

# Visit Aveiro

January 9, 2024

## 1 Project Description

Visit Aveiro is a mobile application that enables users to explore and find diverse locations according to their preferred categories. The app offers an intuitive interface, enabling users to effortlessly search through various places like eateries, historical landmarks, green spaces, and entertainment spots, all neatly categorized for easy navigation. Its standout feature is an immersive camera functionality, which superimposes navigational pointers onto the live view of the user's surroundings, effortlessly leading them to their chosen destination.

## 1.1 App Flow

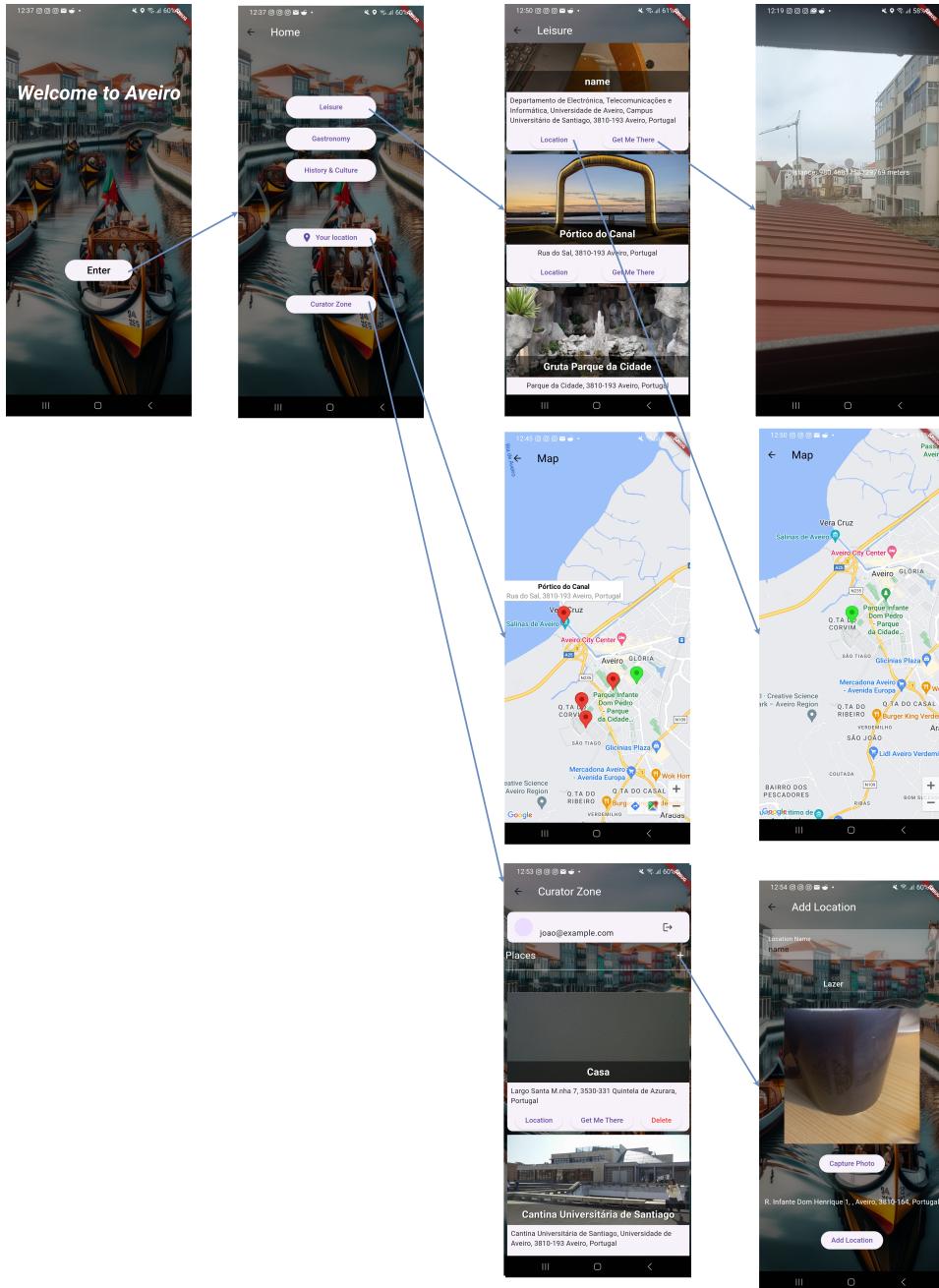


Figure 2: Flow of the app

## 1.2 Technical and Functional Requirements

- **Category-Based Exploration:** Locations are categorized in **Leisure**, **Gastronomy** and **History and Culture**, for facilitating exploration.
- **Location Details:** Within the app, each location is displayed with its photo and address under specific categories. Users can then access the map to view and navigate to the location of interest.
- **Immersive Camera Functionality:** When viewing a location on the map within the app, users can select the 'Immersive Camera' feature for augmented reality-based navigation assistance to their destination.
- **Curator Account:** A person in charge of content management and quality assurance. In other words, has the privilege of **Adding**, **Updating** and **Deleting** a location.
- **Data Management and Storage:** Efficient storage of data locally on the user's device for offline access and seamless data synchronization between the local storage and cloud storage, ensuring data consistency and availability across devices.

