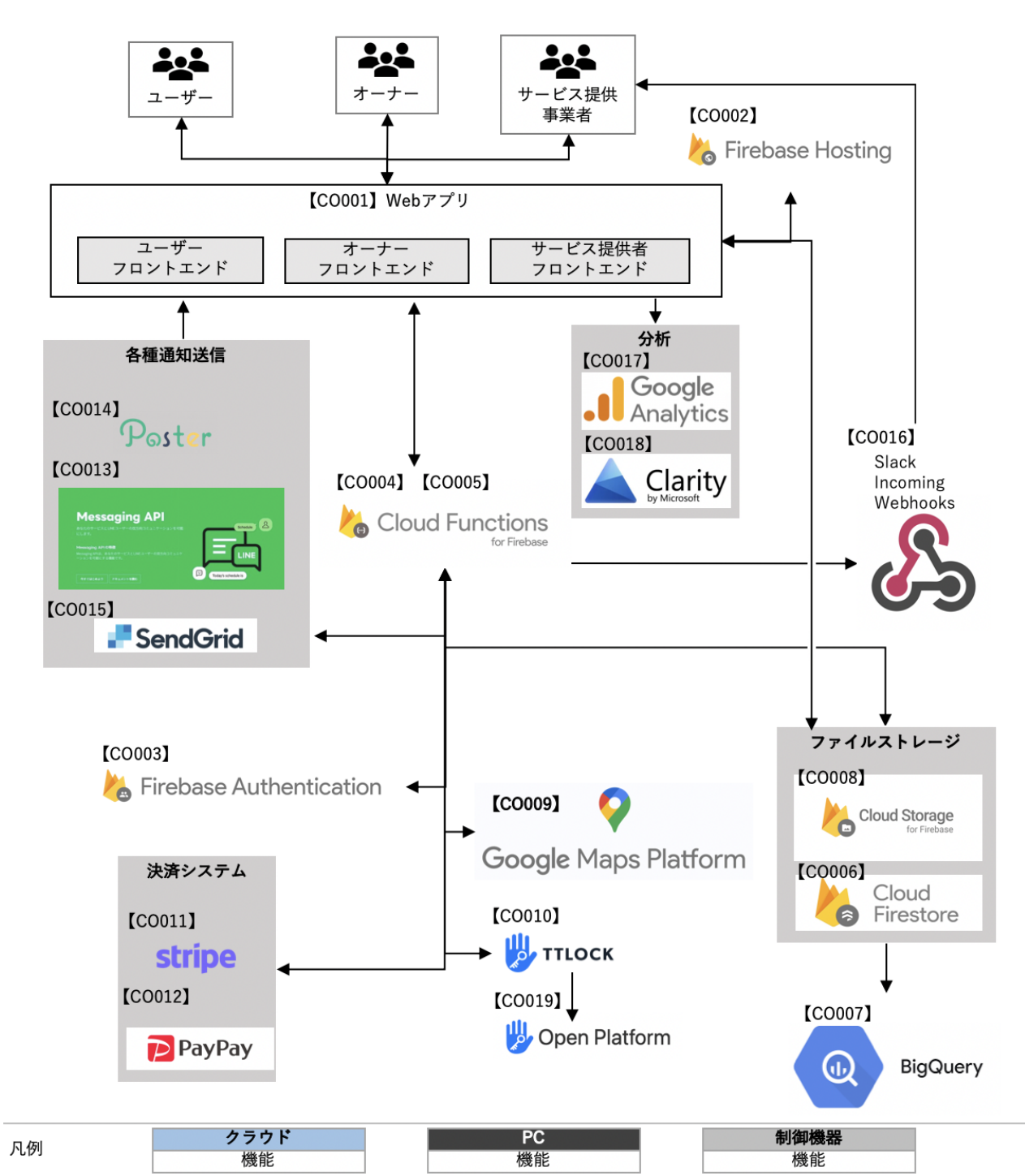


図 2-46 システムコンポーネント図

## 2-2-2. システムコンポーネント一覧

表 2-6 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	Web アプリ	HTML・CSS・JavaScript	<ul style="list-style-type: none"> <li>● Web ブラウザ上で動作するフロントエンドアプリケーション。</li> </ul>
CO002	ホスティングサーバー	Firebase Hosting	<ul style="list-style-type: none"> <li>● Web アプリをホスティングしているサーバー。</li> <li>● Google の提供する信頼性の高いホスティングサービスである Firebase Hosting を使用している。</li> </ul>
CO003	ユーザー認証システム	Firebase Authentication with Identity Platform	<ul style="list-style-type: none"> <li>● Web アプリのログインやメールアドレス及び電話番号の確認を実現するシステム。</li> <li>● Google の提供する信頼性の高いユーザー認証システムである Firebase Authentication with Identity Platform を使用している。</li> </ul>
CO004	API サーバー	Cloud Functions for Firebase	<ul style="list-style-type: none"> <li>● Web アプリからリクエストを受け、各種タスクを行う API サーバー。</li> <li>● Google の提供する信頼性の高いサーバーレスなバックエンド実行サービスである Cloud Functions for Firebase を使用している。</li> </ul>
CO005	バッチサーバー	Cloud Functions for Firebase	<ul style="list-style-type: none"> <li>● 定期的に各種タスクを行うバッチサーバー。</li> <li>● Google の提供する信頼性の高いサーバーレスなバックエンド実行サービスである Cloud Functions for Firebase を使用している。</li> </ul>
CO006	データベース	Cloud Firestore	<ul style="list-style-type: none"> <li>● ユーザー情報や予約情報などの各種データを保存するデータベース。</li> <li>● Google の提供する信頼性の高いサーバーレスの NoSQL ドキュメントデータベースである Cloud Firestore を使用している。</li> </ul>
CO007	分析基盤	BigQuery	<ul style="list-style-type: none"> <li>● 予約情報などを集計して分析する分析基盤。</li> <li>● Google の提供する信頼性の高いサーバーレスなエンタープライズデータウェアハウスである BigQuery を使用している。</li> </ul>
CO008	ストレージ	Cloud Storage	<ul style="list-style-type: none"> <li>● ユーザーの運転免許証の画像や車両の画像などのファイルを保存するストレージ。</li> <li>● Google の提供する信頼性の高いオブジェクトストレージサービスである Cloud Storage for Firebase を使用している。</li> </ul>
CO009	Google マップ表示サービス	Google Maps Platform API	<ul style="list-style-type: none"> <li>● Google マップを表示するサービス。</li> <li>● マップで車両の位置の表示を行う。</li> </ul>

CO010	キーボックス管理サービス	TTLock API	<ul style="list-style-type: none"> <li>● スマートキーボックスの管理を行うサービスとの接続 API。</li> <li>● 予約の利用開始時にスマートキーボックスのパスコードの発行を行う TTLock Open Platform との接続を行う。</li> </ul>
CO011	クレジットカード決済代行サービス	Stripe API	<ul style="list-style-type: none"> <li>● クレジットカードによる決済を代行するサービス。</li> <li>● 予約終了後に請求を発行し、ユーザーが支払を行う。</li> </ul>
CO012	PayPay 決済サービス	PayPay Open Payment API	<ul style="list-style-type: none"> <li>● PayPay による決済をするサービス。</li> <li>● 予約終了後に請求を発行し、ユーザーが支払を行う。</li> </ul>
CO013	LINE メッセージ送信サービス	LINE Messaging API	<ul style="list-style-type: none"> <li>● LINE にメッセージを送信するサービス。</li> <li>● 各種通知（審査完了通知・予約リマインド通知等）を送信する。</li> </ul>
CO014	LINE ユーザー管理サービス	Poster API	<ul style="list-style-type: none"> <li>● LINE のユーザーを高度に管理するサービス。</li> <li>● LINE のユーザーに対してタグ付けを行う。</li> </ul>
CO015	メール送信サービス	SendGrid API	<ul style="list-style-type: none"> <li>● メールを送信するサービス。</li> <li>● 各種通知（審査完了通知・予約リマインド通知等）を送信する。</li> </ul>
CO016	Slack メッセージ投稿サービス	SlackIncoming Webhooks	<ul style="list-style-type: none"> <li>● Slack のチャンネルにメッセージを投稿するサービス。</li> <li>● 必要な業務通知を行う。</li> </ul>
CO017	ウェブ解析ツール	Google Analytics 4	<ul style="list-style-type: none"> <li>● Google が提供するウェブ解析ツール。</li> <li>● Web アプリのユーザーの行動を分析する。</li> </ul>
CO018	行動分析ツール	Microsoft Clarity	<ul style="list-style-type: none"> <li>● Microsoft が提供する行動分析ツール。</li> <li>● Web アプリのユーザーの行動を分析する。</li> </ul>
CO019	キーボックス管理サービス	TTLock Open Platform	<ul style="list-style-type: none"> <li>● スマートキーボックスのパスコードや管理者ロールの付与等の機能を提供する SaaS サービス。</li> <li>● TTLock API を通じて本システムと接続している。</li> </ul>

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

#### 【CO001】 Web アプリ

- ベンダー
  - 自社開発
- 公式サイト
  - <https://ourcar.app/>
- 概要

- 共同使用を通じた車両の提供における必要な機能及び無人で 24 時間貸出ができるシステムを提供する。
- 主な機能
  - 本ドキュメントの 2 章を参照。

#### 【CO002】 Firebase Hosting

- ベンダー
  - Google
- 公式サイト
  - <https://firebase.google.com/products/hosting>
- 概要
  - 高速で安全な静的及び動的な Web コンテンツのホスティングサービスである。グローバル CDN を利用してコンテンツを配信し、SSL 証明書の自動管理やカスタムドメインのサポートも提供する。開発者は簡単に Web サイトやアプリの公開・更新が可能である。
- 主な機能
  - 高速なコンテンツ配信： グローバルに分散した CDN を通じて、低遅延でコンテンツを配信できる。
  - SSL 自動管理： 無料の SSL 証明書を自動で発行・更新し、HTTPS 通信を標準で提供する。
  - カスタムドメイン対応： 独自ドメインを簡単に設定でき、ブランドイメージに合わせたサイト運用が可能。
  - シングルページアプリ対応： リライトルールを設定することで、SPA のルーティングを適切に処理できる。

#### 【CO003】 Firebase Authentication with Identity Platform

- ベンダー
  - Google
- 公式サイト
  - <https://firebase.google.com/docs/auth?hl=ja>
- 概要
  - Web やモバイルアプリに安全かつ簡単にユーザー認証機能を追加できるサービスである。メール・パスワード認証、電話番号認証、OAuth プロバイダー（Google など）に対応し、セキュアなユーザー管理を実現する。
- 主な機能
  - 多様な認証方式のサポート： メール・パスワード、電話番号、OAuth プロバイダーを簡単に組み込める。
  - 簡単なユーザー管理： ユーザー登録、ログイン、ログアウト、パスワードリセットなどの操作を容易に実装できる。
  - セキュリティ強化： トークンの自動更新や多要素認証（MFA）に対応し、不正アクセスを防止できる。
  - クロスプラットフォーム対応： iOS、Android、Web で統一された API を提供し、一貫した認証体験を実現する。

#### 【CO004】 Cloud Functions for Firebase

- ベンダー
  - Google
- 公式サイト
  - <https://firebase.google.com/products/functions>
- 概要
  - Firebase と Google Cloud のインフラを活用し、サーバーレスでバックエンドコードを実行できる環境である。イベントドリブンの関数を記述して、Firebase の各種サービスや HTTP リクエストに応じた処理を自動で行える。
  - 本システムでは、Web アプリからリクエストを受け、各種タスクを行う API サーバーとして利用する。
- 主な機能
  - イベント駆動型処理： Firebase Authentication や Cloud Firestore などのイベントに連動して関数を実行できる。
  - HTTP トリガー対応： REST API のエンドポイントとして関数を公開し、外部からの HTTP リクエストに応答可能。
  - 自動スケーリング： リクエストに応じて関数が自動的にスケールアップ・ダウンし、効率的なリソース管理を実現。
  - ロギングとデバッグ： Cloud Logging と統合され、関数の動作状況をモニタリング及びデバッグできる。

#### 【CO005】 Cloud Functions for Firebase

- ベンダー
  - Google
- 公式サイト
  - <https://firebase.google.com/products/functions>
- 概要
  - Firebase と Google Cloud のインフラを活用し、サーバーレスでバックエンドコードを実行できる環境である。イベントドリブンの関数を記述して、Firebase の各種サービスや HTTP リクエストに応じた処理を自動で行える。
  - 本システムでは、定期的に各種タスクを行うバッチサーバーとして利用する。
- 主な機能
  - イベント駆動型処理： Firebase Authentication や Cloud Firestore などのイベントに連動して関数を実行できる。
  - HTTP トリガー対応： REST API のエンドポイントとして関数を公開し、外部からの HTTP リクエストに応答可能。
  - 自動スケーリング： リクエストに応じて関数が自動的にスケールアップ・ダウンし、効率的なリソース管理を実現。
  - ロギングとデバッグ： Cloud Logging と統合され、関数の動作状況をモニタリング及びデバッグできる。

#### 【CO006】 Cloud Firestore

- ベンダー
  - Google
- 公式サイト
  - <https://firebase.google.com/products/firestore>
- 概要
  - スケーラブルでリアルタイムな NoSQL ドキュメントデータベースサービスである。柔軟なデータ構造と強力なクエリ機能を備え、モバイル・Web・サーバーアプリケーションのデータ管理を効率的に行える。
- 主な機能
  - リアルタイム同期： データの変更を即座にクライアントに反映し、リアルタイムの共同作業を実現。
  - 柔軟なデータモデル： ドキュメントとコレクションによる階層構造で、複雑なデータを直感的に管理可能。
  - 強力なクエリ機能： フィルタリングやソート、複合クエリに対応し、高度な検索が行える。
  - オフライン対応： ネットワークが不安定な環境でもデータの読み書きが可能で、自動同期される。

#### 【CO007】 BigQuery

- ベンダー
  - Google
- 公式サイト
  - <https://cloud.google.com/bigquery>
- 概要
  - Google Cloud が提供するサーバーレスで高速な大規模データ分析プラットフォームである。SQL に似たクエリ言語を用い、ペタバイト規模のデータセットに対して迅速にクエリを実行できる。データウェアハウスとしての利用やリアルタイム分析に適している。
- 主な機能
  - 高速なクエリ処理：分散型のインフラにより大規模データの複雑な分析を短時間で実行できる。
  - スケーラブルなストレージ： データ量に応じて自動的にスケールし、大量データを効率的に格納可能。
  - 標準 SQL 対応： 親しみやすい SQL ライクな言語でクエリを記述でき、多様な分析ニーズに対応。
  - セキュリティとアクセス制御： IAM と連携し、きめ細かなデータアクセス管理が可能。

#### 【CO008】 Cloud Storage

- ベンダー
  - Google
- 公式サイト
  - <https://firebase.google.com/products/storage>
- 概要

- 大容量のファイル（画像、動画、音声、その他のユーザー生成コンテンツなど）を安全かつスケーラブルに保存・配信できるクラウドストレージサービスである。Firebase Authentication と連携し、細かいアクセス制御が可能である。
- 主な機能
  - 大容量ファイルの保存・配信： 画像や動画などのメディアファイルを効率的にアップロード・ダウンロード可能。
  - セキュアなアクセス管理： Firebase Authentication と統合し、ユーザーごとにアクセス権限を細かく設定できる。
  - 高い可用性とスケーラビリティ： Google のインフラ上で動作し、大量のトラフィックやデータに対応可能。
  - 簡単なアップロード SDK： Web やモバイル向けの SDK で、ファイルアップロードの実装が容易。

#### 【CO009】 Google Maps Platform API

- ベンダー
  - Google
- 公式サイト
  - <https://developers.google.com/maps>
- 概要
  - Google の地図サービスを Web やモバイルアプリに組み込むための一連の API 群である。地図表示、ジオコーディング、ルート案内、場所情報検索など多彩な機能を提供し、位置情報を活用したアプリケーション開発を支援する。
- 主な機能
  - 地図の表示とカスタマイズ： インタラクティブな地図を Web ページやアプリに埋め込み、スタイルや操作をカスタマイズ可能。
  - ジオコーディングと逆ジオコーディング： 住所から座標への変換や、座標から住所情報の取得を行える。
  - ルート検索とナビゲーション： 車、徒歩、公共交通機関などの最適ルートを計算し案内できる。
  - 場所情報検索： 店舗や施設などの詳細情報を取得し、ユーザーの位置情報サービスを強化できる。

#### 【CO010】 TTLock API

- ベンダー
  - Guangzhou Oribo Electronic Technology Co., Ltd. （広州欧瑞博電子科技有限公司）
- 公式サイト
  - <https://euopen.ttlock.com/document/doc?urlName=userGuide%2FekeyEn.html>
- 概要
  - スマートロック製品を管理・制御するための API 群である。ドアロックの施錠・解錠、ユーザー権限管理、アクセス履歴の取得などを遠隔から操作可能にし、スマートホームや施設管理システムにおけるアクセスコントロールを実現する。
- 主な機能
  - リモート施錠・解錠： スマートロックの状態を遠隔で操作し、アクセス管理を効率化できる。

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- ユーザー管理： アクセス権限の設定やユーザーの追加・削除を API で柔軟に制御可能。
- アクセス履歴取得： ロックの利用状況や操作履歴を取得し、セキュリティ管理を強化する。
- スケジュール管理： 指定時間帯のみアクセス可能な権限設定や一時的なパスコードの発行を行える。

### 【CO011】 Stripe API

- ベンダー
  - Stripe, Inc.
- 公式サイト
  - <https://docs.stripe.com/development?locale=ja-JP>
- 概要
  - オンライン決済やサブスクリプション管理をはじめとする多様な金融取引をウェブやモバイルアプリに組み込むための API 群である。安全かつ柔軟に決済機能を実装でき、世界中の多通貨・多決済手段に対応する。
- 主な機能
  - 支払処理： クレジットカード、デビットカード、ウォレット、銀行振込など多様な支払手段に対応。
  - サブスクリプション管理： 定期課金の作成、更新、キャンセルを API で操作可能。
  - 顧客管理： 顧客情報の登録、保存、支払履歴の管理を行える。
  - セキュリティ機能： PCI DSS 準拠の安全なトークン化や 3D セキュア認証、詐欺防止ツールを提供。

### 【CO012】 PayPay Open Payment API

- ベンダー
  - PayPay 株式会社
- 公式サイト
  - <https://developer.PayPay.ne.jp/>
- 概要
  - PayPay のキャッシュレス決済を自社のサービスやアプリに組み込むための API である。QR コード決済や支払の開始、決済状況の確認などの機能を提供し、スムーズなキャッシュレス決済体験を実現する。
- 主な機能
  - 決済の開始： QR コード生成や決済リクエストを API 経由で行い、ユーザーに支払を促せる。
  - 決済状況の確認： 支払完了やキャンセルなどのステータスをリアルタイムに取得できる。
  - 取引管理： 取引の履歴取得やキャンセル処理など、決済後の管理操作をサポート。
  - 認証・セキュリティ： API リクエストは署名付きで安全にやり取りされ、不正アクセスを防止する。

### 【CO013】 LINE Messaging API

- ベンダー
  - LINE ヤフー株式会社

- 公式サイト
  - <https://developers.line.biz/ja/docs/messaging-api/overview/>
- 概要
  - LINE プラットフォーム上でボットを作成・運用するための API である。メッセージの送受信、リッチコンテンツの配信、ユーザー管理などを行い、ユーザーと自動的かつ双方向にコミュニケーションを取ることができる。
- 主な機能
  - メッセージ送受信： テキスト、画像、動画、スタンプ、位置情報など多様なメッセージ形式に対応。
  - リッチメッセージ・Flex メッセージ： カスタマイズ可能なリッチコンテンツを利用して魅力的な UI を提供できる。
  - ユーザー管理： ユーザーのプロフィール取得やブロック・友だち追加の管理が可能。
  - Webhook 対応： イベント通知を受取、リアルタイムで応答処理を行える。

#### 【CO014】 Poster API

- ベンダー
  - 株式会社モスコソリューションズ
- 公式サイト
  - <https://poster.ooo/>
- 概要
  - LINE 公式アカウントのマーケティング・コミュニケーションを支援するサードパーティーAPI である。メッセージ配信の自動化や顧客管理、効果測定などの機能を提供し、企業の LINE 活用を効率化する。
- 主な機能
  - メッセージ配信自動化： 定期配信やシナリオ配信など、LINE 公式アカウントからのメッセージ送信を自動化できる。
  - 顧客管理： ユーザー属性や行動履歴に基づくセグメント管理が可能。
  - 効果測定・分析： 配信結果の統計情報を取得し、キャンペーンの効果を可視化できる。
  - API 連携： 他システムとの連携により、柔軟なマーケティング施策の構築を支援。

#### 【CO015】 SendGrid API

- ベンダー
  - Twilio SendGrid, Inc.
  - 株式会社構造計画研究所が正規販売代理店（日本国内代理店）
- 公式サイト
  - グローバルサイト
    - ◇ <https://www.twilio.com/ja-jp/products/email-api>
  - 日本国内代理店サイト
    - ◇ <https://sendgrid.kke.co.jp/>
- 概要

- SendGrid のメール配信サービスをプログラムから利用できる API である。大量メールの送信やメールテンプレートの管理、配信状況のモニタリングなどを自動化し、信頼性の高いメール配信を実現する。
- 主な機能
  - メール送信： シンプルな API で大量のトランザクションメールやマーケティングメールを送信可能。
  - テンプレート管理： 動的テンプレートの作成・更新・利用ができ、パーソナライズされたメールを送れる。
  - 配信状況の追跡： 配信成功率や開封率、クリック率などの詳細なレポートを取得可能。
  - イベント通知： 配信失敗やバウンス、スパム報告などのイベントを Webhook でリアルタイムに受け取れる。

#### 【CO016】 Slack Incoming Webhooks

- ベンダー
  - Slack Technologies, Inc.
- 公式サイト
  - <https://api.slack.com/messaging/Webhooks>
- 概要
  - 外部アプリケーションやサービスから Slack チャンネルへメッセージを投稿するためのシンプルな HTTP エンドポイントである。JSON 形式のペイロードを送信するだけで通知や情報共有が可能になる。
- 主な機能
  - 簡単なメッセージ送信： 事前に発行した Webhook URL に POST リクエストを送るだけでメッセージを投稿できる。
  - カスタマイズ可能なメッセージ： テキストだけでなく、添付ファイルやブロック形式のリッチメッセージも送信可能。
  - チャンネル指定： 特定のチャンネルやユーザーに対してメッセージを送信できる。
  - セキュリティ： Webhook URL の管理でアクセス制限が可能だが、URL の漏えいには注意が必要。

