Oregon State University
College of Engineering
Electrical Engineering and Computer Science

Software Innovation Lab

Master's Project Report

Anush Suresh Kumar

HawkerHub

A community-driven hawker discovery platform

Defended 15th March, 2023

Commencement June, 2023

Abstract

Technology has played a significant role in transforming the way businesses operate in recent years. It has helped businesses reach an extensive consumer base, improve operations, and increase revenue. Thanks to online food-delivery systems, restaurants can now reach customers who may not have been able to visit their brick-and-mortar locations. In developing countries, however, because of the low literacy rate and lack of awareness, the adoption of technology by small businesses, hawkers, and street vendors is slow. These small businesses, street vendors, and hawkers need their community's support to sustain and grow.

HawkerHub is a crowd-sourced, community-driven hawker discovery and indexing mobile application focused on enabling technology for street vendors, hawkers, and small businesses. This application, built on a mobile platform, aims to gather current and accurate information on these vendors to locate them quickly. It is an all-inclusive mobile application with an easy-to-use UI and simple and rapid contribution steps. By adopting these technologies, hawkers can focus on what they do best - serving the community while leaving the rest to technology.

Acknowledgments

I wish to express my deepest gratitude to my adviser, Dr. Will Braynen, for his invaluable guidance and encouragement. He has always been supportive of my efforts and compassionate with me. His wisdom and work ethic will continue to inspire me in both my professional and personal life.

I want to thank my committee members, Dr. Mike Bailey and Dr. Arash Termehchy, for making time in their busy schedules. Their support and feedback have been very valuable.

Thanks to Bikram Pandit for his continuous support and patience during the technical review process.

Finally, I would like to thank my parents, sister, and family for their unconditional love and support. Thank you, everyone.

Table of Contents

List of Figures

1. Introduction

In the last five years, street vendor businesses have grown by about 9% in the US [1]. These numbers are much more significant in developing nations. It was reported in 2022 that about 5 million street vendor establishments exist in India [2]. Hawkers and street vendors sell goods and services on the streets, sidewalks, or other public places. They typically operate without a permanent storefront or shop and may move from location to location to find customers.

Both hawkers and street vendors are often found in urban areas and are essential to the informal economy. They are often associated with selling food, clothing, jewelry, and other small items. They provide necessary goods and services to their local communities and can be an essential source of income for those who participate in this type of work.

Figure 1.1: Arkhipov, Roman. Street Vendors of New York. August 13, 2016. Unsplash. https://unsplash.com/photos/4hxEOapTQGU.

Developing countries face several challenges regarding technology adoption and literacy among hawkers. Many hawkers in developing countries need help accessing computers or smartphones, making it challenging to adopt technology-based solutions. This is particularly true in rural areas, where access to the internet and other technology infrastructure is often limited.

Hawkers and street vendors often suffer from limited literacy and digital skills. They may struggle to adopt new technology solutions without adequate training and support and fully realize their potential benefits.

Even when technology is available, the high cost of devices and services can be a significant barrier to adoption for many small businesses in developing countries. This can be particularly challenging for those with limited financial resources.

Providing access to affordable and user-friendly technology solutions and adequate training and support is vital to overcome the above-mentioned challenges. Local communities need to contribute to enabling and fostering trust in technology among these small businesses, hawkers, and street vendors.

2. Motivation

Support of the local community is essential for its businesses to flourish and sustain. Big establishments have abundant resources and are equipped with the tools and human resources to stay current with technological advancements and cater to the needs current generation. However, small establishments like hawkers, street vendors, and food carts, especially in developing countries, largely depend on their existing customer base to keep their businesses running. They are often under pressure to adopt current trends even though they are not comfortable with its operation and fall prey to its vulnerabilities. One such incident was the Quick Response (QR) code scam in India [3].

In India, the Unified Payments Interface (UPI) has become a popular mode of payment for millions of people. QR code scams are a form of UPI payment fraud where scammers create fake QR codes and trick users into scanning them to make payments. These fake QR codes then redirect users to a payment gateway controlled by the scammers, resulting in the transfer of funds to their accounts. Because UPI is very prevalent, small businesses were forced to incorporate them and fall prey to scams.

On the other hand, consider a situation where people have recently moved to a new area or even tourists exploring the area. They face many challenges, such as language barriers, accessibility issues, trustability, and cost constraints. Hence, these people resort to convenient yet expensive solutions like consulting a moving company to set up their new home or using online delivery platforms to fulfill their needs. Even though it can be convenient, they are not utilizing the cheaper and faster local businesses near them primarily for the above reasons. This situation creates a need for a platform to index local small businesses that are not enabled with technology tools and education to be discovered, providing adequate details to overcome the aforementioned challenges.

Social media platforms are very popular these days. There is a large amount of data about hawkers and street vendors scattered across multiple social media platforms. This information has been shared by social media influences, food bloggers, and content creators.

However, it is hard to consolidate this information as one must follow multiple platforms and subscribe to these specific creators. A single platform where people can contribute is essential for data accuracy and to have the ability to search.

By having a searchable social platform for small businesses like hawkers, street vendors, and food trucks, we can help our local community sustain and grow. Especially in situations like the COVID-19 pandemic, which immensely affected these businesses because they could not adapt to the current situation, it is a strong motivation for such a platform. Residents of Singapore have initiated one such initiative [4].

Singapore is known for its hawker centers, open-air complexes with multiple stalls selling affordable and delicious food. However, the rise of modernization and the pandemic led to the closure of some of these centers. In response, the community mapped out the locations of the remaining hawker centers and stalls in a shared google map so that they could be swiftly discovered.

The mapping efforts began with volunteers who started a “Hawker Mapping” project. The project aims to create a comprehensive map of all hawker centers and stalls in Singapore, including their locations, opening hours, and menus. The information is crowdsourced from the public, who can submit updates and corrections through the project’s website. When writing this report, the project had about 1.5 million views. This demonstrates how the local community can play a crucial role in preserving small businesses in the face of modernization.

3. Existing Solutions

3.1 Google Maps

Google Maps is a multi-platform mapping service launched by Google in 2005. It allows users to search for locations, get driving directions, explores businesses and landmarks, and even view street-level imagery. It has made it easier for businesses to connect with customers by allowing them to create a business profile and manage their online presence. It has benefited establishments that may not have the resources to develop their website or digital marketing strategy. However, in the case of small businesses like hawkers, street vendors, and food trucks, there are some shortcomings.

Limited Visibility: It prioritizes established businesses and well-known landmarks, making it difficult for smaller, informal businesses like street vendors to stand out. This can limit their visibility to potential customers who may not be aware of their presence.

Lack of information: Google Maps typically only provides basic information about a business, such as its name, address, and hours of operation. For street vendors and hawkers, more information may be needed to convey the unique aspects of their business, such as the types of food they sell or the atmosphere of their location.

In some cases, hawkers and street vendors may not have a fixed location and move around frequently, making it difficult to represent them accurately on a map. Additionally, street vendors' types of foods and products can vary widely depending on the culture and location, making providing accurate information about their businesses challenging.

Overall, Google Maps has become an indispensable tool for navigating the world around us, whether exploring a new city or trying to find the nearest coffee shop. Its accuracy, comprehensive mapping data, and user-friendly interface have made it a favorite among millions of people worldwide. Hence, the HawkerHub platform heavily depends on Google Maps, adding more essential features for hawkers.

3.2 Online Delivery Platforms

Online delivery platforms, known as food delivery apps, have become increasingly popular in recent years. These platforms allow users to order food and deliver it to their doorstep, often within minutes. Some popular service providers in the United States include Uber Eats, DoorDash, GrubHub.

These platforms also benefit restaurants by providing them with an additional revenue stream. Restaurants can partner with delivery platforms to expand their customer base and reach a wider audience. Additionally, online delivery platforms often provide marketing and promotional support for their restaurant partners, helping them to increase visibility and attract new customers.

However, there are also some potential drawbacks to online delivery platforms. One issue is these platforms' high fees, which can burden smaller restaurants. Additionally, there are concerns about the working conditions and compensation for delivery drivers, who may not receive benefits or a fair wage.

Online delivery platforms are typically designed for businesses with a fixed location and a set menu. For hawkers and street vendors who may move around frequently and offer a broader range of food options, it cannot be easy to represent their business on these platforms accurately.

Delivery logistics can be difficult for street vendors and hawkers who may not have a dedicated delivery person or vehicle. This can make it challenging for them to offer delivery services, which can limit their potential customer base.

Overall, while online delivery platforms can be a helpful tool for established restaurants and food businesses, they may not be the best option for smaller, informal businesses like hawkers and street vendors. These businesses may need to explore alternative marketing and sales strategies more compatible with their business model and resources.

