MOBILE TOUR GUIDE APPLICATION WITH REAL-TIME ATTRACTION SPOT RECOGNITION USING VLM AND AR FOR KAMPAR

BY

PANG JING THEAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

FEBRUARY 2026

COPYRIGHT STATEMENT

This Final Year Project report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Computer Science (Honours) at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to express my sincere attitude and appreciation to my supervisor, Prof. Ts. Liew Soung Yue, for giving me the opportunity to work on this project. As this is my first time developing mobile application with recognition features, I am truly thankful for his invaluable guidance and advice throughout the project.

I would also like to thank my family for their love, support, and continuous encouragement throughout my studies.

Finally, I would like to express my appreciation to my significant other for his unconditional love and support, and for always standing by my side during hard times.

ABSTRACT

Kampar is a town in Perak, Malaysia, that has many historical places often overlooked by tourists. Most tourists do not know the history and information about the places they visit. To learn more, they often rely on internet search engines such as Google and Google Lens. However, this can be a time-consuming process and yields inaccurate results. To address these issues, this project proposes a mobile tour guide application with recognition features for Kampar. By utilising modern technologies, this application aims to enhance cultural and historical tourism by making the experience more interactive and educational. Developing a mobile application offer seamless accessibility, as people frequently carry their devices. The main feature of this mobile application is to function as a virtual tour guide when users visit these places. It integrates pre-trained models such as Vision-Language Models (VLM) for real-time recognition. The application is designed to identify specific points of interest or distinct structural features rather than the general vicinity. When an attraction spot is identified, it provides Augmented Reality (AR) overlays with the name of attraction spot. Users can tap the labels to view more detailed information. By offering an accessible and immersive tour guide experience, the application promotes Kampar's cultural places to both tourists and locals. The recognition feature focuses on two places: UTAR Grand Hall (Dewan Tun Dr. Ling Liong Sik) and Kampar Seng Fatt Temple. Overall, this project involves pre-trained models, including vision-language model for object detection and image classification, along with augmented reality (AR) for content delivery.

Area of Study: Mobile Application Development

Keywords: Vision-Language Model, Augmented Reality, Attraction Spot Recognition, Mobile Tour Guide Application, Zero-Shot Object Detection, Few-Shot Image Classification

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

VLM	Vision-Language Model
AR	Augmented Reality
CLIP	Contrastive Language-Image Pretraining
FastSAM	Fast Segment Anything Model
ZSL	Zero-Shot Learning
FSL	Few-Shot Learning

Chapter 1
Introduction

This chapter includes background information, problem statements, motivation, objectives, scope, direction and contribution.

1.1 Background Information

Kampar is a town located in Perak, Malaysia. It has a lot of cultural heritage and natural attractions for tourists to visit. However, tourism experience in Kampar is limited due to the lack of interactive and informative guidance. Therefore, tourists may overlook some attraction spots or do not have the sources to understand backstories of the places they visit.

Currently, tourists who are interested in knowing the history of the attraction spots in Kampar often rely on manual search for historical information using search engines or visual search tools such as Google Lens. While these tools can provide helpful information, they are not optimized for a seamless tourism experience. For example, search engines return results from various sources, which require users to spend time reading and verifying the information accuracy. Similarly, visual search tools may misidentify attraction spots or provide results that combine relevant and irrelevant information, making it harder for tourists to obtain reliable historical or cultural information quickly.

With these limitations, a mobile tour guide application could play a crucial role in improving cultural tourism experience in Kampar. The mobile application could help both tourists and locals to gain information about attraction spots by providing reliable and informative context. This not only connects people to the past but also connects people to other cultures, allowing them to discover similarities and differences of places in historical context [1].

1.2 Problem Statement and Motivation

1.2.1 Problem Statement

1. **Lack of interactive and accurate information sources**

   Kampar has many historical and cultural attractions, but the tourism experience in Kampar is hindered by lacking interactive and accurate sources. Hence, tourists may overlook attraction spots or unable to understand the historical background of places they visit.

2. **Difficulties in obtaining in-depth and reliable information**

   Visitors and even locals find difficulties in obtaining in-depth and reliable information about the attraction spots. Without guidance, they can only rely on online sources, which often lack detailed or trustable information. This affects their tourism experience.

3. **Challenges in custom model training**

   Several senior projects have developed mobile tour guide applications to address these issues. To recognise attraction spots, they trained custom image classification and object detection model [2]-[3]. While this approach is feasible, it requires extensive data and training time to achieve high accuracy.

1.2.2 Motivation

The motivation of this project is to help both visitors and locals understand the history of places in Kampar better. By developing a mobile tour guide application, this project aims to improve tourism experience by providing engaging and informative content of the places visited. The integration of pre-trained models will allow faster and accurate recognition of attraction spots. Augmented Reality (AR) will be used for content delivery in an interactive and visually immersive way, making the tourism experience more fun, engaging and educational for users.

1.3 Objectives

The purpose of this project is to develop a mobile application that showcasing pre-trained vision-language models can be integrated without additional training or fine-tuning for recognition tasks. This project aims to recognise two places in Kampar and deliver contextual information.

The objectives of this project:

1. **To develop a mobile tour guide application for Kampar:**
   Develop a mobile tour guide application that can perform recognition on selected locations and deliver accurate, in-depth historical and cultural information.

2. **To enhance tourism experience and provide cultural benefits:**
   The application can benefit both tourists and locals by providing access information and knowledge about attraction spots in Kampar. This will improve their tourism experience and assist cultural education.

3. **To integrate pre-trained models for recognition:**
   Use pretrained models for attraction spot recognition to reduce the need for large datasets and extensive time to train models.

1.4 Project Scope and Direction

This project aims to develop a mobile tour guide application that improves tourism experience in Kampar through real-time attraction spot recognition and deliver content using AR. This application will access mobile device's camera to perform real-time attraction spot recognition and deliver information to users using AR overlays in an interactive way.

The scope of the mobile application includes:

1. **Real-Time Attraction Spot Recognition**

   This project integrates pre-trained models to achieve attraction spots recognition. Two pre-trained models will be integrated for object detection and image classification. Additionally, real-time recognition will be achieved by capturing frames every 2 seconds.

2. **Augmented Reality (AR) Overlays**

   Once an attraction spot is recognised, virtual labels will be displayed to provide the predicted attraction spot name and allow users to tap and view detailed information.

3. **Target Locations**

   The attraction spot recognition features focus on two locations: UTAR Grand Hall and Kampar Seng Fatt Temple.

1.5 Contributions

This project contributes by developing a mobile tour guide application that integrates pre-trained models and Augmented Reality (AR) to provide access to historical and cultural information. By using a pre-trained VLM, it enables attraction spot recognition using fewer reference images compared to traditional approaches, reducing the need for collecting large training datasets. Once recognised, the application will show the results by using AR overlays, enhancing user engagement and improving understanding of the attraction spot.

1.6 Report Organization

This report is organised into 7 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Chapter 7 Conclusion. The first chapter is the introduction of this project which includes background information, problem statement and motivation, objectives, scope and direction, contributions, and report organisation. The second chapter is the literature review carried out on existing works to evaluate the strengths and weaknesses and compare with the proposed system. The third chapter is discussing the overall system methodology of this project. The fourth chapter is about the system design, including diagrams such as flowcharts. Furthermore, the fifth chapter includes the implementation of system, and the sixth chapter involve evaluation of the system. Lastly, the seventh chapter concludes the whole project.

Chapter 2
Literature Review

Currently, there is no mobile tour guide application that integrates VLM for real-time attraction spot recognition. However, there are several approaches that have been utilised to recognise attraction spots. There are also some applications that can be used to recognise objects and display the relevant information.

2.1 Review of the Existing Work

Previous studies did not integrate any AI models for attraction spot recognition, most of them performed attraction spot recognition by using their own trained model.

2.1.1 On-site Attraction Recognition for Kampar Temples

In [2], the author developed a mobile tour guide application focused on temples in Kampar. The main feature of the application is on-site attraction recognition, users can get information based on their visual input. To achieve this, they trained their own model for attraction spot recognition. There are 2 models produced, which are image classification model and the object detection model. They create their own dataset by collecting and labelling images of attraction spots, then proceed to train the models. These models were trained using the TensorFlow library. The image classification model was used to categorize the images, and the object detection model was used to perform multiple attraction spots recognition and real-time multiple attraction spots detection. The models were tested after training, the accuracy of the image classification model was 93.51% using 77 images downloaded from the Internet and the accuracy of the object detection model was about 79.47% using 455 images. The application supports image upload, image capture and live detection, and it can recognise both single and multiple attraction spots.

Strengths

High Accuracy

In [2], they achieved high accuracy for both the image classification and object detection model, obtaining 93.51% and 79.47% respectively. These results showed that the custom-trained models could recognise attraction spots for the targeted locations.

Live Detection

The application developed supports image upload, image capture and real-time recognition of attraction spots. This provides an immersive tourism experience by delivering information immediately during site visits.

Weaknesses

Require Large and Labelled Dataset

In [2], they took 2327 images physically for their dataset. They need to label these images with names and classify them according to the attraction spots manually. This process is highly time-consuming and requires considerable manual effort, especially when there are many locations that need recognition.

Limited Scalability

The model trained for specific locations requires manual labelling and model retraining when it needs to add new attraction spot recognition. This results in scalability challenges, it is difficult to quickly expand the system to include other attraction spots without some time and resources. The process of capturing images physically and manually is also labour-intensive, which makes it challenging to cover multiple attraction spots in a short amount of time.

2.1.2 On-Site Augmented Reality - Kampar Churches

In [3], the author developed an Android mobile tour guide application using augmented reality technology. The application focuses on Kampar Churches site, it provides historical and cultural information to users based on its database. The features include login, an overview of popular tourist attractions in Kampar, live attractions recognition, non-real-time attractions recognition, a mini-game and user settings. To recognise attraction spots, they train a model using libraries such as Mediapipe model makers and Tensorflow. They create their own dataset consisting of 3333 images with different angles and lighting conditions. For the object detection model, the MobileNet V2, a predefined model in Mediapipe model maker, is selected for its lower latency. After training, the model is evaluated, and it achieved an average precision of 90.07%.

Strengths

Provide Interactive Experience

The project in [3] provides interactive experience by using augmented reality (AR) to enhance user engagement. The application presents the historical and cultural information onto real-world view of Kampar Churches.

Weaknesses

Require Large and Labelled Dataset

In [5], they used 3333 images for training their model. The images were collected and labelled manually, which is time-consuming and require intensive manual efforts. When adding a new attraction spot, large amounts of images need to be collected again for the model to recognise.

2.1.3 Tourist Attractions Classification using ResNet

This paper developed a smart tourism application that can classify tourist attractions [6]. In this study, they only focused on tourist attractions in Jakarta and Depok. The datasets were taken from Google's search engine with variations in angle, lighting and size. During data preprocessing, they convert the dataset into the RGB (Red Blue Green) model. After that, the images were resized to 150x200 to reduce the running time of the classification process. For data training and testing, they use k-fold cross-validation to split the data into training and testing datasets. The k-fold cross-validation will divide the data into k sections, and the data will be almost the same size. They perform k-fold cross-validation several times, with each subsample only being used exactly once. The outcomes of each k will be summed and averaged. They perform k-fold cross-validation until k = 4. The ResNet50 is used as a method to classify images. ResNet50 is a residual network with 50 layers. Although there are also other types of ResNet, they choose ResNet50 because it has fewer layers, and it can run faster during inference.

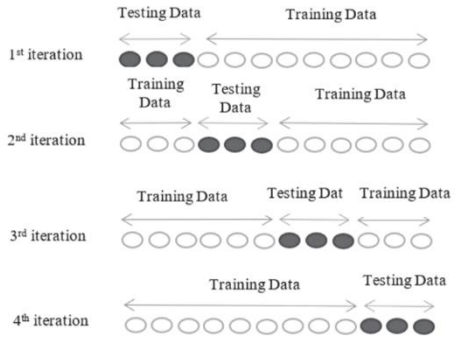


Figure 2.1 k-fold cross validation

Strengths

High Accuracy

ResNet50 ensures faster inference, proven that deep learning model works well on image classification tasks.

Weaknesses

Use of Internet Source Images

Online images with different quality may affect the image classification.

2.1.4 Augmented Reality-Based On-Site Tour Guide

In [7], the authors developed an Augmented Reality (AR) tour guide system to enhance tourist experience at cultural heritage sites, focusing on Gyeongbokgong Palace in Korea. This system overlays animated 3D virtual characters onto real-world scenes by tracking simple geometric primitives such as rectangles. There are four components in the proposed AR tour guide framework: context awareness, augmentation, input agent and output agent. To provide context-aware information, the system recognises specific locations by identifying the wooden tablets that display the name of a building. The k-nearest neighbour (KNN) algorithm is used for recognizing images by detecting feature points in the captured image, computing their descriptors and matching them to descriptors of reference images. This recognition process enables the system to provide context-aware historical information that is relevant to the identified building.

Strengths

Provide Interactive Experience

The tour guide system provides an interactive and immersive experience for users by overlaying animated 3D characters onto real-world scenes. This enhances the visiting experience and user engagement by encouraging users to interact actively with the cultural heritage sites.

Weaknesses

Not Support on Mobile Devices

In [7], the developed AR system is limited to specific devices such as laptop, pen-tablet and palmtop, which exclude mobile devices. In the user evaluation part of the paper, one of the questions asked, “Would it be better if the AR tour guide were available on mobile phones?” Majority of the users responded yes, expressing their preference for mobile-based services due to the convenience, portability and accessibility of mobile phones.

2.1.5 Recognition of Tourist Attractions

This research paper suggests using machine learning methods to recognize specific locations [8]. The convolutional neural networks (CNN) are used for feature extraction, and the support vector machine (SVM) is used to output the predicted attraction. The authors created a custom dataset containing 10000 images that further split into 7000 training images, 2000 validation images and 1000 test images. There are 1000 images for each attraction from different angles and under different weather conditions. For feature extraction, the authors used the Places-CNN model, which is a pre-trained model on scene recognition tasks. The authors decided to train separate classifiers for day and night since they found out there are differences between day and night images of the same attraction. They also use an additional binary classifier for determining which model to use at test time. The pixel value ratio of the image was computed by counting the total number of pixels of each pixel value and then dividing them by the total number of pixels in the image. Two classifiers, SoftMax and SVM were trained in this research. The SoftMax classifier outputs probabilities for each class, indicating the input image belongs to each tourist attraction. The SVM classifier is used to classify the image as day or night.