#### 【CO017】 Google Analytics 4

- ベンダー
  - Google
- 公式サイト
  - <https://marketingplatform.google.com/intl/ja/about/analytics/>
- 概要
  - Web サイトやアプリのアクセス解析サービスである。訪問者の行動データを収集・分析し、ユーザーの利用状況やマーケティング効果を可視化することで、サイトの改善や戦略立案を支援する。
- 主な機能
  - リアルタイム解析： 現在の訪問者数や行動をリアルタイムで把握できる。
  - ユーザー行動分析： ページビュー、セッション、直帰率、滞在時間など詳細な行動データを提供。
  - 集客経路分析： オーガニック検索、広告、SNS など各チャンネルの流入状況を把握。

- カスタムレポート・ダッシュボード： 分析目的に応じたレポートやダッシュボードを作成可能。

#### 【CO018】 Microsoft Clarity

- ベンダー
  - Microsoft
- 公式サイト
  - <https://clarity.microsoft.com/>
- 概要
  - Web サイトのユーザー行動解析ツールである。ヒートマップやセッションリプレイ機能を提供し、訪問者の操作を視覚的に分析できる。サイトの使いやすさ改善や問題発見に役立つ。
- 主な機能
  - セッションリプレイ： ユーザーの画面操作を録画し、実際の操作を再生して分析できる。
  - ヒートマップ： クリックやスクロールの分布を視覚化し、ユーザーの注目ポイントを把握可能。
  - パフォーマンス測定： ページの読み込み速度や動作状況の分析が行える。
  - ユーザー属性分析： 地域やデバイス、ブラウザ情報などのユーザーデータを集計。

【CO019】 TTLock Open Platform

- ベンダー
  - Guangzhou Oribo Electronic Technology Co., Ltd. (広州欧瑞博電子科技有限公司)
- 公式サイト
  - <https://open.sciener.com/>
- 概要
  - TTLock のスマートロック製品を他のソフトウェアやサービスと統合するために提供されている、開発者向けの統合プラットフォーム。
  - これにより、企業や開発者は、TTLock のハードウェア（スマートロック）を活用して、独自のモバイルアプリケーション等の自社サービスにその機能を組み込むことが可能。
- 主な機能
  - パスコードの発行と管理：期間限定、一度限り、又は永久に利用できる権限を遠隔で生成し、ユーザーに共有できる。
  - パスコードと IC カードの管理：遠隔で一時的なパスコードや、繰り返し使えるパスコードを発行可能。IC カードや指紋、リモコンなどの登録・管理が可能。
  - 遠隔でのロック操作：Wi-Fi ゲートウェイを介して、物理的に離れた場所からスマートロックの施錠・解錠を遠隔で行うことができる。
  - 操作履歴の取得：誰が、いつ、どのようにスマートロックを開錠したか（パスコード、指紋、電子鍵など）の記録を取得できる。
  - ユーザーと権限の管理：複数のユーザーアカウントを作成し、それぞれのユーザーに特定のスマートロックのアクセス権限を付与したり、変更したりできる。
  - API と SDK：REST API や、iOS/Android 向けの SDK (Software Development Kit) が提供されており、簡単に独自のアプリケーションに機能を統合できる。

## 2-3. ハードウェア (HW)

### 2-3-1. ハードウェアアーキテクチャ

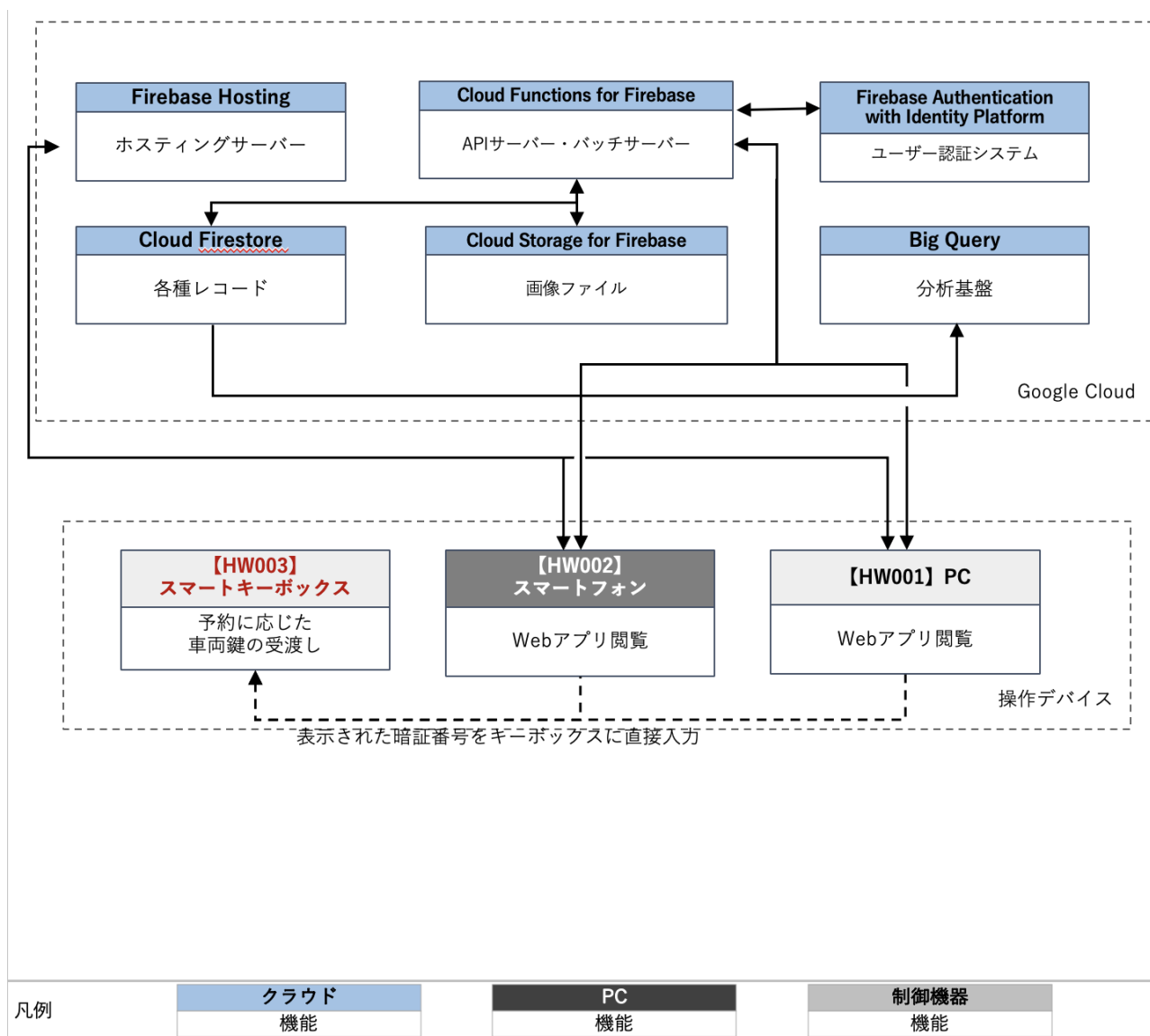


図 2-47 ハードウェアアーキテクチャ

## 2-3-2. ハードウェア一覧

表 2-7 ハードウェア一覧

※朱文字：新規開発・既存改修

ID	種別	ベンダー	品番	用途
HW001	PC	—	型式全般	Web アプリユーザーのアクセス端末。 対応 OS： Windows/macOS、対応ブラウザ： Chrome/Safari。
HW002	スマートフォン	—	型式全般	Web アプリユーザーのアクセス端末。 対応 OS： Windows/macOS、対応ブラウザ： Chrome/Safari。
HW003	スマートキーボックス	Dongguan Qishengda Technology Co., Ltd	—	車両鍵の受渡し。

### 2-3-3. ハードウェアの詳細

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

#### 【HW001】 PC

- 本ハードウェアの概要
  - Web システムにアクセスするための PC
- 本ハードウェアを提供するベンダー
  - 型式全般
- 本ハードウェアの仕様・スペック
  - -
- イメージ



図 2-48 PC

Apple 公式 HP : <https://www.apple.com/jp/mac/>

#### 【HW002】 スマートフォン

- 本ハードウェアの概要
  - Web システムにアクセスするためのスマートフォン
- 本ハードウェアの仕様・スペック
  - -
- イメージ



図 2-49 スマートフォン

Apple 公式 HP : <https://www.apple.com/jp/mac/>

**【HW003】 スマートキーボックス<機能改修>**

- 本ハードウェアの概要
  - 車両鍵を格納し、予約にひも付くパスコードで解錠する
  - 車両の外部に取り付けられ、無人で車両鍵の受渡しが可能
- 本ハードウェアを提供するベンダー
  - Dongguan Qishengda Technology Co., Ltd
- 本ハードウェアの仕様・スペック
  - OTG による携帯等からの給電受付機能
  - IP65 程度の防水機能
  - 物理的な詰まりが発生しないボックス設計
- イメージ



図 2-50 スマートキーボックス

## 2-4. データインターフェース (IF)

### 2-4-1. データアーキテクチャ

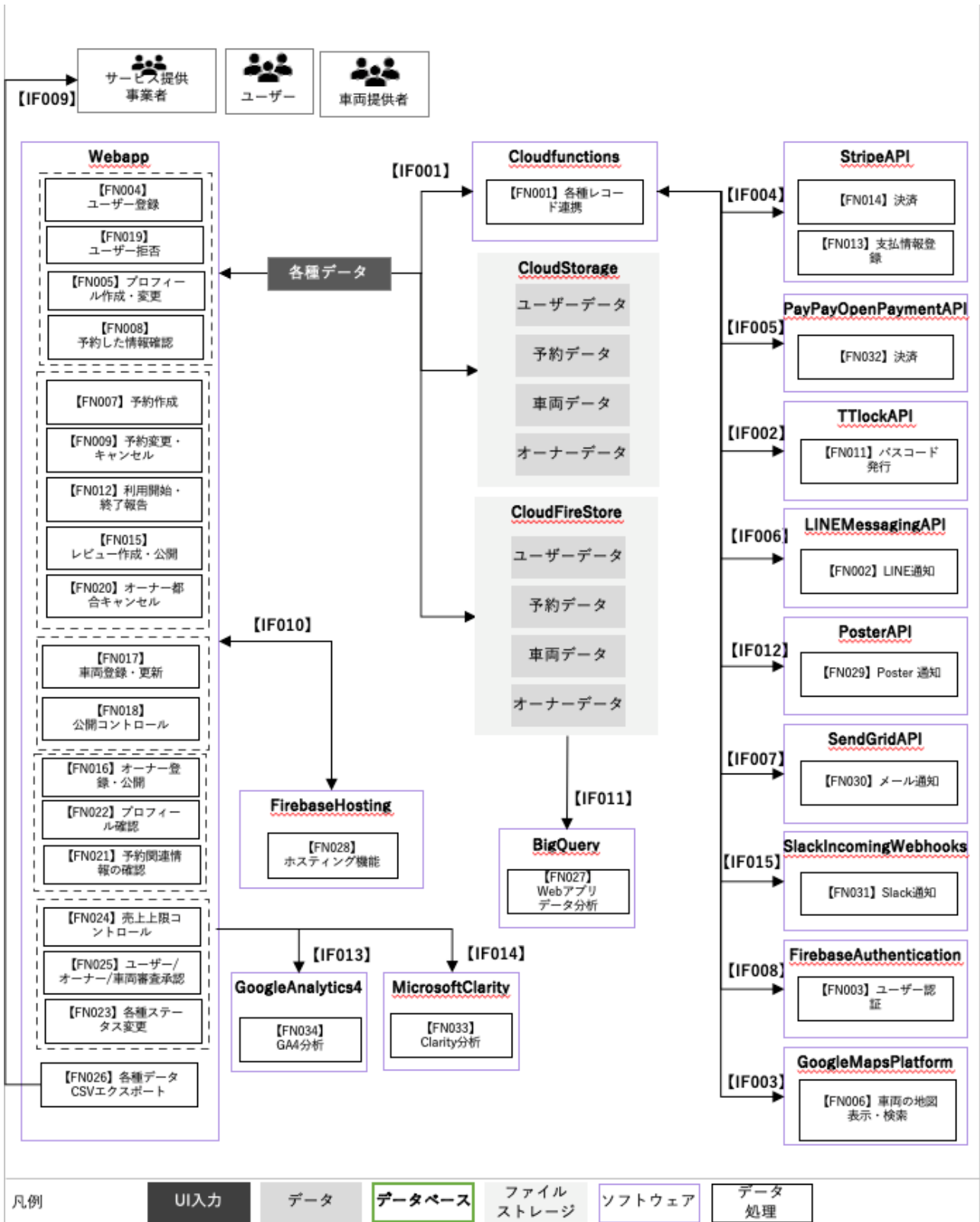


図 2-51 データアーキテクチャ

## 2-4-2. データインターフェース一覧

表 2-8 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	ユーザーデータの作成・取得 IF	FN001	FN004、005
IF002	キーボックスパスコード発行 API (TTLock API)	FN001	FN011
IF003	車両の地図表示 API (Google Maps Platform API)	FN001	FN006
IF004	クレジットカード決済 API (Stripe API)	FN001	FN013、014
IF005	PayPay 決済 API (PayPay Open Payment API)	FN001	FN032
IF006	LINE メッセージ送信 API (LINE Messaging API)	FN001	FN002
IF007	メール送信 API (SendGrid API)	FN001	FN030
IF008	ユーザー認証 IF (Firebase Authentication with Identity Platform)	FN001	FN003
IF009	各種データのファイル出力 IF	FN026	—
IF010	ページ取得 IF (Firebase Hosting)	FN004、005、008、007、 009、012、015、016、 017、018、019、020、 021、022、023、024、 025、026	FN004、005、008、 007、009、012、015、 016、017、018、019、 020、021、022、023、 024、025、026
IF011	Cloud Firestore と BigQuery の連携 IF	CO006、008	FN027
IF012	LINE ユーザー管理 API (Poster API)	FN002	FN001
IF013	ウェブ解析ツール連携 IF (Google Analytics 4)	FN027	FN001
IF014	行動分析ツール連携 IF (Microsoft Clarity)	FN027	FN001
IF015	Slack メッセージ投稿 IF (Slack Incoming Webhooks)	FN002	FN001
IF016	Cloud Storage イベントトリガー	CO004、005	CO008
IF017	Cloud Functions プログラムアクセス	CO004、005	FN006
IF018	予約データの作成・取得 IF	FN001	FN007、FN009、 FN012、FN020、 FN008、FN021

共同使用契約に基づくカーシェアリングシステム システム設計書

IF019	レビューデータの作成・取得 IF	FN001	FN015
IF020	車両データの作成・取得 IF	FN001	FN017、 FN018、 FN019、 FN024
IF021	車両提供者データの作成・取得 IF	FN001	FN016、 FN022
IF022	各種データ承認 IF	FN001	FN023、 FN025

### 2-4-3. データインターフェースの詳細

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

#### 【IF001】 ユーザーデータの作成・取得 IF

- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、ユーザー登録や予約作成をする際や、ユーザー情報や予約情報の閲覧をする際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】 各種通知送信
  - 【FN003】 ユーザー認証
  - 【FN004】 ユーザー登録
  - 【FN005】 プロフィール作成・変更
  - 【FN006】 車両の地図表示・検索
  - 【FN010】 パスコード発行
  - 【FN011】 キーボックス解錠
  - 【FN013】 支払情報登録
  - 【FN014】 クレジット決済
  - 【FN027】 Web アプリデータ分析
- プロトコル・データ形式
  - ライブラリ【SL005】 Firebase（firebase）の実装に依存
    - ◇ <https://github.com/firebase/firebase-js-sdk>
- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-9 ユーザーデータ

No.	カラム名	概要	データ種
1	userId	ユーザーID	テキスト
2	type	ユーザー種別	テキスト
3	state	承認ステータス	テキスト
4	email	メールアドレス	テキスト
5	phoneNumber	電話番号	テキスト
6	lastName	名字	テキスト
7	firstName	名前	テキスト
8	lastNameKatakana	ミョウジ	テキスト
9	firstNameKatakana	ナマエ	テキスト
10	dateOfBirth	生年月日	テキスト
11	occupation	職業	テキスト
12	industry	業種	テキスト
13	country	国籍	テキスト
14	postCode	郵便番号	テキスト
15	address	住所	テキスト
16	passportEntryDate	日本国上陸日（国際免許ユーザーのみ）	テキスト
17	licenseFormat	国際免許種（国際免許ユーザーのみ）	テキスト
18	licenseAddress	免許に記載の住所	テキスト
19	licenseNumber	免許証番号	テキスト
20	licenseType	免許種別	テキスト
21	licenseTypes	免許種別コード（国際免許ユーザーのみ）	テキスト
22	licenseTransmission	運転可能なトランスミッション	テキスト
23	licenseIssueDate	免許発行日	テキスト
24	licenseExpirationDate	免許有効期限	テキスト
25	campaignCode	キャンペーンコード	テキスト
26	invitedBy	紹介コード	テキスト
27	createdAt	作成日時	テキスト
28	updatedAt	更新日時	テキスト
29	licenseImage	免許証画像	JPEG
30	impressedBy	認知のきっかけ	テキスト
31	impressedByOtherDetails	その他きっかけ	テキスト
32	attractedBy	サービスを選んだ理由	テキスト
33	attractedByOtherDetails	その他の理由	テキスト
34	situation	使用を想定するユースケース	テキスト
35	situationOtherDetails	その他のユースケース	テキスト

36	createdAt	作成日時	テキスト
37	updatedAt	更新日時	テキスト
38	profile_image	プロフィール画像	JPEG
39	profile_text	プロフィール文章	テキスト
40	profile_rate	総合評価	数字
41	licenseColor	免許証カラー	テキスト
42	drivingExperiencedTime	運転歴	数字

【IF002】キーボックスパスコード発行 API (TTLock API)

- 本インターフェースの概要
  - Web アプリのバックエンドにおいて、予約に対するスマートキーボックスのパスコードを発行するための API
- 本インターフェースを利用する機能
  - 【FN001】各種レコード連携
- プロトコル・データ形式
  - アクセストークンの取得
    - ◇ <https://euopen.ttlock.com/document/doc?urlName=cloud%2Foauth%2FgetAccessTokenEn.html>
  - パスコードの発行
    - ◇ <https://euopen.ttlock.com/document/doc?urlName=cloud%2Fpasscode%2FgetEn.html>

【IF003】車両の地図表示 API (Google Maps Platform API)

- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、Google マップの表示や車両のピン表示をするための API
- 本インターフェースを利用する機能
  - 【FN001】各種レコード連携
- プロトコル・データ形式
  - ライブラリ【SL008】Google Maps JavaScript API React Wrapper (@googlemaps/react-wrapper) の実装に依存
    - ◇ <https://github.com/googlemaps/react-wrapper>

【IF004】クレジットカード決済 API (Stripe API)

- 本インターフェースの概要
  - Web アプリのバックエンドにおいて、利用終了した予約に対して請求を発行し、Web アプリのフロントエンドにおいて、クレジットカードによる支払をするための API。
- 本インターフェースを利用する機能
  - 【FN001】各種レコード連携
- プロトコル・データ形式
  - 支払方法の設定
    - ◇ ライブラリ【SL009】Stripe Node.js Library (stripe) 【SL010】Stripe.js の実装に依存

- <https://github.com/stripe/stripe-node>
- <https://github.com/stripe/stripe-js>
- ◇ 関連ドキュメント
  - <https://docs.stripe.com/payments/setup-intents>
- 請求の作成及び支払
  - ◇ ライブラリ 【SL009】 Stripe Node.js Library (stripe) 【SL010】 Stripe.js (stripe-js) の実装に依存
    - <https://github.com/stripe/stripe-node>
    - <https://github.com/stripe/stripe-js>
  - ◇ 関連ドキュメント
    - <https://docs.stripe.com/payments/payment-intents>

#### 【IF005】 PayPay 決済 API (PayPay Open Payment API)

- 本インターフェースの概要
  - Web アプリのバックエンドにおいて、利用が終了した予約に対して請求を作成し、PayPay のアプリ又は Web サイトにおいて、PayPay による支払をするための API。
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
- プロトコル・データ形式
  - 請求の作成及び支払用 URL の発行
    - ◇ ライブラリ 【SL 011】 PayPay OPA SDK (@PayPayopa/PayPayopa-sdk-node) の実装に依存
      - <https://github.com/PayPay/PayPayopa-sdk-node>
    - ◇ 関連ドキュメント
      - <https://developer.PayPay.ne.jp/products/docs/Webpayment>

#### 【IF006】 LINE メッセージ送信 API (LINE Messaging API)

- 本インターフェースの概要
  - Web アプリのバックエンドから、各種通知 (審査完了通知・予約リマインド通知等) を、対象ユーザーに LINE メッセージとして送信するための API。
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
- プロトコル・データ形式
  - LINE メッセージの送信
    - ◇ ライブラリ 【SL012】 LINE Messaging API SDK (@line/bot-sdk) の実装に依存
      - <https://github.com/line/line-bot-sdk-nodejs>
    - ◇ 関連ドキュメント
      - <https://developers.line.biz/ja/reference/messaging-api/>

#### 【IF007】 メール送信 API (SendGrid API)

- 本インターフェースの概要
  - Web アプリのバックエンドから、各種通知 (審査完了通知・予約リマインド通知等) を対象ユーザー

ーにメールとして送信するための API。

- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
- プロトコル・データ形式
  - メール送信
    - ◇ ライブラリ 【SL013】 SendGrid Mail SDK (@sendgrid/mail) の実装に依存
      - <https://github.com/sendgrid/sendgrid-nodejs>
    - ◇ 関連ドキュメント
      - [https://sendgrid.kke.co.jp/docs/API\\_Reference/Web\\_API\\_v3/index.html](https://sendgrid.kke.co.jp/docs/API_Reference/Web_API_v3/index.html)