4. Proposed Solution

A mobile platform built on Google Maps for discovering and collecting data with tailor-made features for small businesses, street vendors, and hawkers. The platform enables local communities to contribute information and aid in the growth of these businesses. The mobile application will have an easy-to-use user interface to minimize time to contribute data.

Figure 4.1: HawkerHub application logo

The HawkerHub platform is scoped for the Android mobile platform and powered by backend services designed and deployed using AWS free tier services to be cost-effective. Further, only the client-side or the customer-side application is prioritized, and business-side or the hawker-side features will be prioritized in later phases (to enable features such as live hawker tracking, verified businesses, and live updates).

5. Technology Stack

HawkerHub is developed using a cross-platform mobile application development framework. This provides scope for future development to deploy on other platforms with less development effort. The application is powered by a serverless backend service hosted using Amazon Web Services (AWS) Lambda and other AWS service offerings. HawkerHub application also uses services from the Google Maps Platform for Geocoding, Geolocation, Places API, and GoogleMaps SDK. The following defines each of these technologies.

5.1 Flutter and Dart

Flutter is an open-source mobile app development framework released by Google in 2017 [5]. It allows developers to build native mobile apps for iOS and Android from a single codebase using the Dart programming language.

Dart is an object-oriented programming language that Google also developed [6]. It was designed to be easy to learn and use, with a syntax similar to other popular programming languages like Java and JavaScript. Dart is known for its fast performance, which makes it a good choice for building high-performance apps.

One of the main benefits of using Flutter and Dart is the ability to build native apps for multiple platforms from a single codebase. This not only saves time and effort for developers but also allows for easier maintenance and updates to the app over time.

Flutter and Dart also offer a rich set of pre-built widgets and tools, which makes it easier for developers to build beautiful and responsive user interfaces. Additionally, Flutter's hot reload feature allows developers to see app changes in real time, speeding up the development process and improving overall productivity.

Overall, Flutter and Dart are potent tools for mobile app development. They offer many benefits, including cross-platform development, fast performance, and a rich set of pre-built widgets and tools.

5.2 Amazon Web Services (AWS)

5.2.1 AWS Serverless Application Model (SAM)

AWS SAM (Serverless Application Model) is a framework for building serverless applications on AWS [7]. It simplifies defining and deploying serverless applications using AWS Lambda, API Gateway, and other AWS services. With AWS SAM, developers can define their serverless applications in a YAML or JSON template and then package and deploy them using the AWS SAM CLI (Command Line Interface) [8]. AWS SAM also provides local development and testing tools, making building and iterating on serverless applications easier before deploying them to production.

5.2.2 AWS S3 (Simple Storage Service)

Amazon S3 (Simple Storage Service) is an object storage service offered by AWS [9]. It is designed to store and retrieve data from anywhere on the web. S3 is widely used for backup and recovery, data archiving, big data analytics, and content delivery. S3 is highly scalable, durable, and secure. It allows users to store objects (files) of any size and access them from anywhere using HTTP or HTTPS protocols. Users can also control access to their objects by defining permissions for each object or bucket (a container for objects) using AWS Identity and Access Management (IAM) policies.

5.2.3 AWS Lambda

AWS Lambda is a serverless computing service provided by AWS [10]. It allows developers to run code without the need to provision or manage servers. With AWS Lambda, developers can create functions that automatically scale to meet demand, run code responding to events, and only pay for the computing time they consume. AWS Lambda supports multiple programming languages, including Node.js, Python, Java, and C#. Developers can trigger their Lambda functions in response to events from other AWS services, such as S3, DynamoDB, and API Gateway. Lambda functions can also be integrated with third-party services and custom workflows.

5.2.4 Dynamo DB

Amazon DynamoDB is a fully-managed NoSQL database service provided by AWS [11]. It is designed to provide fast and predictable performance with seamless scalability. DynamoDB offers a key-value and document data model, which allows users to store and retrieve any amount of data from anywhere in the world. DynamoDB is highly available and durable, providing automatic multi-site replication and backup to help protect against data loss. It also offers flexible querying and indexing options to support various use cases. DynamoDB is often used for web and mobile applications, gaming, IoT, ad tech, and other applications that require low-latency, high-throughput data storage.

5.2.5 Google Maps Platform

Google Maps Platform is a suite of APIs and SDKs offered by Google that enables developers to integrate Google Maps into their applications [12]. Businesses, governments, and individuals widely use the platform for mapping and location-based services. It includes a range of products, such as Maps, Routes, and Places, that allows developers to add maps, directions, and location-based services and supports various programming languages, including Java, Python, and JavaScript, and can be used on platforms like mobile and web.

6. User Interface Design

6.1 Market Survey - Design and Usability Study

A survey is conducted to understand how other applications have designed user interaction around maps. Also, understand the commonality among feature placements that can be adopted into the HawkerHub application.

The survey will analyze the following characteristics among our several applications:

- The landing page
- Features on the landing page
- Search Interaction
- Placement of the search feature
- Rendering results
- Map Interaction
- Interactive features (Eg: Zoom In/Out, Home location)

Note: The survey is conducted in the context of Android applications only.

List of surveyed applications:

- Google Maps [13]
- Waze [14]
- Uber [15]
- Lyft [16]

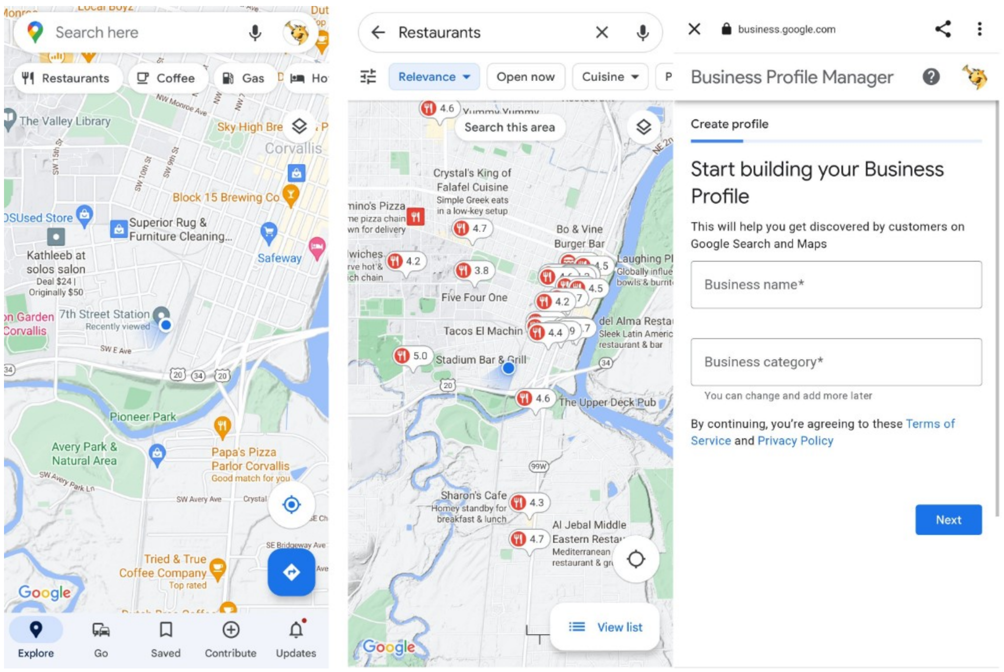

Search here
Restaurants
Coffee
Gas
Explore
Go
Saved
Contribute
Updates
Restaurants
Relevance
Open now
Cuisine
Search this area
View list
business.google.com
Business Profile Manager
Create profile
Start building your Business Profile
This will help you get discovered by customers on Google Search and Maps
Business name*
Business category*
You can change and add more later
By continuing, you're agreeing to these Terms of Service and Privacy Policy
Next

