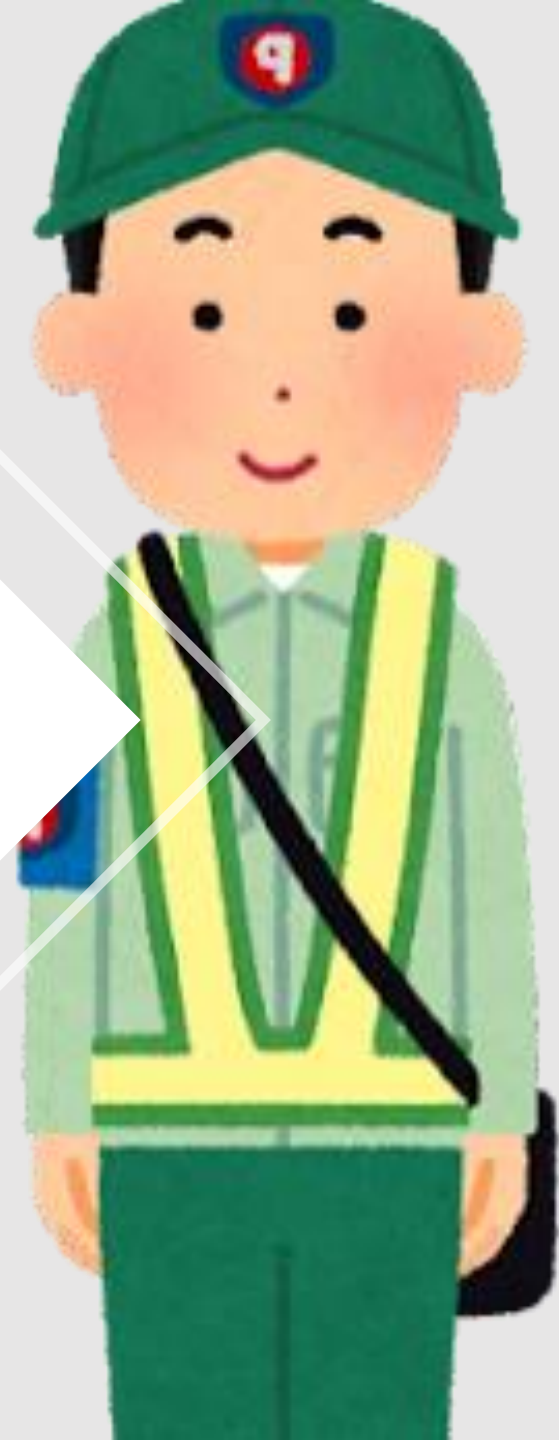




A2

警備会社に向けた
駐車違反取り締まり効率化
システム開発プロジェクト



■ 業務分析

駐車監視員という、違反駐車を取り締まる方がいます。

（このような緑色の服を着た人です。）

次ページからはこの駐車監視員の業務分析を行います。



警備会社
駐車監視員

■ 業務分析

警察



駐車違反の監視業務

業務を委託する！

警備会社
駐車監視員



駐車違反の監視業務の概要:

- ・ 警察から委託されたエリアの巡回
- ・ 駐車違反を見つけたら、規定に則って取り締まり、記録する
- ・ 記録データを警察に送信する

業務分析

違反車両の取り締まりの現在の一連の流れ

- 駐車違反を発見したら、30分程様子を見る。
- 30分後も駐車していた場合「**放置車両違反**」と見なし、記録を開始する
- **携帯端末**を用い、「**違反日時、ナンバー、場所、放置態様、現場写真**」を記録する



- **4INCH THERMAL PRINTER**を用い、**標章**を印刷し、違反車両に貼る



背景・課題・目標



プロジェクトの背景

- ・ 駐車監視員(通称: 緑のおじさん)と呼ばれる仕事がある。
- ・ 彼らは指定エリア内で取り締まりを行い、違反駐車を見つけたら記録を行う。
- ・ 現在は端末を用いてはいるが「手作業」で記録を行っている。
- その結果、記録の誤入力が発生し、違反取り締まりが無効になるケースがある