#### 【IF008】 ユーザー認証 IF (Firebase Authentication with Identity Platform)

- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、Google 認証等のユーザー認証やメールアドレスの確認を行うための IF。
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
- プロトコル・データ形式
  - Google 認証
    - ◇ ライブラリ 【SL005】 Firebase (firebase) の実装に依存
      - <https://github.com/firebase/firebase-js-sdk>
    - ◇ 関連ドキュメント
      - <https://firebase.google.com/docs/auth/Web/google-signin?hl=ja>
  - LINE 認証
    - ◇ ライブラリ 【SL005】 Firebase (firebase) の実装に依存
      - <https://github.com/firebase/firebase-js-sdk>
    - ◇ 関連ドキュメント
      - <https://firebase.google.com/docs/auth/Web/openid-connect?hl=ja>
  - パスワード認証
    - ◇ ライブラリ 【SL005】 Firebase (firebase) の実装に依存
      - <https://github.com/firebase/firebase-js-sdk>
    - ◇ 関連ドキュメント
      - <https://firebase.google.com/docs/auth/Web/password-auth?hl=ja>
  - メールアドレスの確認
    - ◇ ライブラリ 【SL005】 Firebase (firebase) の実装に依存
      - <https://github.com/firebase/firebase-js-sdk>
    - ◇ 関連ドキュメント
      - <https://firebase.google.com/docs/auth/Web/manage-users?hl=ja>
  - 電話番号の確認
    - ◇ ライブラリ 【SL005】 Firebase (firebase) の実装に依存
      - <https://github.com/firebase/firebase-js-sdk>
    - ◇ 関連ドキュメント

- <https://firebase.google.com/docs/auth/Web/phone-auth?hl=ja>

【IF009】各種データのファイル出力 IF

- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、ユーザー情報や予約情報等の各種データを、ファイルとして出力するための IF。
- 本インターフェースを利用する機能
  - 【FN001】各種レコード連携
- ファイルの形式
  - CSV

【IF010】ページ取得 IF (Firebase Hosting)

- 本インターフェースの概要
  - Web ブラウザーにおいて、Web アプリのフロントエンドをホスティングサーバーから取得する際の IF。
- 本インターフェースを利用する機能
  - 【FN004】ユーザー登録
  - 【FN007】予約作成
  - 【FN008】予約した情報確認
  - 【FN009】予約変更・キャンセル
  - 【FN012】利用開始・終了報告
  - 【FN015】レビュー作成・公開
  - 【FN016】車両提供者登録・更新
  - 【FN017】車両登録・更新
  - 【FN018】公開コントロール
  - 【FN019】ユーザー拒否
  - 【FN020】車両提供者都合キャンセル
  - 【FN021】予約関連情報確認
  - 【FN022】プロフィール確認
  - 【FN023】各種ステータス変更
  - 【FN024】売上上限コントロール
  - 【FN025】ユーザー/車両提供者/車両審査・承認
- プロトコル・データ形式
  - プロトコル
    - ◇ HTTPS
  - メソッド
    - ◇ GET
  - レスポンスデータ形式
    - ◇ HTML・CSS・JavaScript
  - 文字コード

◇ UTF-8

【IF011】 Cloud Firestore と BigQuery の連携 IF

- 本インターフェースの概要
  - Cloud Firestore に書き込まれたデータを BigQuery にストリーミングする際の IF。
- 本インターフェースを利用する機能
  - 【FN027】 Web アプリデータ分析
- プロトコル・データ形式
  - Firebase の拡張機能の実装に依存
    - ◇ <https://extensions.dev/extensions/firebase/firestore-bigquery-export>

【IF012】 LINE ユーザー管理 API (Poster API)

- 本インターフェースの概要
  - Web アプリのバックエンドにおいて、LINE ユーザーに対してタグ等を付与して管理する API。
- 本インターフェースを利用する機能
  - 【FN002】 各種通知送信
- プロトコル・データ形式
  - ユーザー情報の取得
    - ◇ <https://poster.ooo/poster-dev-doc/get-user-info/>
  - タグの追加
    - ◇ <https://poster.ooo/poster-dev-doc/tag-add-api/>
  - タグの削除
    - ◇ <https://poster.ooo/poster-dev-doc/tag-remove-api/>

【IF013】 ウェブ解析ツール連携 IF (Google Analytics 4)

- 本インターフェースの概要
  - Web アプリのフロントエンドから、Google Analytics 4 にメトリクスを送信する IF。
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
  - 【FN027】 Web アプリデータ分析
- プロトコル・データ形式
  - ライブラリ 【SL005】 Firebase (firebase) の実装に依存
    - ◇ <https://github.com/firebase/firebase-js-sdk>
  - 関連ドキュメント
    - ◇ <https://firebase.google.com/docs/analytics?hl=ja>
- 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

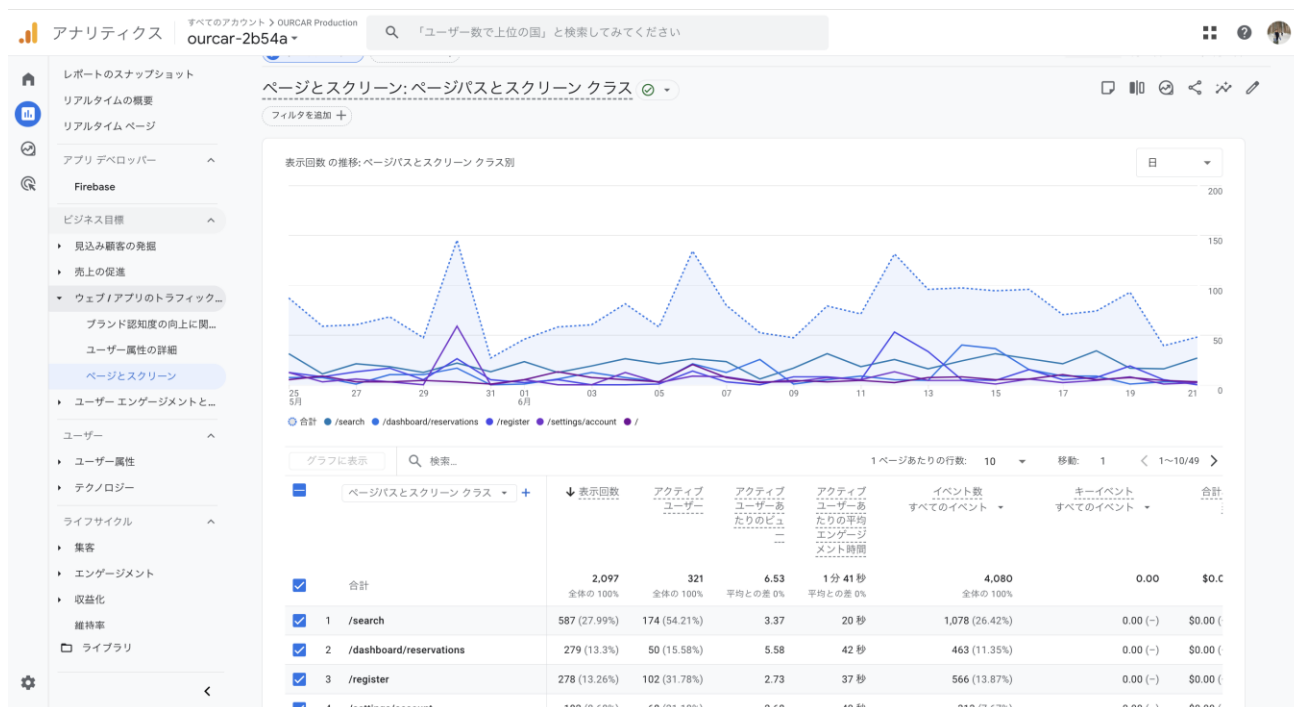


図 2-52 GoogleAnalytics 管理画面

表 2-10 GoogleAnalytics データ

No.	セッションの参照元/メディア	データ種
1	アクティブユーザー	数値
2	セッション	数値
3	キーイベント	数値
4	エンゲージのあったセッション数	数値
5	セッション当たりの平均エンゲージメント時間	数値
6	エンゲージメントのあったセッション数 (1 アクティブユーザーあたり)	数値
7	セッション当たりのイベント数	数値
8	エンゲージメント率	数値
9	イベント数	数値
10	合計収益	数値
11	セッションキーイベントレート	数値

【IF014】 行動分析ツール連携 IF (Microsoft Clarity)

- 本インターフェースの概要
  - Web アプリのフロントエンドから、Microsoft Clarity にメトリクスを送信する IF。
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
  - 【FN027】 Web アプリデータ分析
- プロトコル・データ形式
  - トラッキングコードの実装に依存
    - ◇ <https://learn.microsoft.com/en-us/clarity/setup-and-installation/clarity-setup>

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- 本インターフェースの概要
  - Clarity を通して取得するユーザーの Web アプリ内での行動のデータ。画面録画やヒートマップで数値及び行動を計測する
- 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



図 2-53 Clarity 管理画面

### 【IF015】 Slack メッセージ投稿 IF (Slack Incoming Webhooks)

- 本インターフェースの概要
  - Web アプリのバックエンドから、Slack チャンネルにメッセージを投稿する IF。
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
  - 【FN002】 各種通知送信
- プロトコル・データ形式
  - ライブラリ 【SL014】 Slack Incoming Webhooks SDK (@slack/Webhook) の実装に依存
    - ◇ <https://github.com/slackapi/node-slack-sdk/tree/main/packages/Webhook>

### 【IF016】 Cloud Storage イベントトリガー

- 本インターフェースの概要
  - Cloud Firestore のデータベースイベントをトリガーとして自動的に実行されるサーバーレス関数。これにより、リアルタイムにデータベースの変更を検知し、カスタムロジックを実行する。
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
  - 【FN004】 ユーザー登録
  - 【FN005】 プロフィール作成・変更
  - 【FN007】 予約作成

- 【FN008】 予約した情報確認
- 【FN009】 予約変更・キャンセル
- 【FN012】 利用開始・終了報告
- 【FN015】 レビュー作成・公開
- 【FN016】 車両提供者登録・更新
- 【FN017】 車両登録・更新
- 【FN018】 公開コントロール
- 【FN019】 ユーザー拒否
- 【FN020】 車両提供者都合キャンセル
- 【FN021】 予約関連情報確認
- 【FN022】 プロフィール確認
- 【FN023】 各種ステータス変更
- 【FN024】 売上上限コントロール
- 【FN025】 ユーザー/車両提供者/車両審査・承認
- 【FN026】 各種データ CSV エクスポート
- プロトコル・データ形式
  - Storage イベントトリガー
    - ◇ Cloud Storage で特定のイベント（例：新しいファイルがバケットに追加された）が発生すると、そのイベント情報が Functions に通知され、関数が実行される。
    - ◇ Firestore の場合と同様に、この内部的なプロトコルは Google Cloud のインフラによって管理されており、開発者が直接操作することはない。
    - ◇ 関数は、HTTP リクエストを受け付けるようにデプロイされるのではなく、Storage イベントをリッスンするようにデプロイされる。
  - イベントデータ（JSON）：
    - ◇ Cloud Functions に渡される Firestore イベントのデータは、JSON 形式で提供される。
    - ◇ この JSON オブジェクトには、イベントの種類（作成、更新、削除）、対象のドキュメントのパス、及び変更されたドキュメントのデータ（更新前のデータ oldValue と更新後のデータ newValue）が含まれる。
- 【IF017】 Cloud Function プログラムアクセス
- 本インターフェースの概要
  - Cloud Storage バケット内でファイルが作成、更新、又は削除されたときに自動的にトリガーされるサーバーレス関数。これにより、画像のリサイズ、ファイルのウイルススキャン、メタデータの抽出などの処理を自動化が可能。
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
  - 【FN004】 ユーザー登録
  - 【FN005】 プロフィール作成・変更

- 【FN007】 予約作成
- 【FN008】 予約した情報確認
- 【FN009】 予約変更・キャンセル
- 【FN012】 利用開始・終了報告
- 【FN015】 レビュー作成・公開
- 【FN016】 車両提供者登録・更新
- 【FN017】 車両登録・更新
- 【FN018】 公開コントロール
- 【FN019】 ユーザー拒否
- 【FN020】 車両提供者都合キャンセル
- 【FN021】 予約関連情報確認
- 【FN022】 プロフィール確認
- 【FN023】 各種ステータス変更
- 【FN024】 売上上限コントロール
- 【FN025】 ユーザー/車両提供者/車両審査・承認
- 【FN026】 各種データ CSV エクスポート
- プロトコル・データ形式
  - Storage イベントトリガー
    - ◇ Cloud Storage で特定のイベント（例：新しいファイルがバケットに追加された）が発生すると、そのイベント情報が Functions に通知され、関数が実行される。
    - ◇ Firestore の場合と同様に、この内部的なプロトコルは Google Cloud のインフラによって管理されており、開発者が直接操作することがある。
    - ◇ 関数は、HTTP リクエストを受け付けるようにデプロイされるのではなく、Storage イベントをリッスンするようにデプロイされる。
  - イベントデータ（JSON）
    - ◇ Cloud Functions に渡される Storage イベントのデータも、JSON 形式で提供される。
    - ◇ この JSON オブジェクトには、ファイル名、バケット名、ファイルサイズ、MIME タイプ、タイムスタンプなどのファイルのメタデータが含まれる。ファイルの中身自体は同オブジェクトに含まれる。
- 【IF018】 予約データの作成・取得 IF
- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、予約作成をする際や、予約情報の閲覧をする際のインターフェース
  - 予約作成時の契約対象具体特定のための表記公開
    - ◇ 予約が作成された時点で、車両提供者・共同使用者共に相互の本名・連絡先（電話番号）・住所を確認することができる。予約された当該車両に関してもサービスページから、車両提供者・共同使用者の双方から確認できる。（ユーザーデータ・車両提供者データから引用）
    - ◇ これによって、車両の提供者と共同使用者がお互いに契約相手を相互に特定できる

- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
  - 【FN007】 予約作成
  - 【FN008】 予約した情報確認
  - 【FN009】 予約変更・キャンセル
  - 【FN012】 利用開始・終了報告
  - 【FN020】 車両提供者都合キャンセル
  - 【FN021】 予約関連情報確認
- プロトコル・データ形式
  - ライブラリ【SL005】 Firebase (firebase) の実装に依存
    - ◇ <https://github.com/firebase/firebase-js-sdk>
- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-11 予約データ

No.	カラム名	概要	データ種
1	reservationId	予約 ID	テキスト
2	carId	車両 ID	テキスト
3	ownerId	車両提供者 ID	テキスト
4	userId	ユーザーID	テキスト
5	Version	データバージョン	テキスト
6	Type	契約形態	テキスト
7	State	利用ステータス	テキスト
8	datetimeFrom	予約開始時間	テキスト
9	datetimeTo	利用終了時間	テキスト
10	Charge	利用料金合計	テキスト
11	timeCharge	時間料金	テキスト
12	distanceCharge	距離料金	テキスト
13	couponInfo.couponId	クーポン ID	テキスト
14	protectionInfo.plan	プロテクトプラン	テキスト
15	sharedInsurance	カーシェア専用保険	テキスト
16	additionalDrivers	共同運転者	テキスト
17	startReport.odometer	利用開始時総走行距離メーター数値	テキスト
18	startReport.reportedAt	利用開始報告日時	テキスト
19	endReport.odometer	利用終了時総走行距離メーター数値	テキスト
20	endReport.refueling	給油有無	テキスト
21	endReport.refuelingPrice	給油金額	テキスト
22	endReport.reportedAt	利用終了報告日時	テキスト
23	paymentInfo.method	支払方法	テキスト
24	createdAt	作成日時	テキスト
25	updatedAt	更新日時	テキスト

表 2-12 利用終了時アンケート

No.	カラム名	本インターフェースの概要	データ種
1	reservationId	予約 ID	テキスト
2	carId	車両 ID	テキスト
3	ownerId	車両提供者 ID	テキスト
4	userId	ユーザーID	テキスト
5	recommendation	NPS	テキスト
6	Purpose	利用目的	テキスト
7	purposeOtherDetails	他利用目的詳細	テキスト
8	opinionsAndRequests	意見・要望	テキスト
9	createdAt	作成日時	テキスト
10	updatedAt	更新日時	テキスト

【IF019】 レビューデータの作成・取得 IF

- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、レビュー作成をする際や、レビュー情報の閲覧をする際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】 各種レコード連携
  - 【FN015】 レビュー作成・公開
- プロトコル・データ形式
  - ライブラリ【SL005】 Firebase (firebase) の実装に依存
    - ✧ <https://github.com/firebase/firebase-js-sdk>
- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-13 レビューデータ\_ユーザーから車両提供者へのレビュー

No.	カラム名	概要	データ種
1	-	作成したユーザーID	テキスト
2	-	評価された車両提供者 ID	テキスト
3	-	総合評価	テキスト
4	-	車内クリンリネス	テキスト
5	-	車の管理状態	テキスト
6	-	価格	テキスト
7	-	情報の正確さ/わかりやすさ	テキスト
8	-	コメント	テキスト
9	-	サービスへのコメント (非公開)	テキスト
10	-	作成日時	テキスト
11	-	更新日時	テキスト

表 2-14 レビューデータ\_車両提供者からユーザーへのレビュー

No.	カラム名	概要	データ種
1	-	作成した車両提供者 ID	テキスト
2	-	評価されたユーザーID	テキスト
3	-	総合評価	テキスト
4	-	車の使用マナー	テキスト
5	-	報告の正確さ	テキスト
6	-	コメント	テキスト
7	-	サービスへのコメント（非公開）	テキスト
8	-	作成日時	テキスト

【IF020】 車両データの作成・取得 IF

- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、車両登録をする際や、車両情報の閲覧をする際のインターフェース
  - 定期的に各種タスクを行うバッチサーバー。提供する当該車両を登録する際、車両提供者が車両管理方法についての方式を下記から選択する。
    - ◇ 車両提供者による一元管理（ユーザーにおける日常点検管理）
      - 使用前の日常点検管理とは冷却水量/エンジンオイル/タイヤの空気圧・取付状態・溝・損傷/バッテリー液量/ブレーキ液量/ウィンドウォッシャー液量残量・清浄噴霧/パーキングブレーキの引きしろの確認を指す
      - ◇ その他自由記載：記載どおりの管理がなされているか、車両提供者への定期的な確認を行う
  - 共同使用者は当該車両への初回利用時にその管理方法に関して合意し、共同使用契約を締結する。これによって、ユーザー・車両提供者間でのあらかじめの合意を形成することができる
  - ◇ 一元管理の場合は車両提供者が車両の管理を一元的に行う
- 本インターフェースを利用する機能
  - 【FN001】 各種レコード連携
  - 【FN017】 車両登録・更新
  - 【FN018】 公開コントロール
  - 【FN019】 ユーザー拒否
  - 【FN024】 売上上限コントロール
- プロトコル・データ形式
  - ライブラリ 【SL005】 Firebase (firebase) の実装に依存
    - ◇ <https://github.com/firebase/firebase-js-sdk>
- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-15 車両データ

No.	カラム名	概要	データ種
1	carId	車両 ID	テキスト
2	ownerId	車両提供者 ID	テキスト
3	Type	契約形態	テキスト
4	State	公開ステータス	テキスト
5	Label	ラベル	テキスト
6	manufacturer	製造メーカー	テキスト
7	Model	車両モデル	テキスト
8	Use	自家用事業用の別	テキスト
9	Class	車両クラス	テキスト
10	bodyType	車両本体の型	テキスト
11	ridingCapacity	乗車可能人数	テキスト
12	loadingCapacity	積載可能重量	テキスト
13	grossWeight	総重量	テキスト
14	transmission	AT/MT	テキスト
15	Handle	運転席の場所区別（左右）	テキスト
16	Gasoline	使用燃料	テキスト
17	driveSystem	4WD/FF/FR	テキスト
18	fuelEfficiency	燃費	テキスト
19	Pets	ペット利用可否	テキスト
20	Smoking	喫煙可否	テキスト
21	rearNumberPlateDepth	後部リアナンバーの高さ	テキスト
22	createdAt	作成日時	テキスト
23	updatedAt	更新日時	テキスト
24	deniedUser	拒否したユーザーID	テキスト
25	initalCostCar	車両本体購入時費用	数値
26	initalCostOthers	その他車両本体取得費用	数値
27	runningCostEstimated	想定年間維持費用	数値
28	runningCostActual	年間維持費用実費	数値
29	yearlyDrivingDistance	年間想定走行距離	数値
30	actualDrivingDistance	走行距離実績	数値
31	totalDrivingDistanceCurrent	現状総走行距離	数値
32	odometer	総走行距離メーターの画像	画像
33	registeredDrivingDistance	登録時の総走行距離	数値
34	totalSharingFee	累計受取共同使用料	数値
35	limitSharingFee	上限受取共同使用料	数値
36	dateOfPurchase	購入年月	日付
37	carMaintenanceAgreement	車両管理方法の合意	テキスト

【IF021】車両提供者データの作成・取得 IF

- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、車両提供者登録をする際や、車両提供者情報の閲覧をする際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】各種レコード連携
  - 【FN016】車両提供者登録・更新
  - 【FN022】プロフィール確認
- プロトコル・データ形式
  - ライブラリ【SL005】Firebase (firebase) の実装に依存
    - ◇ <https://github.com/firebase/firebase-js-sdk>
- 本インターフェースを利用してやり取りを行うデータの詳細

表 2-16 車両提供者データ

No.	カラム名	概要	データ種
1	ownerId	車両提供者 ID	テキスト
2	type	法人個人区分	テキスト
3	state	承認ステータス	テキスト
4	corporateNumber	法人番号	テキスト
5	name	事業者名	テキスト
6	email	メールアドレス	テキスト
7	phoneNumber	電話番号	テキスト
8	postCode	郵便番号	テキスト
9	address	住所	テキスト
10	bankName	金融機関名	テキスト
11	bankBranchName	支店名	テキスト
12	bankAccountType	口座種別	テキスト
13	bankAccountNumber	口座番号	テキスト
14	bankAccountHolder	口座名義	テキスト
15	createdAt	作成日時	テキスト
16	updatedAt	更新日時	テキスト

【IF022】各種データ承認 IF

- 本インターフェースの概要
  - Web アプリのフロントエンドにおいて、レビュー登録をする際や、レビュー情報の閲覧をする際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】 各種レコード連携
- 【FN023】 各種ステータス変更
- 【FN025】 ユーザー/車両提供者/車両審査・承認
- プロトコル・データ形式
  - ライブラリ【SL005】 Firebase (firebase) の実装に依存
    - ◇ <https://github.com/firebase/firebase-js-sdk>
- 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
  - IF001・IF018・IF019・IF020・IF021 参照