Figure 6.1: Google Maps application screenshots

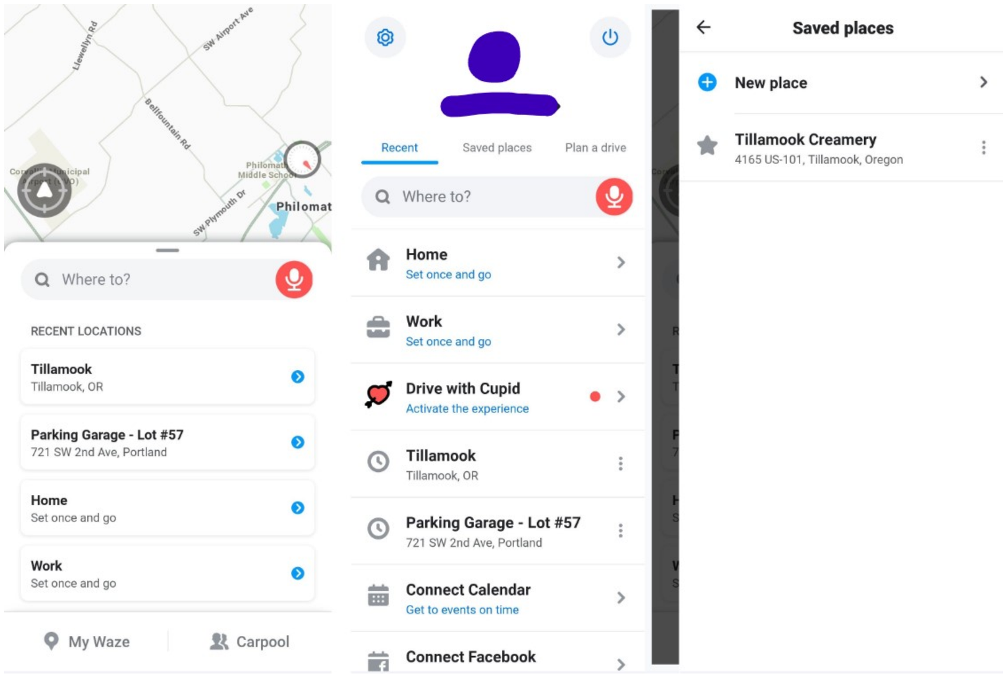


Figure 6.2: Waze application screenshots

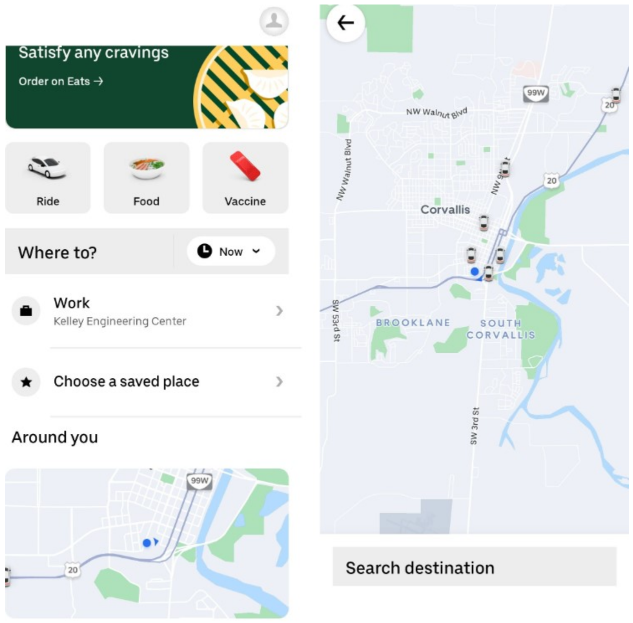


Figure 6.3: Uber application screenshots

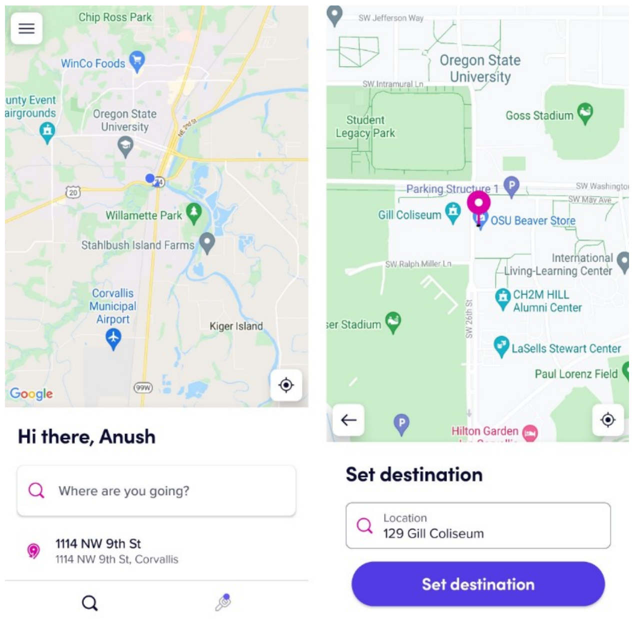


Figure 6.4: Lyft application screenshots

6.1.1 The Landing Page

In most of the applications, the screen is rendered with maps with a current location at the center of the screen and zoomed in to cover a certain radius of the area.

The majority provided the following features:

1. Search Bar
2. Current Location Button
3. Get Directions Button
4. Quick links to explore certain categories (Eg: Restaurants)
5. Bottom bar above the navigation bar with other options

6.1.2 Search Interaction

On using the search bar in the majority of the applications, it lists the user's recent search history and accepts input. And, on interacting with the quick search buttons (Category based), it opens are drawer of results and also plots the result on the map with their ratings on the location pin.

6.1.3 Map Interaction

Google Maps was the most user-friendly and feature-rich of all the applications surveyed. The list of features is as follows:

- It provided pinch to zoom In/Out functionality.
- Auto-focuses to map while interacting (full screen), hides other options which use maps.

- Auto reset to current location button is provided.
- Double tap zooms into the map.
- Tapping on a location opens relevant information if available (like a business).
- Provides multiple terrain views.

In the case of Uber and Lyft applications, they did not provide multiple terrain views, and tapping on a specific location on the map had no action.

Based on the above analysis and commonalities, HawkerHub mockups were developed using Figma.

6.2 Figma Prototypes

Figma is a cloud-based design and prototyping tool used for creating user interfaces, web and mobile applications, and other digital products. Additionally, Figma provides developers with easy access to design specifications and assets, streamlining the handoff process. HawkerHub application was designed using Android's Material 3 design specification [17].

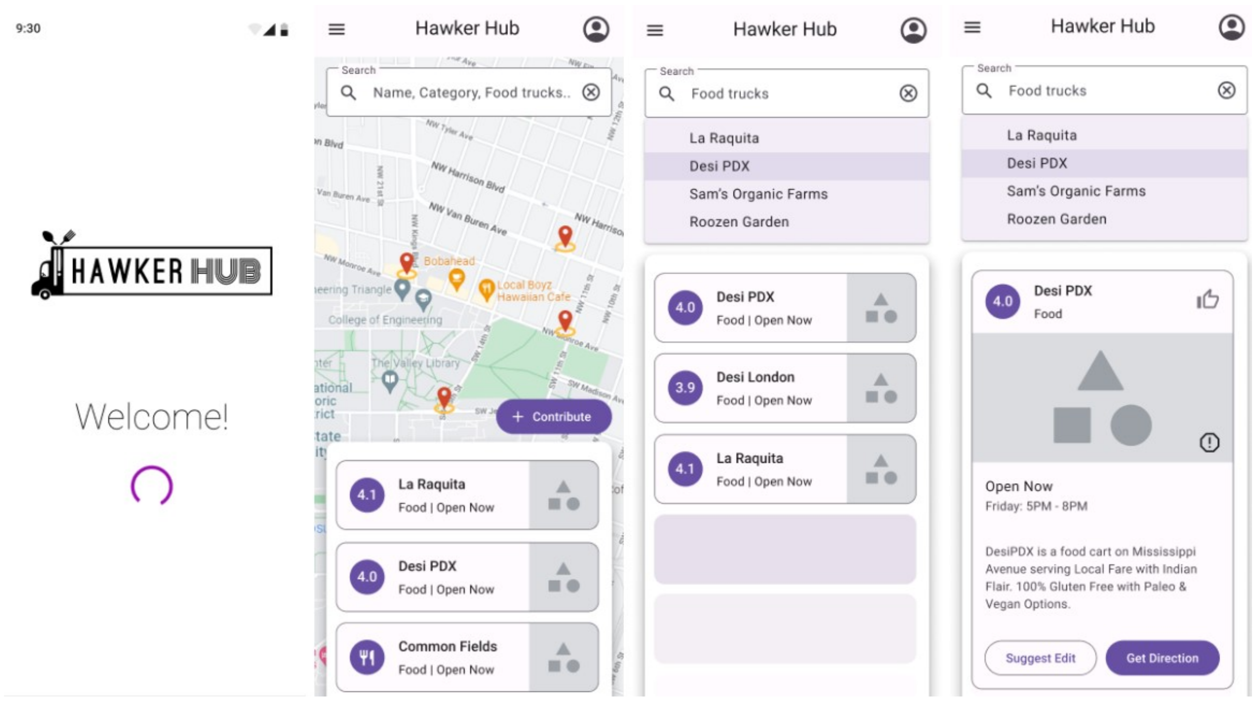
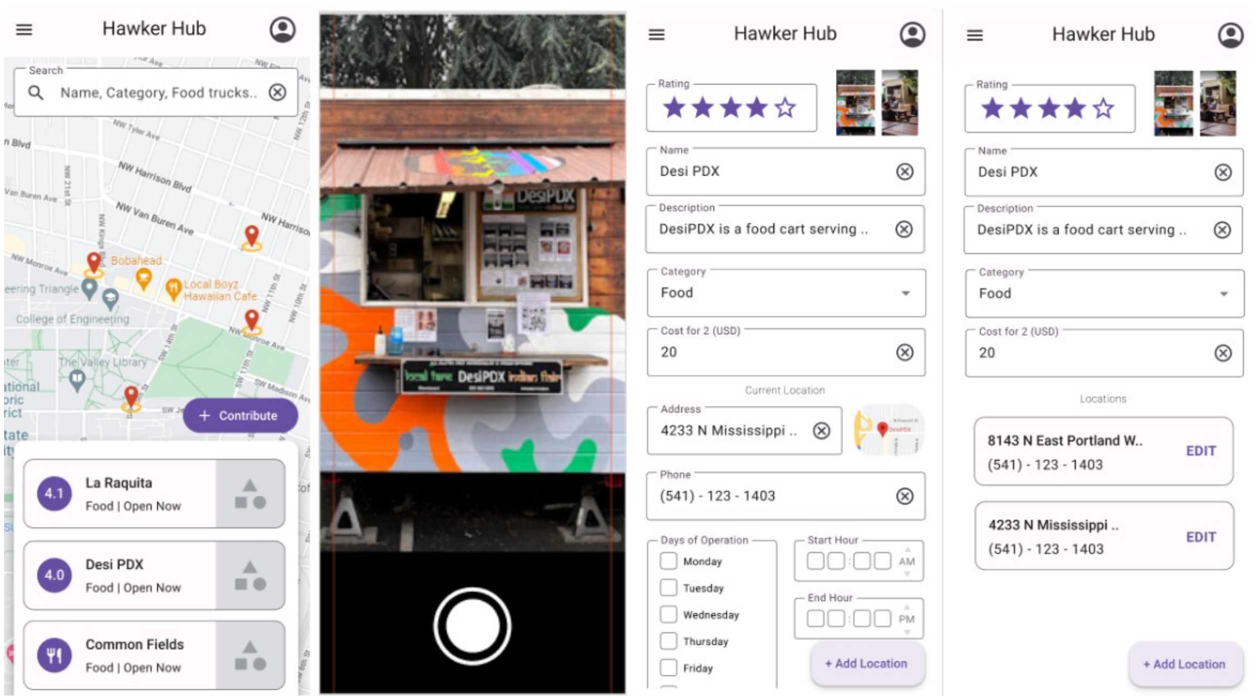


