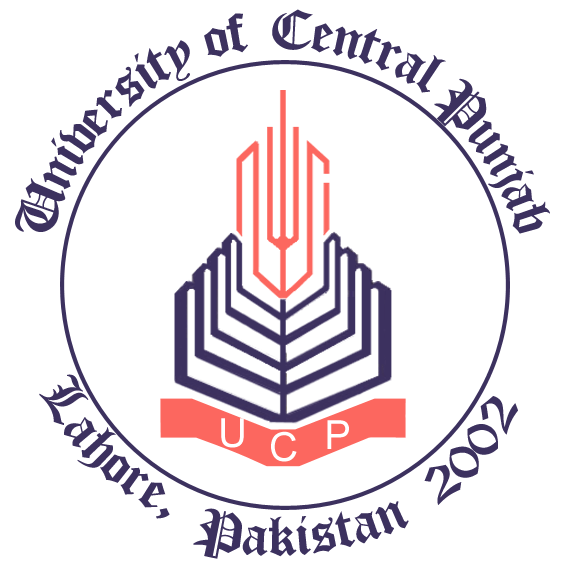
BSCS FINAL PROJECT

Landmark Detection



Advisor: DR. ADNAN N QURESHI

Presented by:

**Group ID: F19BS048**

Student Reg# Student Name

L1F16BSCS0228 MUHAMMAD SUFYAN ASHRAF

L1F16BSCS0255 NOMAN AFZAL

L1F16BSCS0435 SAAD ASIF

**Faculty of Information Technology**

**University of Central Punjab**

**Landmark Detection**

By

MUHAMMAD SUFYAN ASHRAF

NOMAN AFZAL

SAAD ASIF

Thesis/Project submitted to

Faculty of Information Technology,

University of Central Punjab,

Lahore, Pakistan.

in partial fulfillment of the requirements for the degree of

**BACHELOR OF SCIENCE**

**IN**

**COMPUTER SCIENCE**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Project Advisor |  | Manager Projects |

**Abstract**

In today’s world computers are much smarter than humans but there are things where humans are still superior such as Image Classification and object recognition. These are developing fields. Are these fields reach the point to automate those jobs where humans are good at? Tourist guides are generally human and like all humans, they make mistakes like spreading false information and facts due to their lack of knowledge. To stop the spread of false information and provide accurate historical information and facts, landmark detection came into existence to create a digital tourist guide. Initially, data of the historical landmark was gathered. For the image classification to detect historical landmark different machine learning models were trained and tested. In the end, the best performing model (VGG19) was selected. To improve its accuracy different computer vision techniques like data augmentation were used. After obtaining the above 90 percent accuracy we start developing our application. All historical information is gathered through authentic resources. In conclusion, we are able to build an application that is almost as good as a human tourist guide in terms of image classification and provides authentic information about historical landmarks.

**Dedicated to**

**My beloved Parents**

**And**

**My all Teachers**

**Acknowledgements**

All praise to Allah Almighty, who is the creator of this universe, who guides us in the ocean of darkness. Who showers His blessings upon us, which enable us to accomplish this project. All respect and love for Holy Prophet Muhammad (PBUH) who enable us to recognize our creator.

I found no words to describe sweet sensation for my parents, especially my mother who has been a source of inspiration and encouragement to me throughout my life. My parent’s guidance, support, and encouragement are the basis of my all achievements. May Allah Almighty shower His Blessings upon my parents and my family.

The most sincere and deep sense of obligation is to our Advisor, Dr. Adnan N Qureshi, for his friendly advice, noble guidance and sympathy attitude during the whole project. Due to his vast experience and cooperation, we became able to complete our project. He is very hardworking and knowledgeable.

