

# 日大文理案内アプリ

5422016 墨 泰我, 5422020 池田 悠星, 5422074 馬場 光希, 5422083 古賀 日南乃

October 2024

## 1 作品の目的

我々は、大学案内アプリを作成する。本アプリケーションの目的は、オープンキャンパスや学校見学に訪れる人々や新入生に対し、快適で分かりやすいキャンパス案内を提供することである。

オープンキャンパスや学校見学の訪問者は、広大なキャンパスでどの建物にどの施設があるのかを把握するのが難しい場合が多い。そのため、アプリケーションを活用して現在地に基づくナビゲーションや建物・施設の詳細情報をリアルタイムで提供することで、効率的にキャンパスを見学できるようになると考える。

また、入学したばかりの新入生は、キャンパスの広さや建物の配置に慣れておらず、授業や主要な施設への移動に不安を感じることが考えられる。そのため、アプリケーションの機能を通じて、授業の教室や主要施設の場所をすぐに確認できるようにすることで、新入生がより早くキャンパスに馴染み、安心して大学生活を過ごすことができるようサポートできると考える。

## 2 作品構想

本アプリケーションは、オープンキャンパスや学校見学の訪問者、新入生に向けて、キャンパス案内を行う。ユーザーの現在地は GPS を用いて取得し、現在地から任意の目的地までのルートを表示する。

また、キャンパスツアー機能を搭載しており、予め決められた場所と順序で主要施設を巡るツアーを提供する。オープンキャンパスや学校見学の訪問者に向けて、キャンパス内の主要施設を巡るツアーや、各学科ごとに行われる、新入生に向けたキャンパスツアーを用意し、それぞれに適した案内を行う。

制作に余裕があれば、ユーザーに属性や求めている内容を選択してもらい、それに基づいたツアーを提供する機能を考えている。また、提供するツアーをおすすめツアーとして提示し、編集機能で訪れる順番の並び替えや、訪問先の追加ができるようにしたいと考えている。

本アプリは、誰でも自分のスマートフォンで利用できるよう、iOS と Android の両方に対応することを目指している。

## 3 作品の独創的な点

本アプリケーションの独創的な点は、大学キャンパス内での複数地点を順次案内するルート案内機能にある。多くのキャンパス案内アプリは、ユーザーを現在地から目的地まで最短ルートで案内する機能（以降単一案内とする）を採用している。例えば、立命館大学、早稲田大学、東洋大学、同志社大学の大学アプリケー

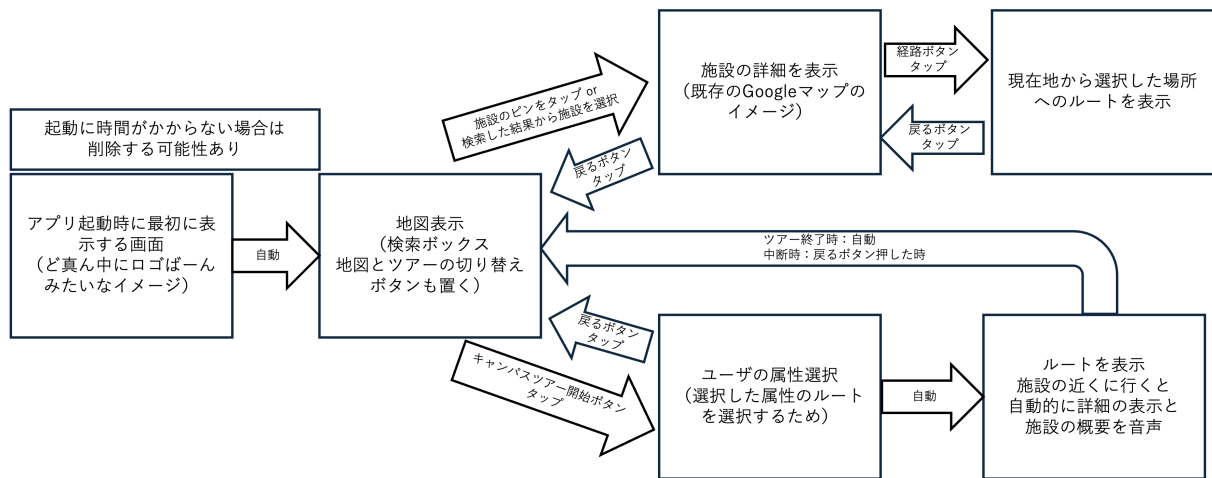


図1 画面遷移フロー

ションではキャンパスを案内する機能として、単一案内を採用していた。

しかし、単一案内ではユーザーが各施設を個別に訪れることになるため、キャンパス全体の構成や関係性を把握することが困難となる。この結果、キャンパスや学科の全体像を掴むことができず、それに対する理解が浅くなってしまう可能性がある。