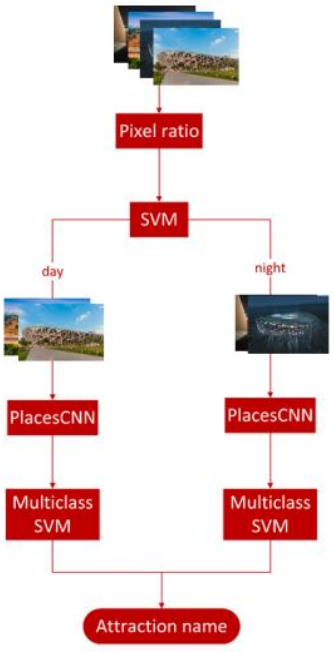


Figure 2.2 Complete Classification Pipeline

Strengths

Consideration of Environmental Conditions

In the study, they trained the classifiers separately for day and night. Moreover, the dataset they used contained 10,000 images of multiple attractions captured from diverse angles and under different weather conditions. The variety of dataset enhances the reliability of the model recognition performance.

Weaknesses

Large Dataset

Although they achieved good recognition results, collecting 10,000 images for training requires significant time and effort. This dependency makes it difficult when includes new attraction spots.

2.1.6 Google Lens

Google Lens is an application that lets people search for what they see [4]. It combines computer vision, machine learning and cloud-based processing to analyse images and provide contextual information [5]. It can understand what people are looking at, and it can copy or translate text, identify plants and animals, explore locales or menus, etc. Google Lens uses convolutional neural networks (CNNs) trained on massive datasets to perform recognition. Using a camera or photos in devices, Google Lens can discover visually similar images and related contents, then gather all relevant information from all over the internet. Google Lens compares objects in images to other images, ranking them according to their similarity with the original image. Moreover, Google Lens will return more accurate results for identifying places and landmarks if users agree to let it access their location.

Strengths

Robust Accuracy

Google Lens is trained by large datasets, and it can search for visually similar images and related information all over the internet. It not only can recognize objects, but also landmarks, text, animals, plants, etc.

Showcase of using Google Lens to identify UTAR Grand Hall:

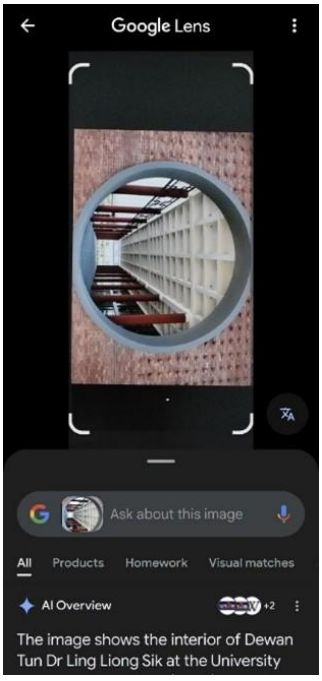


Figure 2.3 Image Search using Google Lens

Figure 2.4 Results from Google Lens

2.2 Proposed System

To address the limitations and weaknesses of previous work, the proposed solution will focus on developing a mobile tour guide application for Kampar and emphasize real-time attraction spot recognition, convenience and scalability. This solution will integrate advanced technology such as vision-language model (VLM) and augmented reality (AR) to provide seamless and interactive tour guide experience for users.

1. **Real-Time Attraction Spots Recognition Using Pre-Trained Models**

   Instead of training model, this project integrates pre-trained models such as VLM for attraction spots recognition. VLM such as CLIP is designed to understand both visual features and textual descriptions, capable to identify parts of an attraction such as Moongate in UTAR Grand Hall and statues in temples. Attraction spots can be identified using few-shot learning (FSL) approach, by providing the model with a small number of reference images and text descriptions of attraction spots.

2. **Reduce Manual Labelling and Data Collection**

   Since using VLM only requires a small number of images, it reduces the need for manual labelling and intensive image collection. Using VLM reduces the time and effort for preparing the dataset and can focus on delivering a functional and user-friendly application.

3. **Mobile Augmented Reality Experience**

   This project uses AR overlays to display the contents on users' screens. When the attraction spots are recognized, AR elements display on screen, creating an interactive and immersive experience for users. This allows users to explore historical places in an engaging and informative way, enhancing their tourism experience.

4. **Highly Scalable**

   The use of VLM enables the application to be scalable. Since VLMs do not require retraining the model for each new attraction spot, a small number of images and descriptions can be provided to VLM to learn and recognize new attraction spots. This simplifies the process of adding new attraction spots for recognition and reduces development time. As a result, the application can be easily scaled to include more attraction spots without the need to retrain models and large datasets.

2.2.1 Limitations and Proposed Solutions

Limitations	Proposed Solution
Require large and labelled dataset	Utilizing the object detection and classification ability of pre-trained VLMs reduces the work of labelling by providing small amounts of reference images and text prompts
Lack of scalability when adding new attraction spot	Require small number of images of new added attraction spots
Not support on mobile devices	A mobile application designed for mobile devices

Table 2.1 Limitations and Proposed Solutions

Chapter 3

System Methodology/Approach

3.1 Methodology

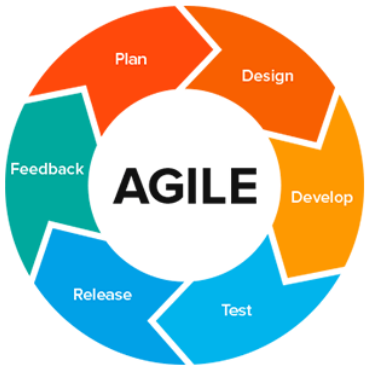


Figure 3.1 Agile Methodology

This project adopted Agile methodology for system development. Agile is based on principles that emphasizes iterations and incremental deliveries, where the system is break down into small and manageable task, and developed incrementally. The core functions will be implemented and delivered first, allowing user to test the application and provide feedback [9].

Core functions in this project:

1. Real-Time Attraction Spot Recognition
2. Augmented Reality (AR) Content Delivery

Each of these functions will be implemented, delivered tested incrementally throughout the development process, following Agile methodology iterative and incremental approach.

3.2 System Architecture Diagram

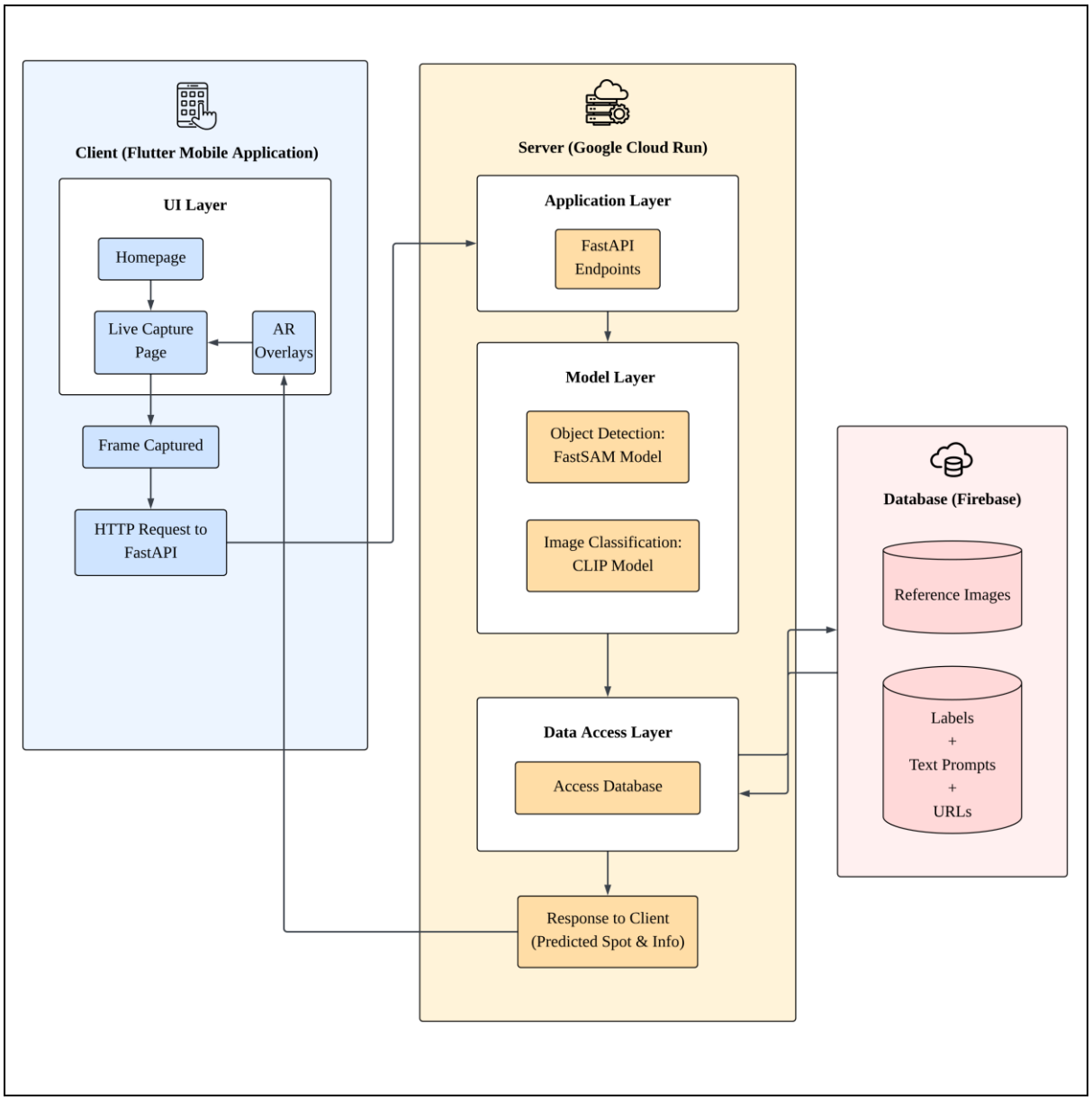


Figure 3.2 System Architecture Diagram

The system architecture utilises the combination of the **client-server** and **layered architectures**. In this design, the mobile application acts as the client, and the FastAPI backend acts as the server hosted in the cloud. The client captures images through the camera, sends HTTP requests to the backend and displays AR overlays for user interaction. For the server, it processes the requests from clients by performing attraction spot recognition and returns the predicted results and information to the client.

Within the server, layered architecture is applied to separate concerns and improve maintainability. The application layer handles HTTP requests through the FastAPI endpoint. The model layer integrates pretrained FastSAM and CLIP model to perform recognition by calculating similarities. The data access layer retrieves the referenced images, text and URLs from the data layer. In the database layer, it contains reference images, text descriptions, labels and URLs for each attraction spot. Lastly, the server returns the prediction results and related data to the client to display it with AR overlays.

The combination of architectures provides **separation of responsibilities**, **scalability** and **flexibility** to the system.

3.3 User Requirements

3.3.1 Functional Requirements

1. The user can view the onboarding pages on first launch.
2. The user can view a list of places on the home page.
3. The user can select place.
4. The user can tap the view map button.
5. The user can view the name of the attraction spot detected.
6. The user can tap the labels to view information of the attraction spot.
7. The user can close the webpage after viewing.
8. The user can return to the home page and select another place.

3.3.2 Non-functional Requirements

1. The user can use the mobile application on Android devices that support ARCore.
2. The user can navigate the application easily.
3. The user can view the webpages of attraction spots without leaving the application.

3.4 Use Case Diagram and Use Case Description

3.4.1 Use Case Diagram

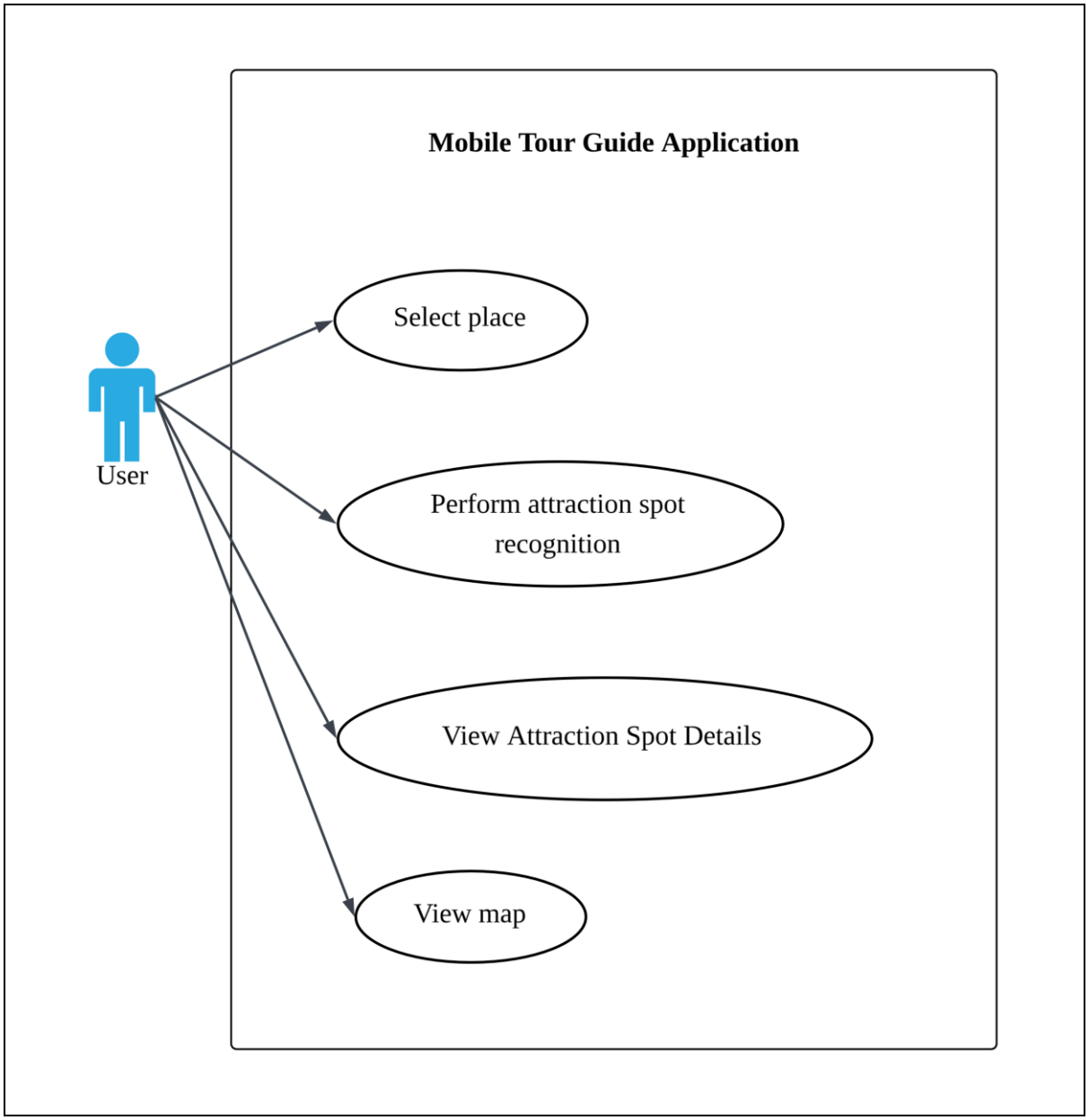


Figure 3.3 Use Case Diagram

3.4.2 Use Case Description

1. Select Place

Use Case Name: Select place	ID: 1	Importance Level: High
Primary Actor: User	Use Case Type: Detail, essential	
Stakeholders and Interests: User – wants to select a place		
Brief Description: The use case describes the steps of user selects a place in Home page.		
Trigger: User launches the application. Type: Internal		
Relationships: Association: User Include: Not Applicable Extend: Not Applicable Generalization: Not Applicable		
Normal Flow of Events: 1. User launches application. 2. System display Home page. 3. User scrolls the list of places. 4. User selects a place.		
Sub Flows: Not applicable		
Alternate/Exceptional Flows: 2a. First time application launch, display Onboarding pages.		