Figure 6.5: Landing page and Hub exploration flow



Search: Name, Category, Food trucks..

La Raquita 4.1 Food | Open Now

Desi PDX 4.0 Food | Open Now

Common Fields Food | Open Now

Rating: ★★★★☆

Name: Desi PDX

Description: DesiPDX is a food cart serving ..

Category: Food

Cost for 2 (USD): 20

Address: 4233 N Mississippi ..

Phone: (541) - 123 - 1403

Days of Operation: Monday, Tuesday, Wednesday, Thursday, Friday

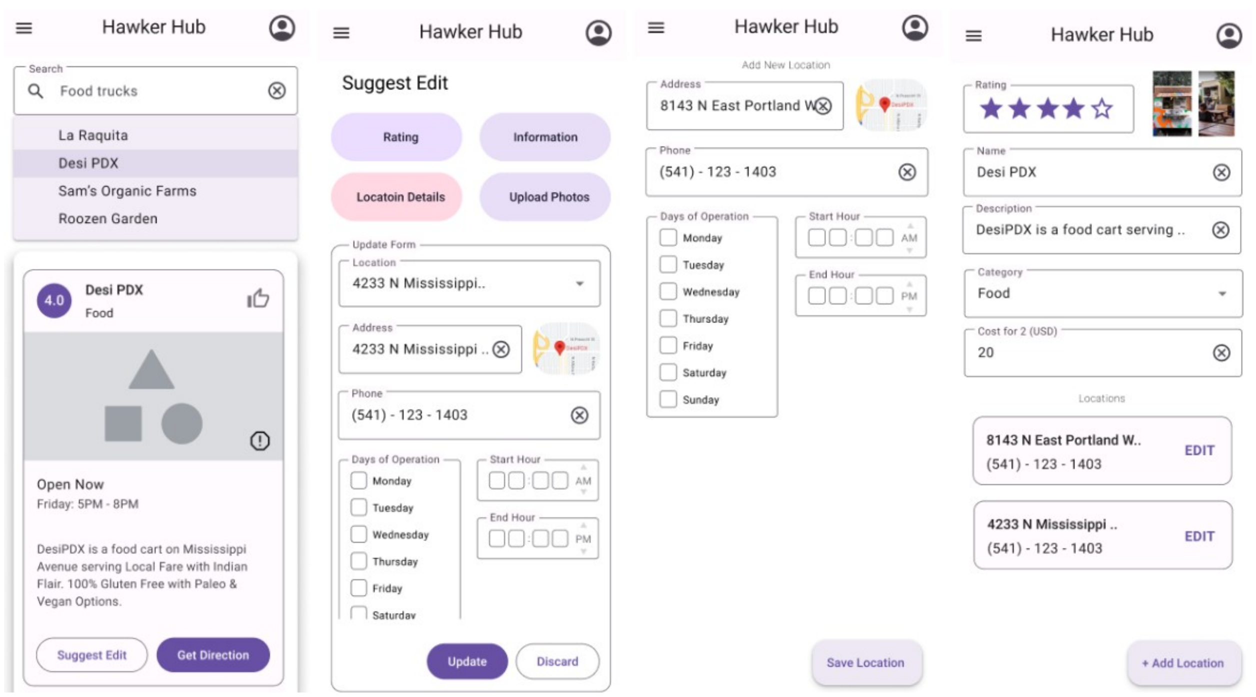
Start Hour: AM

End Hour: PM

8143 N East Portland W.. (541) - 123 - 1403

4233 N Mississippi .. (541) - 123 - 1403

Figure 6.6: Hub contribution flow



Search: Food trucks

La Raquita

Desi PDX

Sam's Organic Farms

Roozen Garden

Desi PDX 4.0 Food

Open Now

Friday: 5PM - 8PM

DesiPDX is a food cart on Mississippi Avenue serving Local Fare with Indian Flair. 100% Gluten Free with Paleo & Vegan Options.

Location: 4233 N Mississippi..

Address: 4233 N Mississippi ..

Phone: (541) - 123 - 1403

Days of Operation: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

Start Hour: AM

End Hour: PM

Address: 8143 N East Portland W

Phone: (541) - 123 - 1403

Days of Operation: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

Rating: ★★★★☆

Name: Desi PDX

Description: DesiPDX is a food cart serving ..