## 2-5. ユーザーインターフェース (UI)

### 2-5-1. 画面遷移図

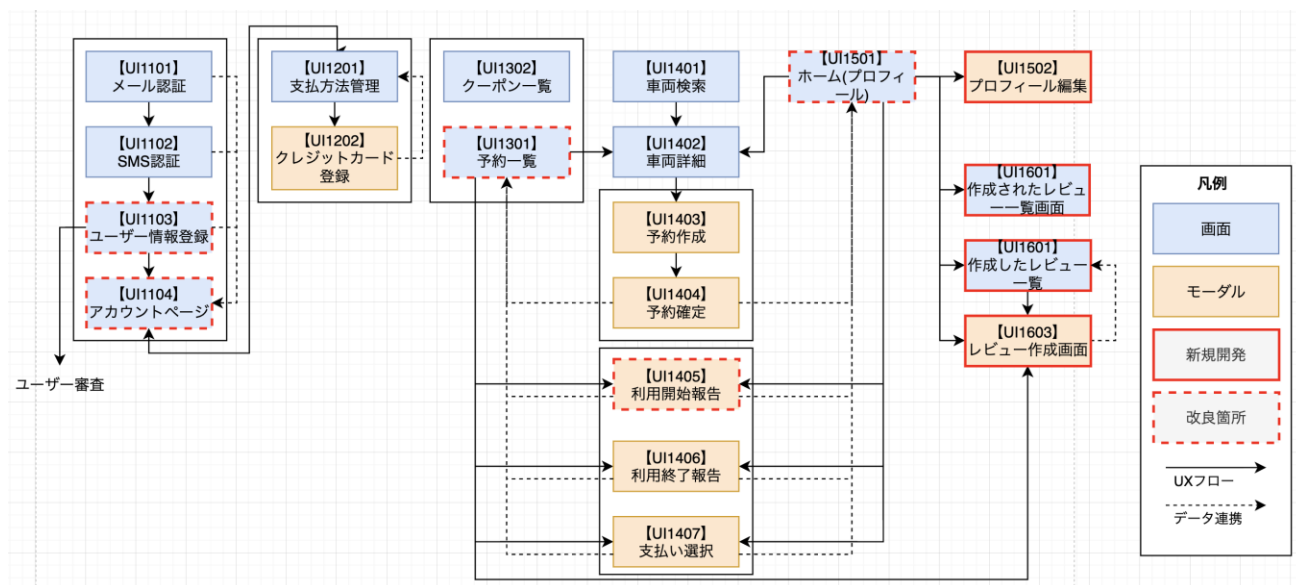


図 2-54 【HW001・HW002】 PC・スマートフォン\_ユーザー画面遷移図

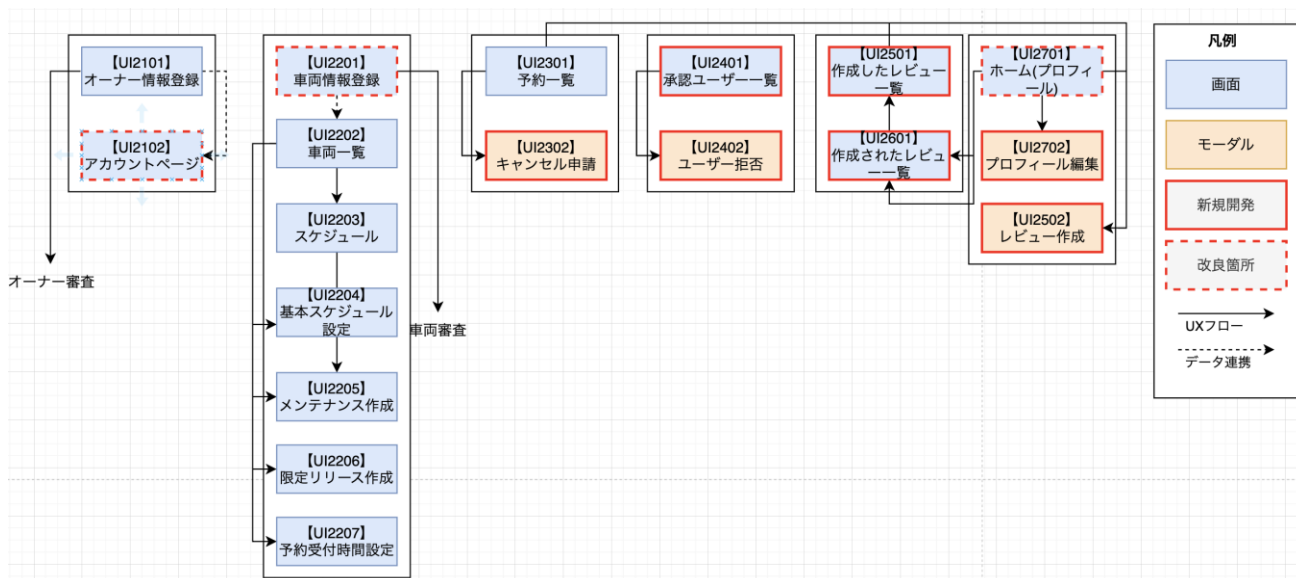


図 2-55 【HW001・HW002】 PC・スマートフォン\_車両提供者画面遷移図

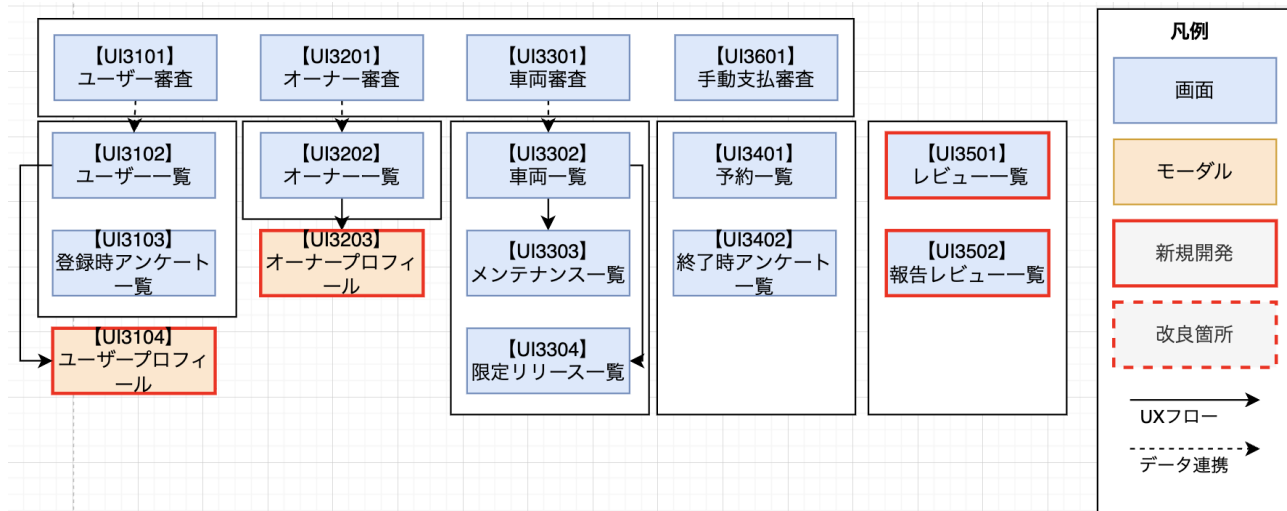


図 2-56 【HW001・HW002】PC・スマートフォン\_サービス提供事業者管理画面遷移図

## 2-5-2. ユーザーインターフェース一覧

表 2-17 【HW001・HW002】 PC・スマートフォン画面一覧\_ユーザー画面

※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)
UI1100	アカウント認証画面	● 既存の LINE・Google・パスワード方式を選択し、アカウントとひも付け、ログイン時その方法を選択することでより簡易にサービスにアクセスできる。	FN003
UI1101	メール認証画面	● ユーザー登録の本人確認としてメールを通した認証メールを送信・確認する。	FN003、FN004
UI1102	SMS 認証画面	● ユーザー登録の本人確認として SMS を通した認証を確認する。	FN003、FN004
UI1103	ユーザー情報登録画面	● 名前や免許証画像等の個人情報を登録する。	FN003、FN004
UI1104	アカウントページ画面	● 登録した個人情報を確認・更新できる。	FN004
UI1201	支払方法管理画面	● 利用に対する支払方法を登録できる。(主にクレジットカード)	FN013
UI1202	クレジットカード登録画面	● クレジットカードを登録できる。	FN013
UI1301	予約一覧画面	● ユーザー自身が作成した予約を一覧で確認できる。	FN008、FN009
UI1302	クーポン一覧	● ユーザーが保持するクーポンが一覧で確認できる。	FN008
UI1401	車両検索画面	● 地図上から使用したい車両を検索できる。 ● 日時等の条件でフィルタリング可。	FN006
UI1402	車両詳細画面	● 車両の料金や設備等詳細情報を確認できる。	FN007
UI1403	予約作成画面	● 日時やプロテクトプラン等を選択し、利用予約が作成できる。	FN007
UI1404	予約確定画面	● 予約内容を確認し、確定する。	FN007
UI1405	利用開始報告画面	● 日常点検及び利用開始時の車体写真やメーター等の入力を行う。	FN010、FN011、FN012
UI1406	利用終了報告画面	● 日常点検及び終了時時の車体写真やメーター等の入力を行う。	FN011、FN012
UI1407	支払い画面	● 支払方法を選択し料金を支払う。	FN014
UI1501	ホーム (プロフィール)	● 自身のプロフィール情報及びやることリストや、直近の予約などを確認できる。	FN005、FN015、FN022

UI1502	プロフィール編集画面	● 車両提供者及びユーザーが閲覧できるユーザー自身のプロフィール情報を編集できる。	FN005、N022
UI1601	作成したレビュー一覧画面	● ユーザー自身が車両提供者に作成したレビューを確認できる。	FN015、FN022
UI1602	作成されたレビュー一覧	● 車両提供者からユーザーに作成されたレビューを確認できる。	FN015、FN022
UI1603	レビュー作成画面	● 実際に利用した車両・車両提供者に対してレビューを作成できる。	FN015

表 2-18 【HW001・HW002】 PC・スマートフォン画面一覧\_車両提供者画面

ID	画面名	説明	画面を表示した機能 (ID)
UI2101	車両提供者登録画面	● 車両提供者としての口座や連絡先等の個人情報を登録できる。	FN016
UI2102	アカウントページ画面	● 登録した個人情報を確認・更新できる。	FN016
UI2201	車両情報登録画面	● 提供する車両の設備や車検を基にした車両情報等を登録する。	FN017
UI2202	車両一覧画面	● 登録した車両について詳細も含めて一覧で確認できる。	FN017
UI2203	スケジュール画面	● 車両ごとの利用状況やメンテナンス等のスケジュール状況をカレンダー形式で把握できる。	FN018
UI2204	基本スケジュール設定画面	● 貸出可否のスケジュールを事前に設定できる。	FN018
UI2205	メンテナンス作成画面	● 基本スケジュール以外の一時的な利用不可ブロックを作成できる。	FN018
UI2206	限定リリース作成画面	● 基本スケジュール以外の一時的な利用可能ブロックを作成できる。	FN018
UI2207	予約受付時間設定画面	● 利用予約の受付時間を設定できる（開始時刻から何時間前まで受付可能等）。	FN018
UI2301	予約一覧画面	● 作成された予約の一覧を確認できる。	FN021
UI2302	キャンセル申請画面	● 車両提供者都合でのキャンセルが可能。理由と共にユーザーに伝えられる。	FN020
UI2401	承認ユーザー一覧画面	● 予約を入れたユーザーの一覧を確認できる。	FN019
UI2402	ユーザー拒否画面	● 予約を作成し自動承認されたユーザーについて後から共同使用の予約を拒否することができる。	FN019
UI2501	作成したレビュー一覧	● 自身がユーザーに対して作成したレビュー	FN015

		を一覧で確認できる。	
UI2502	レビュー作成画面	● ユーザーへのレビューを作成できる。	FN015
UI2601	作成されたレビュー一覧	● ユーザーから自身へのレビューを一覧で確認できる。	FN015
UI2701	ホーム（プロフィール）	● 車両提供者が最初に確認するページ。	FN005
UI2702	プロフィール編集画面	● 車両提供者及びユーザーが閲覧できる車両提供者自身のプロフィール情報を編集できる。	FN005

表 2-19 【HW001・HW002】PC・スマートフォン画面一覧\_サービス提供管理画面

ID	画面名	説明	画面を表示した機能 (ID)
UI3101	ユーザー審査画面	● ユーザー登録を申請した個人情報の正当性を審査する。	FN025
UI3102	ユーザー一覧画面	● 登録したユーザーについて詳細を含めて一覧で確認できる。	FN023
UI3103	登録時アンケート一覧画面	● 登録した際のサービスに対してのアンケートを確認できる	FN027
UI3104	ユーザープロフィール	● ユーザーのプロフィールを確認できる。	FN022
UI3201	車両提供者審査画面	● 車両提供者登録を申請した個人情報の正当性を審査する。	FN025
UI3202	車両提供者一覧画面	● 登録した車両提供者について詳細を含めて一覧で確認できる。	FN023
UI3203	車両提供者プロフィール	● 車両提供者のプロフィールを確認できる。	FN022
UI3301	車両審査画面	● 車両登録申請した車両情報の正当性を確認し審査する。	FN025
UI3302	車両一覧画面	● 登録した車両について詳細を含めて一覧で確認できる。	FN023
UI3303	メンテナンス一覧画面	● 基本スケジュール以外の一時的な利用不可ブロックを作成できる。	FN023
UI3304	限定リリース一覧画面	● 基本スケジュール以外の一時的な利用可能ブロックを作成できる。	FN023
UI3401	予約一覧	● 作成された予約の一覧を確認できる。	FN023
UI3402	終了時アンケート一覧	● ユーザーの利用終了時のアンケートを一覧で確認できる。	FN027
UI3501	レビュー一覧	● ユーザー・車両提供者が作成したレビューを一覧で確認できる。	FN023
UI3502	報告レビュー一覧	● ポリシー違反レビューの報告があったも	FN023

共同使用契約に基づくカーシェアリングシステム システム設計書

		の一覧で確認できる。	
UI3601	手動支払審査	● 電子決済による支払状態を確認し、承認する。	FN023

### 2-5-3. ユーザーインターフェースの詳細

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

#### 【UI1100】アカウント認証画面

- PC 画面

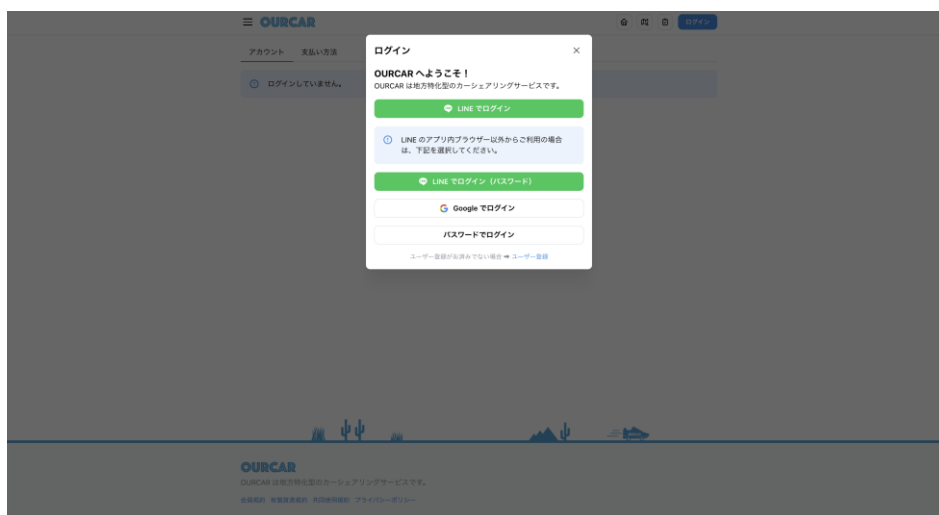


図 2-57 アカウント認証画面

- スマートフォン画面



図 2-58 アカウント認証画面

- 本画面の概要

➤ ユーザーアカウント作成時、ログイン時に認証に使用するサービスを選択できる

- 本画面から利用する機能

【FN003】ユーザー認証

【UI1101】メール認証画面

- PC 画面

OURCAR へようこそ！  
まずはメールアドレスとパスワードの設定を完了させてください。

必須 メールアドレス

必須 パスワード  
10文字以上で設定してください。半角英小文字・半角数字・記号をそれぞれ1文字以上含めてください。

必須 パスワード (確認)

確定

OURCAR  
OURCAR は地方特化型のカーシェアリングサービスです。  
会員規約 有償貸借規約 共同使用規約 プライバシーポリシー

図 2-59 メール認証画面

- スマートフォン画面

OURCAR へようこそ！  
まずはメールアドレスとパスワードの設定を完了させてください。

必須 メールアドレス

必須 パスワード  
10文字以上で設定してください。半角英小文字・半角数字・記号をそれぞれ1文字以上含めてください。

必須 パスワード (確認)

確定

OURCAR  
OURCAR は地方特化型のカーシェアリングサービスです。  
利用規約 会員規約 プライバシーポリシー

ホーム・予約一覧 地図から探す ダッシュボード アカウント

図 2-60 メール認証画面

- 本画面の概要
  - ユーザーアカウント登録に使用するメールアドレスの認証を行う
- 本画面から利用する機能
  - 【FN003】ユーザー認証
  - 【FN004】ユーザー登録

【UI1102】SMS 認証画面

- PC 画面



図 2-61 SMS 認証画面

- スマートフォン画面



図 2-62 SMS 認証画面

- 本画面の概要
  - LINE・Google・パスワード方式を選択し、アカウントとひも付け、ログイン時その方法を選択することでより簡易にサービスにアクセスできる。
- 本画面から利用する機能
  - 【FN003】ユーザー認証
  - 【FN004】ユーザー登録

【UI1103】ユーザー情報登録画面 <既存改修>

- PC 画面

図 2-63 ユーザー情報登録画面

- スマートフォン画面

図 2-64 ユーザー情報登録画面

- 本画面の概要
  - ユーザーアカウントにひも付く氏名・住所等の個人情報を入力する
- 本画面から利用する機能
  - 【FN003】ユーザー認証
  - 【FN004】ユーザー登録

【UI1104】アカウントページ画面 <既存改修>

● PC 画面



図 2-65 アカウントページ画面

● スマートフォン画面

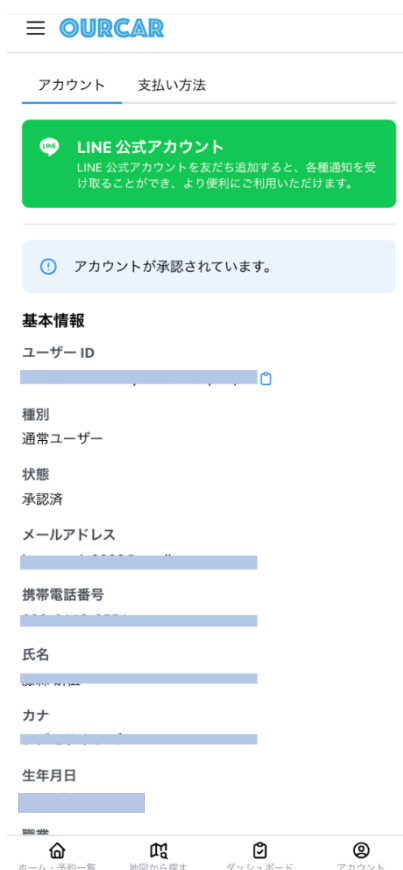


図 2-66 アカウントページ画面

- 本画面の概要
  - 登録した個人情報及び審査状況が確認できる
- 本画面から利用する機能
  - 【FN004】ユーザー登録

## 【UI1201】支払方法管理画面

- PC 画面

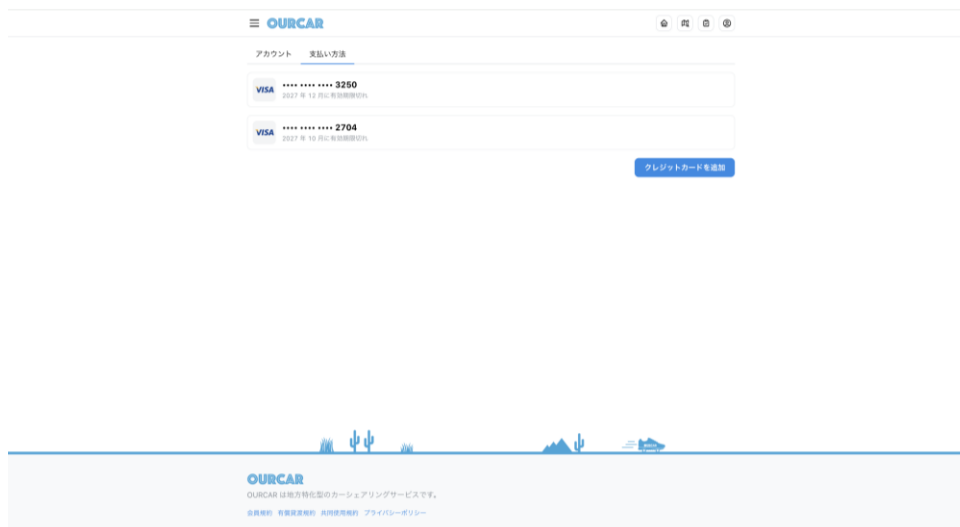


図 2-67 支払方法管理画面

- スマートフォン画面



図 2-68 支払方法管理画面

- 本画面の概要
  - クレジットカード等過去に登録しているカード等の情報を確認できる
- 本画面から利用する機能

➤ 【FN013】支払情報登録

【UI1202】クレジットカード登録画面

● PC 画面



図 2-69 クレジットカード登録画面

● スマートフォン画面



図 2-70 クレジットカード登録画面

● 本画面の概要

- 支払に使用するクレジットカードを事前に登録できる

● 本画面から利用する機能

- 【FN013】支払情報登録

【UI1301】予約一覧画面 <既存改修>

● PC画面

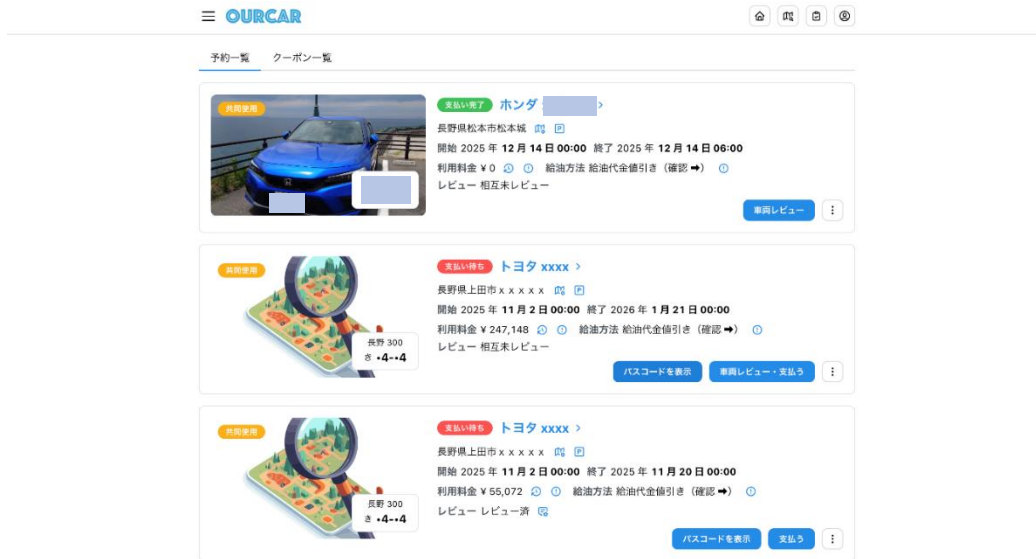


図 2-71 予約一覧画面

● スマートフォン画面



図 2-72 予約一覧画面

- 本画面の概要
  - ユーザーが作成した予約を一覧形式で確認できる
- 本画面から利用する機能
  - 【FN008】予約した情報確認
  - 【FN009】予約変更・キャンセル