Table 3.1 Use Case Description for Select Place in Home Page

2. Perform Attraction Spot Recognition

Use Case Name: Perform attraction spot recognition	ID: 2	Importance Level: High
Primary Actor: User	Use Case Type: Detail, essential	
Stakeholders and Interests: User – wants to perform attraction spot recognition		
Brief Description: The use case describes the steps of user performs attraction spot recognition.		
Trigger: User selects a place. Type: Internal		
Relationships: Association: User Include: Not Applicable Extend: Not Applicable Generalization: Not Applicable		
Normal Flow of Events: 1. User selects a place. 2. System display Live Capture page with camera view. 3. User scans surroundings. 4. System captures frame every 2s and sends to server. 5. System return prediction results. 6. System display labels with attraction spot names.		
Sub Flows: Not applicable		
Alternate/Exceptional Flows:		
2a. System detects no camera permission. 4a. Server is not ready, display loading message. 5a. Prediction results is empty, return to Step 4 of Normal Flow.		

Table 3.2 Use Case Description for Perform Attraction Spot Recognition

3. View Attraction Spot Details

Use Case Name: View attraction spot details	ID: 3	Importance Level: High
Primary Actor: User	Use Case Type: Detail, essential	
Stakeholders and Interests: User – wants to view the details of the attraction spot		
Brief Description: The use case describes the steps of user tap and view the details of attraction spot.		
Trigger: User taps the label. Type: Internal		
Relationships: Association: User Include: Not Applicable Extend: Not Applicable Generalization: Not Applicable		
Normal Flow of Events: 1. User taps the label. 2. System opens the URL in the in-app browser. 3. User views the details information. 4. User closes the webpage. 5. System returns user to Live Capture page.		
Sub Flows: Not applicable		
Alternate/Exceptional Flows: Not applicable		

Table 3.3 Use Case Description for View Attraction Spot Details

4. View Map

Use Case Name: View map	ID: 4	Importance Level: High
Primary Actor: User	Use Case Type: Detail, essential	
Stakeholders and Interests: User – wants to view the locations of the attraction spots on a map to navigate or plan their route		
Brief Description: The use case describes the steps of user view the locations of places in external application.		
Trigger: User taps "View Map" button. Type: Internal		
Relationships: Association: User Include: Not Applicable Extend: Not Applicable Generalization: Not Applicable		
Normal Flow of Events: 1. User taps "View Map" button. 2. Systems opens URLs using external browser. 3. Use case ends.		
Sub Flows: Not applicable		
Alternate/Exceptional Flows: Not applicable		

Table 3.4 Use Case Description for View Map

3.5 Timeline

3.5.1 FYP1

Week/Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Revise Proposal Report												
Research VLM Models												
Prepare Sample Images												
Test VLM Models Performance												
Explore Comparison Logic												
Select VLM Model												
Capture New Reference Images												
Improve Comparison Logic												
Sketch Wireframes												
Develop Homepage UI												
Design System Flowchart												
Develop Attraction Pages UI												
Access Device's Camera in App												
Design System Architecture												
Develop Send Frames Function												
Integrated FastAPI												
Handling Issues of Non-Attraction Spot Recognition												
Display Prediction Result in App												
Test Application												
Report Writing												

Figure 3.4 FYP1 Gantt Chart

3.5.2 FYP2

Week/Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
Revise FYP1 Report													
Planning for FYP2													
Redesign Mobile Application UI													
Configure Firebase													
Store Reference Images/Text Prompts in Firebase													
Transform Text Results Display to Labels													
Research for Better Recognition Pipeline													
Plan New Recognition Pipeline													
Research & Explore Pre-Trained Object Detection Models													
Integrate Object Detection Model													
Collect New Images													
Store New Images/Text Prompts in Firebase													
Code Showcase Webpages													
Store URLs in Firestore													
Resolve Multiple Labels on ONE Object													
Create Dockerfile for Cloud Hosting Backend													
Deploy Backend in Google Cloud Run													
Configure Cloud Run													
Deploy Revisions													
Test Application													
Report Writing													

Figure 3.5 FYP2 Gantt Chart

Chapter 4

System Design

4.1 System Flowchart

Figure 4.1 to Figure 4.4 illustrate the overall system flow. Figure 4.1 shows the main flow of the mobile application, starting with launching the application to displaying the home page. While Figure 4.2 further illustrates the flow of home page, including flow of viewing the map and selecting a place. In Figure 4.3 continue with the flow of Live Capture page after users navigates from home page, it shows the flow of attraction spot recognition feature, starting with frame capture to label display, followed by tapping labels and webpage opening webpage in the in-app browser. Lastly, Figure 3.4 shows the frame process module by the backend and the return of prediction results.