これに対して本アプリケーションでは、地点1から地点2、地点3と複数の場所を連続して案内する機能（以降順次案内とする）を採用する。この機能により、ユーザーはただ目的地に向かうのではなく、特定の学科に関連する施設や研究室を順に巡ることで、より深くその学科について理解できることが期待される。また、オープンキャンパスや学校見学などに参加する高校生に対しては、学科関連施設だけでなく、学食や体育館、学生課といった主要な施設も順次案内することが可能である。

## 4 詳細

本アプリケーションは React Native と呼ばれる開発用フレームワークを用いて、JavaScript で開発を行う予定である。React Native は iOS と Android の両方の開発を同時に行えるクロスプラットフォームの開発用フレームワークである点を踏まえて選択した。

このアプリケーションは主に、地図表示モードとキャンパスツアーモードの2つの機能を実装する。それぞれの機能の詳細は4.2節および4.3節に記述する。4.1節ではアプリケーションに表示する画面の種類とその関係性について述べる。

### 4.1 画面の種類と関係性

本アプリケーションの画面遷移フローを図1に示す。

まず最初に、起動時はこのアプリケーションのアイコンを表示したのち、地図表示モードとなりキャンパス内の地図や現在地などを表示する。本プロジェクトではこの画面をホーム画面と呼ぶ。次に、任意の施設を選択した際は、施設の詳細を表示する。その後、経路というボタンを押した際に、現在地から選択した施設までの経路を検索し、表示する。施設詳細画面と経路画面では戻るボタンを押した際にひとつ前の画面に遷移

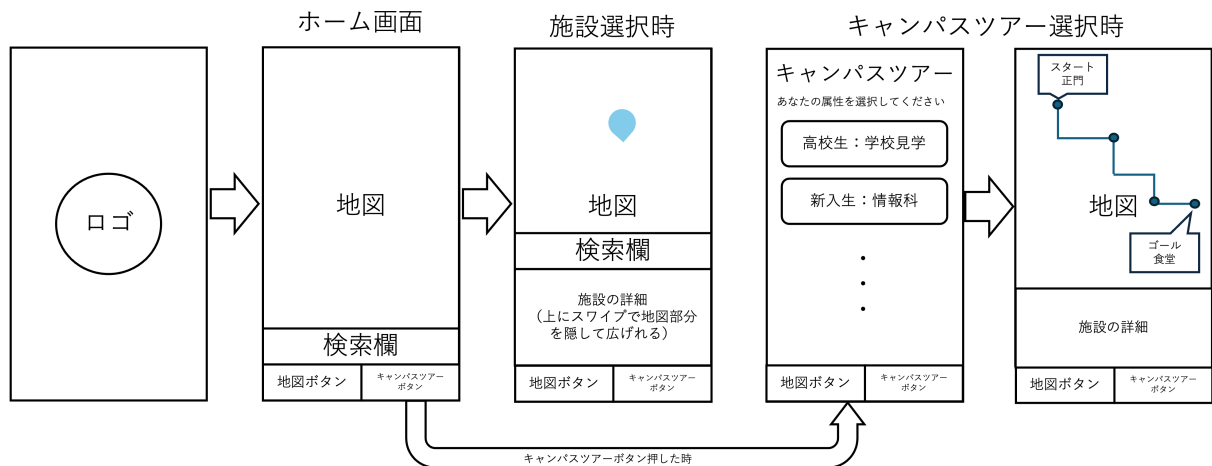


図2 画面設定

する。

ホーム画面には、地図表示モードとキャンパスツアーモードを切り替えられるボタンが配置されており、キャンパスツアーボタンを押した際にキャンパスツアーモードへと遷移する。キャンパスツアーモードでは最初にユーザーの属性を選択する画面を表示する。これはユーザーが高校生なのか新入生なのか、また新入生の場合は何学科なのかを選択することでユーザーの属性に合ったツアーを提供するためである。属性選択後、ルートを表示する。この画面では現在地からツアー開始地点が離れている場合はそこまでの経路も表示する。属性選択画面では戻るボタンを押すことでホーム画面へ遷移する。ツアー中は中断ボタン（バツボタンを想定）を押す、またはゴール地点にユーザーが辿り着いた際にホーム画面へと遷移する。