【UI1302】クーポン一覧

● PC 画面

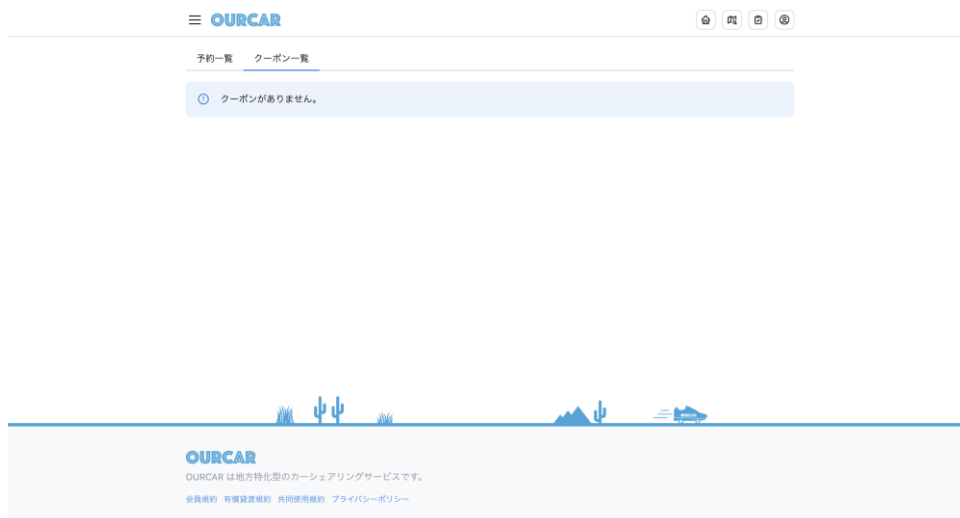


図 2-73 クーポン一覧

● スマートフォン画面



図 2-74 クーポン一覧

- 本画面の概要
  - ユーザーに付与されたクーポンを一覧形式で確認できる
- 本画面から利用する機能
  - 【FN008】予約した情報確認

【UI1401】車両検索画面

- PC 画面

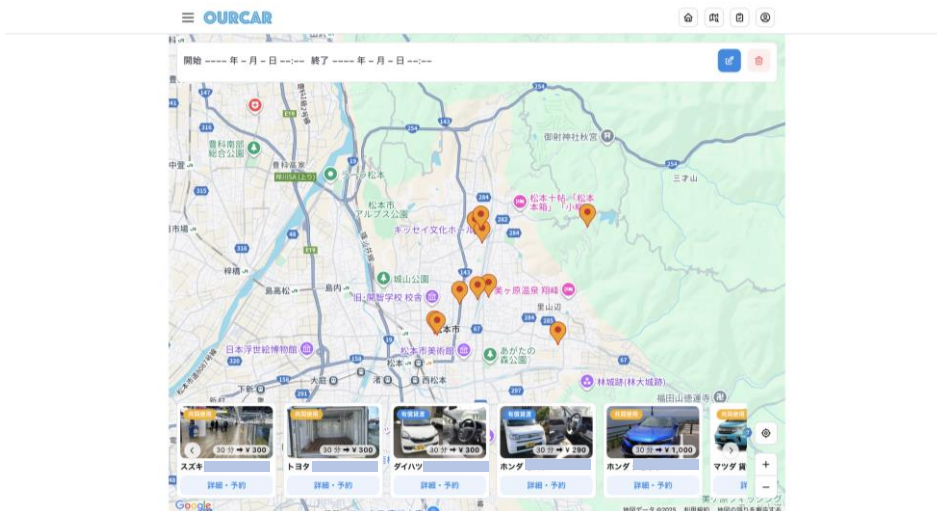


図 2-75 車両検索画面

- スマートフォン画面

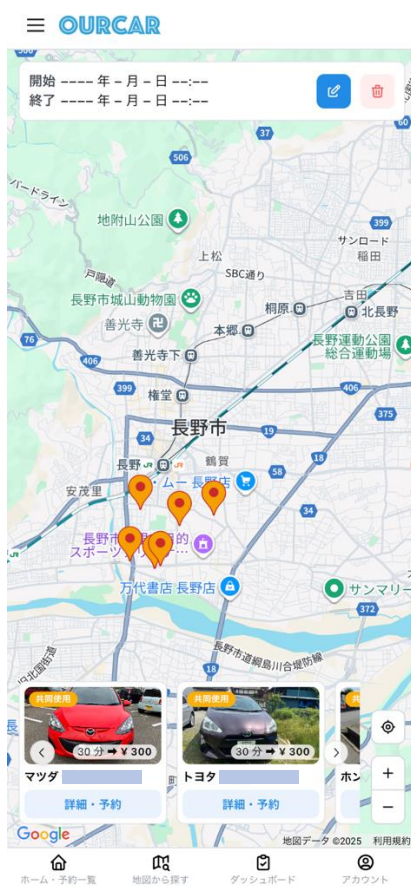


図 2-76 両検索画面

- 本画面の概要
  - 登録車両を地図から確認でき、時間等によってフィルター表示することができる
- 本画面から利用する機能
  - 【FN006】車両の地図表示・検索

【UI1402】車両詳細画面

● PC 画面



図 2-77 車両詳細画面

● スマートフォン画面



図 2-78 車両詳細画面

- 本画面の概要
  - 選択した車両の詳細条件やスケジュールを確認できる
- 本画面から利用する機能
  - 【FN007】予約作成

### 【UI1403】 予約作成画面

- PC 画面

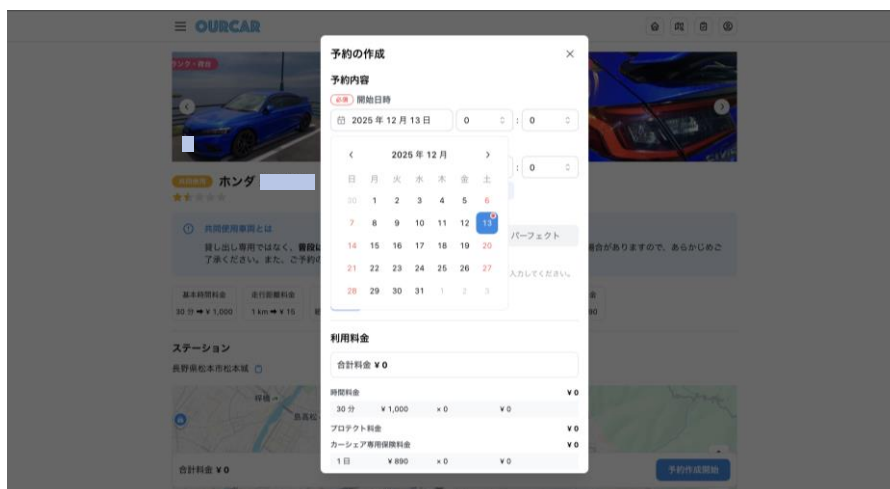


図 2-79 予約作成画面

- スマートフォン画面



図 2-80 予約作成画面

- 本画面の概要
  - 対象の車両の時間を指定して予約を作成、確定させる

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面から利用する機能
  - 【FN007】予約作成

### 【UI1404】予約確定画面

- PC 画面

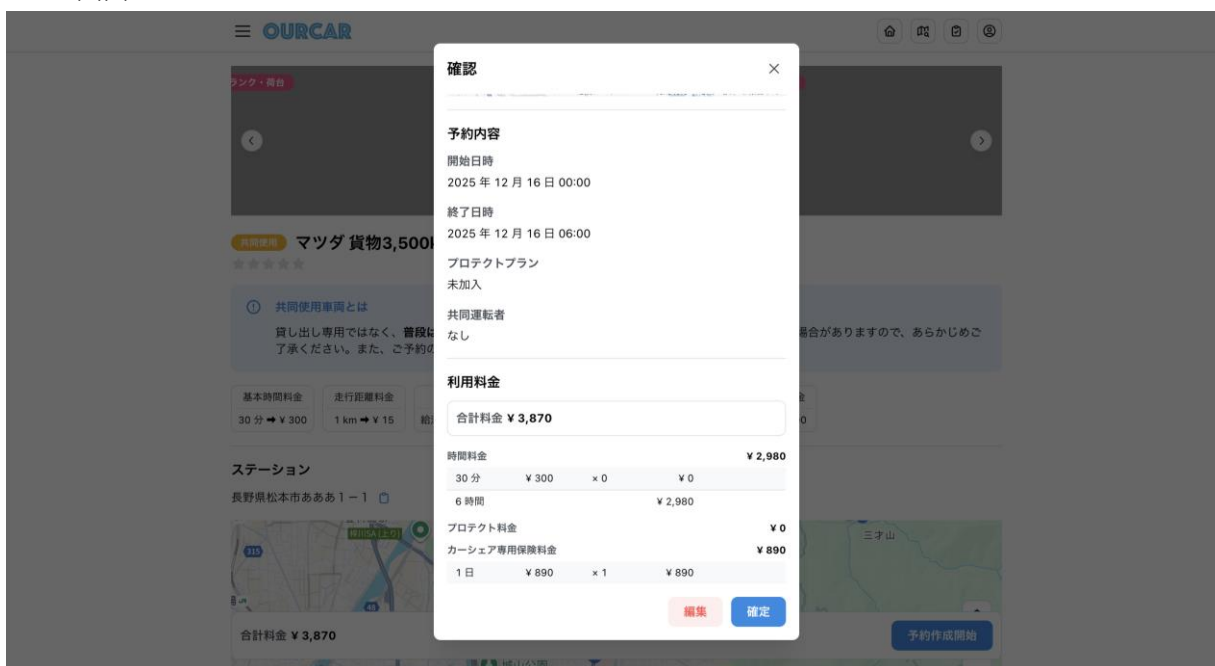


図 2-81 予約確定画面

- スマートフォン画面



図 2-82 予約確定画面

- 本画面の概要
  - 作成した予約を確定させる
- 本画面から利用する機能
  - 【FN007】予約作成

【UI1405】利用開始報告画面<既存改修>

- PC画面

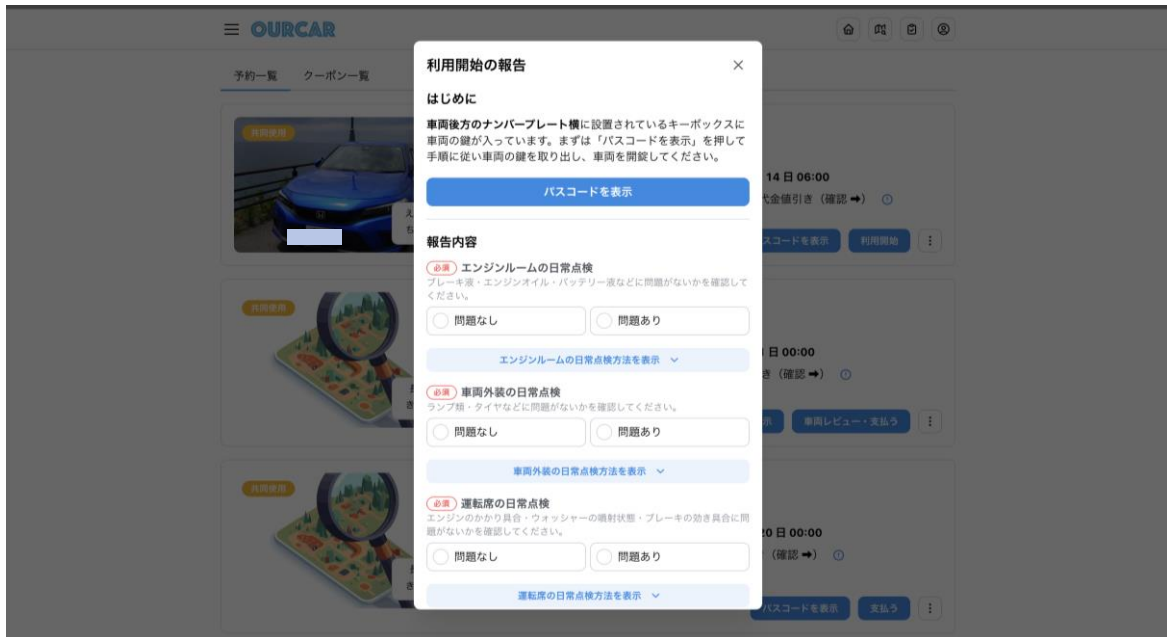
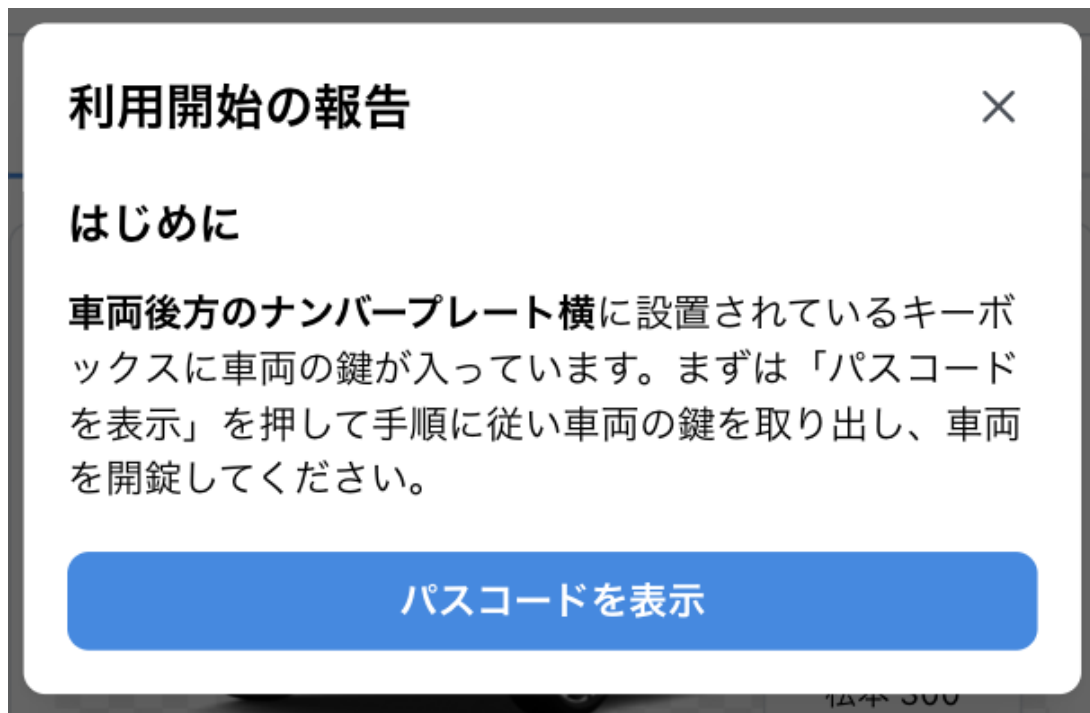


図 2-83 利用開始報告画面

- スマートフォン画面



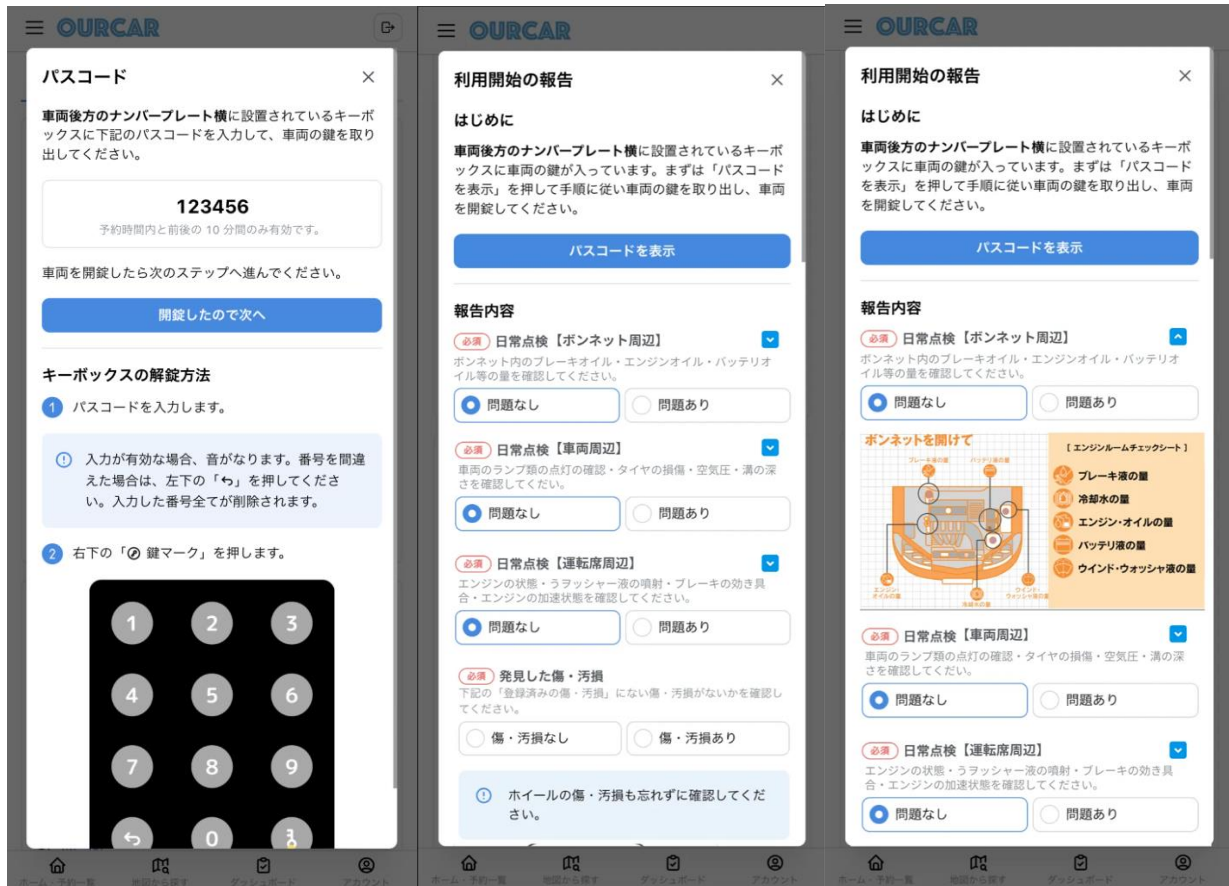


図 2-84 利用開始報告画面

- 本画面の概要
  - 車両の利用開始時に車両状態の確認、及びスマートキーボックスのパスコード閲覧・解錠、総走行距離メーターの数値報告を行う
- 本画面から利用する機能
  - 【FN010】パスコード発行
  - 【FN011】キーボックス解錠
  - 【FN012】利用開始・終了報告

【UI1406】利用終了報告画面

● PC 画面



図 2-85 利用終了報告画面

● スマートフォン画面

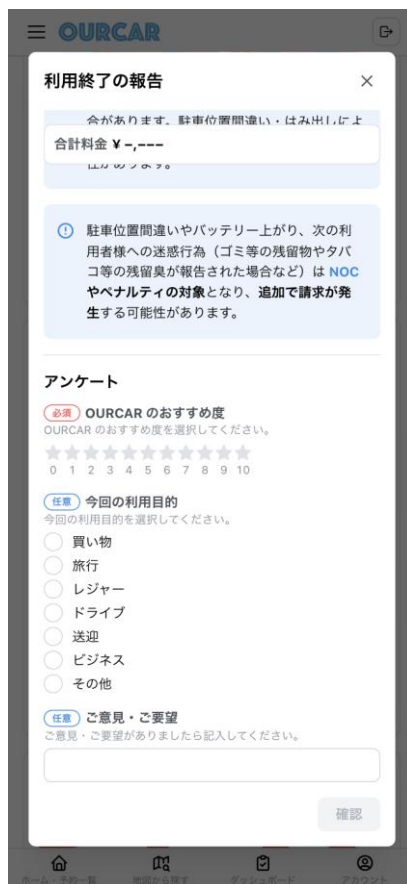


図 2-86 利用終了報告画面

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面の概要
    - 車両を駐車後、車両の状態を写真と共に報告する
  - 本画面から利用する機能
    - 【FN011】キーボックス解錠
    - 【FN012】利用開始・終了報告
- 【UI1407】支払い画面
- PC 画面