**List of Figures**

Figure 3.1 Use-case diagram

Figure 4.1 Architecture Diagram 1

Figure 4.2 Architecture Diagram 2

Figure 4. 3 Activity Diagram

Figure 4.4 Component Diagram

Figure 4.5 Class Diagram

Figure 4.6 ER Diagram

Figure 4.7 ER Tables

Figure 4.8 Context Diagram

Figure 4.9 Zero level Diagram

Figure 4.10 Level one Diagram Expanding Process 5

Figure 4.11 Level one Diagram Expanding Process 2

Figure 4.12 Level one Diagram Expanding Process 6

Figure 4.13 Sequence Diagram

Figure 4.14 Collaboration Diagram

Figure 5.1 Swimlane diagram

Figure 5.2 Camera Screen

Figure 5.3 Detected successfully Screen

Figure 5.4 Details Screen

Figure 5.5 unsuccessful scan Screen

Figure 5.6 Recommending food Screen

Figure 5.7 Recommending place to visit Screen

Figure 5.8 Training Accuracy vs Validation Accuracy Graph

Figure 5.9 Training Loss vs Validation Loss Graph

Figure 8.1 User Manual Screen 1

Figure 8.2 User Manual Screen 2

Figure 8.3 User Manual Screen 3

Figure 8.4 User Manual Screen 4

Figure 8.5 User Manual Screen 5

**List of Tables**

Table 3.1                     Use Case 1

Table 3.2                     Use Case 2

Table 3.3                     Use Case 3

Table 3.4                     Use Case 4

Table 4.1                     Decision Table 1

Table 4.2                     Decision Table 2

Table 4.3                     Decision Table 3

Table 5.1                     Confusion matrix

Table 6.1                     Test Case 1

Table 6.2                     Test Case 2

Table 6.3                     Test Case 3

Table 6.4                     Test Case 4

Table 6.5                     Test Case 5

Table 6.6                     Summary of All Test Results

Table 7.1                     Project Completion Status

Table 7.2                     Objective(s)/Target(s) Status

Table of Contents

Abstract i

Dedication ii

Acknowledgements iii

List of Figures iv

List of Tables vi

List of Illustrations vii

Table of Contents viii

Chapter 1. Introduction 1

1.1 Product (Problem Statement) 1

1.2 Background 1

1.3 Objective(s)/Aim(s)/Target(s) 2

1.4 Scope 2

1.5 Business Goals 2

1.6 Challenges 3

1.7 Learning Outcomes 3

1.8 Nature of End Product 3

1.9 Related Work/ Literature Survey/ Literature Review 3

1.10 Document Conventions 4

Chapter 2. Overall Description 5

2.1 Product Features 5

2.2 Functional Description 5

2.3 User Classes and Characteristics 5

2.4 Design and Implementation Constraints 5

2.5 Assumptions and Dependencies 5

Chapter 3. System Requirements 6

3.1 Functional Requirements 6

3.1.1 Name of Use-Case 1 6

3.1.2.1 Name of Use-Case 2 7

3.1.2.2 Name of Use-Case 3 8

3.1.2 Name of Use-Case 4 9

3.1.3 Requirements Analysis and Modeling 10

3.2 Nonfunctional Requirements 11

3.2.1 Performance Requirements 11

3.2.2 Safety Requirements 11

3.2.3 Security Requirements 11

3.2.4 Additional Software Quality Attributes 11

3.3 Other Requirements 11

Chapter 4. Technical Architecture 12

4.1 Application and Data Architecture 13

4.2 Component Interactions and Collaborations 18

4.3 Design Reuse and Design Patterns 22

4.4 Technology Architecture 22

4.5 Architecture Evaluation 23

Chapter 5. Detailed Design and Implementation 24

5.1 Component-Component Interface 24

5.2 Component-External Entities Interface 24

5.3 Component-Human Interface 24

5.4 Screenshots/Prototype 25

5.4.1 Workflow 25

5.4.2 Screens 26

5.5 Additional Information 32

5.6 Other Design Details 32

Chapter 6. Test Specification and Results 34

6.1 Test Case Specification 34

6.2 Summary of Test Results 39

Chapter 7. Project Completion Status/Conclusion 40

References 42

Appendix A Glossary 43

Appendix C Deployment/Installation Guide 44

Appendix D User Manual 45

# Introduction

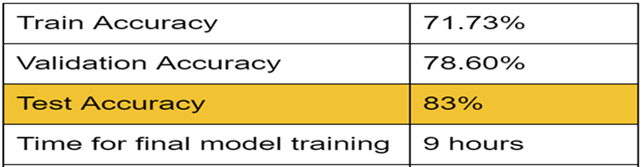
## Product (Problem Statement)

While traveling people are very curious to explore new places. Some of the time they do not know the landmark that they are visiting or most of the people take pictures in front of the famous landmarks, but they don’t know about it. This software will help the people to know the name and information of those landmarks. The application is tailored towards the tourist who wants to go to historical places and try the cuisine. The recommendation of places to visit next and food to try is also added in the application. The software is an android application. The user is able to take a picture of the objects and that will be automatically sent for the processing which then gives name and information about that landmark.