4.1.1 Application Flow - Main Flow

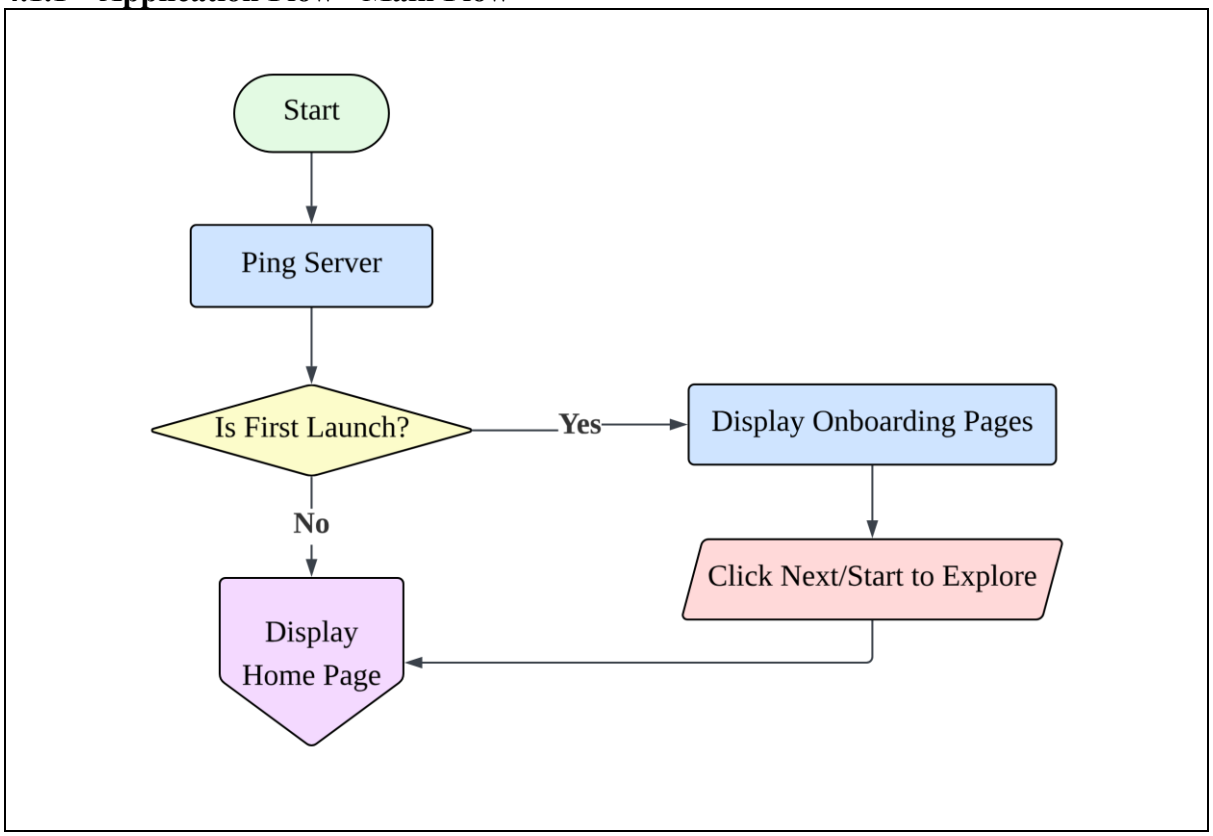


Figure 4.1 Application Flow - Main Flow

4.1.2 Application Flow - Home Page Flow

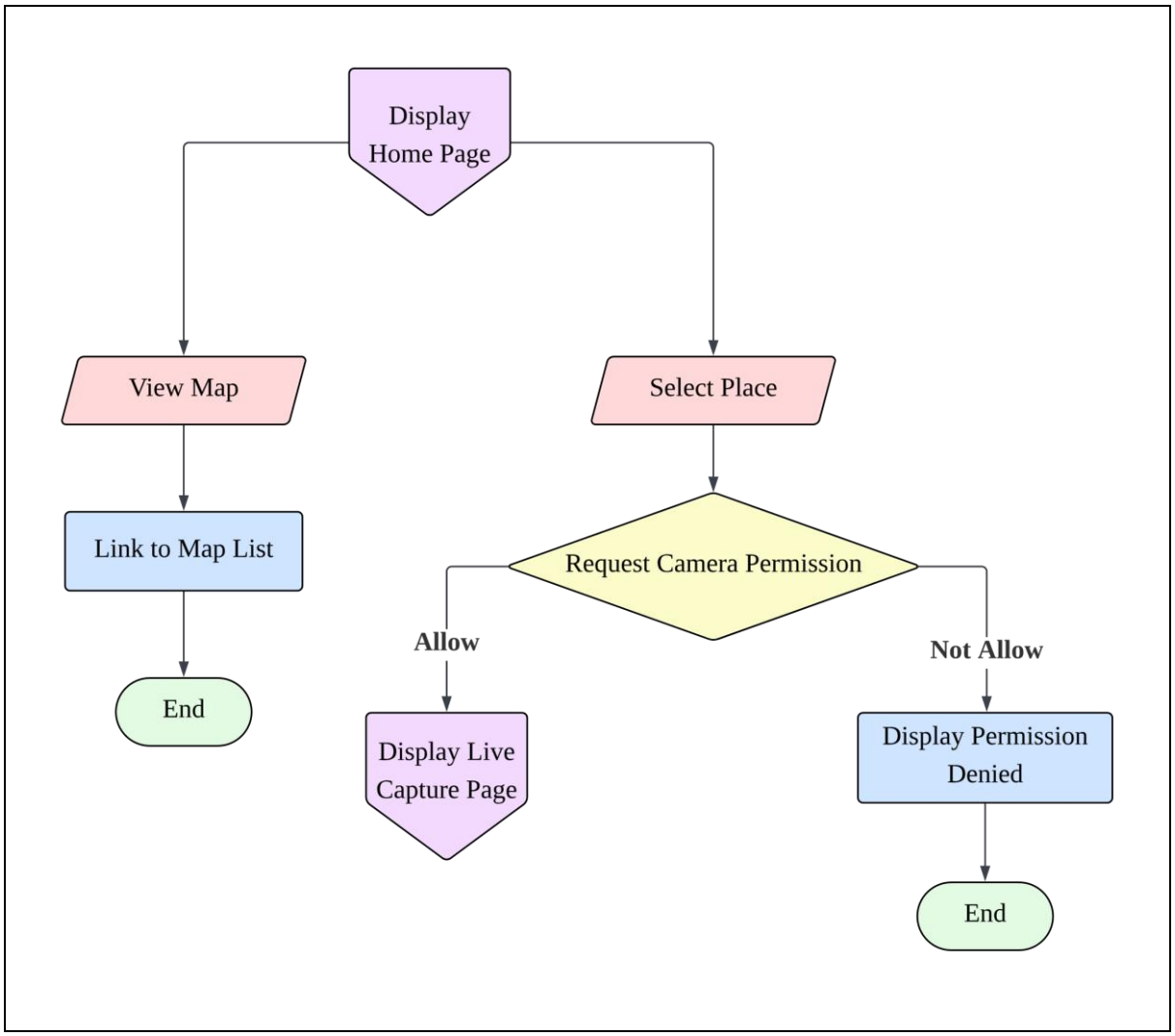


Figure 4.2 Application Flow - Home Page Flow

4.1.3 Application Flow - Live Capture Page

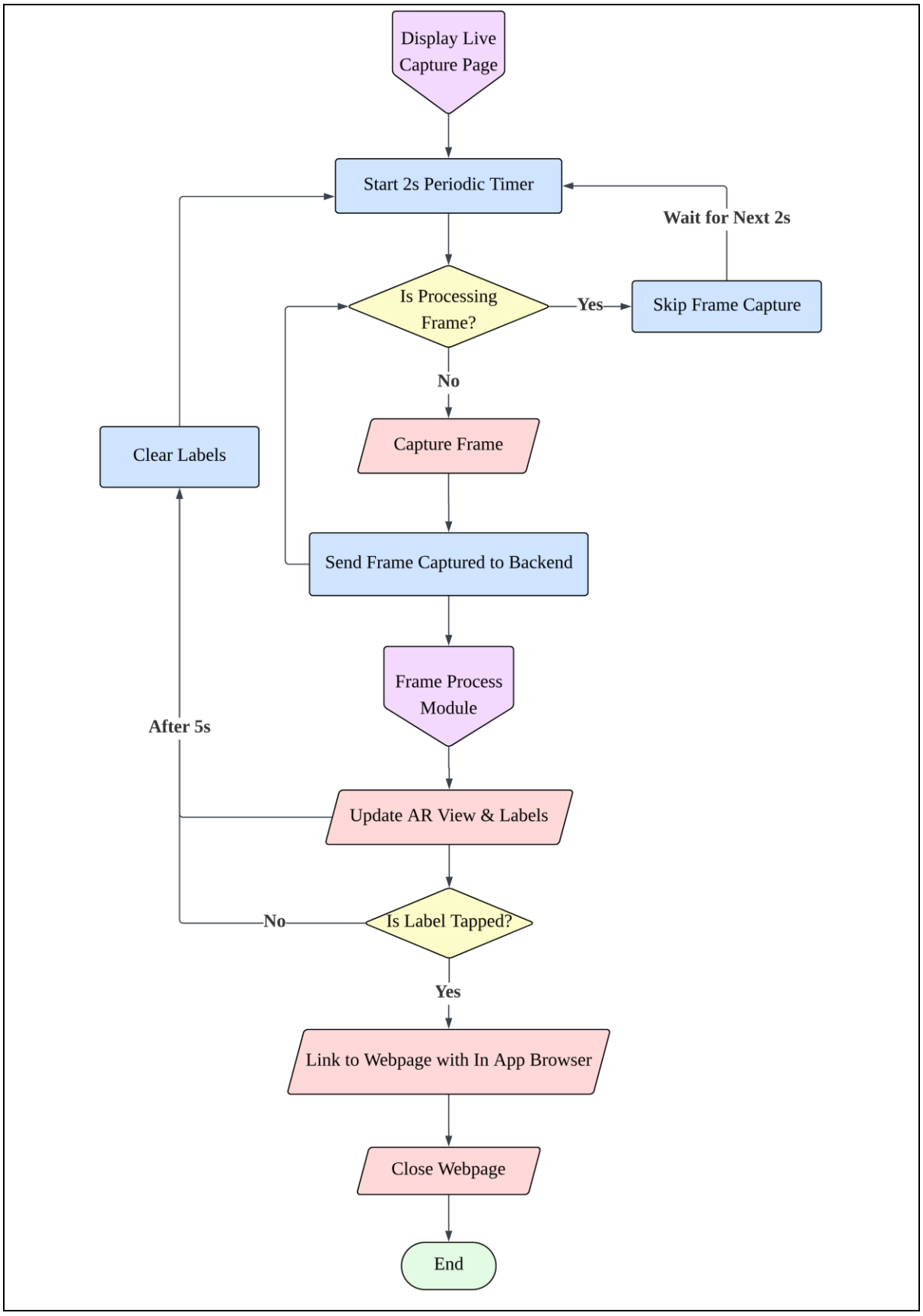


Figure 4.3 Application Flow - Live Capture Page

4.1.4 Backend Flow - Frame Process Module

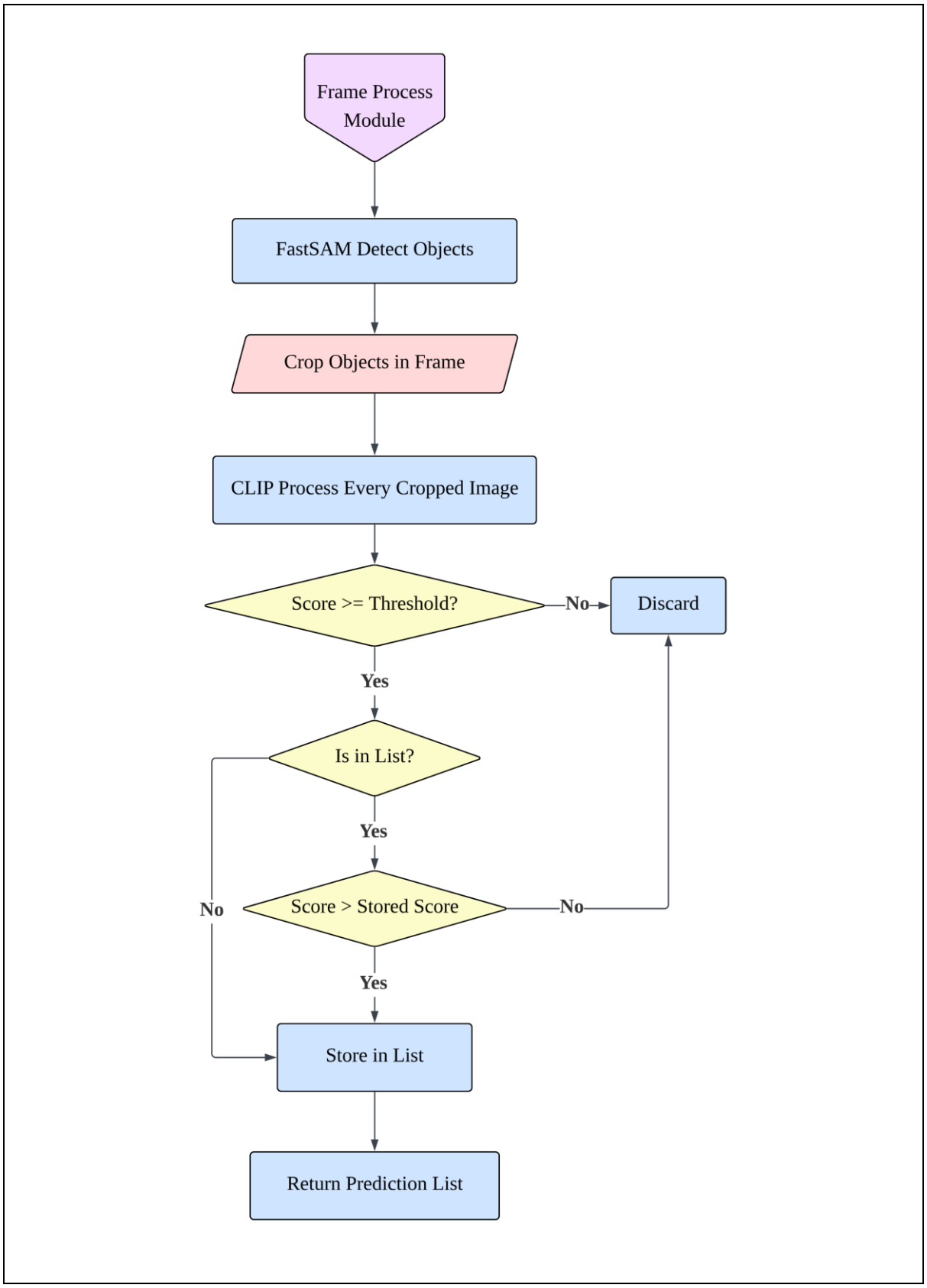


Figure 4.4 Backend Flow - Frame Process Module

4.2 Activity Diagram

4.2.1 Perform Attraction Spot Recognition

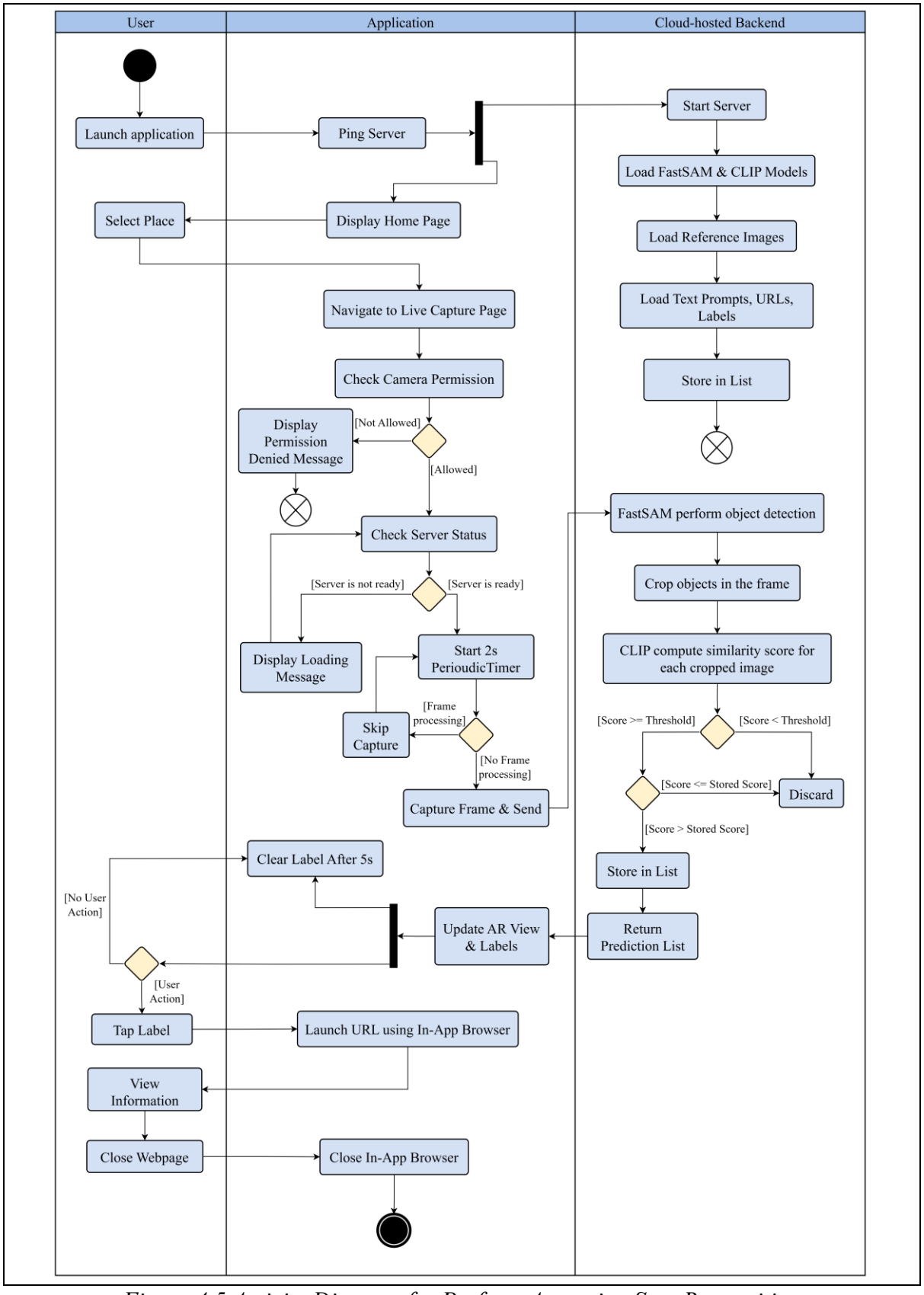


Figure 4.5 Activity Diagram for Perform Attraction Spot Recognition

4.2.2 View Map

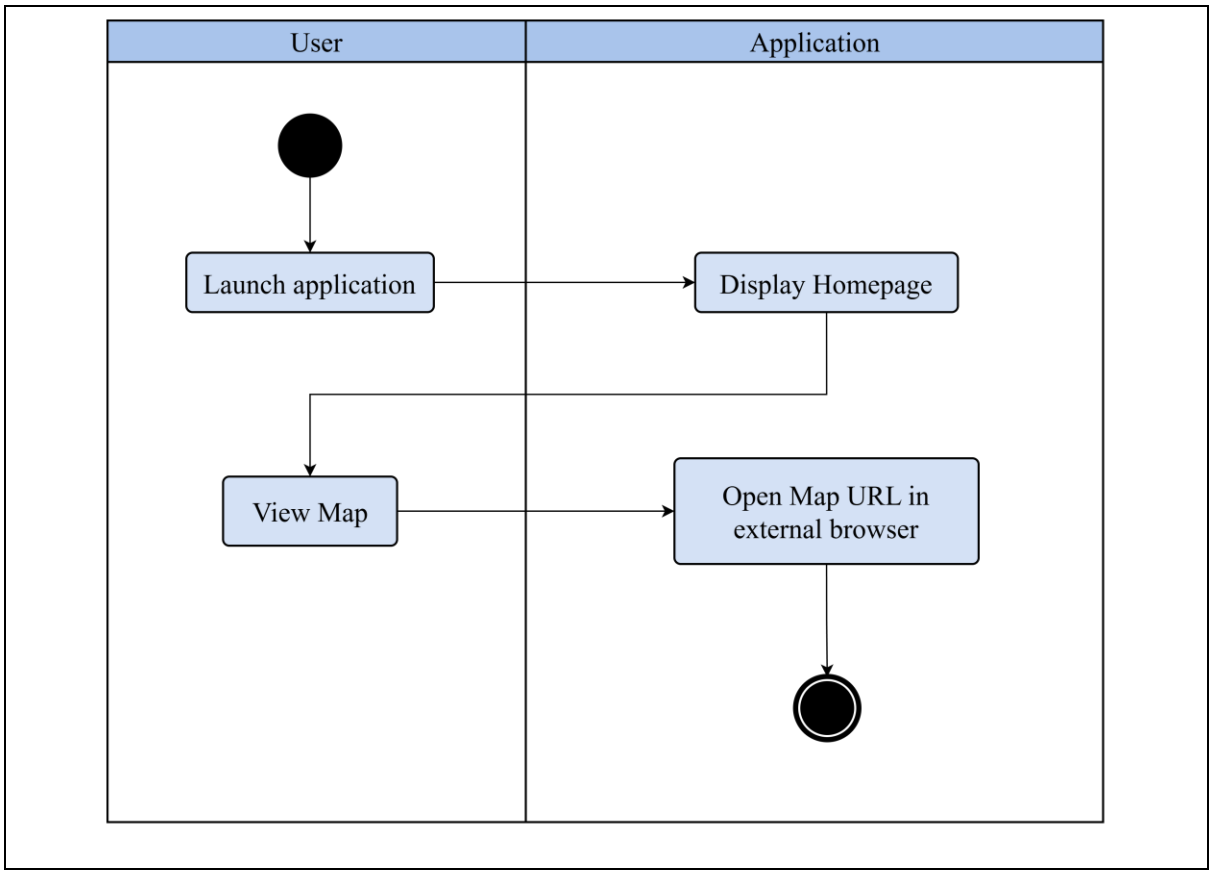


Figure 4.6 Activity Diagram for View Map

Chapter 5

System Implementation

5.1 Hardware Setup

Hardware involved in this project including a laptop and an android mobile device. The laptop is used for developing the whole project, and the mobile device is used to capture the references images and test the developed mobile application.