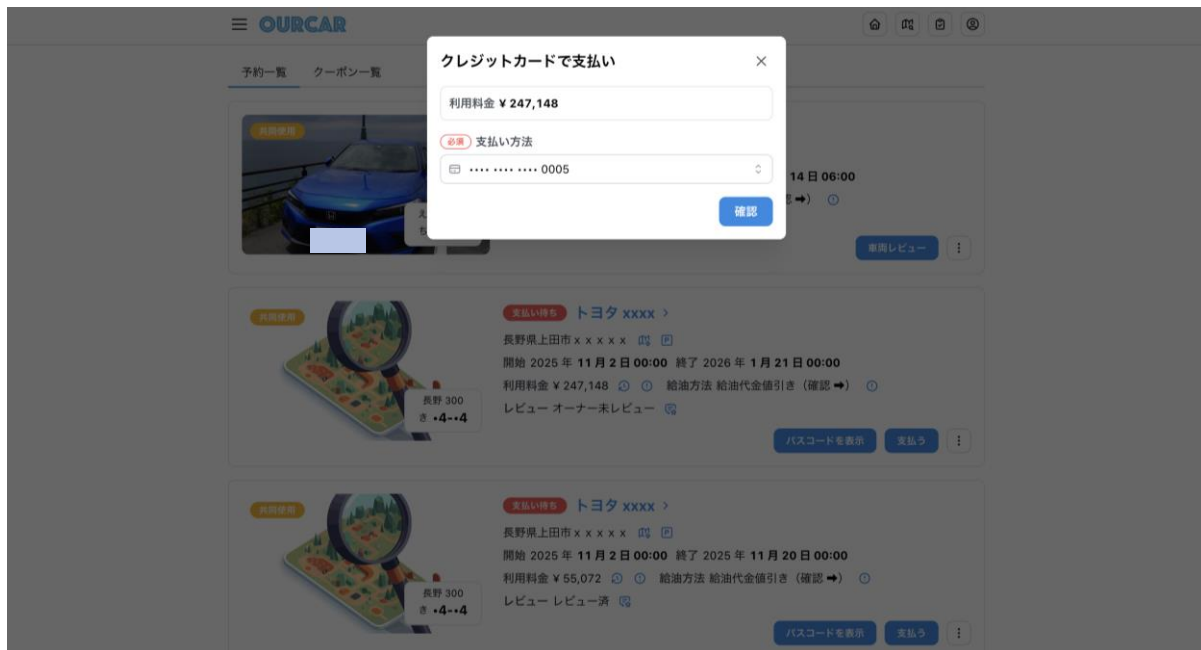


図 2-87 支払い画面

- スマートフォン画面

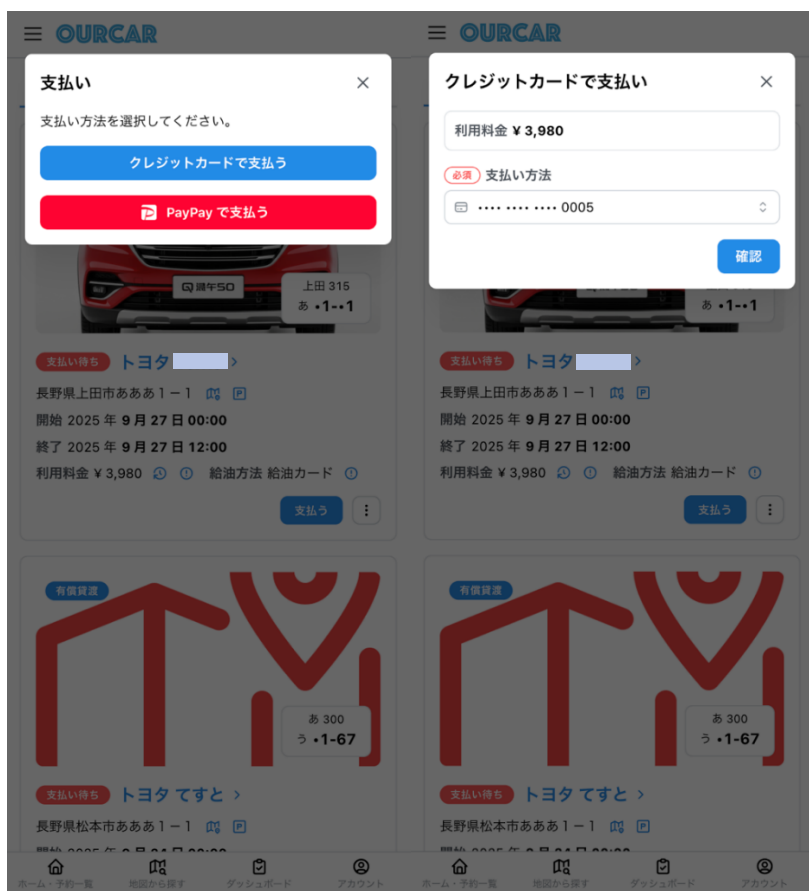


図 2-88 支払い画面

- 本画面の概要
  - 支払方法をクレジットカード又は電子決済で選択する
- 本画面から利用する機能
  - 【FN014】クレジットカード決済

【UI1501】ホーム画面 <既存改修>

● PC 画面



図 2-89 ホーム画面

● スマートフォン画面



図 2-90 ホーム画面

- 本画面の概要
  - Web アプリのホーム画面として機能する
- 本画面から利用する機能

- 【FN015】 レビュー作成・公開
- 【FN005】 プロフィール作成・変更
- 【FN022】 プロフィール確認

【UI1502】 プロフィール編集画面 <新規開発>

● PC画面



図 2-91 プロフィール編集画面

● スマートフォン画面



図 2-92 プロフィール編集画面

- 本画面の概要
  - ユーザープロフィールを編集・更新することができる
- 本画面から利用する機能

- 【FN005】プロフィール作成・変更
- 【FN022】プロフィール確認

【UI1601】作成したレビュー一覧画面 <新規開発>

● PC画面



図 2-93 作成したレビュー一覧画面

● スマートフォン画面



図 2-94 作成したレビュー一覧画面

- 本画面の概要
  - 自分が過去に作成したレビューを一覧形式で確認できる
- 本画面から利用する機能
  - 【FN015】 レビュー作成・公開
  - 【FN022】 プロフィール確認

【UI1602】 作成されたレビュー一覧 <新規開発>

- PC 画面

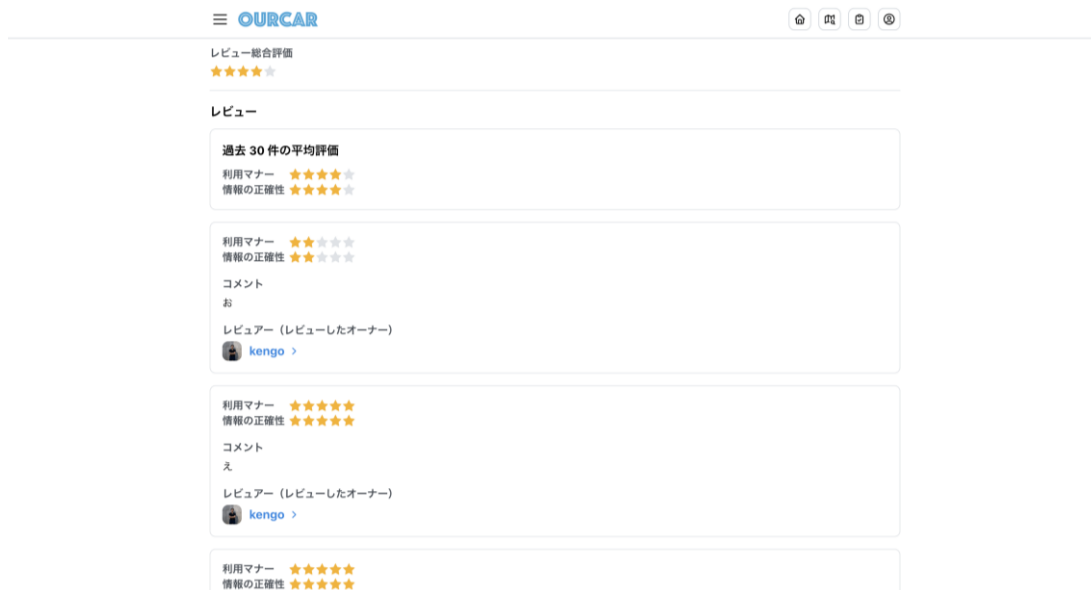


図 2-95 作成されたレビュー一覧

- スマートフォン画面



図 2-96 作成されたレビュー一覧

- 本画面の概要
  - ユーザーの過去の利用に対するレビューが一覧形式で確認できる
- 本画面から利用する機能
  - 【FN015】レビュー作成・公開
  - 【FN022】プロフィール確認

【UI1603】レビュー作成画面<新規開発>

- PC 画面

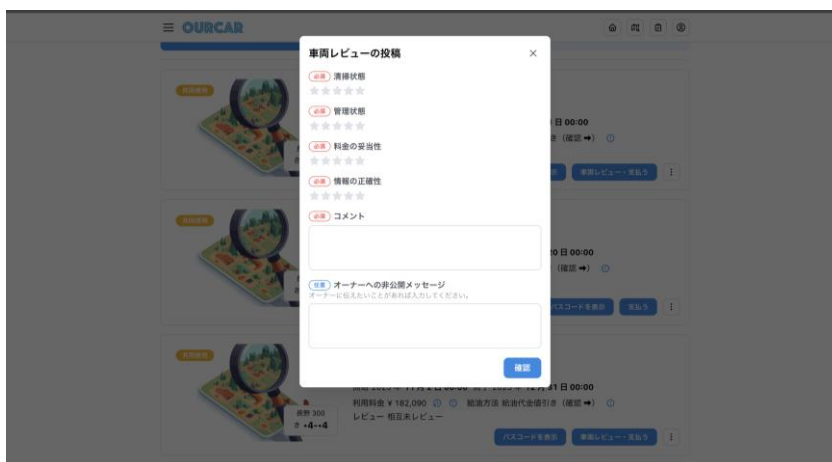


図 2-97 レビュー作成画面

- スマートフォン画面



図 2-98 レビュー作成画面

- 本画面の概要
  - ユーザーが利用した車両に対してレビューを作成する
- 本画面から利用する機能
  - 【FN015】レビュー作成・公開

【UI2101】車両提供者登録画面

- PC 画面



図 2-99 車両提供者登録画面

- スマートフォン画面

図 2-100 車両提供者登録画面

- 本画面の概要
  - 車両提供者アカウント作成に必要な情報（法人・個人の別、連絡先、口座番号等）を入力する
- 本画面から利用する機能
  - 【FN016】車両提供者登録・更新

【UI2102】アカウントページ画面<既存改修>

● PC 画面



図 2-101 アカウントページ画面

● スマートフォン画面

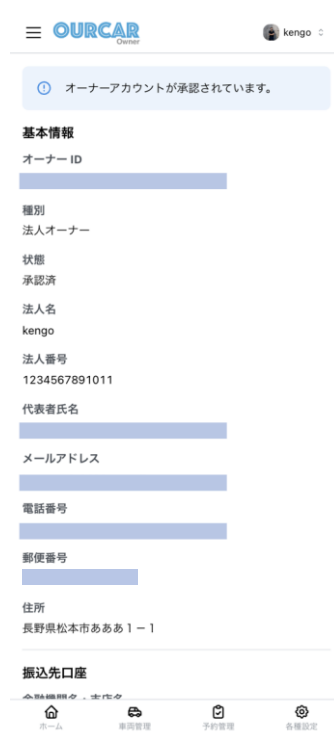


図 2-102 アカウントページ画面

- 本画面の概要
  - 車両提供者アカウント情報の確認・更新ができる
- 本画面から利用する機能
  - 【FN016】車両提供者登録・更新

【UI2201】車両情報登録画面<既存改修>

● PC 画面



図 2-103 車両情報登録画面

● スマートフォン画面



図 2-104 車両情報登録画面

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面の概要
  - 車両登録に必要な情報（車両写真、傷の登録、車検証情報、保険証情報等）を入力する
- 本画面から利用する機能
  - 【FN017】車両登録・更新

### 【UI2202】車両一覧画面

- PC 画面



図 2-105 車両一覧画面

- スマートフォン画面

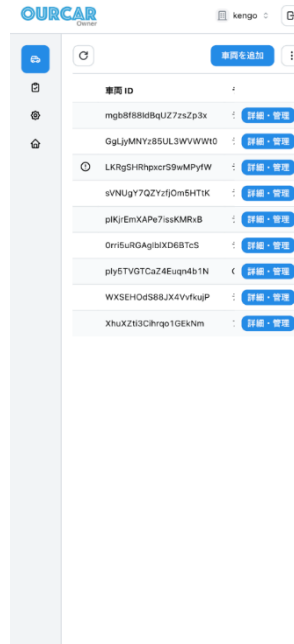


図 2-106 車両一覧画面

- 本画面の概要
  - 車両提供者アカウントにひも付く登録されている車両を一覧形式で確認できる
- 本画面から利用する機能
  - 【FN017】車両登録・更新

## 【UI2203】スケジュール画面

- PC 画面



図 2-107 スケジュール画面

- スマートフォン画面



図 2-108 スケジュール画面

- 本画面の概要
  - 車両の予約状況をスケジュール形式で確認できる
- 本画面から利用する機能
  - 【FN018】公開コントロール

## 【UI2204】基本スケジュール設定画面

- PC画面

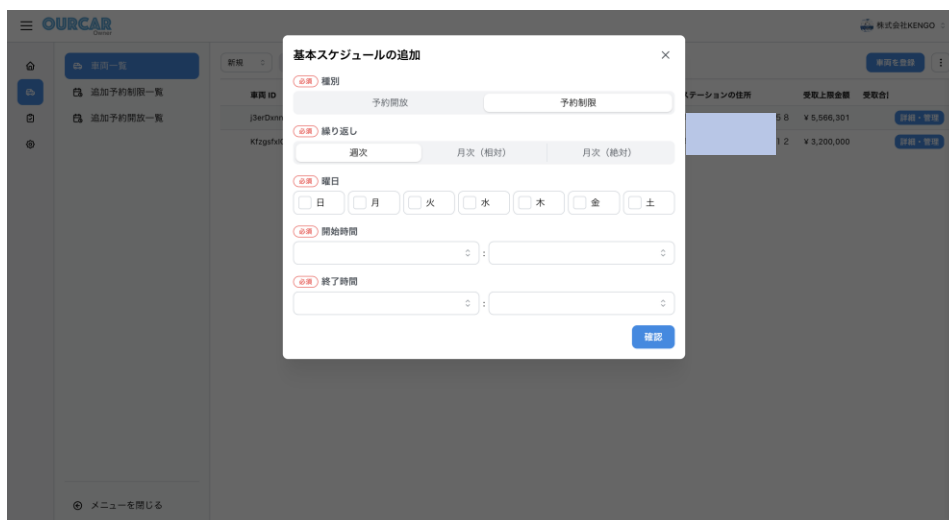


図 2-109 基本スケジュール設定画面

- スマートフォン画面



図 2-110 スケジュール設定画面

- 本画面の概要
  - 利用の可否を曜日、時間、第何週等の種類から指定する
- 本画面から利用する機能
  - 【FN018】公開コントロール

【UI2205】メンテナンス作成画面

- PC 画面

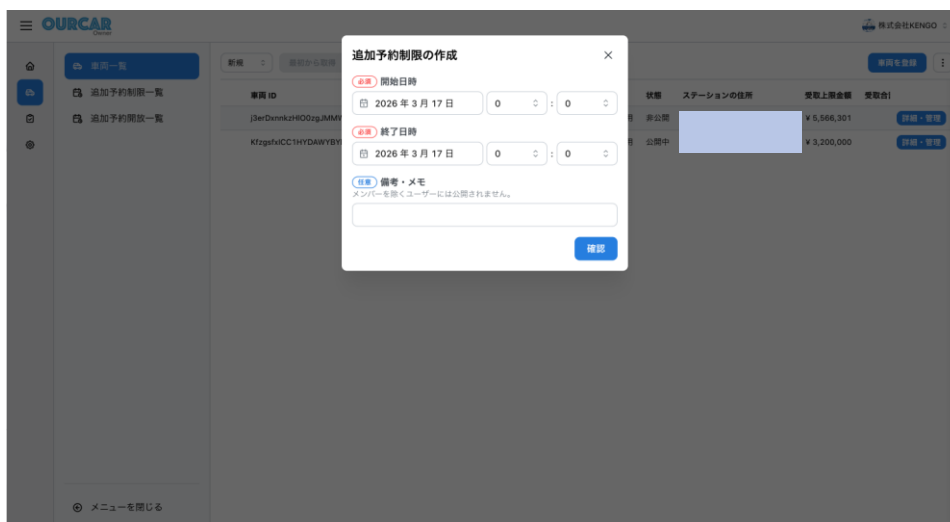


図 2-111 メンテナンス作成画面

- スマートフォン画面



図 2-112 メンテナンス作成画面

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面の概要
  - 予約不可な日時を指定して基本スケジュールとは別に追加する
- 本画面から利用する機能
  - 【FN018】公開コントロール

### 【UI2206】限定リリース作成画面

- PC 画面



図 2-113 限定リリース作成画面

- スマートフォン画面



図 2-114 限定リリース作成画面

- 本画面の概要
  - 予約可能な日時を指定して基本スケジュールとは別に追加する
- 本画面から利用する機能
  - 【FN018】公開コントロール

【UI2207】予約受付時間設定画面

- PC 画面



図 2-115 予約受付時間設定画面

- スマートフォン画面



図 2-116 予約受付時間設定画面

- 本画面の概要
  - 予約開始時間が現時刻から何時間後であれば予約を作成できるようにするかを指定する
- 本画面から利用する機能
  - 【FN018】公開コントロール

## 【UI2301】予約一覧画面

- PC 画面



図 2-117 予約一覧画面

- スマートフォン画面

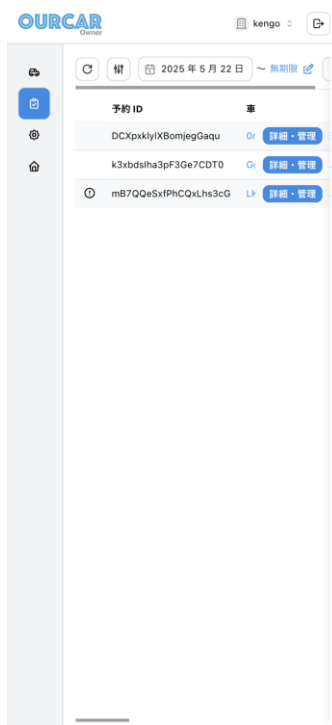


図 2-118 予約一覧画面

- 本画面の概要
  - 登録した車両に対して作成された予約が一覧形式で確認できる
- 本画面から利用する機能
  - 【FN021】予約関連情報確認

## 【UI2302】キャンセル申請画面

- PC 画面



図 2-119 キャンセル申請画面

- スマートフォン画面



図 2-120 キャンセル申請画面

- 本画面の概要
  - 既に作成されている予約に対して車両提供者側の都合によるキャンセルを実行する

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面から利用する機能
  - 【FN020】車両提供者都合キャンセル

### 【UI2401】承認ユーザー一覧画面<新規開発>

- PC 画面

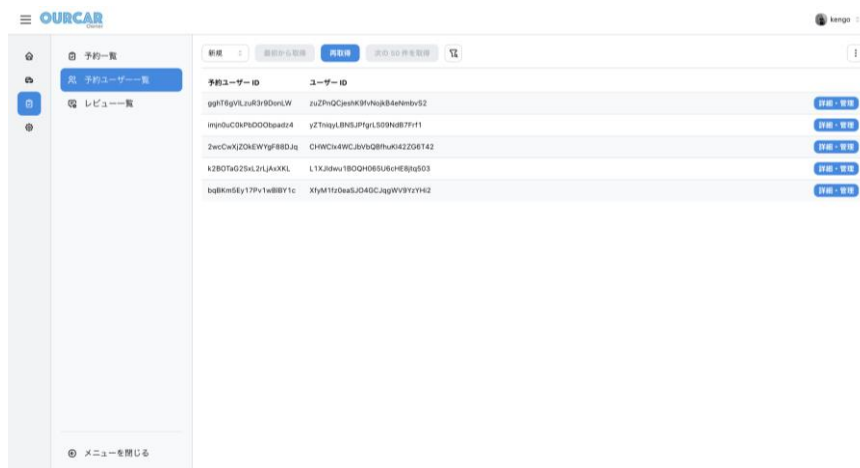


図 2-121 承認ユーザー一覧画面

- スマートフォン画面

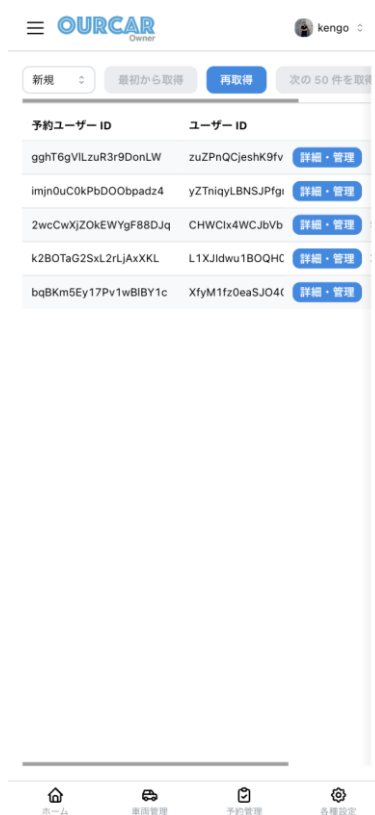


図 2-122 承認ユーザー一覧画面

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面の概要
  - 予約作成したユーザーを一覧形式で確認することができる
- 本画面から利用する機能
  - 【FN019】ユーザー拒否

### 【UI2402】ユーザー拒否画面<新規開発>

- PC 画面

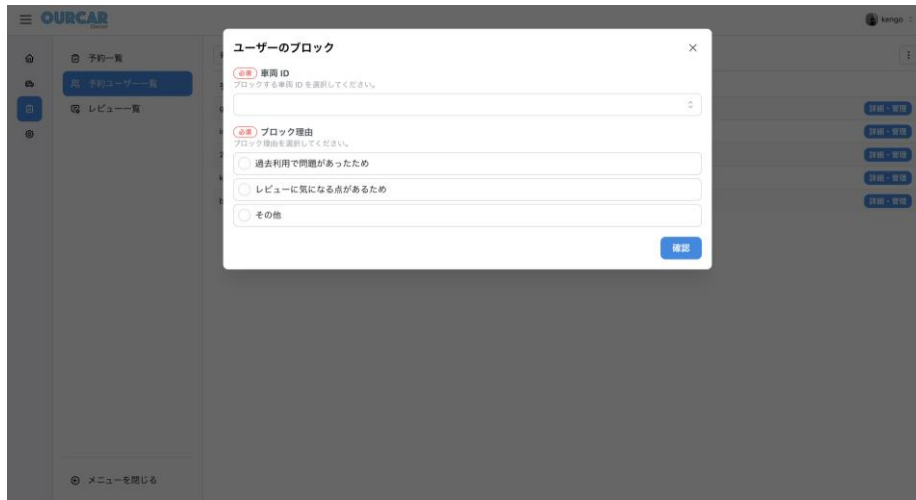


図 2-123 ユーザー拒否画面

- スマートフォン画面



図 2-124 ユーザー拒否画面

- 本画面の概要
  - 予約作成したユーザーを車両提供者側の判断で共同使用契約締結の拒否ができる

共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面から利用する機能
  - 【FN019】 ユーザー拒否