アプリケーションの具体的な画面設定を図2に示す。これはまだ構想段階であるため、ボタンの配置等のデザインは変更する可能性がある。

## 4.2 地図表示モード

このモードでは、キャンパス内の地図の表示、GPSを用いた現在地の表示、施設検索と詳細の表示、任意の施設や場所までの経路の表示を行う。地図の表示と位置情報と取得に関しては以下のツールから適切なものを選択し開発を進めていく。このモードの機能としては、Googleマップ等の既存のマップアプリと同様のものを想定している。ただ、任意の地点までの経路に関しては、経路の提示までにとどまりその後の誘導は現時点では行わない予定である。

使用するツールは以下のものから適切なものを選択する予定である。

### 1. 地図の表示

- (a) Google Map API : <https://developers.google.com/maps/?hl=ja>
- (b) Mapbox: <https://www.kkc.co.jp/service/blog/mapbox/technical-info/article/7159/>
- (c) 自作の地図

### 2. 位置情報の取得

- (a) Google MAP API
- (b) 位置情報 API : [https://developer.mozilla.org/ja/docs/Web/API/Geolocation\\_API](https://developer.mozilla.org/ja/docs/Web/API/Geolocation_API)

### 4.3 キャンパスツアーモード

このモードでは、ユーザーの属性に応じた適切なキャンパスツアーを提供する。属性として現時点ではオープンキャンパス等で大学に訪問した高校生と情報科学科の新入生の2つを設定し、順次対応する学科を増やしていく予定である。具体的なルートはプロジェクトメンバがそれぞれ作成し、地図に反映させていく。このモードでも地図表示モードと同様の地図と現在地の表示を行う。

仕様

1. アプリケーションの機能（必須）
  - (a) 地図の表示と現在地・キャンパス内の施設にピン
  - (b) キャンパス内で現在地から指定した位置までのルート表示
  - (c) 学科ごとの（まずは情報科の）キャンパスツアー
  - (d) 施設の近くに行った際に音声案内（キャンパスツアー使用時は必須）

## 5 制作計画

### 5.1 各メンバーの役割分担

古賀：

- 基本 UI 構築
  - － ホーム画面の開発本アプリケーション内では、複数の機能を実装することを見越している。各機能へ遷移するが、機能終了後、あるいは中断時にホーム画面へと戻る。基本機能への円滑な誘導を実現する。
- デザイン考案
  - － 各フローごとに用意された画面の UI デザインを担当ボタンの配置や、アプリ全体の統一したデザインを通じて、ユーザーフレンドリーなアプリケーションを実現する。

墨：

- キャンパスツアー機能の開発
  - － ユーザーの属性設定画面の開発利用するユーザーの情報を確認し、各項目に対応する音声を紐付けし、属性とデータを接続する。
  - － 経路図 + 説明画面の開発ユーザーの情報に基づいた音声を再生すると同時に、目的地までの経路図を表示する。

池田：

- 施設選択機能の開発
  - － 施設の詳細説明画面の開発選択された施設の詳細説明を表示し、周辺施設情報等の提示機能を実現する。
- スタート画面の作成アプリ起動時の遷移画面を実現する。

馬場:

- 各機能開発の統括
  - Github 上でのプログラム管理プルリクエストの統括.
- 各機能間の接続プログラム
  - 各機能間のイベントハンドラー作成各機能間の接続を行うイベントハンドラーを作成する.

## 5.2 大まかな制作スケジュール

本プロジェクトの進行スケジュールは以下の通りである.

10 月:仕様決定この月では,プロジェクトの全体的な仕様を確定する.各機能の詳細な設計や動作方針を決め,全体の開発計画を立てる.

11 月:プログラム作成各メンバーが担当する機能のプログラム作成を開始する.UI デザインやユーザー機能など,アプリの基礎となる部分を中心に開発を進める.

12 月上旬:プログラム作成・Web サイト作成プログラム作成を引き続き行いながら,Web サイトの作成にも取り掛かる.各機能の統合や UI の微調整を行い,アプリの全体像を完成させる.

12 月下旬:バグ修正・Web サイト作成実装した機能や画面の動作確認を行い,発見されたバグの修正を行う.同時に,Web サイトの最終調整も行い,プロジェクトの完成度を高める.