## Background

Previously, Catherine McNabb and his team used transfer learning for the detection of a landmark **[3]**. They attempted to train CNNs to achieve good accuracy. For this purpose, the Google Cloud Platform was used to use machines with the required capabilities. First, the training image dataset was split into training, validation, and test images, and images were resized before downloading to make the challenge more tractable. For model training, the VGG 16 Neural Network was used with Transfer Learning from ImageNet. They tried variations in the number of layers, optimizers, hyperparameters, image augmentation, batch size, and the number of epochs to improve upon the validation accuracy. And then finally, they used DeLF to deal with the actual test dataset which consisted mostly of unrelated images.

Results



K.Ozaki and S.Yokoo. used resNet for the detection of a landmark **[1]**. Their approach is based on deep convolutional neural networks with metric learning, trained by cosine-softmax based losses. Deep metric learning methods are usually sensitive to noise, and it could hinder learning a reliable metric. To address this issue, they develop an automated data cleaning system. Besides, they devise a discriminative re-ranking method to address the diversity of the dataset for landmark retrieval. Using these methods, they achieved 1st place in the Google Landmark Retrieval 2019 challenge on Kaggle. Both of these teams work to identify if the image is of a landmark or not but our model will predict which landmark it is.

Google image search **[4]** is a tool provided by Google that tries to find the closest image to the input image. It is publicly available, but its accuracy is not so good. The photo location is another tool to find the image location. If the EXIF file contains the location, then we will see latitude and longitude details. But we are not using the GPS systems in our application, so we will visually recognize the image with better accuracy. The visual positioning system is another tool provided by Google. It is close to our system, but it is mainly used for map navigation **[5]**.

## Objective(s)/Aim(s)/Target(s)

Our Application is now able to tell the name and information of the particular landmark. given the input image of any landmark taken by the application, the system can determine the image correctly with the maximum accuracy. The accuracy is more than 85%. The Application is good enough to differentiate between the replica and the original landmark. Currently, our application is limited to detect some of the famous landmarks in Pakistan.

## Scope

Tourism is a fast-growing industry. In 2018, 1.401 billion international tourist arrivals worldwide generated 1.7 trillion dollars. From 2017 to 2018 the tourism industry grew 4%.[2] With the growing tourism, more and more people are traveling and reaching obscure landmarks. They need help knowing about them. Getting local guides for each place is not economical and people also do not want a guide as their privacy is affected. Our application not only identifies landmarks but also provides automatic annotation of it. Historical information will also be available.

## Business Goals

We are introducing our software as a product. We can sell our product to the people involved in tourism. we can sell the product to any organization; they will make it reach the worldwide market. This app can also directly sell to users.

There is no such application yet in our country, so the business scope is big. We will continue building this application in the future, add more features, enhance the performance, and will keep updating according to users’ demand.

## Challenges

These are some challenges we faced:

* The biggest challenge was data gathering, we only want to work with international historical landmarks but the jury wants us to work with Pakistani landmarks. The biggest problem working with them is the lack of availability of data on these landmarks and data gathering is not an easy task. Initially, we gathered some data but soon data gathering was no longer an option due to lockdown.
* Secondly, not knowing machine learning and where to start working was also a challenge. Fortunately, the university was offering a course that helped but it was only a beginner course so we have searched and learned through online resources.

## Learning Outcomes

Learning outcomes include different models of deep learning. Computer vision techniques and data augmentation. Android application development and using a deep learning model in the android application.

## Nature of End Product

The nature of the end product is an application that can detect landmarks.

## Related Work/ Literature Survey/ Literature Review

* K.Ozaki and S.Yokoo. used resNet for the detection of a landmark but they do not proceed with this work. Their approach is based on deep convolutional neural networks with metric learning and they use cosine-softmax activation function. As images in datasets are noisy. so, they clean data automatically. The re-ranking is done to solve a variety of the dataset for landmark retrieval. The problem is that this is not a product but just proof of concept and They will detect only if it is a landmark or not but they do not tell anything about it.[1]
* Google image search **[4]** is a tool provided by Google that tries to find the closest image to the input image. It is publicly available, but its accuracy is not so good. The photo location is another tool to find the image location. If the EXIF file contains the location, then we will see latitude and longitude details. But we are not using the GPS systems in our application, so we will visually recognize the image with better accuracy. The visual positioning system is another tool provided by Google. It is close to our system, but it is mainly used for map navigation **[5]**.