Description	Specifications
Model	Lenovo ideapad Slim 3
Processor	AMD Ryzen 7 5700U with Radeon Graphics
Operating System	Windows 11
Memory	16GB RAM
Storage	512GB SSD

Table 5.1 Specifications of laptop

Description	Specifications
Model	HONOR 400 Pro
Processor	Qualcomm Snapdragon 8 Gen 3
Operating System	MagicOS 10.0 (Android 16)
Memory	12GB RAM + 12GB HONOR RAM Turbo
Storage	512GB

Table 5.2 Specifications of mobile phone

5.2 Software Setup

In this project, there are several software used to develop the frontend mobile application and backend logic.

5.2.1 Software

Visual Studio Code (Version 1.118.1)

Visual Studio Code (VS Code) is used as the Integrated Development Environment (IDE) for this project. It is a code editor that provides a flexible environment for coding, debugging and integrating libraries and frameworks.

This project used it for coding and debugging the mobile application UI and functions and backend logic.

GitHub

GitHub is a cloud-based developer platform which is commonly used to back up, track changes and control versions.

It is used to back up the project code, track incremental deliveries and maintain project history.

Lucid Chart

Lucid Chart is a cloud-based diagramming tool.

It is used to design system diagrams such as flowchart and architecture diagram.

Google Cloud Run

This project utilised Google Cloud Run to host and run the backend logic, including FastAPI and pre-trained models. The backend was deployed using Docker.

Firebase

Firebase was used to store reference images, text prompts, labels and URLs of attraction spots. The data will be loaded while the server starts.

5.2.2 Mobile Application

Dart (Version 3.8.1)

Dart is a programming language commonly used for developing web and mobile apps. It is the primary programming language for developing the mobile tour guide application with the Flutter framework.

- **Framework:**

1. Flutter (Version 3.32.8)

   Flutter is a cross-platform UI framework built on Dart. It is used to develop this tour guide application by its provided widgets and tools to build an interactive UI and allows users to navigate easily.

   Core packages in this project:

Packages	Version	Purpose
camera	0.10.16	Controls the device's camera
permission_handler	11.4.0	Managing request device permissions for camera access
http	1.5.0	Handles network requests, communicating with backend
ar_flutter_plugin_2	0.0.3	Handles AR view
vector_math	2.1.4	Used for 2D and 3D geometry
url_launcher	6.3.2	Launches URLs, used to open in-app browser and external application
flutter_native_splash	2.4.7	Generates splash screen
shared_preferences	2.5.3	Saves simple local data persistently on the device, used to display onboarding pages
flutter_launcher_icons	0.14.4	Generates app icons

Table 5.3 Core Packages for Flutter

5.2.3 Backend

Python (Version 3.13.6)

Python is a programming language which is commonly used for backend development. In this project, it is used for backend and integration of pre-trained models and image processing.

Core libraries used:

Library	Version	Purpose
fastapi	0.135.1	Provides the web framework to define the API routes, handles HTTP request, manage CORS for frontend
ultralytics	8.44.2	Loads and runs the FastSAM model (YOLO-based) to perform object detection and extract bounding boxes from frame captured
transformers	5.3.0	Loads the Hugging Face CLIPModel and CLIPProcessor to generate feature embeddings from text prompts and cropped images
torch	2.10.0	Used to execute models, handle tensors and compute cosine similarities
firebase_admin	7.2.0	Authenticates and connects to Firebase to retrieve Cloud Storage for images and Firestore for labels, text prompt, URLs
pillow	12.1.1	Handles image operations in memory, including convert image format and cropping bounding boxes
python-multipart	0.0.22	Backend dependency required by FastAPI to parse incoming multipart form data
uvicorn	0.41.0	Required to run FastAPI

Table 5.4 Main Libraries for Python

5.3 Mobile Application Development

5.3.1 App Icon and Splash Screen

Figure 5.1 App Icon

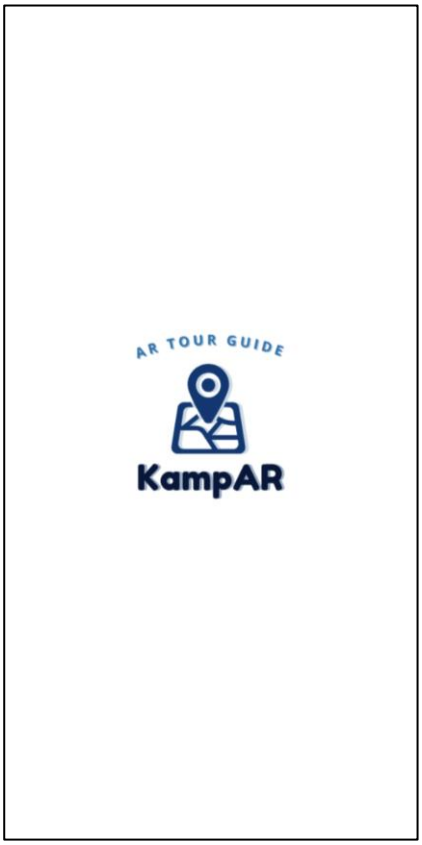


Figure 5.2 Splash Screen

5.3.2 Onboarding Pages

The onboarding pages were designed to help first-time users to understand how the application works. Figure 5.3 to Figure 5.6 illustrates the attraction spot recognition feature, along with view map feature that help users navigate to the places.

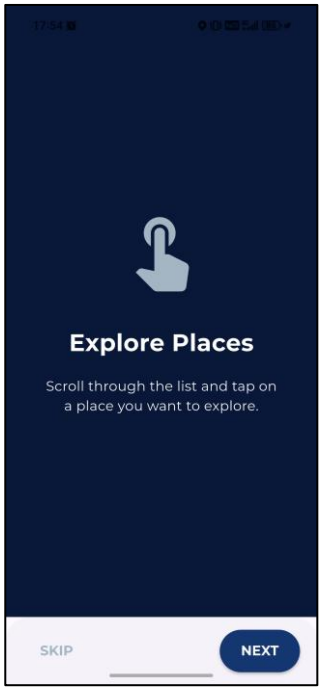


Figure 5.3 Onboarding Pages – Explore Places

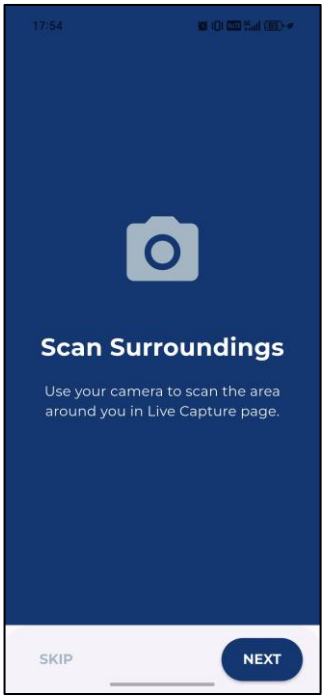


Figure 5.4 Onboarding Pages – Scan Surroundings

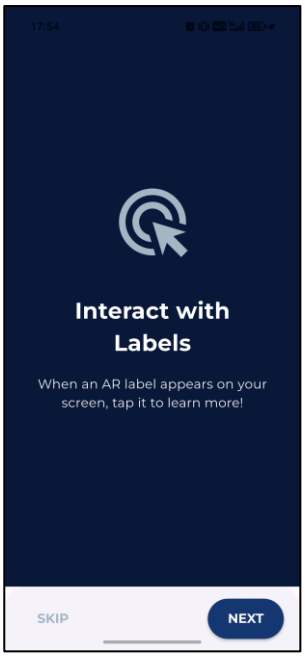


Figure 5.5 Onboarding Pages – Interact with Labels

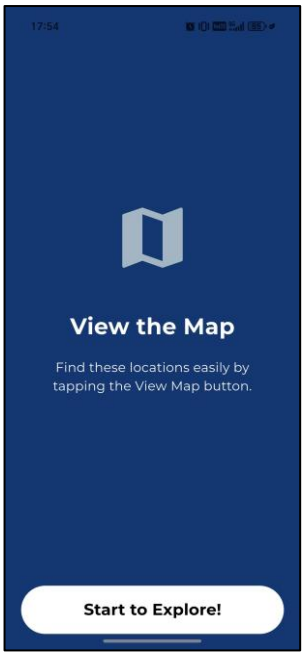


Figure 5.6 Onboarding Pages – View Map

5.3.3 Home Page

Figure 5.7 Home Page

Home page was designed to display a list of place cards, including a floating action button "View Map". The CustomClipper component was used to create the curved paths [11].

5.3.4 Live Capture Page

1. Camera View

Live Capture page uses a live camera feed for attraction spot recognition, displaying the place name help users identify their location. For the recognition features, the backend logic was optimised by separating places, the frame captured will only compare with the relevant reference images and texts. The system captures a frame every 2 seconds, displaying live camera feed while silently captures in the background.

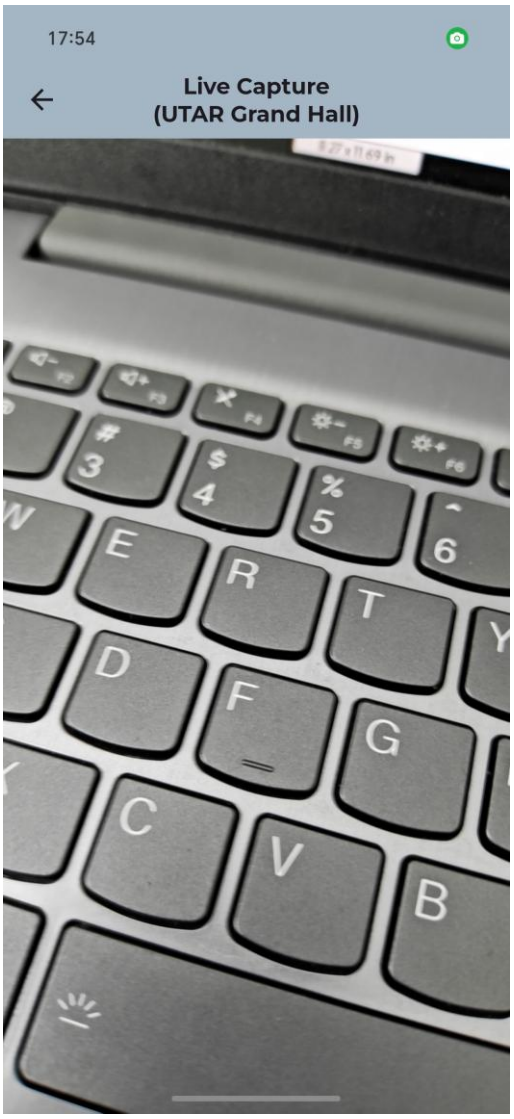


Figure 5.8 Live Capture Page

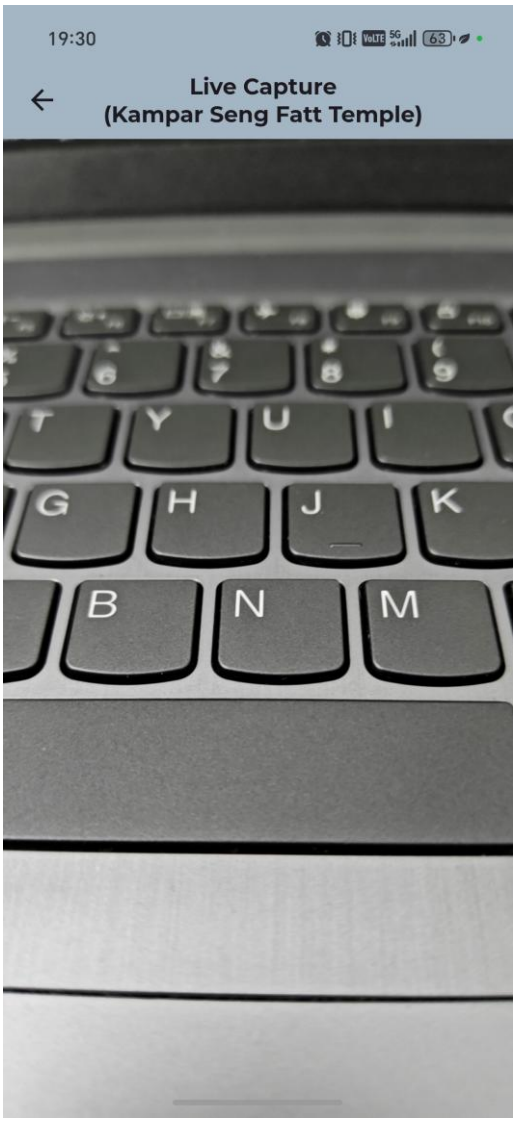


Figure 5.9 Live Capture Page

2. Loading Messages

To enhance the system visibility and user experience, loading message such as “Analysing environment.” and “Waking up server…” were implemented. These messages proactively inform users that the server is starting up, providing clear feedback preventing confusion.