プロジェクトの目的

- ・ 記録をシステムでサポートし、誤入力を防止する
- ・ 作業内容の簡略化

プロジェクトの目標

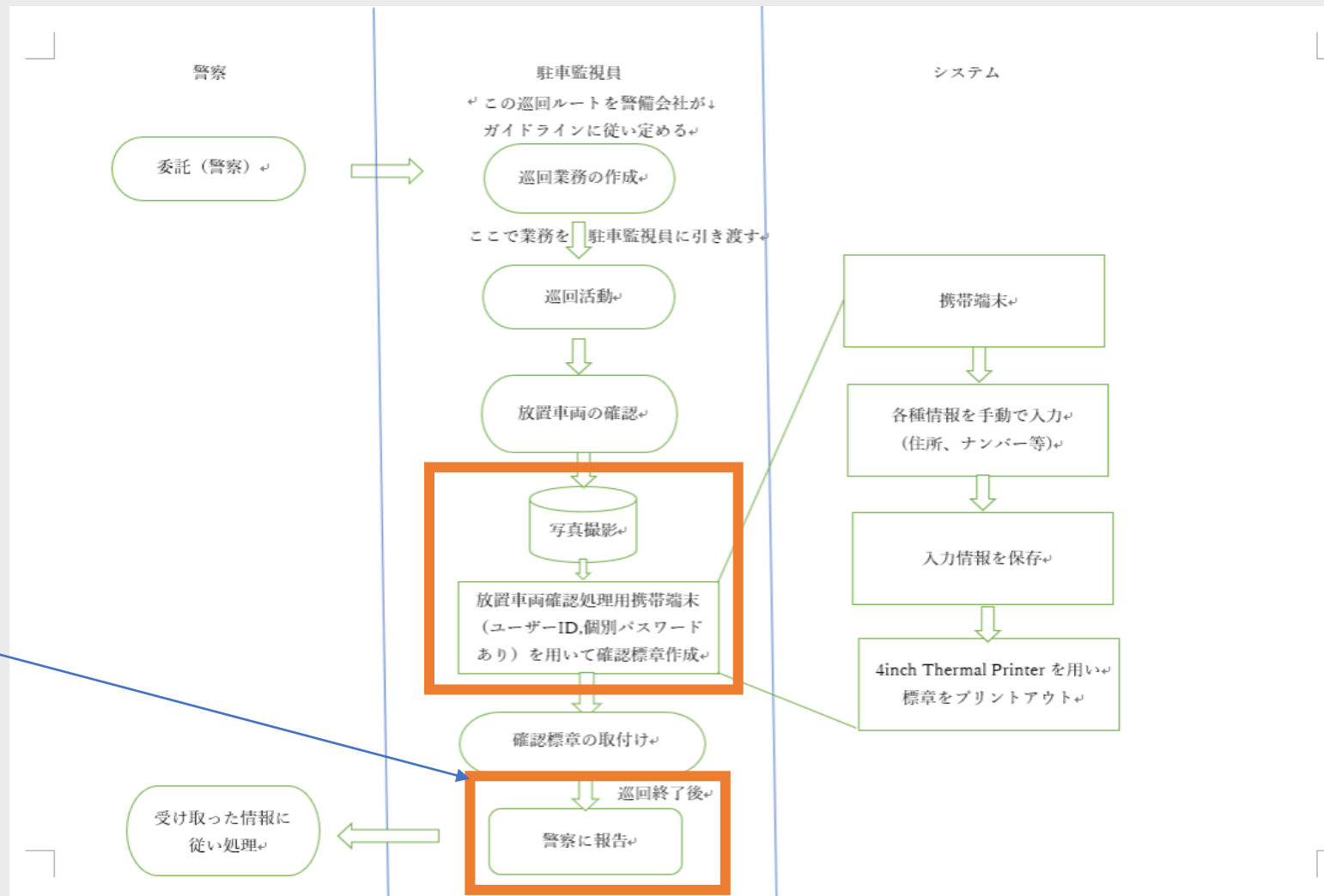
- ・ 誤入力を減らすことを目的とする
Androidアプリ+ 情報管理Webサイトの開発