## Document Conventions

Documents written rules followed:

* Headline font size 18
* Text font size 12
* Font “Arial”
* Line spacing 1
* Sub-Headline text size 14

# Overall Description

## Product Features

The application is a camera application that lets the user take a picture of a landmark. The flash is also available to take pictures in dark. The system predicts an ID against that image and then the system checks the location of the user to find out the landmark is original or replica. if it is a replica. The replica will be displayed otherwise the name of that landmark will be displayed. When the user clicks the Details button. Details related to that landmark will be displayed including its history. When the user clicks the tries cuisine button, The System recommends local food. When the user clicks places you may also like button, The System recommends the nearest historical place. If the system unable to predicts an ID against that image and then the system will ask for user suggestions.

## Functional Description

The application is a digital Guide it will detect landmarks and provide accurate information about them. This will also include historical information, who build those landmarks, and everything related to that. It also correctly identify if is an original landmark or replica of the landmark. The application will also provide the recommendation of the nearest historical landmark and local cuisine to try.

## User Classes and Characteristics

The user class is the general public who has access to an android phone. They can be of any age and education level if they understand the English language. Their location can be anywhere on earth. They can also have a basic knowledge of using apps.

## Design and Implementation Constraints

There is only one way to run a machine learning model natively on an android device as a time of this project which through TensorFlow lite. The model has to be trained in Tensorflow to convert it for Tensorflow lite. Hardware limitations can also be problematic as a training model on a large number of images can require a lot of hardware resources and even with that, the model takes a very long time to train.

## Assumptions and Dependencies

Following are the assumptions:

* The device on which our application is running has a camera and GPS.
* The device is running on android 5.0 or higher.
* The device also has internet access.

# System Requirements

## Functional Requirements

### Detect the landmark

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-1 | |
| **Purpose** | | Detect the landmark | |
| **Priority** | | High | |
| **Pre-conditions** | | User is in the application | |
| **Post-conditions** | | Display Name of landmark | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Point devise toward the landmark | | Show preview |
| **2** | Press Capture Button | | Take an image and send it for prediction. The model will predict the landmark and information of that will be returned |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Point devise toward the non-landmark | | Show preview |
| **2** | Press Capture Button | | Take an image and send it for prediction. The model will unable to predict the landmark and Ask suggestion |

**Table 3.1: UC-1**

### Predict the Replica Image

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-2 | |
| **Purpose** | | Predict The Replica Image | |
| **Priority** | | High | |
| **Pre-conditions** | | User is in the application | |
| **Post-conditions** | | Display replica | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Point devise toward the landmark | | Show preview |
| **2** | User Will Press Capture Button | | Take an image and send it for prediction. The model will predict the landmark but the location does not match. Display replica. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Point devise toward the non-landmark | | Show preview |
| **2** | User Will Press Capture Button | | Take an image and send it for prediction. The model will unable to predict the landmark and Ask suggestion |

**Table 3.2: UC-2**

### Get details of the landmark

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-3 | |
| **Purpose** | | Get details of the landmark | |
| **Priority** | | High | |
| **Pre-conditions** | | User is in the application | |
| **Post-conditions** | | Press the back button | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Point devise toward the landmark | | Show preview |
| **2** | User will Press Capture Button | | Take an image and send it for prediction. The model will predict the landmark and a Dialog box will appear displaying the name of the landmark. |
| **3** | User will Press Details Button | | A new screen will appear displaying details of the landmark. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Point devise toward the non-landmark | | Show preview |
| **2** | User will Press Capture Button | | Take an image and send it for prediction. The model will unable to predict the landmark and Ask suggestion |

**Table 3.3: UC-3**

### Recommendation

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-4 | |
| **Purpose** | | Recommendation | |
| **Priority** | | High | |
| **Pre-conditions** | | User is in the application | |
| **Post-conditions** | | Press Back button | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Point devise toward the landmark | | Show preview |
| **2** | User will Press Capture Button | | Take an image and send it for prediction. The model will predict the landmark and a Dialog box will appear displaying the name of the landmark. |
| **3** | User will Press Details Button | | A new screen will appear displaying details of the landmark. |
| **4** | User will Press Try cuisine Button | | A Dialog box will appear recommending local cuisine. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User will Press Places you may also like Button | | A Dialog box will appear recommending the nearest place like the one you are visiting. |