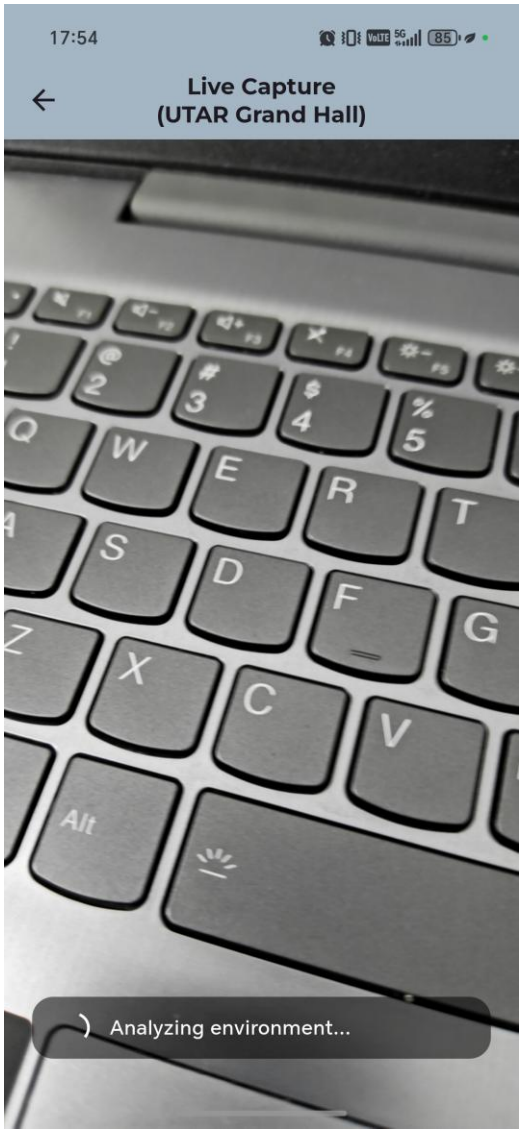


Figure 5.10 Analysing Environment Message

Figure 5.11 Waking Up Server Message

3. Labels Display

Tappable floating labels were designed to display the names of attraction spots. The labels appear on screen as soon as a spot is detected, users can tap the label to view detailed information. To prevent confusion, the labels will be cleared automatically after 5 seconds of inactivity. Additionally, tapping a label will also clear the labels on screen when navigating users to the webpage, ensuring that when users return to the live capture page, the labels are not misaligned to the current live feed.

Figure 5.12 Labels Display

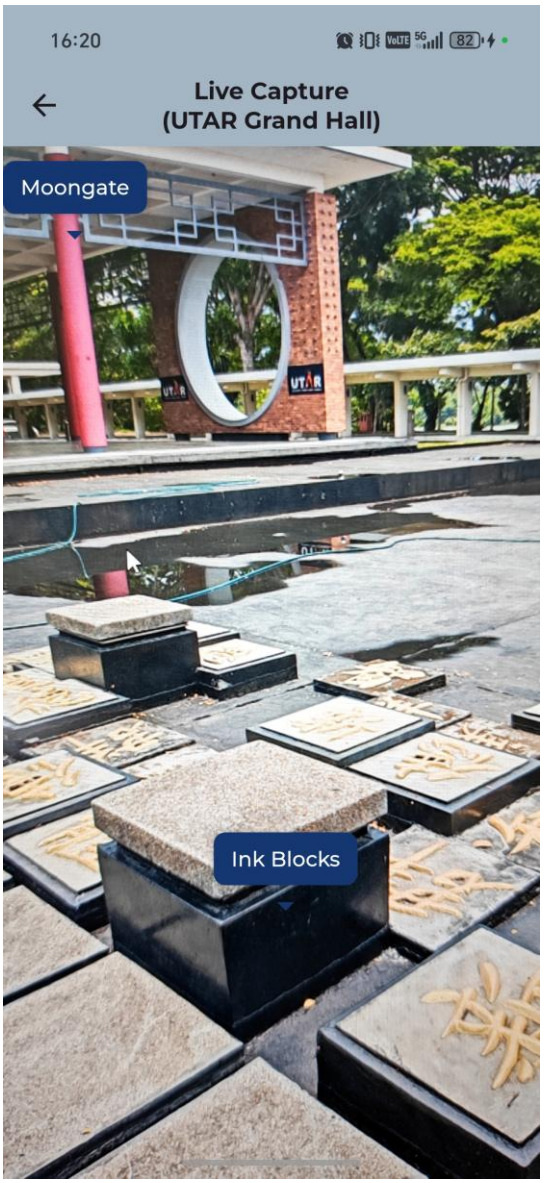


Figure 5.13 Labels Display

4. URL Open in the In-App Browser

To provide a smoother user experience, attraction spot webpages open within an in-app browser using Flutter's url_launcher package [12], allows users to read the details without leaving the application. After viewing the information, users can close the webpage and resume the live capture.

Figure 5.14 In-App Browser View

Figure 5.15 In-App Browser View

5.3.5 View Map

The additional “View Map” feature allows users to view the location of the places. The button redirects users to a shared Google Map list, by opening the URL externally. Utilizing the Google Map list allows users to navigate to these places easily, without implementing a native in-app map.

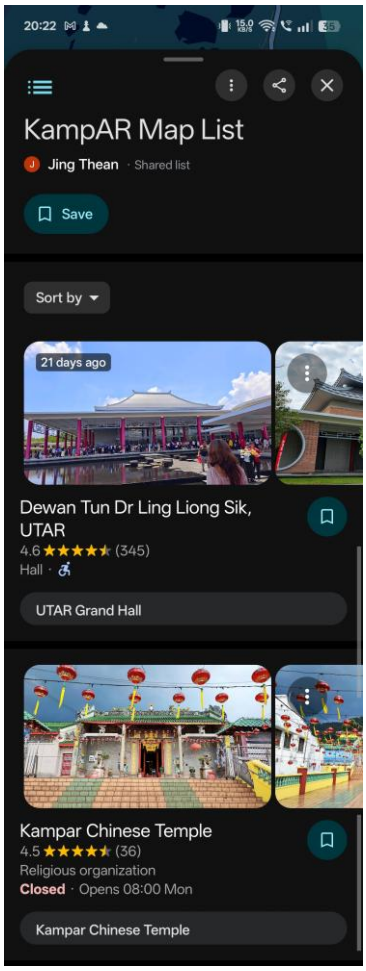


Figure 5.16 Google Map List

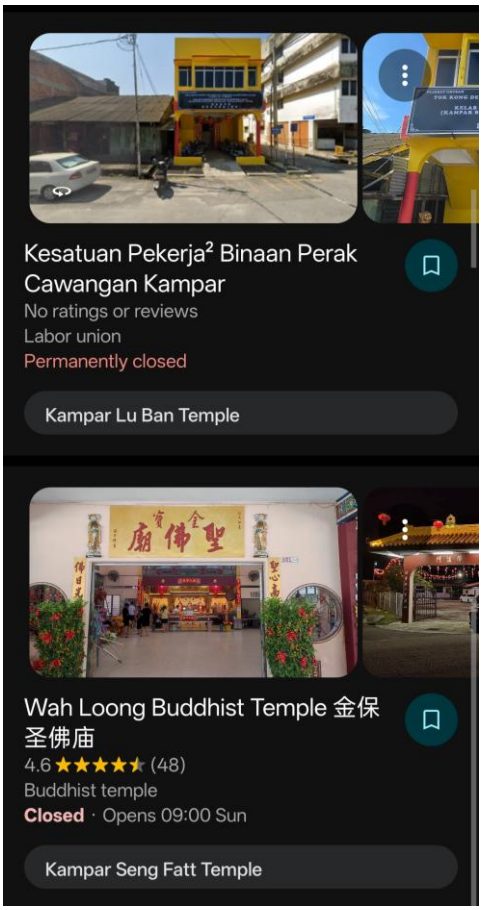


Figure 5.17 Google Map List

5.4 Model Integration

Figure 5.18 illustrates the recognition pipeline integrating **two pre-trained models**: **FastSAM** and **CLIP**. The frame sent from the application will first be processed by FastSAM to detect and crop the objects in the frame. While CLIP used to handle similarity computation for every single image cropped. The output of the recognition pipeline will be the similarity scores.

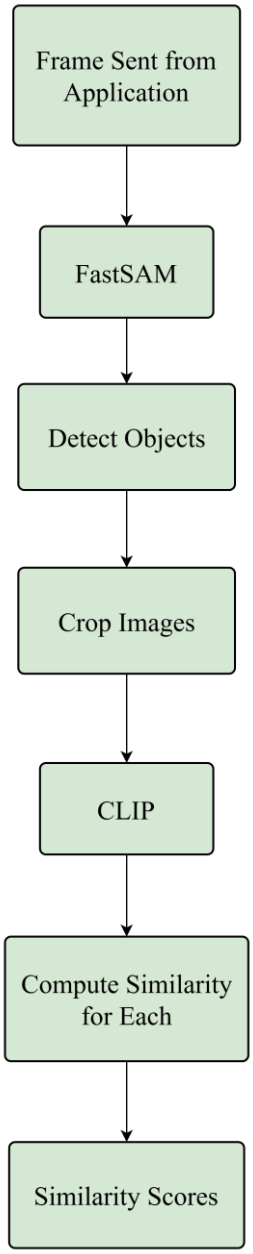


Figure 5.18 Recognition Pipeline

5.4.1 FastSAM

Fast Segment Anything Model (FastSAM) is a Convolutional Neural Network (CNN)-based vision model [13]. In this project, FastSAM was integrated into the recognition pipeline specifically for **object detection** tasks. It was implemented utilising its zero-shot image segmentation capabilities. The model automatically performed segmentation without being provided any reference images, text prompts or prior training sample. Parameters of FastSAM were set instead of using default values to optimise performance for object detection task for this project. As shown in Figure 5.19, the "**conf**" (confidence) and "**iou**" (Intersection over Union) were set to **0.6** and **0.45** respectively. Default value of "conf" is 0.4, the increase to 0.6 will show only high confidence detections. While default value of "iou" is 0.9 means fewer overlapping boxes are removed, decreasing the value will remove more overlapping boxes.

```
fastsam_results = fastsam_model(img, conf=0.6, iou=0.45)
```

Figure 5.19 Setting Confidence and Intersection over Union Thresholds for FastSAM

5.4.2 CLIP

The Contrastive Language-Image Pretraining (CLIP) model is a multimodal architecture designed to learn visual concepts from natural language supervision. It utilises a ViT-B/32 Transformer architecture as image encoder and a masked self-attention Transformer as text encoder, which is trained to maximise the similarity of (image, text) pairs through a contrastive loss function [14]. CLIP was integrated into the recognition pipeline to perform **image classification** using Few-Shot Learning (FSL) techniques [15]. For each attraction spot, the model was provided with number of images ranging from 5 to 15. To perform classification, the model computes the cosine similarity between the generated embeddings to output a confidence score. In this project, the final similarity score is calculated using a weighted combination of image and text prompt, as shown in the following equation:

$$\boldsymbol{Similarity\ Score} = (\mathbf{0.7} * \boldsymbol{I}) + (\mathbf{0.3} * \boldsymbol{T})$$

where I represents the **image embedding similarity score** and T represents **text prompt embedding similarity score**.

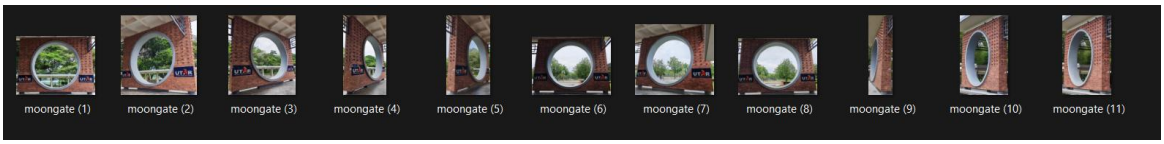


Figure 5.20 Example of Reference Images for Moongate

5.5 Cloud Database and Server

5.5.1 Firebase

Firebase is a Backend-as-a-Service (BaaS) platform that helps to accelerate application development by providing fully managed backend infrastructure. In this project, Cloud Storage and Firestore was used to handle images and data management.

1. Cloud Storage

Firebase Cloud Storage was utilised to manage **reference images**. It built with secure Google Cloud infrastructure, provides robust object storage to store images and videos [16]. Figure 5.21 to Figure 5.23 show the structure of Cloud Storage, demonstrating how images were stored and categorised based on attraction spots and places.