■ 本プロジェクト導入前の駐車監視員の業務フロー

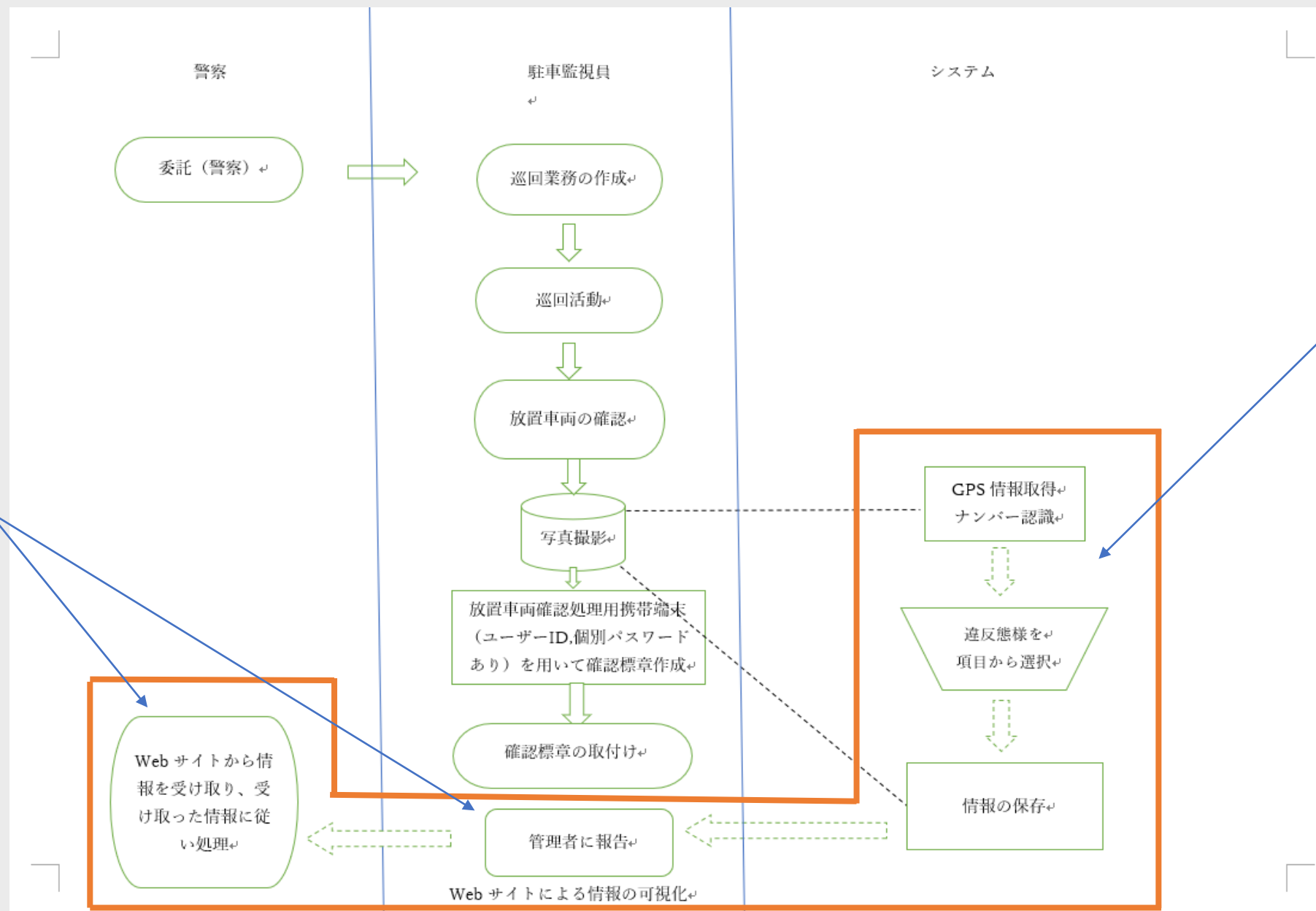
つまり、現在の駐車監視員の業務フローを書き起こすと以下ようになります。

私達A2グループは、**橙枠部分のサポートを行う**ことで**誤入力を防ぐことを目的**にします。

警察・管理会社が**違反情報の記録を確認できるようにサポート**します。



■ 本プロジェクト導入後の駐車監視員の業務フロー



■ 本システムにおけるサポート範囲

違反態様記録に関して

- 主観による判断基準がある (e.g., 駐車禁止、駐停車禁止、私有地駐車)
- 今回のシステムでは手動入力とする。

サポート範囲に関して

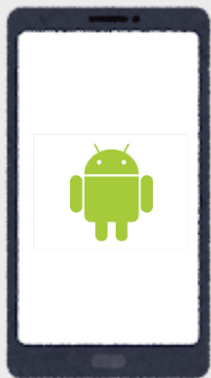
写真入力から確認標章の作成のアウトプット

→pdfファイルとして出力

→運用管理者の報告用のデータの作成、及び閲覧するシステムまで

■ 本システムの概要

駐車監視員



Androidアプリ

- ・ ユーザーログイン認証
- ・ 違反車両のナンバープレートの写真を撮る → ナンバー情報を取得
- ・ 写真を撮ったタイミングで、「GPS情報、日時情報」を取得
- ・ 違反態様情報をガイドラインに従い選択
- ・ 標章の画像を生成、確認の上PDFで保存
- ・ 上記で取得した情報をデータベースにアップロード