【UI2501】 作成したレビュー一覧<新規開発>

- PC 画面



図 2-125 作成したレビュー一覧

- スマートフォン画面

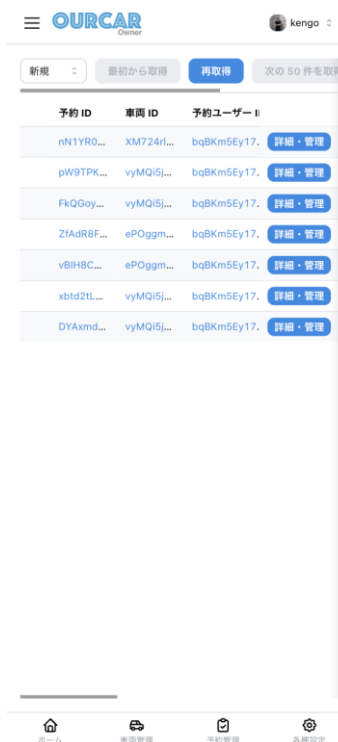


図 2-126 作成したレビュー一覧

- 本画面の概要
  - 過去に車両提供者が作成したレビューが一覧形式で確認できる
  - レビューの詳細（コメントの内容、星評価）も確認できる

## 共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面から利用する機能
  - 【FN015】 レビュー作成・公開

### 【UI2502】 レビュー作成画面<新規開発>

- PC 画面

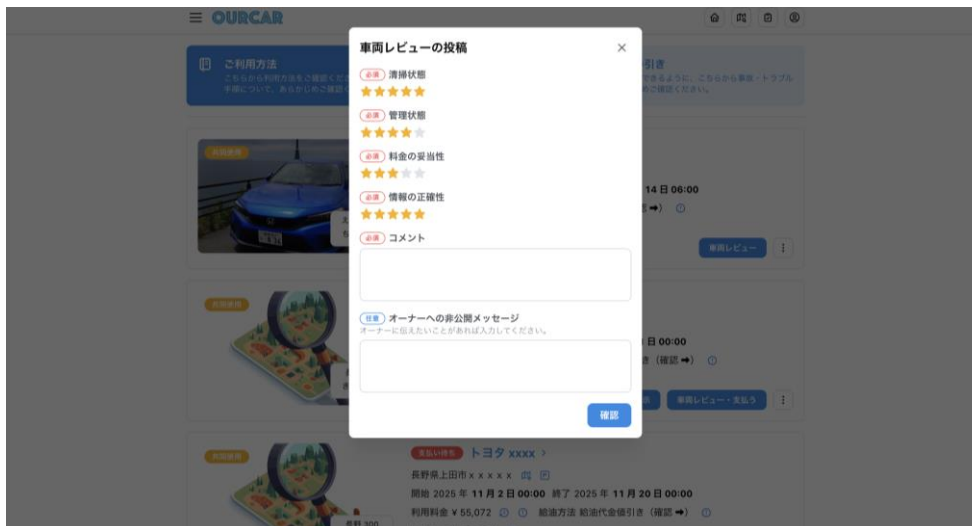


図 2-127 レビュー作成画面

- スマートフォン画面



図 2-128 レビュー作成画面

- 本画面の概要
  - 利用したユーザーに対して利用後の状態を確認後、レビューを作成できる
- 本画面から利用する機能
  - 【FN015】 レビュー作成・公開

【UI2601】作成されたレビュー一覧<新規開発>

- PC画



図 2-129 作成されたレビュー一覧

- スマートフォン画面

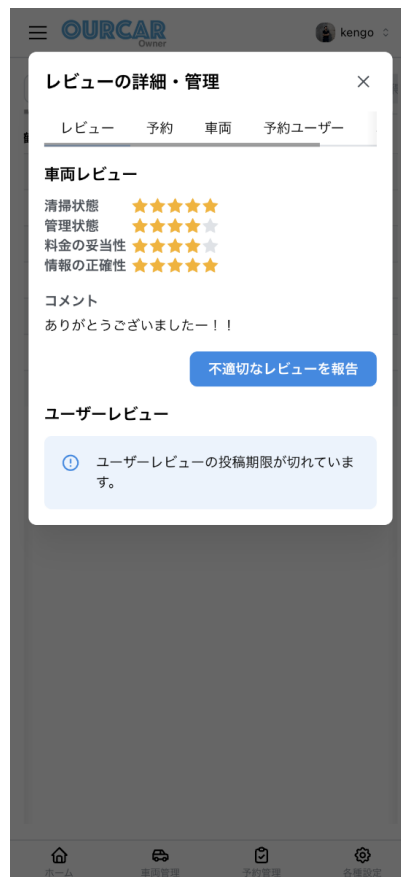


図 2-130 作成されたレビュー一覧

- 本画面の概要
  - 過去に利用したユーザーからのレビューを確認できる
- 本画面から利用する機能
  - 【FN015】レビュー作成・公開

【UI2701】ホーム（プロフィール）画面<既存改修>

● PC 画面



図 2-131 ホーム（プロフィール）画面

● スマートフォン画面



図 2-132 ホーム（プロフィール）画面

- 本画面の概要
  - 車両提供者自身のプロフィールの内容が確認できる
- 本画面から利用する機能
  - 【FN005】プロフィール作成・変更

【UI2702】プロフィール編集画面<既存改修>

● PC画面



図 2-133 プロフィール編集画面

● スマートフォン画面



図 2-134 プロフィール編集画面

- 本画面の概要
  - 車両提供者のプロフィールが作成・変更できる
- 本画面から利用する機能
  - 【FN005】プロフィール作成・変更

【UI3101】ユーザー審査画面

- PC 画面

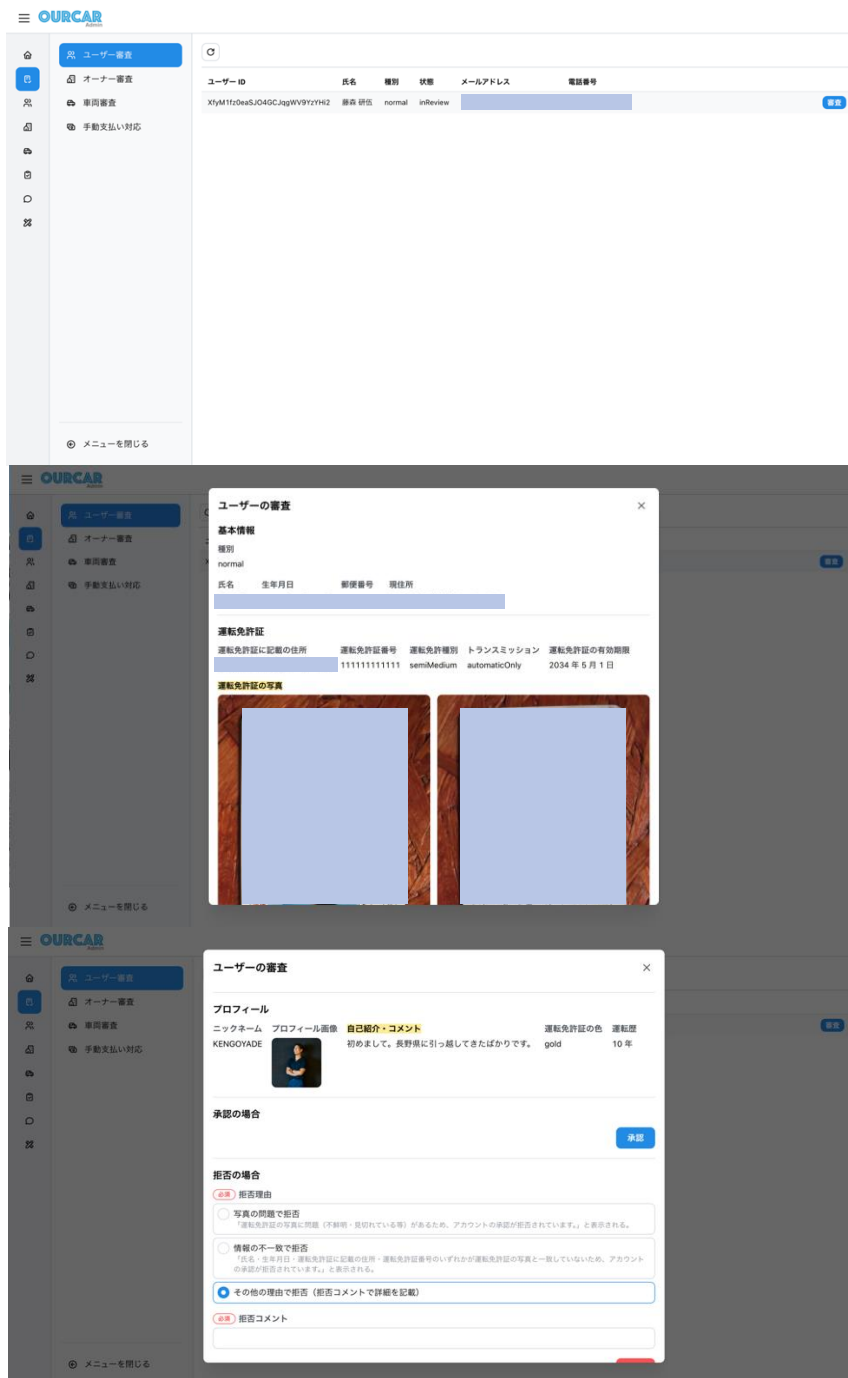


図 2-135 ユーザー審査画面

- スマートフォン画面

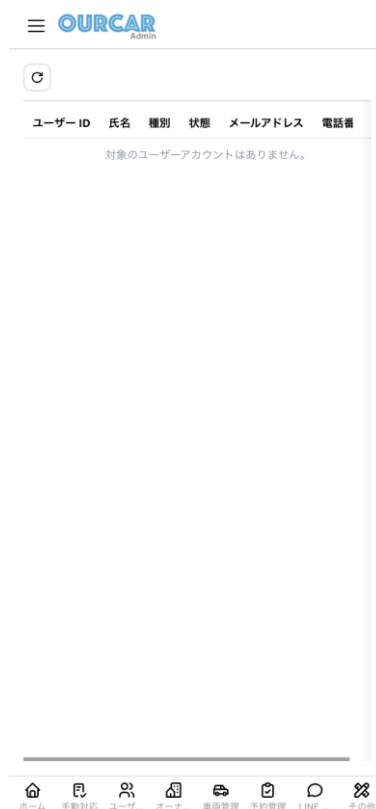


図 2-136 ユーザー審査画面

- 本画面の概要
  - ユーザー登録したユーザーの情報を確認し、承認又は拒否をする
- 本画面から利用する機能
  - 【FN025】ユーザー/車両提供者/車両審査・承認

【UI3102】ユーザー一覧画面

- PC 画面

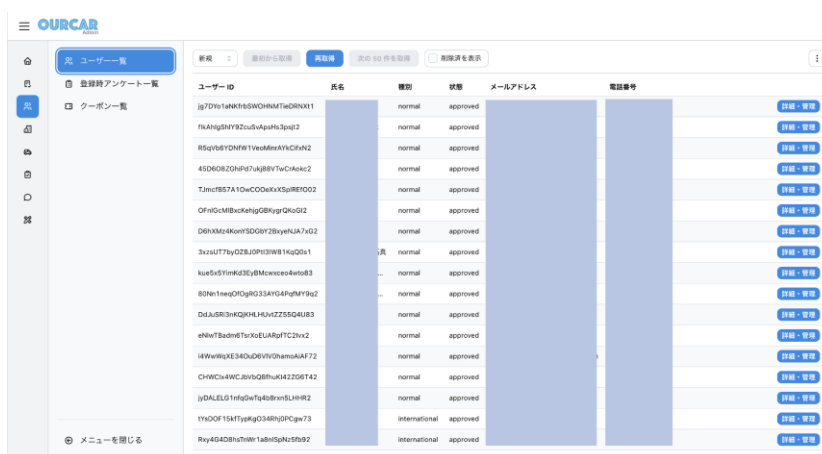


図 2-137 ユーザー一覧画面

- スマートフォン画面

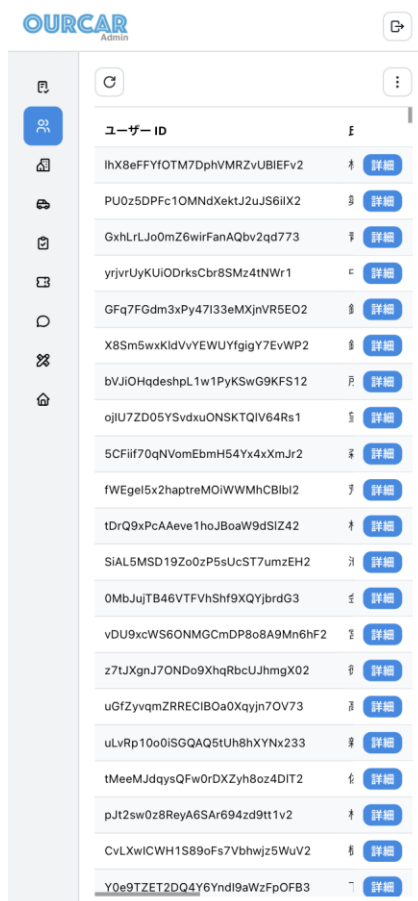


図 2-138 ユーザー一覧画面

- 本画面の概要
  - サービスに登録している全てのユーザーが一覧形式で確認できる
  - ユーザー詳細ページからはステータスを変更することができる
- 本画面から利用する機能
  - 【FN023】各種ステータス変更

【UI3103】登録時アンケート一覧画面

- PC 画面

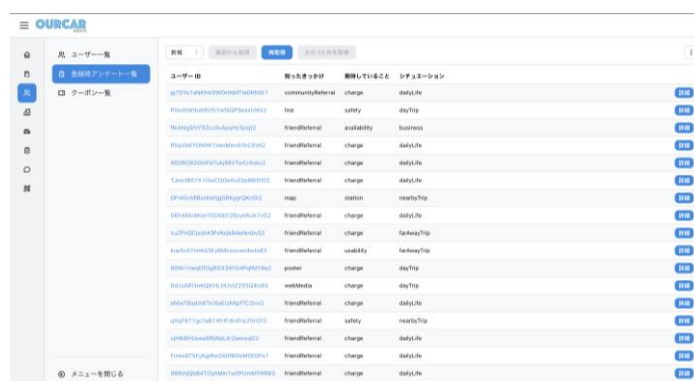


図 2-139 登録時アンケート一覧画面

● スマートフォン画面



図 2-140 登録時アンケート一覧画面

- 本画面の概要
  - ユーザー登録時に入力するアンケート内容が一覧形式で確認できる
- 本画面から利用する機能
  - 【FN027】 Web アプリデータ分析

【UI3104】ユーザープロフィール画面<既存改修>

● PC 画面



図 2-141 ユーザープロフィール画面

● スマートフォン画面



図 2-142 ユーザープロフィール画面

- 本画面の概要
  - 各ユーザーのプロフィールが確認できる
- 本画面から利用する機能
  - 【FN022】プロフィールの確認

【UI3201】車両提供者審査画面

- PC 画面

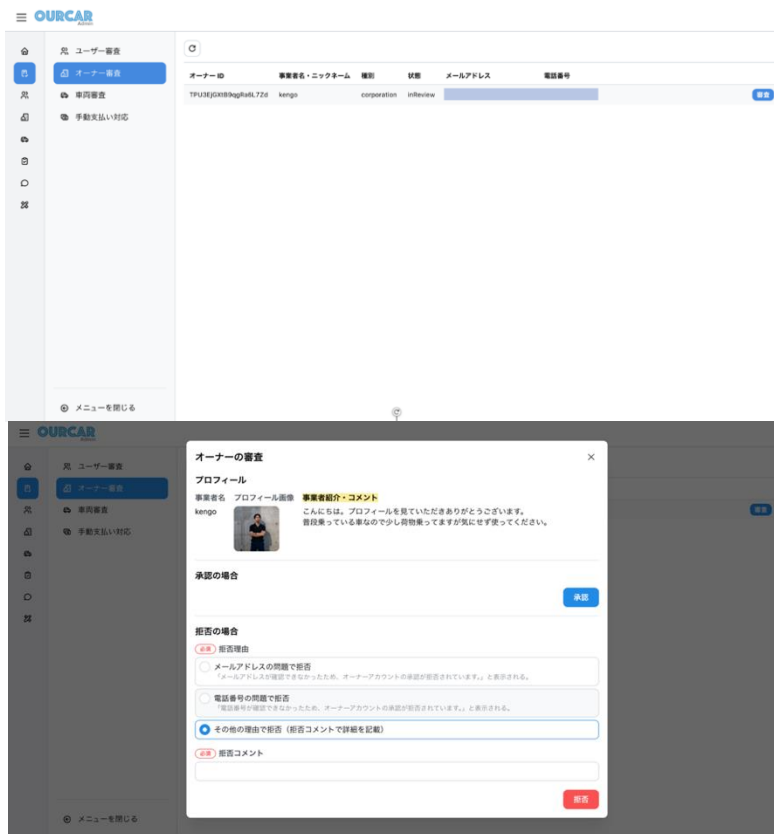


図 2-143 車両提供者審査画面

- 本画面の概要
  - 車両提供者登録に申請したアカウントを一覧で確認できる
  - 申請を確認し、承認・拒否を選択できる
- 本画面から利用する機能
  - 【FN025】ユーザー/車両提供者/車両審査・承認

【UI3202】車両提供者一覧画面

- PC 画面

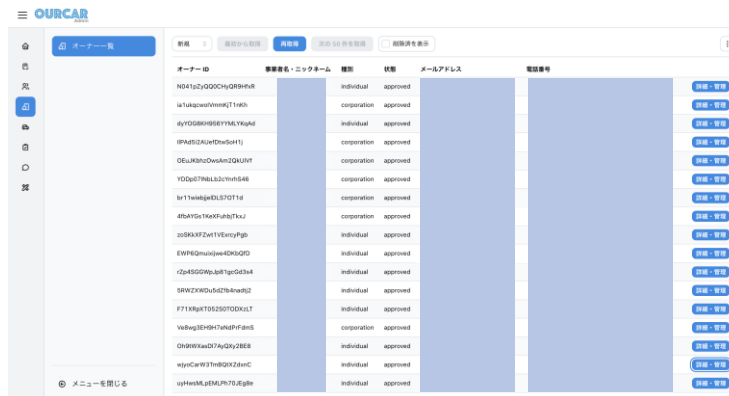


図 2-144 車両提供者一覧画面

- スマートフォン画面



図 2-145 車両提供者一覧画面

- 本画面の概要
  - 車両提供者登録したアカウントが一覧で確認できる
  - 車両提供者詳細画面から車両提供者アカウントのステータスが変更できる
- 本画面から利用する機能
  - 【FN023】各種ステータス変更

【UI3203】車両提供者プロフィール画面<既存改修>

- PC 画面



図 2-146 車両提供者プロフィール画面

- スマートフォン画面



図 2-147 車両提供者プロフィール画面

- 本画面の概要
  - 車両提供者のプロフィールが確認できる
- 本画面から利用する機能
  - 【FN022】プロフィールの確認

【UI3301】車両審査画面

- PC 画面



図 2-148 車両審査画面

- スマートフォン画面



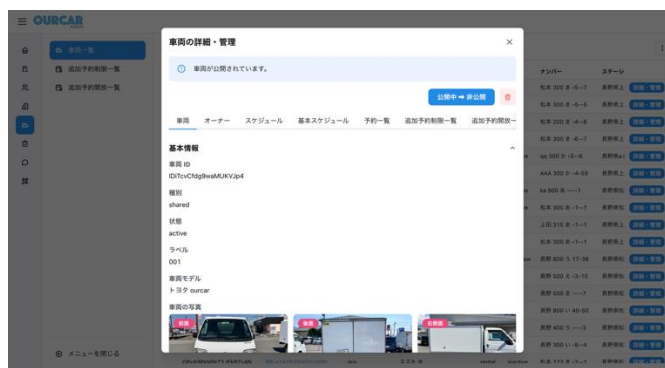


図 2-150 車両一覧画面

- スマートフォン画面



図 2-151 車両一覧画面

- 本画面の概要
  - 登録された車両が一覧形式で確認できる
  - 車両詳細から車両の公開ステータスが変更できる
- 本画面から利用する機能
  - 【FN023】各種ステータス変更