**Table 3.4: UC-4**

### Requirements Analysis and Modeling

### Use-case diagram

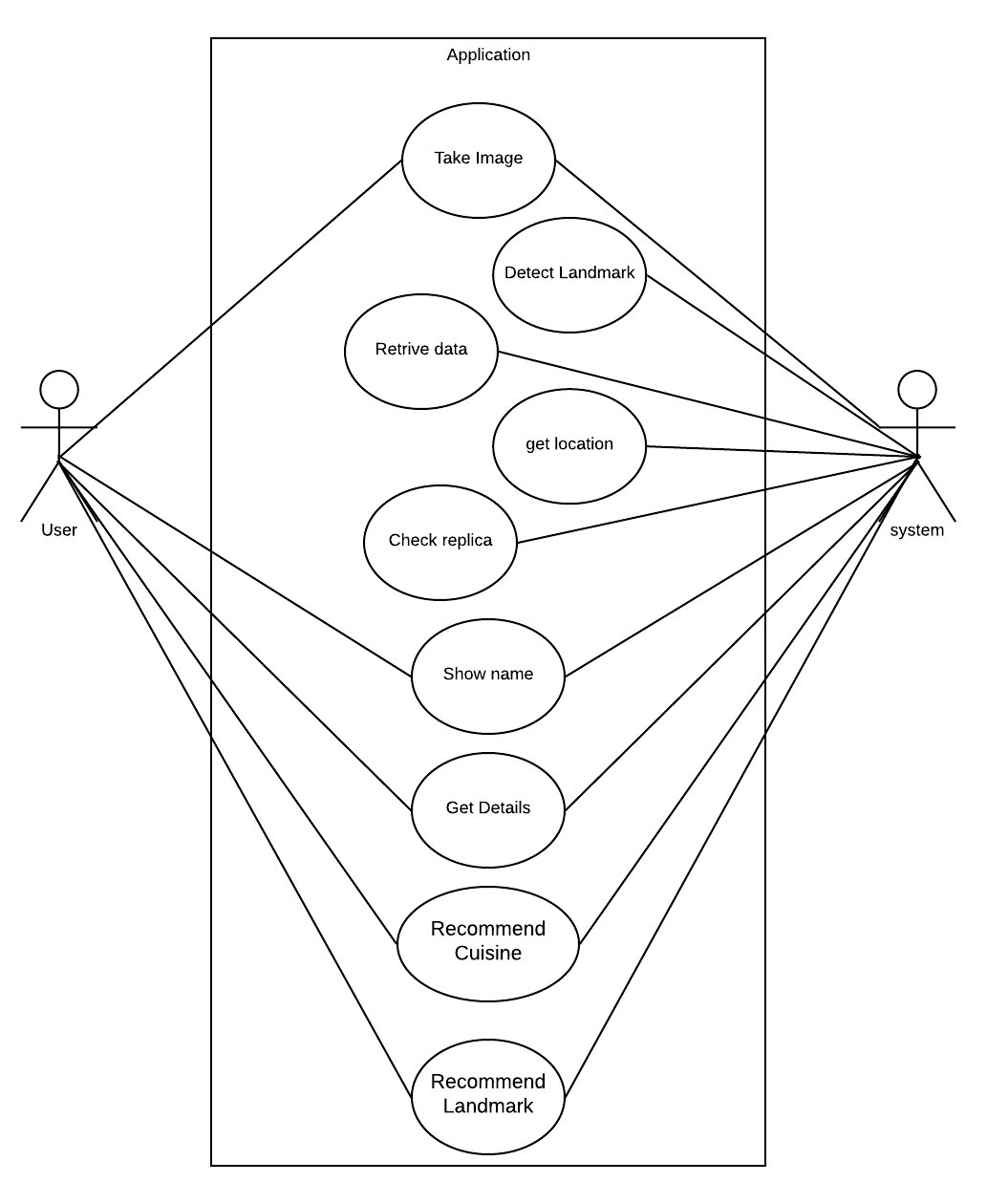
**

Figure 3.1

## Nonfunctional Requirements

### Performance Requirements

The performance of the application depends on the hardware of the device. If the CPU the device is higher or equal than 600 series processor of Snapdragon or equivalent, the application will work just fine, the lower series chips are not made for these machine learning tasks that is why they are too slow. The application required Android 5.0 or higher because it covers 98 percent of the android users. The database response should take no more than one second. Otherwise, the user will consider it slow.