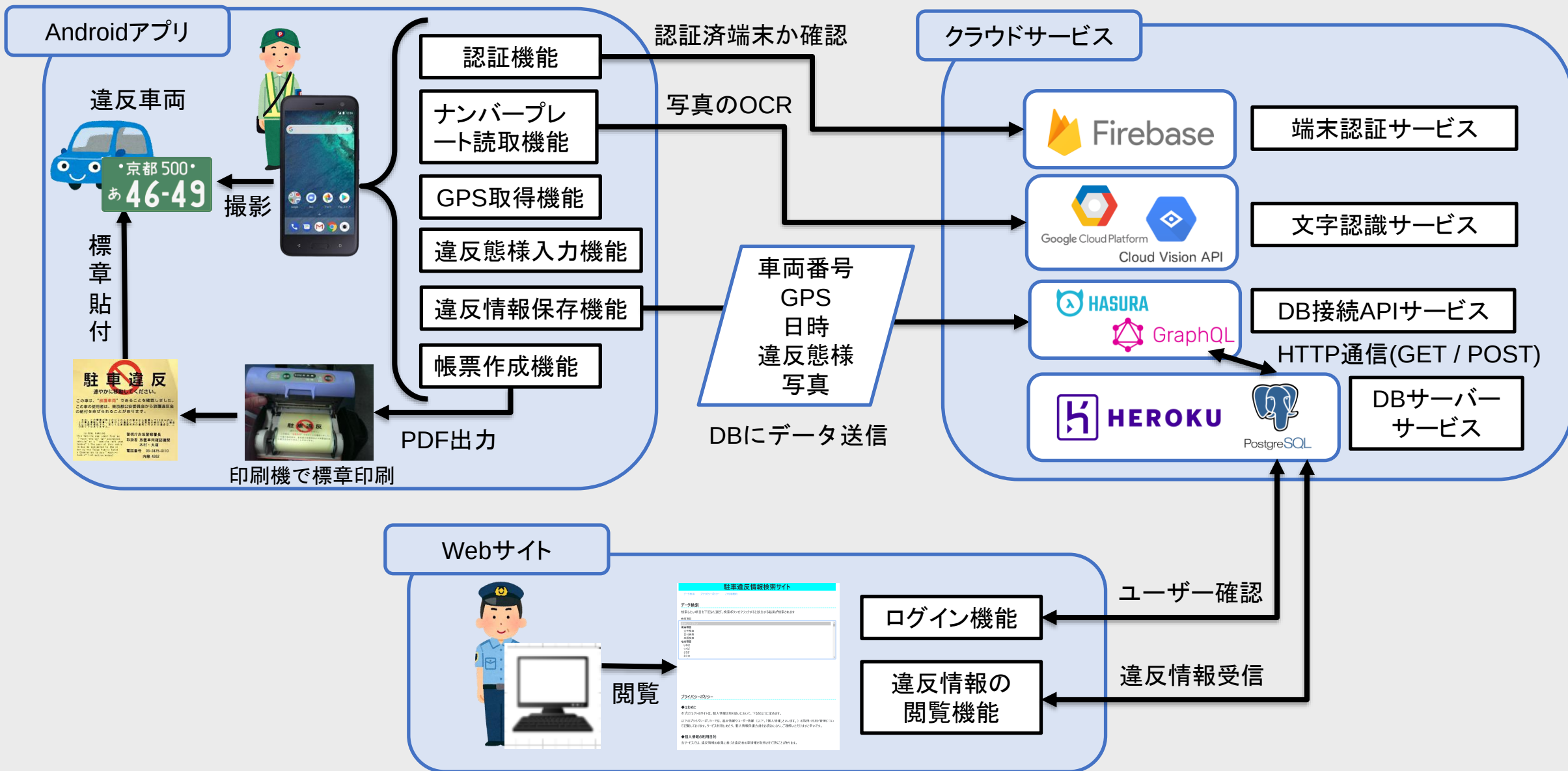
警察
管理会社



Webサイト

- ・ ユーザーログイン認証
- ・ データベースから違反情報を検索
- ・ 「どの地域での違反か」、「何月何日の違反か」等で詳細に検索

システム構成図



■ Google Vision APIを用いた画像のOCR

再掲だが、本システムの主要目的は「**人的要因ミスをなくすこと**」である。

車両ナンバーの人的要因ミスが指摘されていたため、
ナンバーの情報を以下の流れで取得する。

ナンバープレートの写真を撮る

OCR(Optical Character Reader) を行い、画像内のテキストを文字情報とする

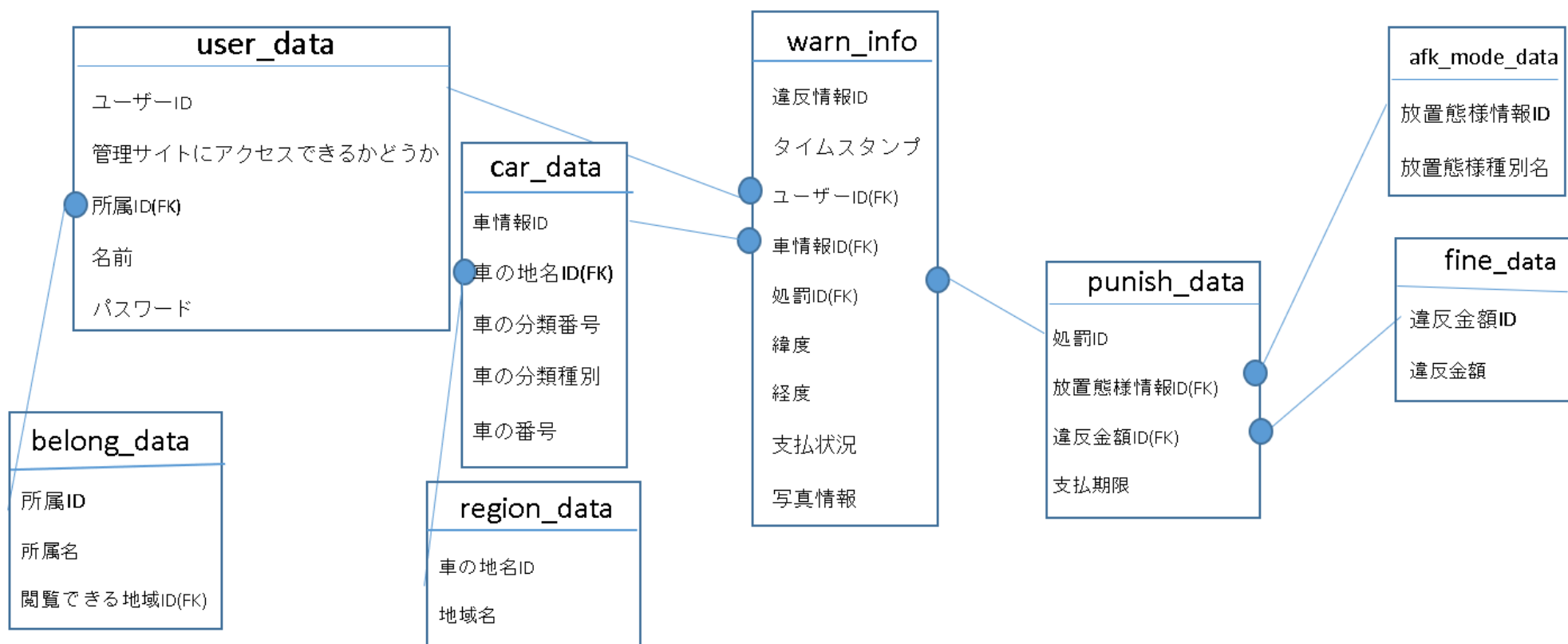


データベース設計と、HTTP通信によるRestful API

今回、違反情報を保存するDBとして、Herokuが提供するHeroku Postgresを利用している。

本プロジェクトでは、実際に運用する時を考え、**100万件入っても効率よくデータが取り出せるように正規化されたリレーション構造を持つ**テーブルを作成している。以下はそのE-R図である。

各種テーブルの詳細は、添付されているDB設計書.xlsxを確認してほしい。



データベース設計と、HTTP通信によるRestful API

そしてHerokuサービス内のDBに、Androidアプリケーションから操作を行うため、Heroku内にGraphQLサーバーを用意し、HTTP通信(GETとPOST)を行うことで、Heroku内部のサービスにアクセスできるようにした。