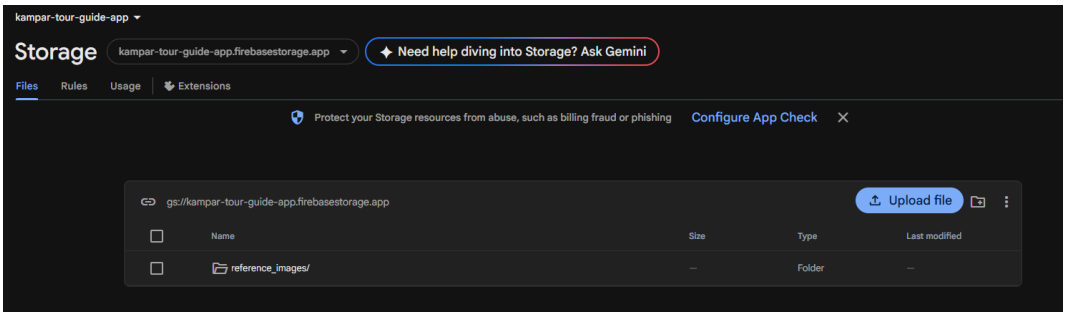


Figure 5.21 Cloud Storage Structure

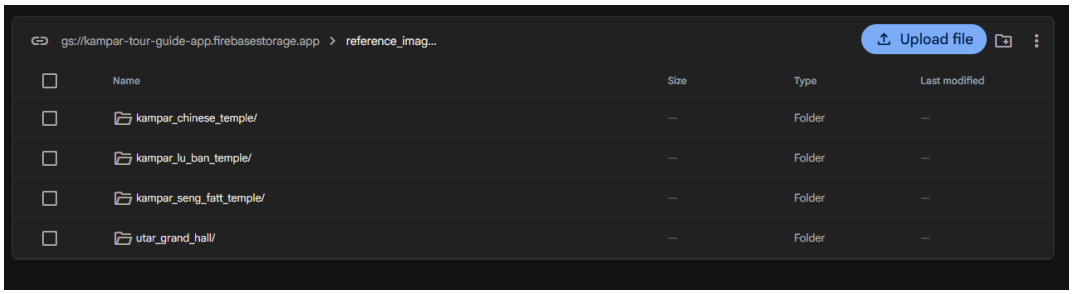


Figure 5.22 Cloud Storage Structure

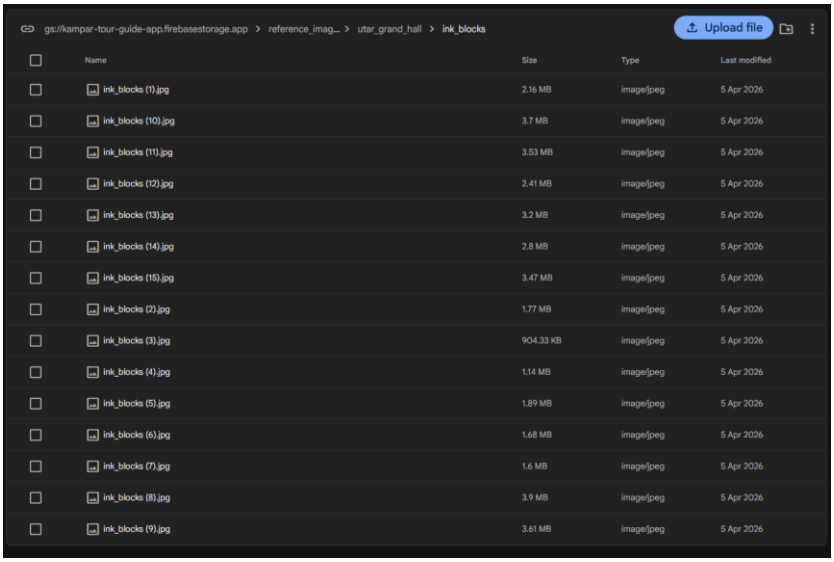


Figure 5.23 Cloud Storage Structure

2. Firestore

Firestore is a scalable and flexible NoSQL database, it offers seamless integration with other services such as Google Cloud Run [17]. In this project, Firestore was used to store **structured text data,** including text prompts, labels and URLs. Figure 5.24 to Figure 5.26 show the structure of Firestore. The data stored is dynamically retrieved by the cloud-hosted backend and subsequently passed to frontend application for display and user interaction.

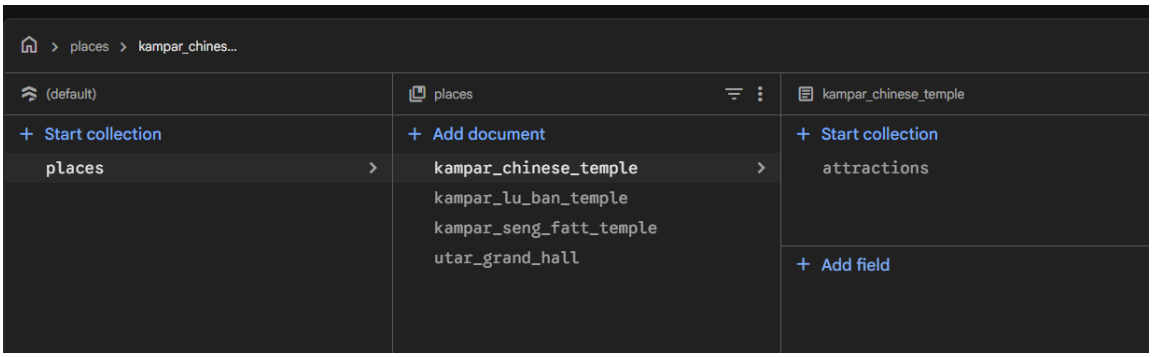


Figure 5.24 Firestore Structure

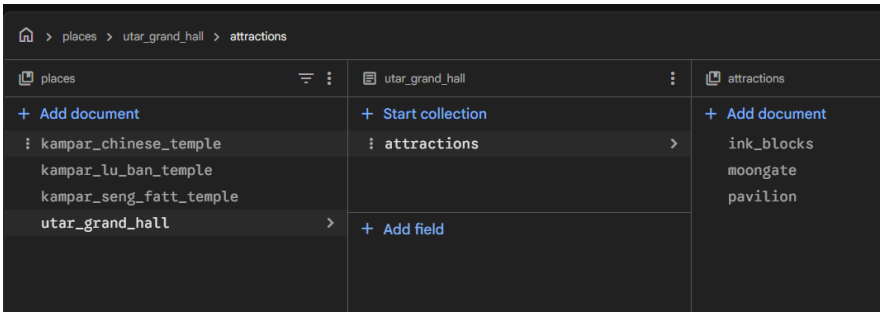


Figure 5.25 Firestore Structure

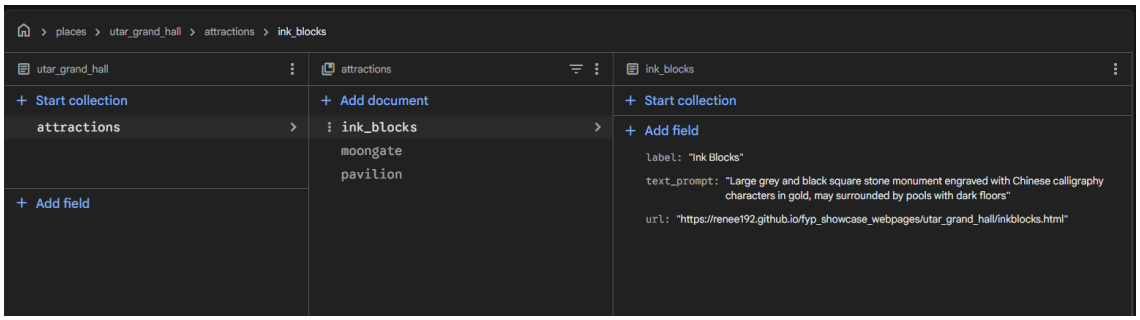


Figure 5.26 Firestore Structure

5.5.2 Google Cloud Run

Google Cloud Run is a fully managed, serverless compute platform designed to execute containerised applications. It allows automatic scaling while automatically provides secure, custom HTTPS endpoints [18]. In this project, Cloud Run utilised as the hosting environment for the backend of application. It was used to **deploy project's Dockerfile** and manage resource allocation such as CPU and memory limits, while also utilised Cloud Run "revision" feature to update the changes.

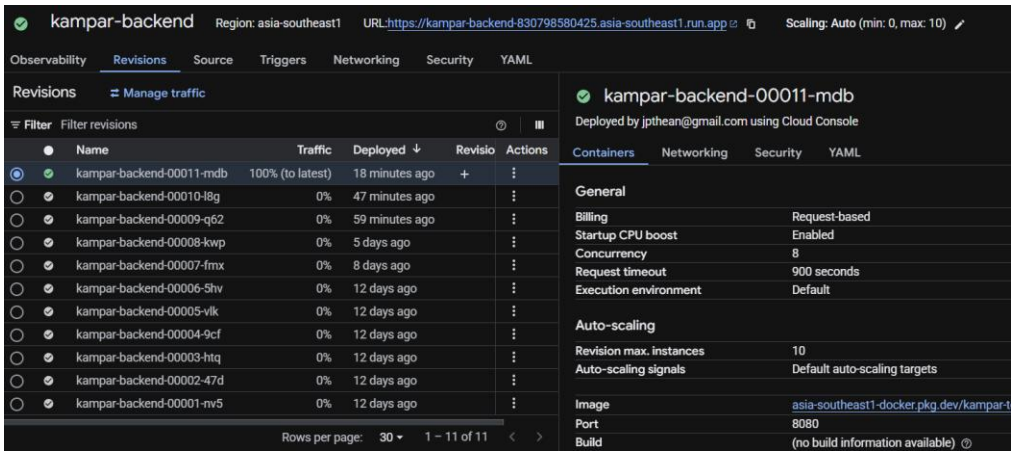


Figure 5.27 Revisions

5.6 Implementation Issues and Challenges

During the implementation of the recognition pipeline, several issues and challenges were encountered and addressed to improve the recognition accuracy.

1. **Redundant Object Detections**

   One of the challenges during the implementation of the recognition pipeline was the object detection model (FastSAM) generating excessive, overlapping bounding boxes for a single object. By setting the parameters “conf” to 0.6 and “iou” to 0.45, which are Intersection over Union (IoU) and confidence score threshold, the object detection part was optimised.

2. **Sensitivity to Background Noise in Classification**

   For the image classification model (CLIP), it is sensitive to the framing and quality of the input image. As the classification process relies on computing similarity scores between images, it requires tightly cropped images of the attraction spot. Excessive background noise may negatively affect accuracy of similarity computations.

Chapter 6

System Evaluation and Discussion

6.1 Use Case Testing

Test Cases 1 and 2 were set up for first use case in Chapter 3 (Select place), with Test Case 1 for normal flow and Test Case 2 for alternate flow.

Test Case ID	T1	**Related Use Case ID**	1	
Use Case	**Test Case**	**Expected Result**	**Actual Result**	**Pass/Fail**
Select place (Normal Flow)	1. Launch the application.	System display Home page.	System display Home page successfully.	Pass
	2. Scroll the list of places	List scrolls smoothly.	Matches expected result.	Pass
	3. Tap to select a place.	System navigates user to Live Capture page.	User navigated to the Live Capture page.	Pass

Table 6.1 Test Case 1

Test Case ID	T2	**Related Use Case ID**	1	
Use Case	**Test Case**	**Expected Result**	**Actual Result**	**Pass/Fail**
Select place (Alternate Flow 2a)	1. Install app and launch it for the first time.	System displays Onboarding pages instead of Home page.	System displays Onboarding pages.	Pass

Table 6.2 Test Case 2

Test Cases 3 to 6 were set up to test the second use case in Chapter 3 (Perform attraction spot recognition). Test Case 3 is for normal flow, while Test Cases 4, 5 and 6 are for alternate flows.

Test Case ID	T3	**Related Use Case ID**	2	
Use Case	**Test Case**	**Expected Result**	**Actual Result**	**Pass/Fail**
Perform attraction spot recognition (Normal Flow)	1. Select a place.	System displays Live Capture page with camera view.	System display camera live feed.	Pass
	2. Scan surroundings.	System captures frame every 2s and send to server silently.	Matches expected result.	Pass
	3. Wait for system processing.	System returns prediction results.	Matches expected result.	Pass
	4. View screen results.	System displays labels with attraction spot names.	System displays labels on spot detected.	Pass

Table 6.3 Test Case 3

Test Case ID	T4	**Related Use Case ID**	2	
Use Case	**Test Case**	**Expected Result**	**Actual Result**	**Pass/Fail**
Perform attraction spot recognition (Alternate Flow 2a)	1. Revoke camera permission for the app in device settings			Pass
	2. Select a place.	System detects and display permission denied message.	Matches expected result.	Pass

Table 6.4 Test Case 4

Test Case ID	T5	**Related Use Case ID**	2	
Use Case	**Test Case**	**Expected Result**	**Actual Result**	**Pass/Fail**
Perform attraction spot recognition (Alternate Flow 4a)	1. Select a place and scan surroundings			Pass
	2. System attempts to send frame to server.	Server is not ready, system displays loading messages.	System displays loading messages.	Pass

Table 6.5 Test Case 5

Test Case ID	T6	**Related Use Case ID**	2	
Use Case	**Test Case**	**Expected Result**	**Actual Result**	**Pass/Fail**
Perform attraction spot recognition (Alternate Flow 5a)	1. Select a place and point camera to blank wall.			Pass
	2. System sends frame to server.	System silently returns to Step 4 in Normal Flow (capture frames) without showing labels.	Matches expected result.	Pass

Table 6.6 Test Case 6

Lastly, Test Case 7 was set up for third use case (View attraction spot details) to test the normal flow.

Test Case ID	T7	**Related Use Case ID**	3	
Use Case	**Test Case**	**Expected Result**	**Actual Result**	**Pass/Fail**
View attraction spot details (Normal Flow)	1. Tap a label on Live Capture page.	System opens URL in an in-app browser view.	System opens in-app browser view.	Pass
	2. View the details information.	Webpage loads and display.	Webpage displays.	Pass
	3. Close the webpage.	System returns user to Live Capture page.	Matches expected result.	Pass

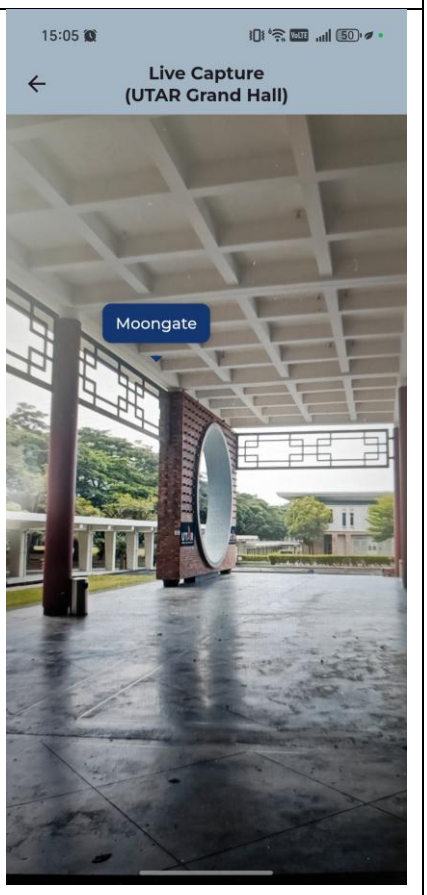
Table 6.7 Test Case 7

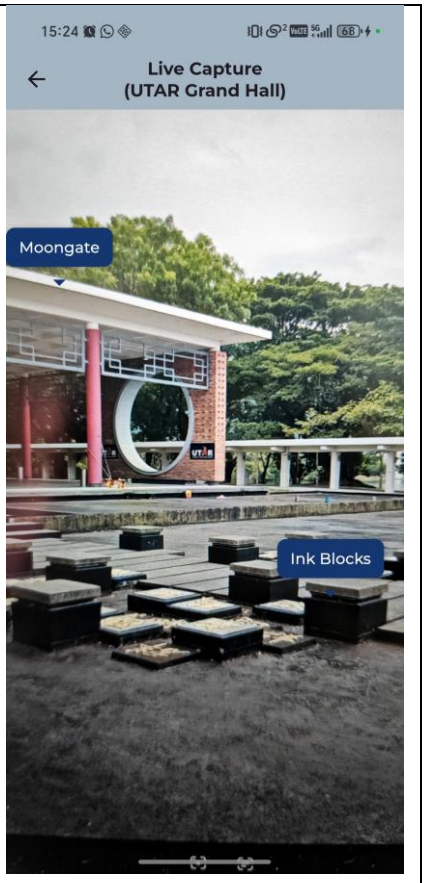
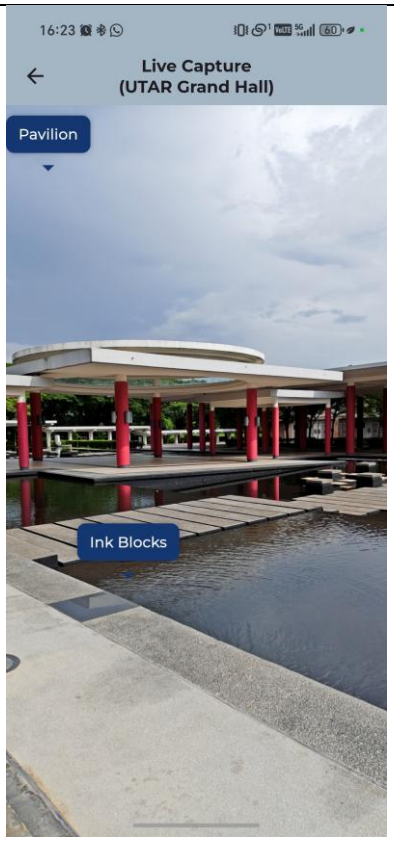
6.2 Real-World Testing

To test the application in real-world scenarios, evaluations were conducted for two locations: **UTAR Grand Hall** and **Kampar Seng Fatt Temple**. For each test case, the input image, expected outcome and the actual result were tabulated as shown in Table 6.8 and Table 6.9.