Category: Food

Cost for 2 (USD): 20

8143 N East Portland W.. (541) - 123 - 1403

4233 N Mississippi .. (541) - 123 - 1403

Figure 6.7: Hub update flow

7. System Architecture

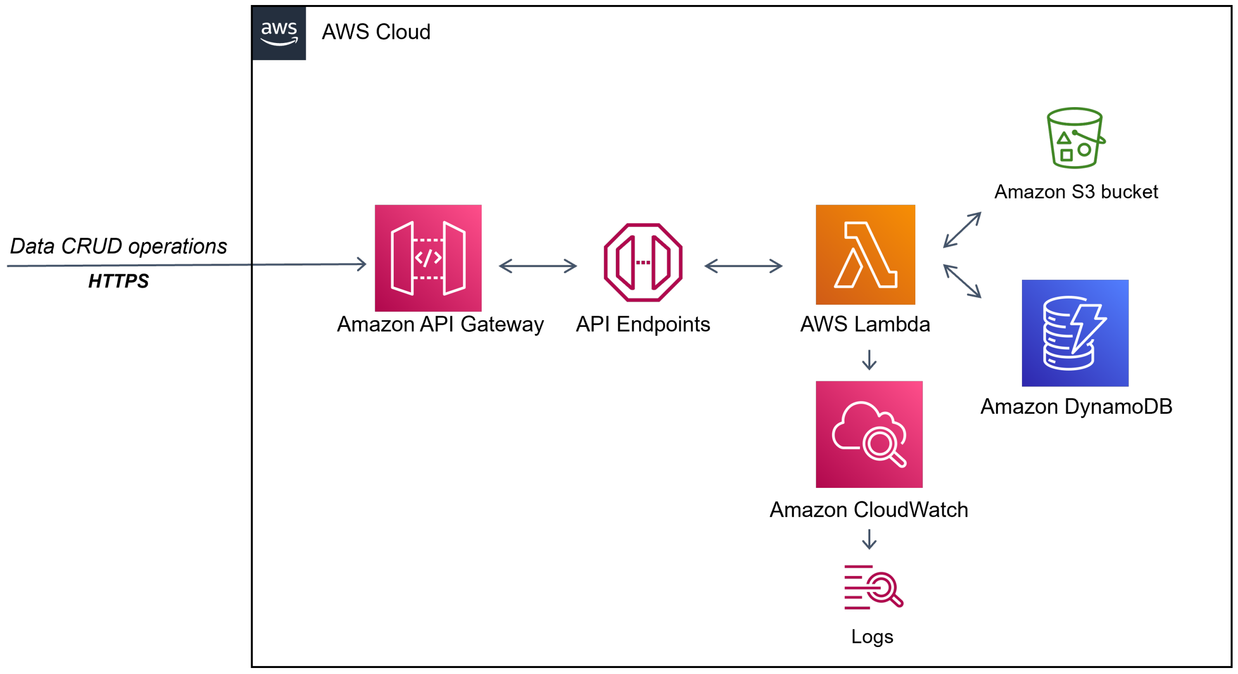


Figure 7.1: System architecture

Request and Response Flow (Creating a Hub)

1. The client initiates an HTTP POST request with a multipart content type (multipart/form-data).

   This request consists of 2 parts.

   (a) Files: Binary, A Hub image/photo

   (b) Body: JSON, Hub details

2. Amazon API Gateway receives the request, configured to accept only certain HTTP verbs and endpoints.

3. The API Gateway then invokes an AWS Lambda function associated with the respective REST service. The Gateway is also responsible for deserialization the request.

4. The AWS Lambda function receives the request details and payload as an event context which is then validated, processed, and the business logic is executed.

5. In the case of creating a Hub, the file (Hub photo) in the request is stored in an AWS S3 bucket, and in return, a public URL is returned.

6. Further, the JSON data is inserted in the respective AWS DynamoDB table, along with the S3 object's public URL returned from the previous step.

7. The final JSON record with the updated public URL is then returned to the client.

Note: Logging and alerting are enabled for traceability and maintenance at each step.

7.1 Resources

7.1.1 Image Resources

Images are stored in S3 bucket and are transported as part of HTTP multipart requests, which allow binaries as files in the request.

Note: The images are compressed at the clientside to save network round-trip and also save on storage costs.

7.1.2 Data Resources

Data is stored as JSON objects in DynamoDB and transported in the HTTP multipart request body. These data are non-relational and suitable for object stores. AWS DynamoDB is chosen because of its cost in the free tier.

7.2 Frontend Architecture

7.2.1 Model-View-ViewModel

Model-View-ViewModel or MVVM is a software design pattern that separates the presentation layer from the business logic and data access layers. It is a popular pattern used in mobile application development, including Flutter.

In Flutter, developers often use the MVVM pattern in combination with the provider package. The provider package is a state management library that provides a convenient way to manage application states across multiple widgets and screens.

The Model layer represents the business logic and data access layer, while the View layer represents the UI. The ViewModel layer acts as an intermediary between the Model and the View layers, providing data and functionality to the UI.

With provider, the ViewModel layer is responsible for managing the application state and providing it to the View layer. It exposes data and functions through a provider object, which the View layer can access.

The View layer interacts with the ViewModel layer through the provider object, requesting data and triggering functions as needed. When the ViewModel layer updates the state, it notifies the View layer through the provider object, triggering a rebuild of the UI.

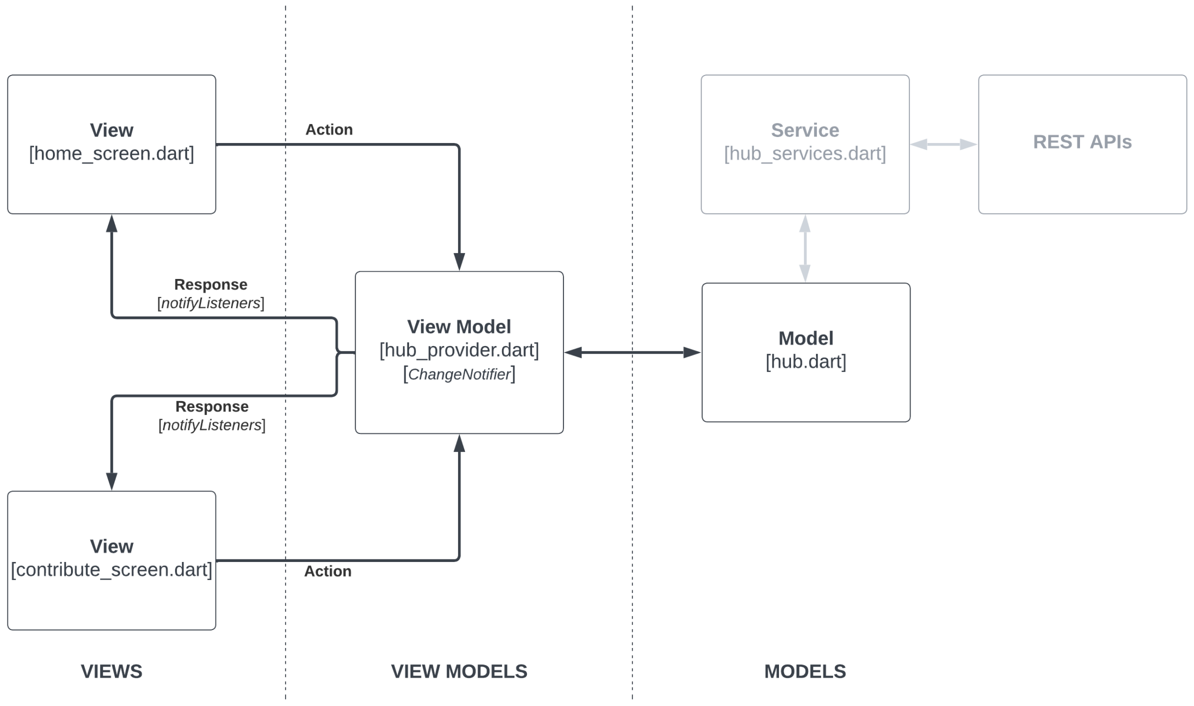


Figure 7.2: MVVM example

7.2.2 Provider State Management

The provider manages the state in a Flutter application [18]. It is built on top of the *InheritedWidget*, which is a widget that allows data to be passed down the widget tree. Providers are a mechanism for sharing data between different parts of the app more efficiently than manually passing data down the widget tree.

The provider state management works by creating a provider that holds the state of the app and then using that to share the state with other widgets in the app. Providers are typically defined as classes that extend the *ChangeNotifier* class, which provides a mechanism for notifying listeners when the state of the provider changes.

When a provider is created, it is added to the widget tree as a parent of the widgets that need access to the state. Widgets can then access the state using the *Provider.of()* method to retrieve it from the widget tree.

Once a widget has access to the provider, it can use the state stored in the provider to render itself. If the state in the provider changes, the provider notifies its listeners (i.e., the widgets that are listening for changes), and the widgets can update themselves accordingly.

In the HawkerHub application, we use only one provider, the *hub_provider.dart*. The whole application's state is stored in this provider. The application is built under the assumption that all the hub details retrieved from the backend are stored in the state, and it depicts the true values (as there are cases where the details are updated).

7.3 Backend Architecture

7.3.1 AWS Lambda Function

HawkerHub's backend is deployed using AWS serverless architecture (NodeJS SDK) and other AWS offerings. Each REST service is defined as a separate AWS Lambda function so that they can be scaled individually without affecting the other service and are cost-effective.

Lambda Functions: getAllHubsFunction, getHubByIdFunction, insertHubFunction, updateHubFunction, deleteHubByIdFunction.