例えば、前ページのregion_dataテーブルの全データを取得したい場合は以下のURLにアクセスする。

https://peteama-apiserver.herokuapp.com/api/rest/region_data

本APIサーバーを作成するにあたっては

オープンソースソフトウェアであるHasuraを用いた。

上記URLで実行されるGraphQL queryを下記に記載する

```
query getRegionData{
  region_data(order_by: {id: asc}) {
    id
    region_name
  }
}
```



```
{
  "region_data": [
    { "id": 0, "region_name": "警察" },
    { "id": 1, "region_name": "札幌" },
    { "id": 2, "region_name": "函館" },
    { "id": 3, "region_name": "旭川" },
    { "id": 4, "region_name": "網走" },
    { "id": 5, "region_name": "小樽" },
    { "id": 6, "region_name": "釧路" },
    { "id": 7, "region_name": "根室" },
    { "id": 8, "region_name": "青森" },
    { "id": 9, "region_name": "弘前" },
    { "id": 10, "region_name": "八戸" },
    { "id": 11, "region_name": "秋田" },
    { "id": 12, "region_name": "山形" },
    { "id": 13, "region_name": "福島" },
    { "id": 14, "region_name": "茨城" },
    { "id": 15, "region_name": "栃木" },
    { "id": 16, "region_name": "群馬" },
    { "id": 17, "region_name": "埼玉" },
    { "id": 18, "region_name": "千葉" },
    { "id": 19, "region_name": "東京" },
    { "id": 20, "region_name": "神奈川" },
    { "id": 21, "region_name": "新潟" },
    { "id": 22, "region_name": "富山" },
    { "id": 23, "region_name": "石川" },
    { "id": 24, "region_name": "福井" },
    { "id": 25, "region_name": "山梨" },
    { "id": 26, "region_name": "長野" },
    { "id": 27, "region_name": "岐阜" },
    { "id": 28, "region_name": "愛知" },
    { "id": 29, "region_name": "三重" },
    { "id": 30, "region_name": "滋賀" },
    { "id": 31, "region_name": "京都" },
    { "id": 32, "region_name": "大阪" },
    { "id": 33, "region_name": "兵庫" },
    { "id": 34, "region_name": "奈良" },
    { "id": 35, "region_name": "和歌山" },
    { "id": 36, "region_name": "徳島" },
    { "id": 37, "region_name": "高松" },
    { "id": 38, "region_name": "香川" },
    { "id": 39, "region_name": "愛媛" },
    { "id": 40, "region_name": "高知" },
    { "id": 41, "region_name": "大分" },
    { "id": 42, "region_name": "福岡" },
    { "id": 43, "region_name": "佐賀" },
    { "id": 44, "region_name": "長崎" },
    { "id": 45, "region_name": "熊本" },
    { "id": 46, "region_name": "大分" },
    { "id": 47, "region_name": "福岡" },
    { "id": 48, "region_name": "佐賀" },
    { "id": 49, "region_name": "長崎" },
    { "id": 50, "region_name": "熊本" },
    { "id": 51, "region_name": "大分" },
    { "id": 52, "region_name": "福岡" },
    { "id": 53, "region_name": "佐賀" },
    { "id": 54, "region_name": "長崎" },
    { "id": 55, "region_name": "熊本" },
    { "id": 56, "region_name": "大分" },
    { "id": 57, "region_name": "福岡" },
    { "id": 58, "region_name": "佐賀" },
    { "id": 59, "region_name": "長崎" },
    { "id": 60, "region_name": "熊本" },
    { "id": 61, "region_name": "大分" },
    { "id": 62, "region_name": "福岡" },
    { "id": 63, "region_name": "佐賀" },
    { "id": 64, "region_name": "長崎" },
    { "id": 65, "region_name": "熊本" },
    { "id": 66, "region_name": "大分" },
    { "id": 67, "region_name": "福岡" },
    { "id": 68, "region_name": "佐賀" },
    { "id": 69, "region_name": "長崎" },
    { "id": 70, "region_name": "熊本" },
    { "id": 71, "region_name": "大分" },
    { "id": 72, "region_name": "福岡" },
    { "id": 73, "region_name": "佐賀" },
    { "id": 74, "region_name": "長崎" },
    { "id": 75, "region_name": "熊本" },
    { "id": 76, "region_name": "大分" },
    { "id": 77, "region_name": "福岡" },
    { "id": 78, "region_name": "佐賀" },
    { "id": 79, "region_name": "長崎" },
    { "id": 80, "region_name": "熊本" },
    { "id": 81, "region_name": "大分" },
    { "id": 82, "region_name": "福岡" },
    { "id": 83, "region_name": "佐賀" },
    { "id": 84, "region_name": "長崎" },
    { "id": 85, "region_name": "熊本" },
    { "id": 86, "region_name": "大分" },
    { "id": 87, "region_name": "福岡" },
    { "id": 88, "region_name": "佐賀" },
    { "id": 89, "region_name": "長崎" },
    { "id": 90, "region_name": "熊本" },
    { "id": 91, "region_name": "大分" },
    { "id": 92, "region_name": "福岡" },
    { "id": 93, "region_name": "佐賀" },
    { "id": 94, "region_name": "長崎" },
    { "id": 95, "region_name": "熊本" },
    { "id": 96, "region_name": "大分" },
    { "id": 97, "region_name": "福岡" },
    { "id": 98, "region_name": "佐賀" },
    { "id": 99, "region_name": "長崎" },
    { "id": 100, "region_name": "熊本" },
    { "id": 101, "region_name": "大分" },
    { "id": 102, "region_name": "福岡" },
    { "id": 103, "region_name": "佐賀" },
    { "id": 104, "region_name": "長崎" },
    { "id": 105, "region_name": "熊本" },
    { "id": 106, "region_name": "大分" },
    { "id": 107, "region_name": "福岡" },
    { "id": 108, "region_name": "佐賀" },
    { "id": 109, "region_name": "長崎" },
    { "id": 110, "region_name": "熊本" },
    { "id": 111, "region_name": "大分" },
    { "id": 112, "region_name": "福岡" },
    { "id": 113, "region_name": "佐賀" },
    { "id": 114, "region_name": "長崎" },
    { "id": 115, "region_name": "熊本" },
    { "id": 116, "region_name": "大分" },
    { "id": 117, "region_name": "福岡" },
    { "id": 118, "region_name": "佐賀" },
    { "id": 119, "region_name": "長崎" },
    { "id": 120, "region_name": "熊本" },
    { "id": 121, "region_name": "大分" },
    { "id": 122, "region_name": "福岡" },
    { "id": 123, "region_name": "佐賀" },
    { "id": 124, "region_name": "長崎" },
    { "id": 125, "region_name": "熊本" },
    { "id": 126, "region_name": "大分" },
    { "id": 127, "region_name": "福岡" },
    { "id": 128, "region_name": "佐賀" },
    { "id": 129, "region_name": "長崎" },
    { "id": 130, "region_name": "熊本" },
    { "id": 131, "region_name": "大分" },
    { "id": 132, "region_name": "福岡" },
    { "id": 133, "region_name": "佐賀" },
    { "id": 134, "region_name": "長崎" },
    { "id": 135, "region_name": "熊本" }
  ]
}
```