The **input images** are the images fed into the recognition system, **expected result** is the correct identification of the attraction spot while the **actual result** is the real-time output, captured via application screenshots.

6.2.1 Recognition Results for UTAR Grand Hall

Input Image	Expected Result	Actual Result
	Moongate	
	Moongate	

	Moongate Ink Blocks	
	Pavilion Ink Blocks	

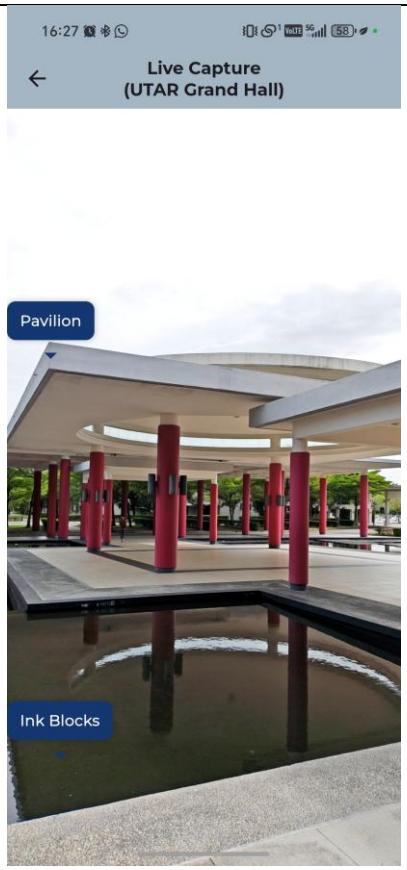
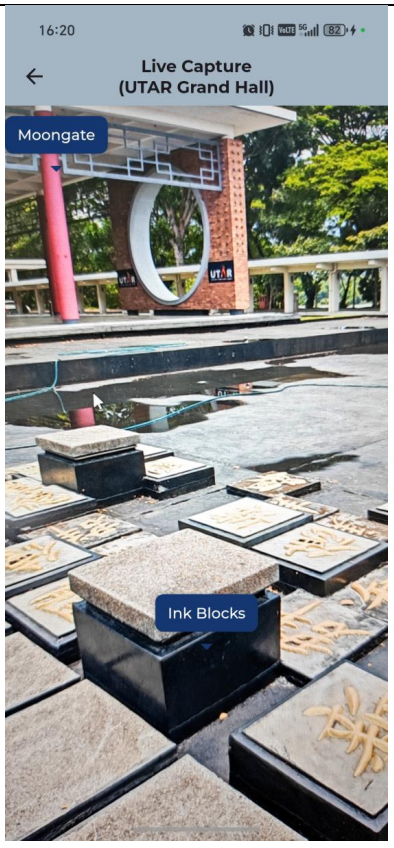
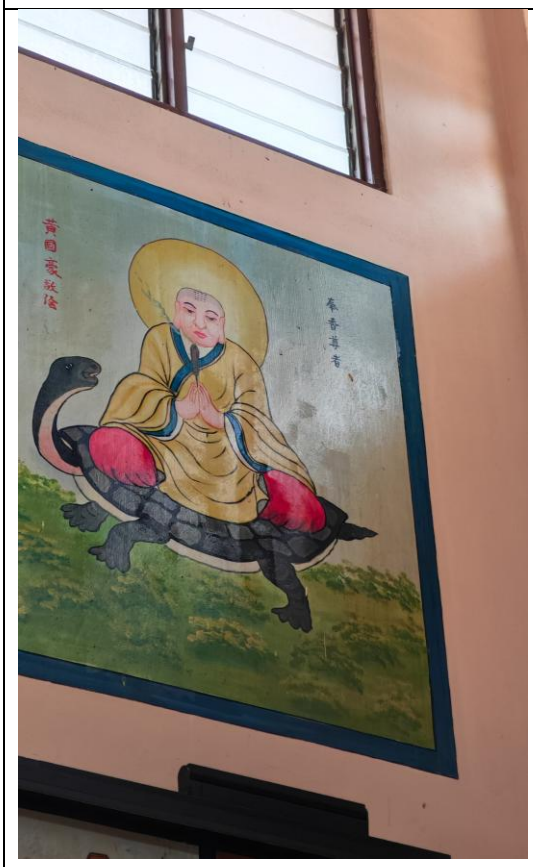
	Pavilion Ink Blocks	
	Moongate Ink Blocks	

Table 6.8 Real-World Testing Results – UTAR Grand Hall

6.2.2 Recognition Results for Kampar Seng Fatt Temple

Input Image	Expected Result	Actual Result
	出巡牌區	
	出巡牌區	

	奉香尊者	 14:17 Live Capture (Kampar Seng Fatt Temple) 奉香尊者
	奉香尊者	 14:18 Live Capture (Kampar Seng Fatt Temple) 奉香尊者

	弥勒尊者	 09:41 Live Capture (Kampar Seng Fatt Temple) 弥勒尊者
	福禄财神图	 10:04 Live Capture (Kampar Seng Fatt Temple) 福禄财神图

	福禄财神图 七仙下凡	
	白无常	

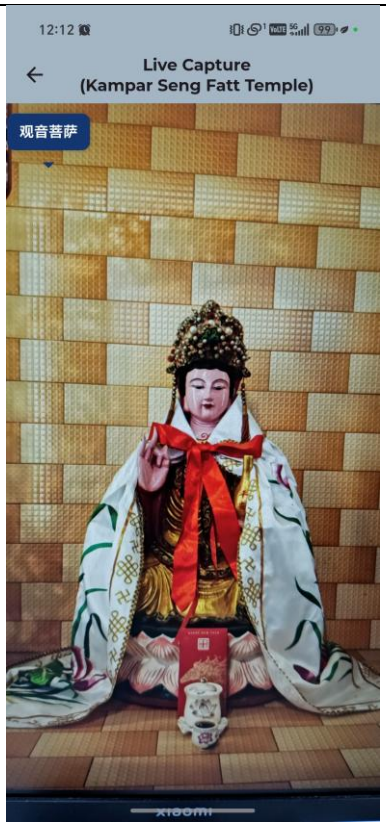
	观音菩萨	
	关帝圣君	

Table 6.9 Real-World Testing Results – Kampar Seng Fatt Temple

6.3 Overall Performance and Objectives Evaluation

Based on the testing results indicates that the three objectives mentioned in Chapter 1 have been achieved successfully.

1. To develop a mobile tour guide application for Kampar
2. To enhance tourism experience and provide cultural benefits
3. To integrate pre-trained models for recognition

The objectives were achieved by the successful development and deployment of the mobile application. The application is specifically designed for locations in Kampar. Furthermore, the testing results verified the interactive components function as expected. The overlays allow users to interact with the application to access cultural information and enhance their tourism experience.

Additionally, the application successfully integrated pre-trained models for recognition feature. As demonstrated by the real-world testing, the models are capable of accurately identifying attraction spots without the need to collect large datasets or extensive manual image labelling. By leveraging the pre-trained models, the application achieved reliable recognition capabilities using a small number of images, ranging from 5 to 15, along with a text prompt for each attraction spot.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

In conclusion, this project successfully developed a mobile tour guide application with real-time attraction spot recognition. The recognition feature was successfully achieved by integrating two pre-trained models: FastSAM for object detection and CLIP for image classification. This outcome of this project demonstrate that pre-trained models possess zero-shot and few-shot learning capabilities, proving that they can be integrated into applications without the need of large, manually labelled datasets.

The developed application provides an engaging platform for tourists and locals to access and understand cultural information interactively. Lastly, this project highlights the leverage of pre-trained models leads to a way for smart tourism and efficient application development.

7.2 Recommendation

Although the mobile tour guide application was developed successfully, there are still several areas for future development. Regarding user experience, adding haptic feedback for label tapping would provide more interactive experience. Additionally, the current 2D label display could be replaced with interactive 3D models. However, implementing 3D models would increase the application's size, could sacrifice its lightweight nature. For recognition, the models used in this project were not fine-tuned. Therefore, fine-tuning the models with small and specialised dataset in future could increase the accuracy and better optimise the system for recognition task.

REFERENCES

[1] A. French, “Why Visit Historical Places?,” Roaming Historian, Sep. 13, 2020. https://roaminghistorian.com/2020/09/13/why-visit-historical-places/

[2] K. W. Teh, “Mobile tour guide application with on-site attraction recognition for Kampar Temples - UTAR Institutional Repository,” Utar.edu.my, 2023, doi: http://eprints.utar.edu.my/5522/1/fyp_CS_2023_TKW.pdf.

[3] Z. Y. Ooi, “Mobile tour guide application with on-site augmented reality - Kampar churches - UTAR Institutional Repository,” Utar.edu.my, 2024, doi: http://eprints.utar.edu.my/6671/1/fyp_CS_2024_OZY.pdf.

[4] Google, “Google Lens - Search What You See,” Google Lens. https://lens.google/howlensworks/

[5] “What is the technology behind Google Lens?,” Milvus.io, 2025. https://milvus.io/ai-quick-reference/what-is-the-technology-behind-google-lens

[6] N. M. Firdaus, D. Chahyati and M. I. Fanany, “Tourist Attractions Classification using ResNet,” 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2018, pp. 429-433.

[7] B.K. Seo, K. Kim, and J.I. Park, “Augmented reality-based on-site tour guide: A study in Gyeongbokgung,” in Computer Vision – ACCV 2010 Workshops, 2010, pp. 276–285, doi: 10.1007/978-3-642-22819-3_28.

[8] Y. Li, X. Li, and C. Yue, “Recognition of Tourist Attractions.” Available: https://cs229.stanford.edu/proj2016/report/LiLiYue-RecognitionOfTouristAttractions-report.pdf

[9] A. Asfour, S. Zain, N. Salleh, and J. Grundy, “Exploring Agile mobile app development in Industrial Contexts: A Qualitative study,” 2019. https://ijtes.net/index.php/ijtes/article/view/1573

[10] Flutter, “Flutter - Beautiful native apps in record time,” *Flutter.dev*, 2024. https://flutter.dev/

[11] “CustomClipper class - rendering library - Dart API,” Flutter.dev, 2026. https://api.flutter.dev/flutter/rendering/CustomClipper-class.html

[12] “url_launcher | Flutter Package,” Dart packages. https://pub.dev/packages/url_launcher

[13] Ultralytics, “Fast Segment Anything Model (FastSAM),” Ultralytics.com, 2023. https://docs.ultralytics.com/models/fast-sam/#overview

[14] "CLIP," Huggingface.co, 2017. https://huggingface.co/docs/transformers/en/model_doc/clip

[15] A. Data, "Few-Shot and Zero-Shot Learning: Teaching AI with Minimal Data," Medium, Jun. 21, 2024. https://medium.com/@sahin.samia/few-shot-and-zero-shot-learning-teaching-ai-with-minimal-data-801603ed40f8

[16] Firebase, "Cloud Storage | Firebase," Firebase, 2019. https://firebase.google.com/docs/storage

[17] Firebase, "Cloud Firestore | Firebase," Firebase, 2019. https://firebase.google.com/docs/firestore

[18] "What is Cloud Run," Google Cloud Documentation, 2026. https://docs.cloud.google.com/run/docs/overview/what-is-cloud-run?_gl=1

POSTER

PANG JING THEAN, 23ACB01046
SUPERVISOR: PROF TS DR LIEW SOUNG YUE

MOBILE TOUR GUIDE APPLICATION

WITH **REAL-TIME ATTRACTION SPOTS RECOGNITION**
USING **VLM** AND **AR** FOR KAMPAR

INTRODUCTION

This project aims to Enhance the cultural tourism experience in Kampar, Perak, by providing an interactive and informative mobile tour guide application to bridge the gap in historical knowledge and natural attractions.

OBJECTIVE

1.To develop a mobile tour guide application for Kampar
2. To enhance tourism experience and provide educational benefits
3. To integrate pre-trained models for recognition

METHODS

1. Leverages pre-trained models such as Vision-Language Model capability with Few Shot learning to achieve attraction spot recognition.
2. Incorporate Augmented Reality technology, providing immersive experience.

CONCLUSION

By leveraging modern technology in a tourism recognition application it can foster a deeper connection between visitors and Kampar's rich cultural heritage.