Each Lambda function depends on an AWS S3 bucket and AWS DynamoDB table for data. An IAM role controls their access, which provides fine-grained control over the resources.

```
insertHubFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: src/handlers/insert-hub.insertHubHandler
      Runtime: nodejs16.x
      Architectures:
        - x86_64
      MemorySize: 128
      Timeout: 100
      Description: Function to insert one hub to a DynamoDB table.
      Policies:
        - DynamoDBCrudPolicy:
            TableName: !Ref HubsTable
        - S3CrudPolicy:
            BucketName: hubs-image
      Environment:
        Variables:
          HUBS_TABLE_NAME: !Ref HubsTable
      Events:
        Api:
          Type: Api
          Properties:
            Path: /hub
            Method: POST
```

Listing 7.1: Definition of the insert hub API's lambda function

7.3.2 AWS S3 Bucket

The S3 bucket stores and retrieves each Hub's image. When registering a Hub, its photo is stored in the S3 bucket, and an S3 public URL is updated in its DynamoDB record. Subsequently, the client application uses the S3 public URL to retrieve the photo. Using these public URLs to serve the images, we can offload additional calls to Lambda functions, saving additional costs and maintenance.

```
HubsImageBucket:
      Type: AWS::S3::Bucket
      Properties:
        BucketName: hubs-image
        AccessControl: PublicRead
```

Listing 7.2: Definition of the S3 bucket

7.3.3 AWS DynamoDB

DynamoDB is a highly scalable datastore available as part of the AWS free tier and always free in limited capacity [19]. DynamoDB is the primary data store of the application. The application's data modeling is done to fit the use cases of a NoSQL object store. Each Hub is stored in a JSON record identified by a unique *hub_id* field.

```
HubsTable:
    Type: AWS::Serverless::SimpleTable
    Properties:
      PrimaryKey:
        Name: hub_id
        Type: String
      ProvisionedThroughput:
        ReadCapacityUnits: 2
        WriteCapacityUnits: 2
```

Listing 7.3: Definition of the DynamoDB table

7.3.4 AWS Serverless Application Model (SAM)

The whole backend is deployed using the SAM framework. This framework allows using Infrastructure as code by defining required resources and policies in a YAML file. Each AWS service mentioned above is configured in the YAML file and deployed using SAM CLI. It internally uses AWS CloudFormation, which is an infrastructure as code (IaC) service that allows you to easily model, provision, and manage AWS and third-party resources. SAM can deploy into multiple regions using different deployment strategies.

The configuration file for the HawkerHub application can be found here (GitHub).

7.3.5 Deployment Details

AWS Details

AWS Region: us-west-2
Stage Environment:
`https://o7bd36fp29.execute-api.us-west-2.amazonaws.com/Stage/hub/`
Production Environment:
`https://o7bd36fp29.execute-api.us-west-2.amazonaws.com/Prod/hub/`

Package Versions

Flutter: 3.3.7
Dart: 2.18.4
DevTools: 2.15.0
Android Minimum SDK Version: 20

8. API Documentation

8.1 RESTful APIs

8.1.1 Search APIs

- Get all Hubs
  - [**GET**] ***/hub***
  - Returns a JSON array of all the hubs in the system.
  - [**GET**] ***/hub/{hub_id}***
  - Returns a JSON of the specified hub ID.

8.1.2 Create API

- Add a Hub
  - [**POST**] ***/hub***
  - Content-type: multipart/form-data
  - Creates and returns the JSON of the newly created hub.

8.1.3 Update API

- Update a Hub
  - [**PUT**] ***/hub***
  - Content-type: multipart/form-data
  - Updates an existing Hub and returns the JSON of the updated hub.

8.1.4 Delete API

- Delete a Hub by id
  - [**DELETE**] ***/hub/{hub_id}***
  - Deletes a Hub and returns the status of the operation.

8.2 Swagger Document

Swagger is a popular tool for documenting APIs. It provides a standardized way to describe an API's endpoints, parameters, and responses. Swagger documentation can be generated automatically based on the API code, making it easy to keep documentation up-to-date. Swagger also provides a web-based user interface for exploring and testing the API, which can be helpful for developers who are integrating with the API.

The swagger YAML file for the HawkerHub application can be found here (GitHub).

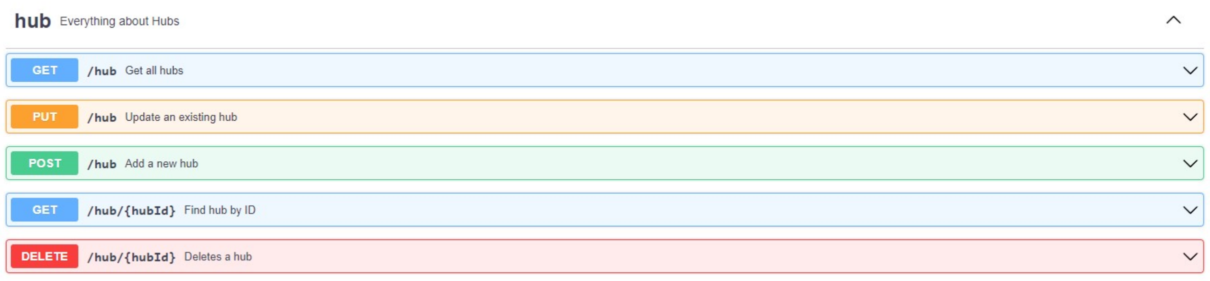

hub Everything about Hubs
GET /hub Get all hubs
PUT /hub Update an existing hub
POST /hub Add a new hub
GET /hub/{hubId} Find hub by ID
DELETE /hub/{hubId} Deletes a hub

Figure 8.1: Swagger document

9. Testing

9.1 Unit Testing

Unit testing is a testing technique where individual functions or methods of the code are tested in isolation from the rest of the application [20]. It involves writing test cases for each function or method and verifying that it produces the expected output for a given input. The primary goal of unit testing is to detect and fix bugs early in the development cycle, making the code more reliable and easier to maintain.

Flutter provides a built-in framework called Flutter Test, allowing developers to write and run unit tests for their Flutter applications. Flutter Test provides APIs that make it easy for developers to write and run tests. The framework allows developers to write tests in either the Dart programming language or Flutter Widget Tests.

Mocking dependencies is an important aspect of unit testing in Flutter. It allows you to isolate the code being tested from external dependencies such as databases, network requests, and other external services. This ensures that the unit tests focus only on the specific behavior of the code being tested, without the interference of external factors.

HawkerHub uses the *mockito* framework to create mock instances of dependent classes. The mocked classes are then injected into the test environment via constructor-based dependency injection. Flutter provides helpful packages such as build_runner, which generates these mock classes.

A test class name in Flutter ends with the "test" keyword, and mocks that need are generated ".mocks.dart" appended to the filename. Further, classes to be mocked are declared using annotation *@GenerateMocks([HttpClient, XFile]).* The mocked classes have a prefix, for example, *MockHttpClient*, which is then used in the tests.

9.2 Widget Testing

Widget testing is a testing technique where developers write tests to verify the functionality and behavior of their UI components, such as buttons, input fields, and widgets [21]. It involves simulating user interactions with the UI components and verifying that they behave as expected. The primary goal of widget testing is to ensure that the UI components are working correctly, free of errors, and providing a smooth user experience.

Flutter Test package provides several methods for widget testing, including.

- pumpWidget: This method builds and displays the widget on the screen, allowing developers to simulate user interactions with the UI components.
- tester.tap: This method simulates a user tapping on the UI component, allowing developers to test the component's behavior.
- tester.enterText: This method simulates a user entering text into an input field, allowing developers to test the field's behavior.
- expect: This method verifies that the UI component behaves as expected, allowing developers to identify and fix UI-related issues.

In conclusion, unit and widget testing is a crucial technique in Flutter development to ensure that the UI components are working as intended and error-free. Flutter Test provides a built-in framework for developers to write and run application widget tests. By following best practices for widget testing, developers can write effective tests and ensure that their UI components are robust, reliable, and provide a smooth user experience.

HawkerHub mobile application has unit and widget test coverage of about **70%**.

10. Application Screenshots

This section presents screenshots of all the screens involved in the various flows. The figure caption provides a brief explanation of the components on the screen.