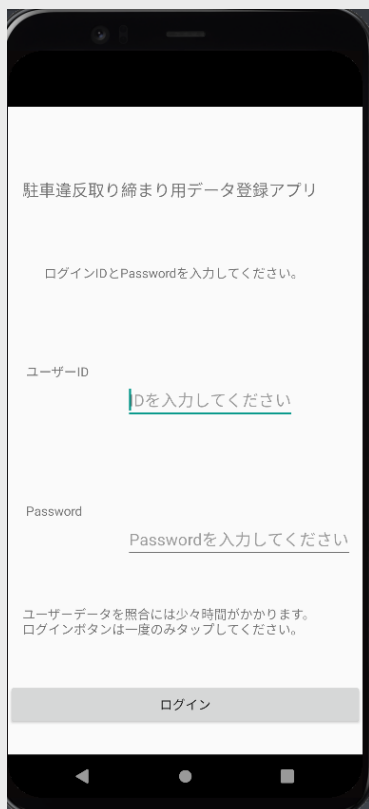
ブラウザで開くとこのようなjsonが取得できる

Androidアプリケーションの画面設計

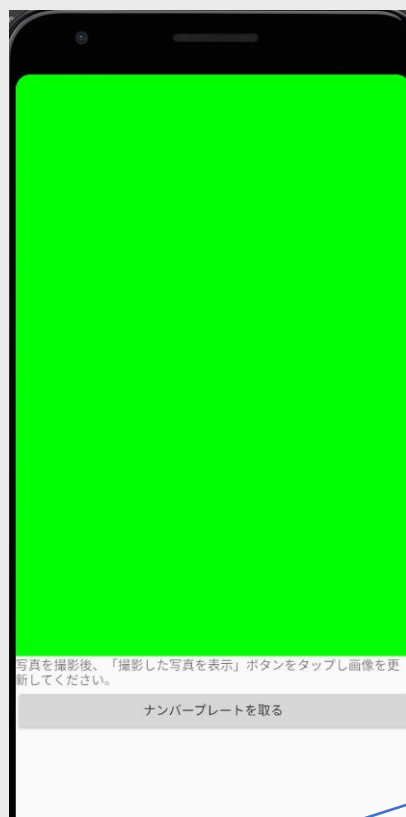
写真



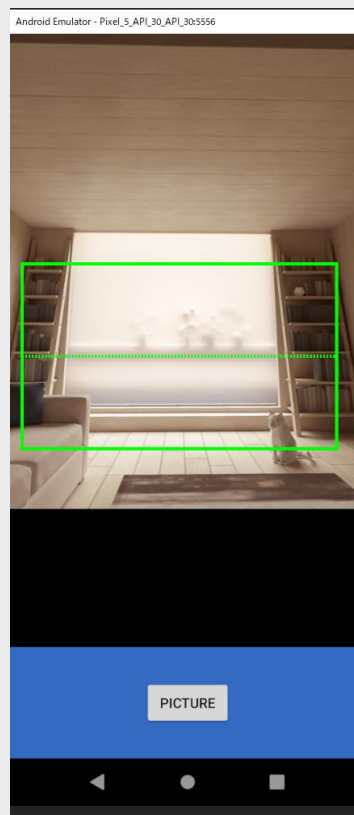
ログイン画面



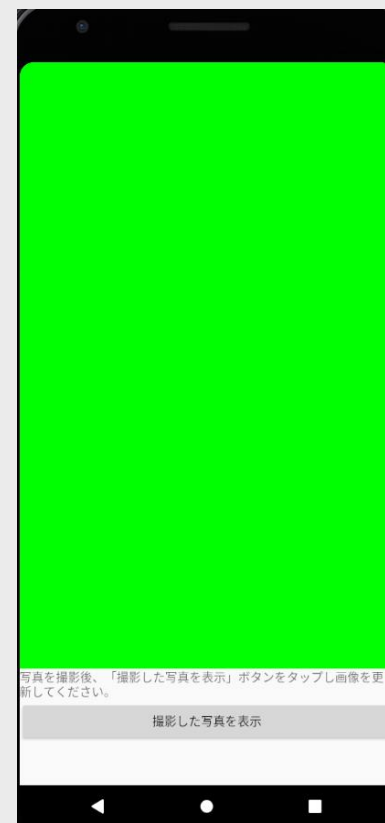
カメラ準備画面



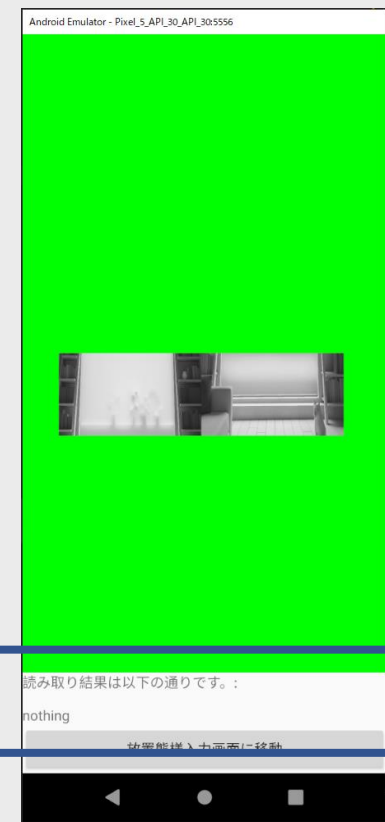
ナンバープレート撮影画面



OCR準備画面



OCR画面



写真を撮る際に、GPS情報と違反日時情報を取得

写真内に文字がないのでnothingと表示されている

Androidアプリケーションの画面設計

ガイドラインにある
選択肢から適した
違反態様を選択
する。

択一形式のため
記入ミスはない。

違反態様入力画面

- ☐ 道路標識により駐（停）車を禁止する場所
- ☐ 停車及び駐車を禁止する場所
- ☐ 法定の駐車禁止場所
- ☐ 無余地場所
- ☐ 停車又は駐車の方法に従わない駐車
- ☐ 時間制限駐車区間（パーキング・メーター設置区間）における駐車
- ☐ 高齢運転者等専用駐車区間における駐車

印刷プレビューに進む

標章画像確認画面

Android Emulator - Pixel_5_API_30_API_305556

放置車両確認標章
(標章番号) 号の使用者 殿
(登録(車両)番号)

駐車違反

速やかに移動してください。

この車は、“放置車両”であることを確認しました。

この車の使用者は、神奈川県 公安委員会から放置違反金の納付を命ぜられることがあります。

なお、この標章が取り付けられた日の翌日から起算して30日以内に、この車を移動し駐車した者がこの違反について反則金の納付をした場合又は公訴を提起され、若しくは家庭裁判所の審判に付された場合は、この限りではありません。

ILLEGAL PARKING 神奈川県 警察署長

This vehicle was identified as "Abandoned" (left unattended) or a "Vehicle left unattended". The user of this vehicle may be subject to the order by the Kanagawa Pref. Police Safety Commission to pay "Abandoned Vehicle" (contraction money).