## 2 Implemented Solution

### 2.1 Architecture Overview

The app utilizes Firebase as a Backend-as-a-Service (BaaS) solution to streamline the implementation and management of authentication, database services, and storage. For facilitating offline data access and storage capabilities, HiveBox has been employed.

The UI is built using a combination of Flutter widgets and custom-built components. The screens are linked using a combination of Blocs and Providers that manage the state and handle data operations. This setup facilitates effective communication between the UI and the underlying logic and the caching for offline use of relevant data.

For location purposes and map viewing, it was used Google Maps API. The location data is handled by a 'LocationBloc', which employs the Bloc pattern for efficient and responsive location services.

The Immersive Navigation feature integrates the device's camera, location services, accelerometer, and magnetometer. These components work together to determine the user's current position and orientation, as well as the location and distance of Points of Interest in a 2D overlay over a live feed of the device's camera.

Our application revolved around a single model, 'local.dart', that describes all of the location's attributes.

The curator page mandates user authentication, and this authentication needs to be consistent across the app. To solve it, we employed a Provider

as an abstraction, managing communication with Firebase Auth and ensuring consistent authentication state.

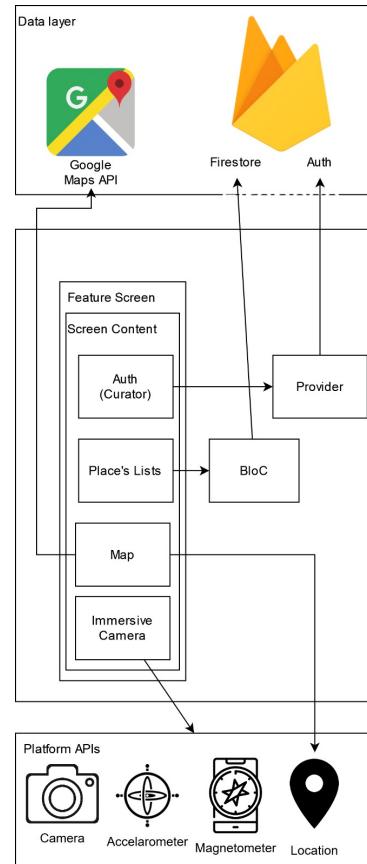


Figure 1: Architecture

### 3 Project Members



João Martins, 93183,  
**MECT**



David Raposo, 93395,  
**MEI**

The objectives for this app were achieved with both members contributing 50

The main UI, item lists, were mainly developed by David while the Curator Screen and authentication were mainly developed by João. Both contributed to app state management solutions (Bloc and Provider).