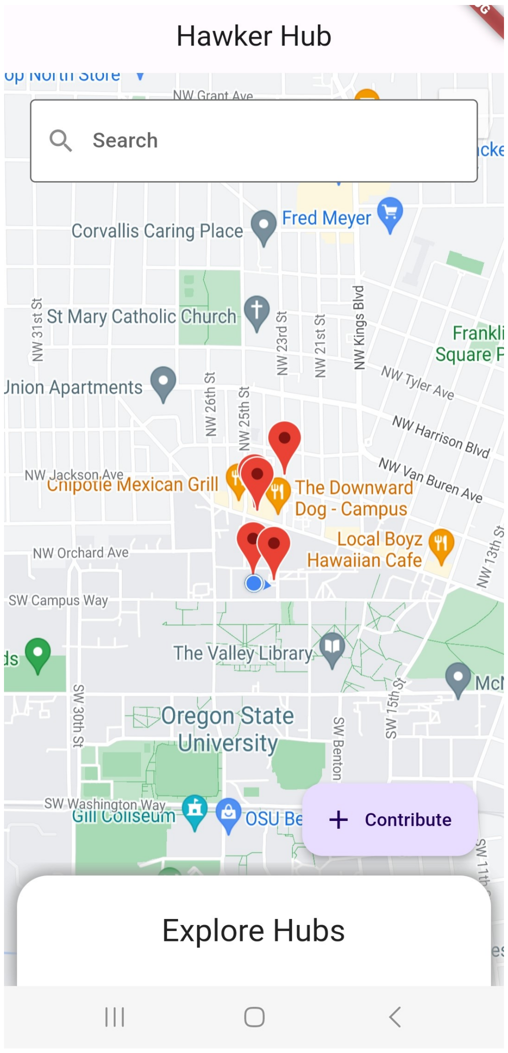
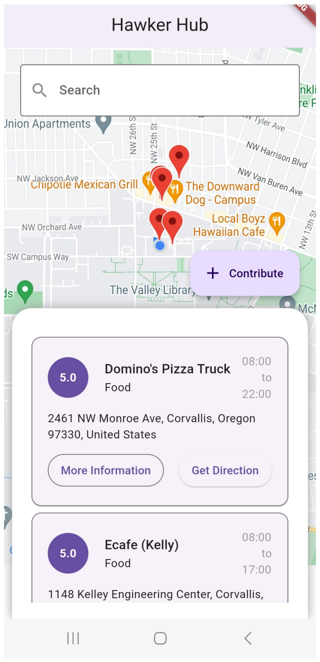


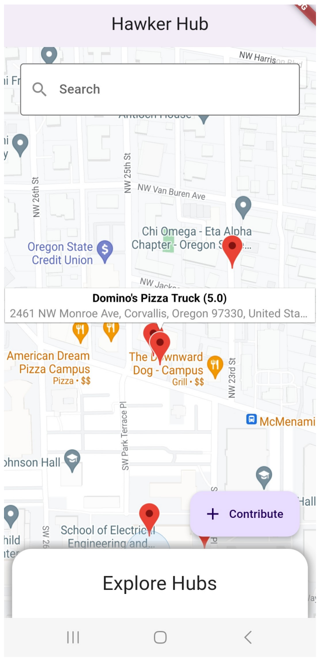
Figure 10.1: Landing page, the first screen presented to the user, contains the search bar, the contribute button, and a swipe-up drawer.



5.0 Domino's Pizza Truck
Food
08:00 to 22:00
2461 NW Monroe Ave, Corvallis, Oregon 97330, United States

5.0 Ecafe (Kelly)
Food
08:00 to 17:00
1148 Kelley Engineering Center, Corvallis,

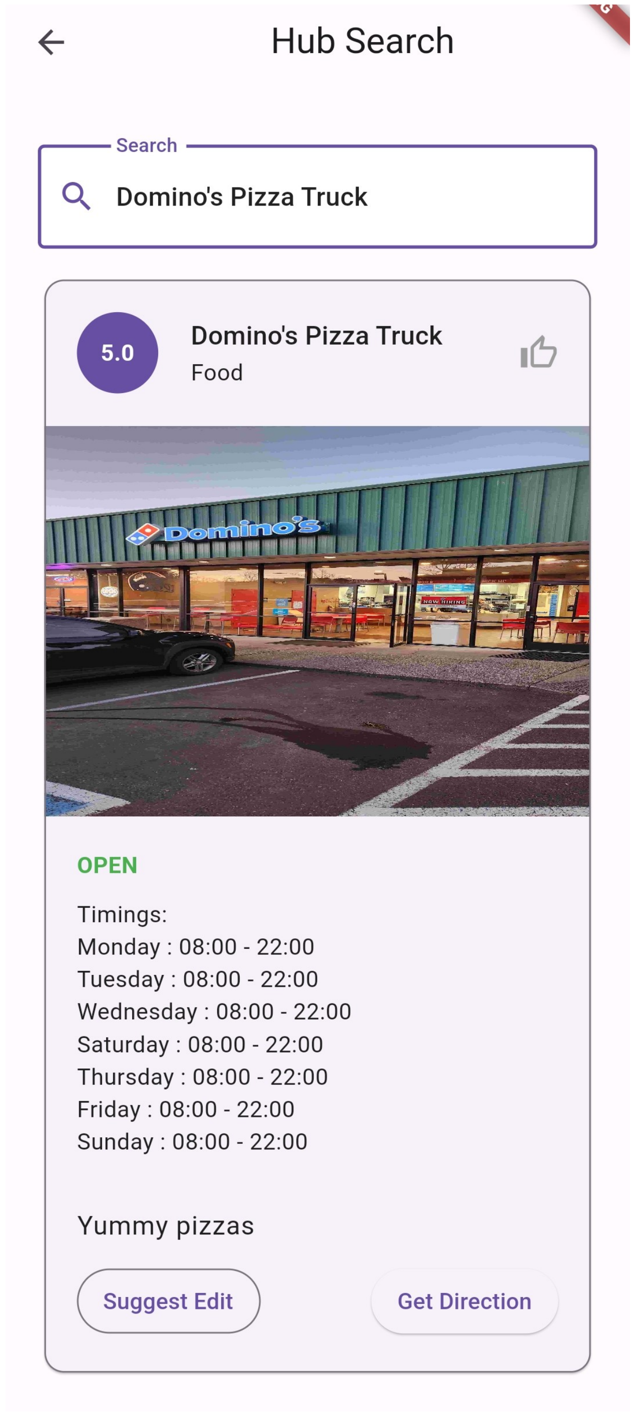
(a) Hubs exploration drawer, a swipe-up Hub exploring drawer that provides more information about Hubs in the area.


Hawker Hub
Search
Oregon State Credit Union
Chi Omega - Eta Alpha Chapter - Oregon S...
American Dream Pizza Campus
Pizza · $$
The Downward Dog - Campus
Grill · $$
McMenami
Johnson Hall
School of Electrical Engineering and
Contribute
Explore Hubs

Domino's Pizza Truck (5.0)
2461 NW Monroe Ave, Corvallis, Oregon 97330, United Sta...

(b) Tooltip, information of the Hub is overlayed using a tooltip when the location icon is clicked.

Figure 10.2: Landing page Interactions


Hub Search

Search

Domino's Pizza Truck

5.0

Domino's Pizza Truck

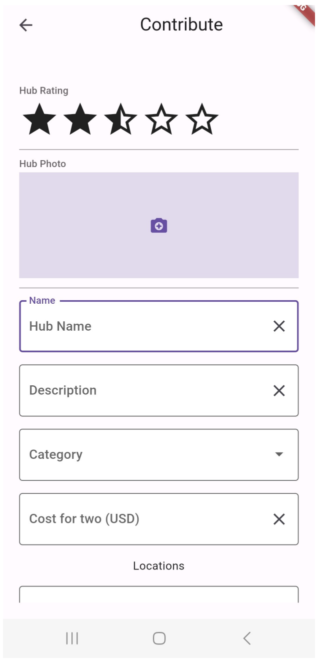
Food

OPEN

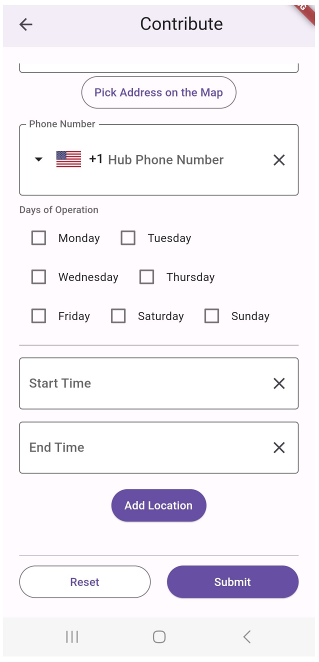
Timings:
Monday : 08:00 - 22:00
Tuesday : 08:00 - 22:00
Wednesday : 08:00 - 22:00
Saturday : 08:00 - 22:00
Thursday : 08:00 - 22:00
Friday : 08:00 - 22:00
Sunday : 08:00 - 22:00

Yummy pizzas

Figure 10.3: Hubs vertical card, more information about the Hub is represented using the vertical card. The screen shows the search interaction.