日時	
違反場所	
状況	
状態	

この車の使用者、運転者その他この車の管理責任者以外の者がこの標章を破壊・汚損し、又は取り壊すと罰せられます。
違反する時は、交通事故防止のため、この標章を取り除いてください。

000000

上記標章画像に問題が無ければ、下にある[PDFを発行する]ボタンを押してください。

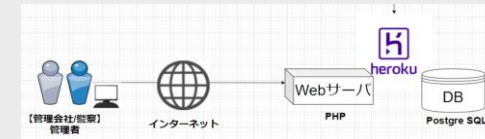
PDFを発行する

違反情報をデータベースに保管



違反車両に貼る標章画像をPDF出力

管理Webサイトの設計



違反情報をデータベースから取得して、検索がかけられるようなWebサイトを作成する。

<https://peteamawebapp.herokuapp.com/>

Login

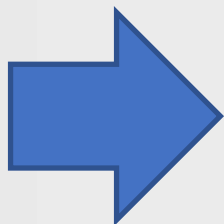
user_id

IDを入力してください

password

passwordを入力してください

LOGIN



駐車違反情報検索サイト

[データ検索](#) [プライバシーポリシー](#) [ご利用規約](#)

データ検索

検索したい項目を下記より選び、検索ボタンをクリックすると該当する結果が検索されます

検索項目

選択してください

検索項目

全件検索

日付検索

単語検索

地方項目

いわき

つくば

とちぎ

なにわ

プライバシーポリシー

◆はじめに

本プロジェクトのサイトは、個人情報の取り扱いにおいて、下記のように定めます。

以下のプライバシーポリシーでは、違反情報やユーザー情報（以下、「個人情報」といいます。）の取得・利用・管理について記載しております。サービス利用にあたり、個人情報保護方針をお読みになり、ご理解いただけますと幸いです。

◆個人情報の利用目的

当サービスでは、違反情報の収集に基づき違反者の車情報を取得させて頂くことがあります。

全件検索

日付検索

地域検索(曖昧検索)

の4種類の検索方法

が実装される。

仮ID: 1 , 仮pass: asdfgh

これで、A2チームの発表を終わります。