【UI3303】メンテナンス一覧画面

- PC 画面

# 共同使用契約に基づくカーシェアリングシステム システム設計書

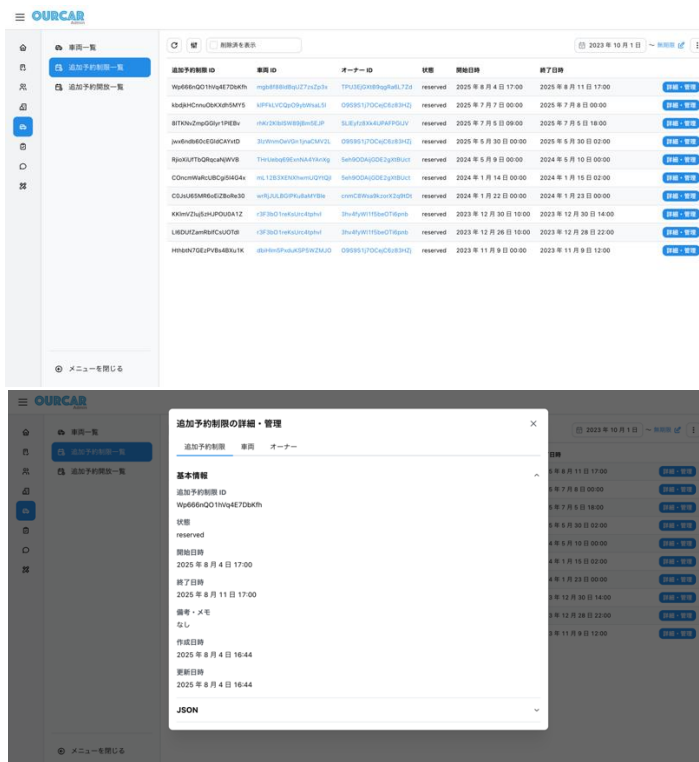


図 2-152 メンテナンス一覧画面

- スマートフォン画面



図 2-153 メンテナンス一覧画面

- 本画面の概要
  - 作成されたメンテナンス一覧が確認できる

共同使用契約に基づくカーシェアリングシステム システム設計書

- メンテナンス詳細画面からステータスが変更できる
- 本画面から利用する機能
  - 【FN023】各種ステータス変更

【UI3304】限定リリース一覧画面

- PC 画面

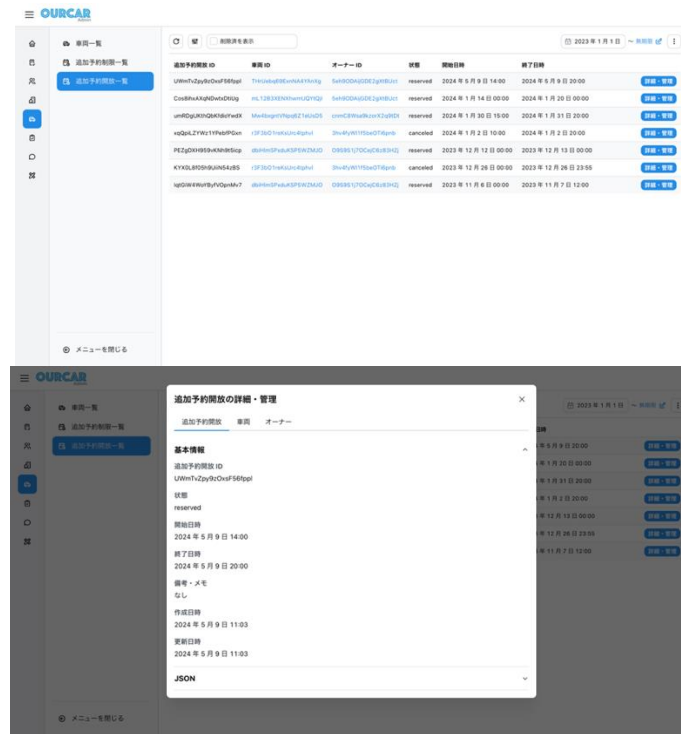


図 2-154 限定リリース一覧画面

- スマートフォン画面



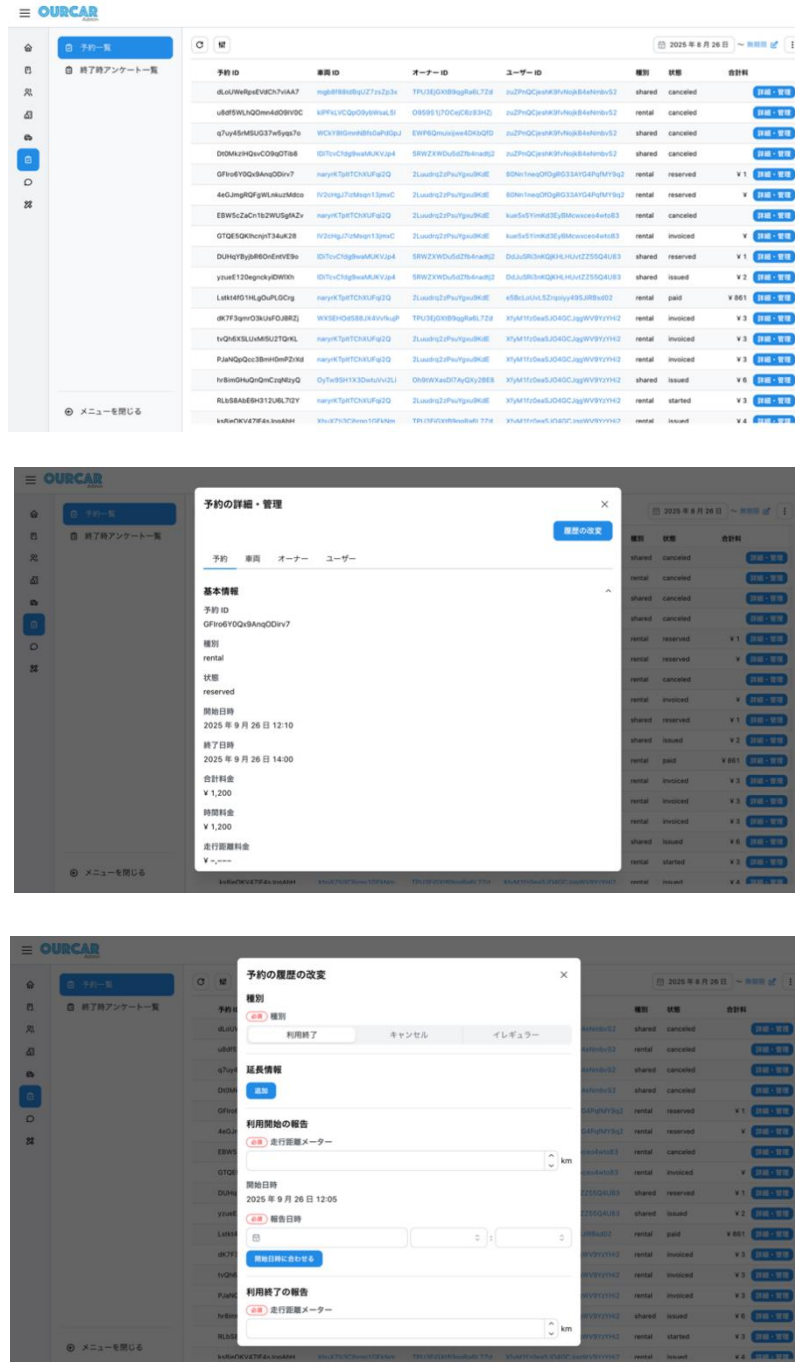
図 2-155 限定リリース一覧画面

共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面の概要
  - 作成された限定リリース一覧が確認できる
  - 限定リリース詳細画面からステータスが変更できる
- 本画面から利用する機能
  - 【FN023】各種ステータス変更

【UI3401】予約一覧

- PC 画面



● 図 2-156 予約一覧

- スマートフォン画面



図 2-157 予約一覧

- 本画面の概要
  - 作成された予約が一覧形式で確認できる
  - 予約詳細からステータスの変更ができる
- 本画面から利用する機能
  - 【FN023】各種ステータス変更

【UI3402】終了時アンケート一覧

- PC 画面

# 共同使用契約に基づくカーシェアリングシステム システム設計書

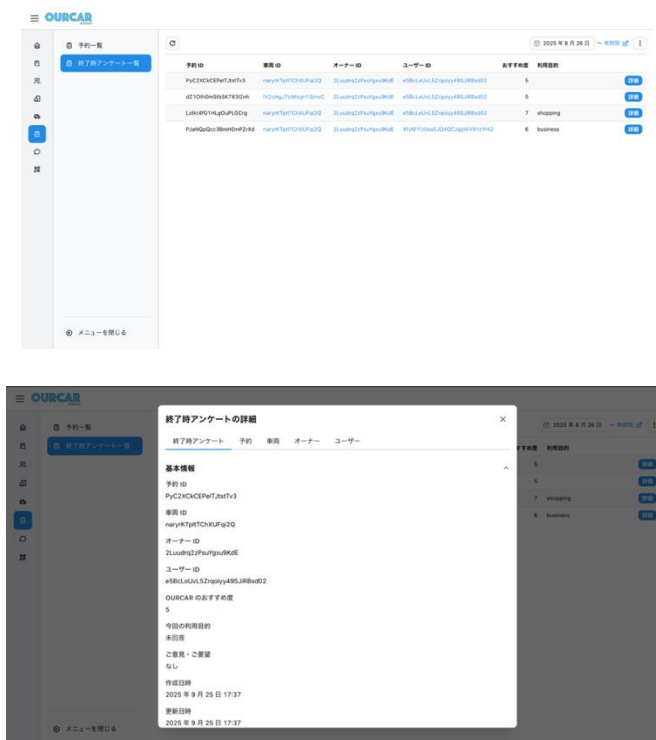


図 2-158 完了時アンケート一覧

- スマートフォン画面

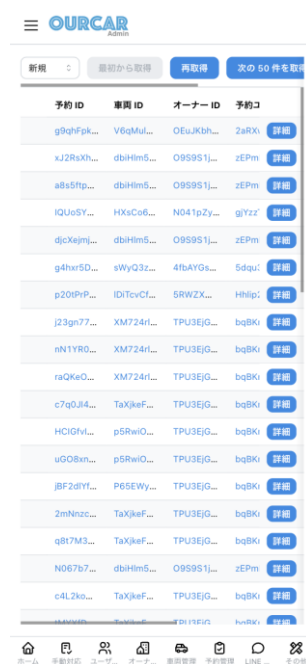


図 2-159 完了時アンケート一覧

- 本画面の概要
  - 利用終了時にユーザーが記入するアンケートが一覧形式で確認できる
- 本画面から利用する機能

➤ 【FN027】 Web アプリデータ分析

【UI3501】 レビュー一覧<新規開発>

● PC 画面

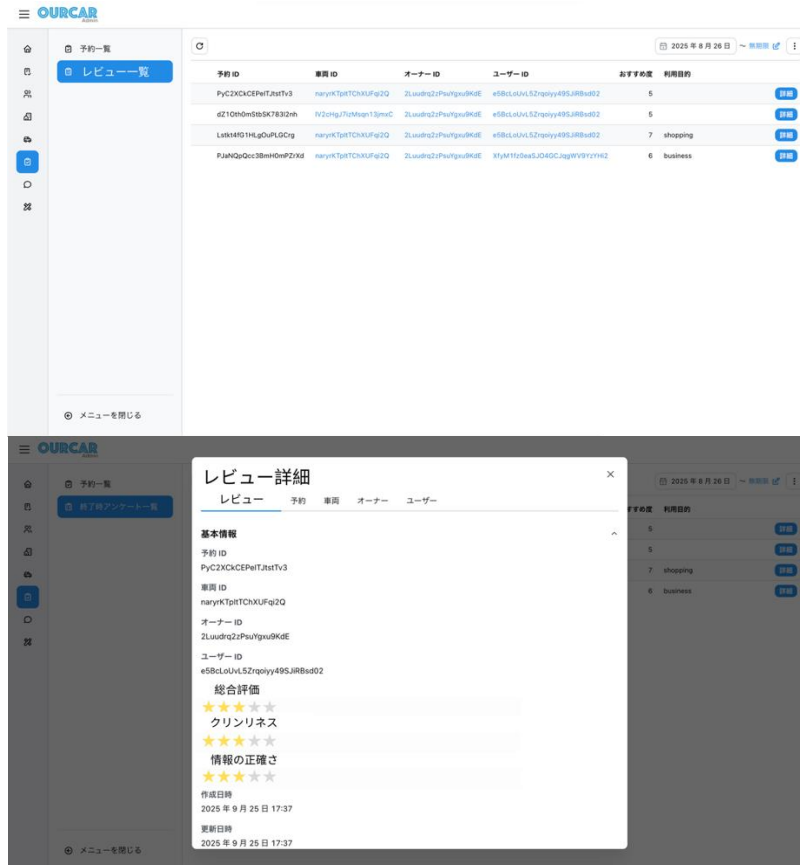


図 2-160 レビュー一覧

- スマートフォン画面



図 2-161 レビュー一覧

- 本画面の概要
  - 予約に対して作成された車両提供者側、ユーザー側のレビューを一覧形式で確認できる
  - レビュー詳細からステータスの変更ができる
- 本画面から利用する機能
  - 【FN023】各種ステータス変更

【UI3502】報告レビュー一覧<新規開発>

● PC 画面

報告 ID	審査 ID	オーナー ID	子約ユーザー ID	ユーザー ID	報告レビュー状態	連携状態	管理状態	料金の妥当性	情報の正確性	ユーザーレビュー状況	
3FN89...	KstCHJ...	dyYGG8...	mu5v8kAts2...	3zuzUT7...	published	2	5	4	4	published	
AM3mw...	RBKQ03...	Qh9WY...	nUD4Jg4u9...	eNwTba...	published	5	5	5	5	expired	
pw9TPK...	vyMQ9...	TPU3EG...	bgkK6Ey17...	XyM1t...	published	3	4	4	5	3	expired
74kky...	2hd0m...	8PA5S...	YnjQw0rth...	6SAngS...	published	1	3	4	2	2	expired
VUwDn...	y8Ln05...	YDdp07...	M5Xny2ge1...	4SD6O8...	published	5	5	5	5	5	published
2knhab...	y8Ln05...	YDdp07...	CNARTPP5a...	XyM1t...	published	4	4	4	4	4	expired
mk7DK...	V6qkL...	OEuJKh...	gH8KERNW...	RSqV6F...	published	1	2	1	2	2	expired
M6vW3...	ylqMda...	br11web...	YmwZM54...	XyM1t...	published	3	3	3	3	3	expired
byctzV...	ylqMda...	br11web...	zBHSWGPo1...	Tjncf8L...	published	4	4	4	4	4	published
AhucyG...	wWYQ3...	4f8AYG...	5dqzCzwG4...	QPhGcM...	published	3	5	3	5	5	deleted
9KUZY...	dbhe85...	O55851...	zPhmMwFV...	7DQgkf...	published	4	4	4	4	4	expired
FxQ0ay...	vyMQ9...	TPU3EG...	bgkK6Ey17...	XyM1t...	published	1	1	1	1	1	expired
ZkAR8F...	eQagm...	TPU3EG...	bgkK6Ey17...	XyM1t...	published	2	2	2	2	2	published
v8H8C...	eQagm...	TPU3EG...	bgkK6Ey17...	XyM1t...	published	3	3	3	3	3	published
x8z2L...	vyMQ9...	TPU3EG...	bgkK6Ey17...	XyM1t...	published	5	5	5	5	5	published
DYAmnd...	vyMQ9...	TPU3EG...	bgkK6Ey17...	XyM1t...	deleted	3	3	3	3	3	deleted
5BL3M...	Oyfe98...	Qh8WY...	5mpz2WqG...	XyM1t...	published	3	3	3	3	3	expired

図 2-162 報告レビュー一覧

● スマートフォン画面

クーポン ID	ユーザー ID
xjPispGTOpl64VOMD6xX	u16RyPcGc
T1qBeMR98ciHwOqg8f6	x3kbNs7Jb
IB0IhnML7zG4Fjg4Iuc2	R6tbFDSYL
MW2mADkAyGdkPMOdpSrM	PyXamOmc
5W2icB9CWnO7kOQD4VGa	ZzPIS2TJKs
XbCOVs4qJc9t3VW2psWD	CQFDans0:
iXbKDTXm11BgosPYeju7	o6kKsr4fcv
HE99sh7v12FErJUEbaB	24G6UHTM
ILlctH7PbyilUqKNizKl	CQFDans0:
SAPluVtszTA0ozkhh4w	sGRUCX05
mSeS1nVMEJilH8GrHn95	BDS9e31dr
XawWdJds3BK9Zkiv2v0dx	TxL38GMJ:
s1xCr7Qwc10GpAQs1aVi	Pb0hr9HUj
2TUsGuQaqq39A1Ggik5H	TxL38GMJ:
LJLGduauOYrgzJKTivrP	vBFouHpm
vKT5Jte5wkJhTQtnfw3v	Rts0KbUg8
BP71S6yuq4eRtALyFcPX	mi9sAk6Rc
UxFP1CPBjUEJapnaRzxn	xyV6hArSe
4KFoFcvTAEZmammaB6Y	NuvZ1Scm
93zxFOQUXypUXmu3KRFP	ZYAaCoPFL
9IEPTQXBcVMFVbt4rpk3	GnU3e1scf

図 2-163 報告レビュー一覧

● 本画面の概要

- 不適切なレビューとしてユーザー・車両提供者から報告があったレビューが一覧形式で確認できる
- レビュー詳細からステータスの変更ができる

共同使用契約に基づくカーシェアリングシステム システム設計書

- 本画面から利用する機能
  - 【FN023】各種ステータス変更

【UI3601】手動支払審査

- PC 画面

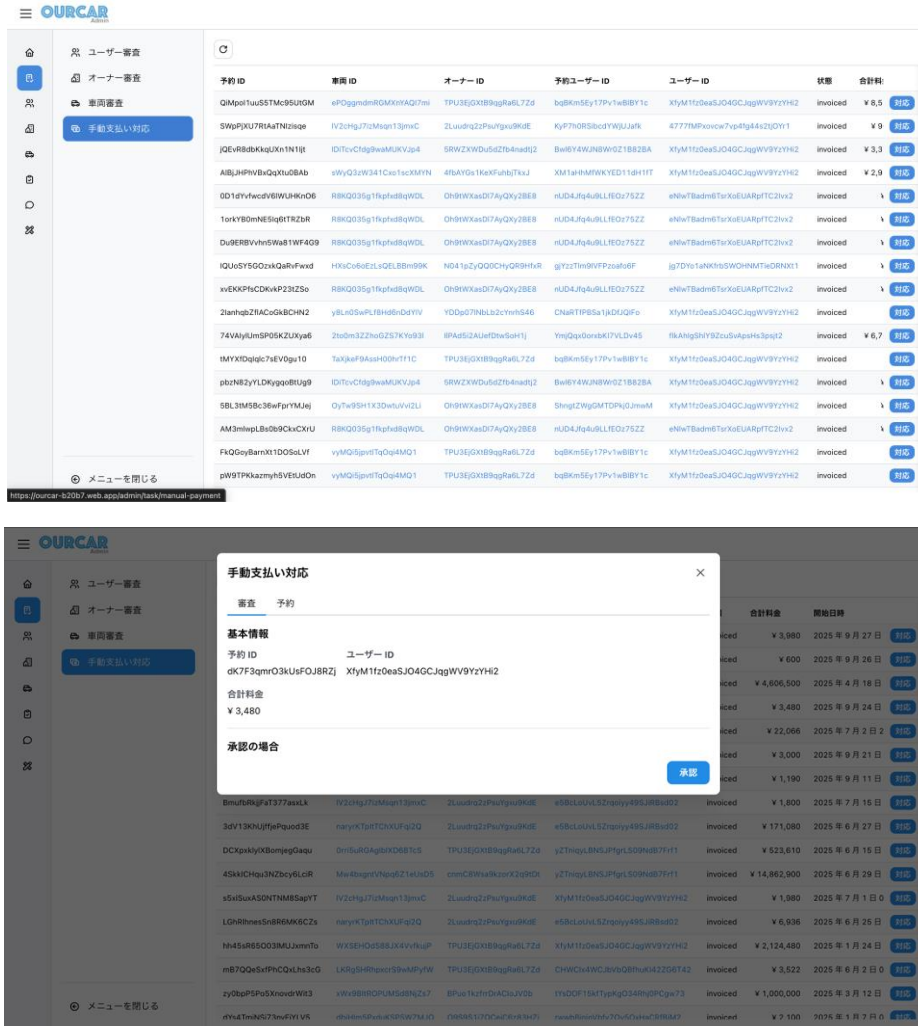


図 2-164 手動支払審査

- スマートフォン画面

図 2-165 手動支払審査



- 本画面の概要
  - PayPay での支払を手動で承認することができる
  - PayPay からの承認が下り次第、自動支払認証が実装可能
- 本画面から利用する機能
  - 【FN023】各種ステータス変更

### 3. 共同使用契約に基づくカーシェアリングシステム：非機能要件 (NF)

#### 3-1. 非機能要件一覧

表 3-1 非機能要件一覧

カテゴリ	ID	非機能項目	要件詳細
セキュリティ	NF001	セキュリティ	<ul style="list-style-type: none"> <li>● セキュリティ強化：トークンの自動更新や多要素認証（MFA）に対応し、Firebase セキュリティルールにのっとり不正アクセスを防止する</li> </ul>
	NF002	認証	<ul style="list-style-type: none"> <li>● Firebase Authentication を通した電話番号・メールアドレス及び免許証画像を用いた本人認証を実施する</li> </ul>
拡張性	NF003	システムスケーラビリティ	<ul style="list-style-type: none"> <li>● スケール可変なサーバーレスサービスを利用して開発し、開発においても適切なブランチ設計を行うことで追加開発にも対応できる環境を構築する</li> </ul>
データバックアップ	NF004	データのバックアップ	<ul style="list-style-type: none"> <li>● ユーザー・車両提供者・車両データ等を保管し、障害発生時にもデータを復旧できる体制を構築する</li> </ul>

## 3-2. 非機能要件の詳細

### 【NF001】セキュリティ

- 本非機能要件の概要
  - ユーザーの個人情報を保存しているデータベース不正アクセスされないようにする
- 設定理由
  - 不正アクセスによる情報漏えいのリスクがあるため

### 【NF002】認証

- 本非機能要件の概要
  - 外部サービスの認証も用いるなら確実な本人確認を行う
- 設定理由
  - なりすまし利用による盗難や事故、違法行為を未然に防ぐため

### 【NF003】システムスケーラビリティ

- 本非機能要件の概要
  - サービスのスケールや追加開発に対応できる環境を構築する
- 設定理由
  - 今後のサービス拡大や方向性の変更による追加開発の際に柔軟に対応できるようにする

### 【NF004】データバックアップ

- 本非機能要件の概要
  - ユーザーデータや車両提供者、車両、予約データ等を Firestore 内でバックアップしておく
    - ◇ 頻度：1分単位でのバックアップを行う
    - ◇ 保存方法：PITR でのデータ保存
    - ◇ データの保存期間：7日間の保存
- 設定理由
  - 障害等でデータが消失した際も、支障なくサービス運営を継続できるようにするため

## 4. 実証調査に利用するデータ (DT)

### 4-1. 実証調査に利用するデータ一覧

表 4-1 実証調査に利用するデータ一覧

※朱文字：本実証で変換・作成するデータ

ID	データ名称	データ形式	出所	データを利用する ID
-	-	-	-	-

### 4-2. 実証調査に利用するデータの詳細

-

## 5. 用語集

用語	定義・説明
Web アプリ	本サービスを提供するための Web アプリケーションシステムのうち、実際にユーザーが画面で確認するユーザーインターフェース。
カーシェア専用保険	共同使用を通した車両使用の際に適用される自動車保険。
共同使用契約	同一の自動車を二以上の者それぞれが、自己の欲求充足のために主体的な立場において使用するための契約。共同使用する使用者が具体的に特定され、自動車の使用及び管理に関してあらかじめの合意が存在しており、使用者それぞれが自動車の使用及び管理に関する権限と責任を有している状態。当該車両を使用する際に支払われる共同使用料は営利を目的とせず、自動車の維持費の範囲内に収める必要がある。
共同使用者	予約作成を通して車両提供者と共同使用契約を締結したユーザー。
共同使用料	共同使用契約を締結して車両を使用する際に、自動車の維持費用の範囲内で共同使用者が車両提供者に支払う金額。
距離料金	走行した距離および燃費に応じて決定される料金。
口座振込手数料	銀行口座に振り込む際に発生する取引手数料。
サービス提供事業者	本サービスを提供・運営する事業者。
サービス手数料	サービス提供事業者が1シェアあたりに取得する手数料金額。
車両提供者	サービス内においてユーザー登録及び車両提供者登録・車両登録を行い、共同使用契約を通した車両提供を行うもの。
時間料金	予約時間に応じて決定される料金。
プロテクト料金	選択したプロテクトプランに応じた料金。
本サービス	Web アプリケーションシステム及びオペレーションを含めた全てのサービス価値。
本システム	本サービスを提供するための Web アプリケーションシステムのうち、機能的な側面を実装しているバックエンドシステム。
ユーザー	サービス内においてユーザー登録を行った者。
利用料金	共同使用料(時間料金)および距離料金やカーシェア専用保険料等を合計したユーザーがサービス提供事業者を支払う金額。
累計受取共同使用料	車両をシェアすることによって車両提供者が受け取る共同使用料の累計金額。



共同使用契約に基づくカーシェアリングシステム 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: 株式会社TRILL.