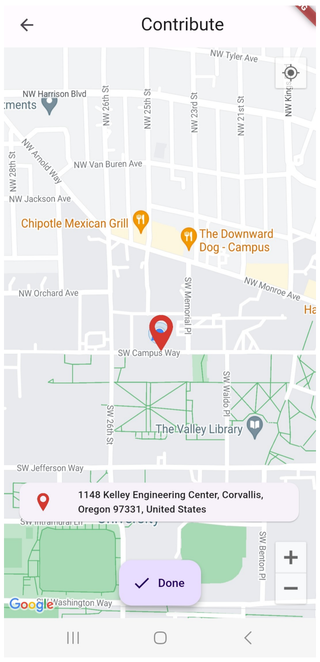


(a)

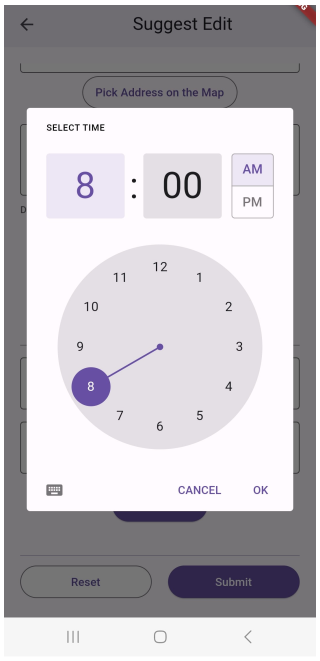


(b)

Figure 10.4: Hub contribution screen, collects all the details required with support to add multiple locations.

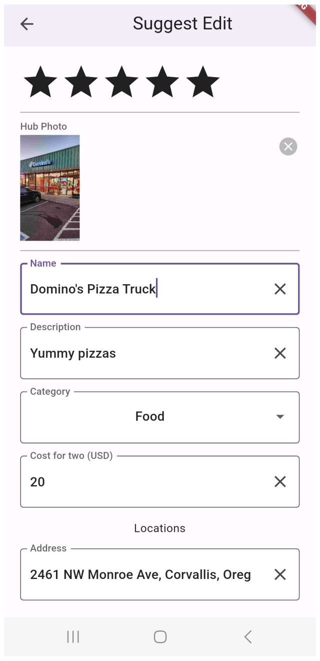

Contribute
Chipotle Mexican Grill
The Downward Dog - Campus
The Valley Library
SW Campus Way
1148 Kelley Engineering Center, Corvallis, Oregon 97331, United States
Done

(a) Address Picker



(b) Time Picker

Figure 10.5: Address picker is used to pinpoint the location of the Hub. The time picker is used to record the working hours of the Hub.


Suggest Edit

★★★★★

Hub Photo

Name: Domino's Pizza Truck

Description: Yummy pizzas

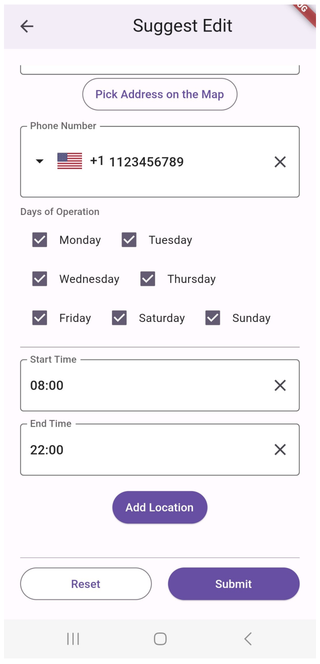
Category: Food

Cost for two (USD): 20

Locations

Address: 2461 NW Monroe Ave, Corvallis, Oreg

(a)


Suggest Edit

Pick Address on the Map

Phone Number: +1 1123456789

Days of Operation

☑ Monday ☑ Tuesday

☑ Wednesday ☑ Thursday

☑ Friday ☑ Saturday ☑ Sunday

Start Time: 08:00

End Time: 22:00

Add Location

Reset Submit

(b)

Figure 10.6: Suggest edit screen, similar to Contribution screen, but with pre-filled data of an existing Hub.

11. Future Developments

Integration with the payment platforms

Handling payments for these small businesses can be simplified and cost-effective by integrating with a third-party payment platform. With this integration, small businesses can enable users to have more payment options and be relieved of maintaining such services.

Introduce revenue models

To serve HawkerHub as a free-to-use platform, and it needs a revenue model to sustain infrastructure and development costs. In the future, advertisements and promotions of businesses could be introduced based on the feedback from its users and businesses on the platforms.

Integration with online delivery platforms

The Hubs on the platform can now be integrated with third-party online delivery platforms to enable doorstep delivery. This way, the hubs can reach more comprehensive customers and are more discoverable on social platforms.

Rewards

Currently, the HawkerHub platform does not require the identities of its contributors for the sole reason of saving time in the contribution process. In the future, we can introduce identities on a pure volunteer basis to enroll in the rewards program. This program can incentivize the members of the community to contribute.

12. Conclusion

Community support is critical for the survival of small businesses. There are thousands of hawkers, street vendors, and small businesses around us. Many are on the verge of closure in the face of modernization. These businesses often face challenges, including fierce competition from larger companies, limited resources, and a need for more visibility. However, when the community rallies behind small businesses, it can make all the difference in preventing their closure.

By supporting small businesses, the community can help create a sustainable local economy and maintain the unique character of their community. Additionally, small businesses often contribute to the community in other ways, such as sponsoring local events, providing job opportunities, and supporting charitable causes. We have seen in the case of Singapore that a small initiative could help hundreds of local businesses.

The HawkerHub as a platform can play a significant role in enabling and creating an outreach for these small businesses, hawkers, and street vendors. By us adopting this platform, hawkers can focus on what they do best - *serving the community while leaving the rest to technology.*

References

[1] "Street vendors in the us - number of businesses 2003–2028." https://www.ibisworld.com/industry-statistics/number-of-businesses/street-vendors-united-states/. [Online; accessed 13-March-2023].

[2] "49.48 lakh street vendors identified in india: Government." https://www.tribuneindia.com/news/nation/49-48-lakh-street-vendors-identified-in-india-government-366768. [Online; accessed 13-March-2023].

[3] "Fraudsters with qr codes give traders nightmares." https://www.thehindu.com/news/national/kerala/fraudsters-with-qr-codes-give-traders-nightmares/article65048726.ece. [Online; accessed 13-March-2023].

[4] "S'porean man, 28, creates map of digitally-disadvantaged hawkers to bring them customers." https://mothership.sg/2021/06/disadvantaged-hawkers-online-map/. [Online; accessed 13-March-2023].

[5] "Flutter documentation." https://docs.flutter.dev/. [Online; accessed 13-March-2023].

[6] "A tour of the dart language." https://dart.dev/guides/language/language-tour. [Online; accessed 13-March-2023].

[7] "Aws serverless application model." https://aws.amazon.com/serverless/sam/. [Online; accessed 13-March-2023].

[8] "Aws sam cli." https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-reference.html#serverless-sam-cli. [Online; accessed 13-March-2023].

[9] "Aws s3." https://aws.amazon.com/s3/. [Online; accessed 13-March-2023].

[10] "Aws lambda." https://aws.amazon.com/lambda/. [Online; accessed 13-March-2023].

[11] "Aws dynamodb." https://aws.amazon.com/dynamodb/. [Online; accessed 13-March-2023].

[12] "Google maps platform." https://mapsplatform.google.com/. [Online; accessed 13-March-2023].

[13] “Google maps - apps on google play.” https://play.google.com/store/apps/details?id=com.google.android.apps.maps&amp;hl=en_US&amp;gl=US&amp;pli=1. [Online; accessed 13-March-2023].

[14] “Waze navigation - apps on google play.” https://play.google.com/store/apps/details?id=com.waze&amp;hl=en_US&amp;gl=US. [Online; accessed 13-March-2023].

[15] “Uber - request a ride - apps on google play.” https://play.google.com/store/apps/details?id=com.ubercab&amp;hl=en_US&amp;gl=US. [Online; accessed 13-March-2023].

[16] “Lyft - apps on google play.” https://play.google.com/store/apps/details?id=me.lyft.android&amp;hl=en_US&amp;gl=US. [Online; accessed 13-March-2023].

[17] “Material design.” https://m3.material.io/. [Online; accessed 13-March-2023].

[18] “Provider, flutter package.” https://pub.dev/packages/provider. [Online; accessed 13-March-2023].

[19] “Aws free tier.” https://aws.amazon.com/free/. [Online; accessed 13-March-2023].

[20] Flutter, “An introduction to widget testing.” https://docs.flutter.dev/cookbook/testing/widget/introduction. [Online; accessed 13-March-2023].

[21] Flutter, “An introduction to unit testing.” https://docs.flutter.dev/cookbook/testing/unit/introduction. [Online; accessed 13-March-2023].