### Safety Requirements

There is no safety required as the product is an application that will run on a system.

### Security Requirements

All the data we have related to landmarks be properly managed and stored with encrypted password storage. There is no Security requirement related to user data as we are not gathering any information on our customers.

### Additional Software Quality Attributes

The availability of the system should be high and software upgrades shall be applied no more than once a month. Maintainability should be very easy and maintenance time shall be less than 15 minutes every 2 weeks. The system is modular in design and every module is tested separately, this makes bug finding less difficult. After that integrated testing is done. Information provided by the application should be correct and authentic and will be gathered from respectable resources. The application interface should be minimalistic and easy to use.

## Other Requirements

The system needs a database to save the name, history, and all related information of landmarks. Data under their ID is stored.

# Technical Architecture

**Architecture Diagram**

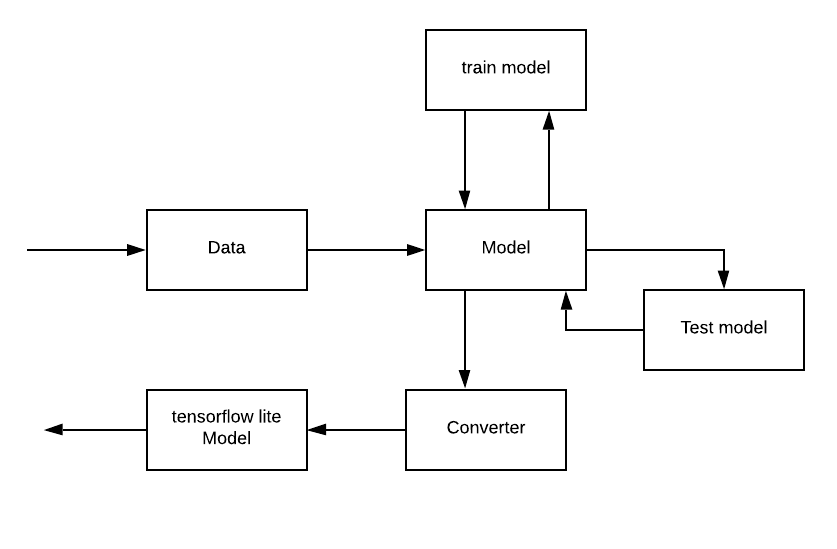
****

Figure 4.1

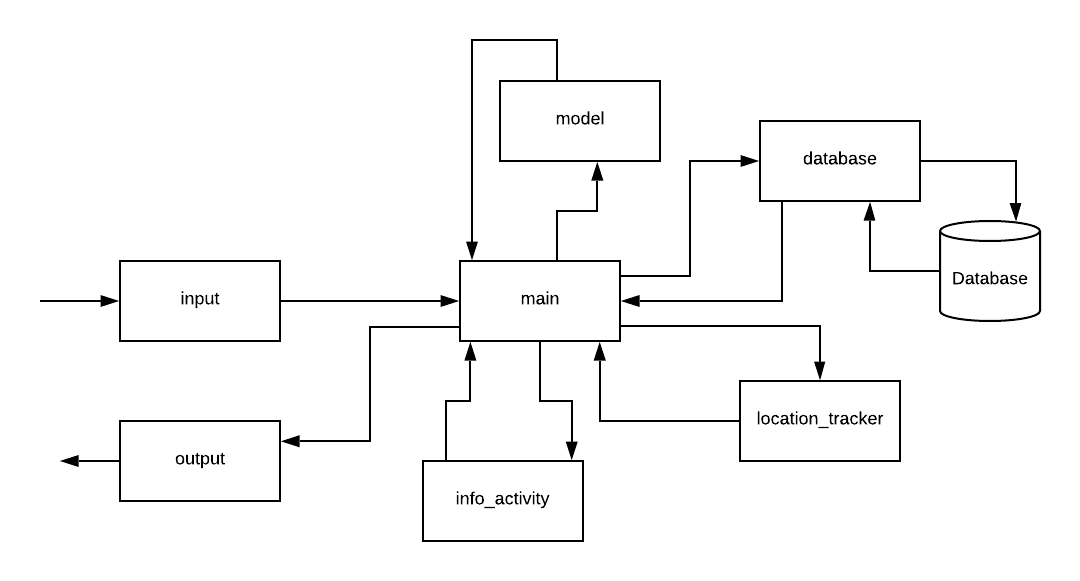
****

Figure 4.2

## Application and Data Architecture

**Activity Diagram**

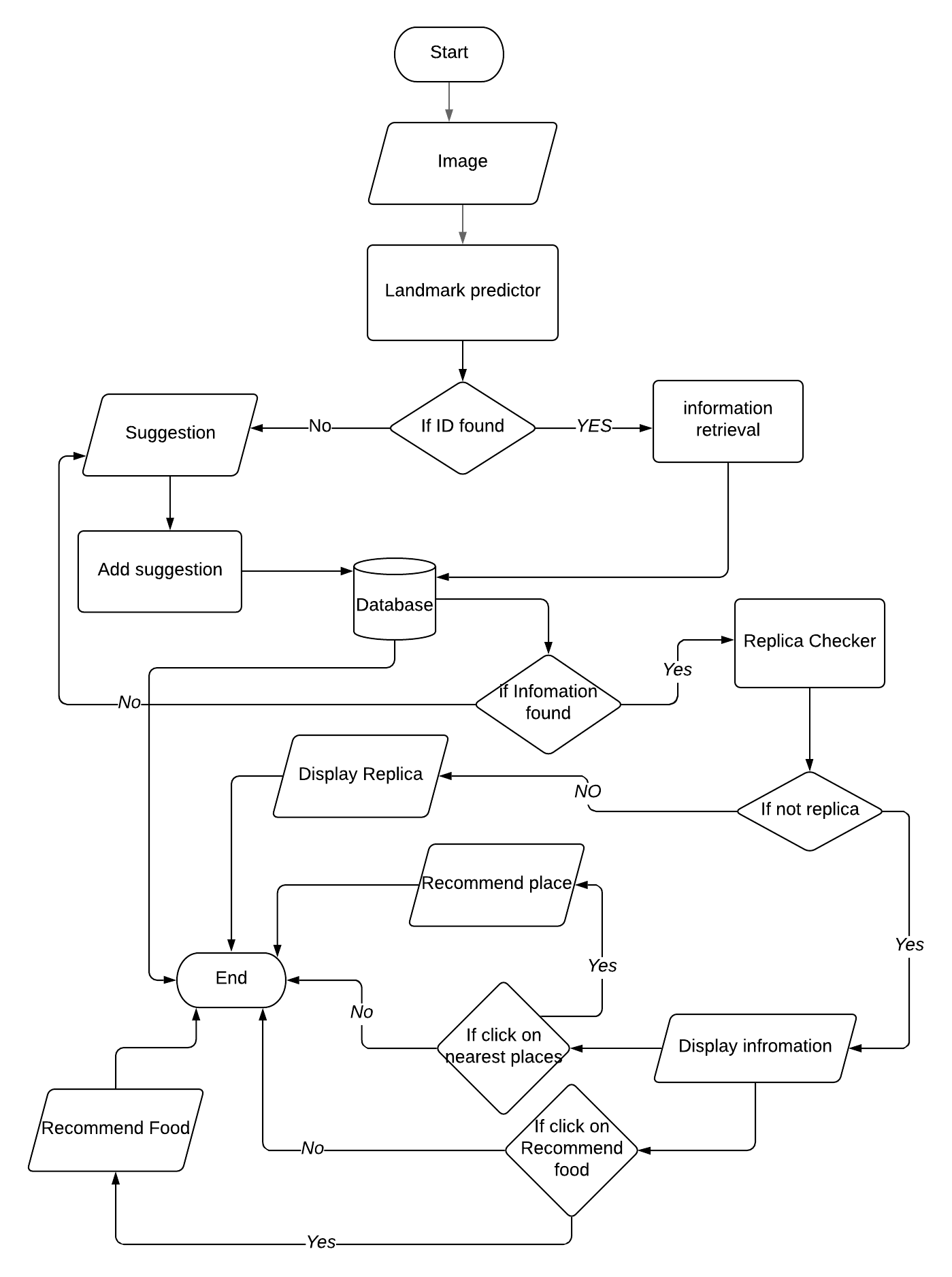
****

Figure 4.3

**Component Diagram**

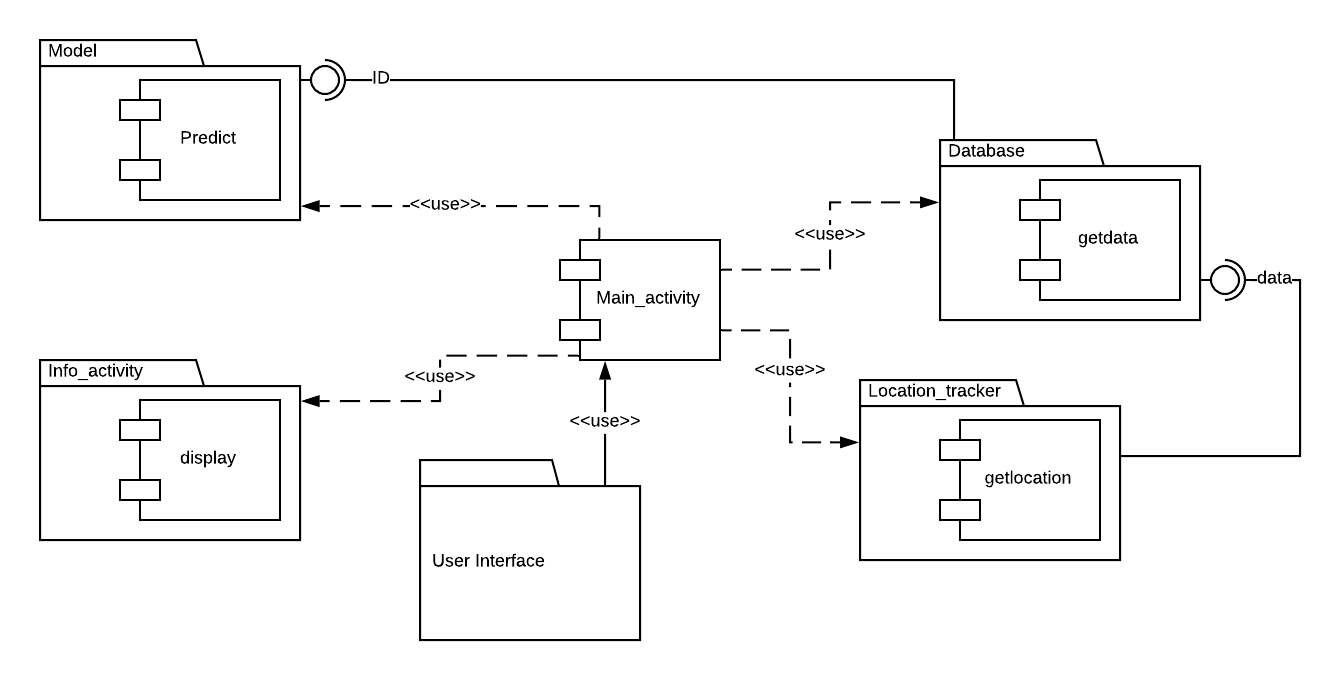


Figure 4.4

**Decision Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Conditions** | **Press the capture button** | **no** | **yes** | **no** | **yes** |
| **Point phone toward the landmark** | **no** | **no** | **yes** | **yes** |
| **Actions** | **Expected result** | **none** | **Not found** | **Not found** | **Found** |
| **Show result/suggestion dialog box** | **none** | **Suggestion dialog box** | **none** | **Result dialog box** |

Table 4.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Conditions** | **Press the cuisine button** | **no** | **yes** | **no** | **yes** |
| **Have internet access** | **no** | **no** | **yes** | **yes** |
| **Actions** | **Expected result** | **none** | **Error** | **none** | **Found** |
| **Show recommendation dialog box** | **none** | **Display no internet access** | **none** | **Result dialog box** |

Table 4.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Conditions** | **Press the “you may like this place” button** | **no** | **yes** | **no** | **yes** |
| **Have internet access** | **no** | **no** | **yes** | **yes** |
| **Actions** | **Expected result** | **none** | **Error** | **none** | **Found** |
| **Show recommendation dialog box** | **none** | **Display no internet access** | **none** | **Result dialog box** |

Table 4.3

A close up of text on a white background

Description automatically generated**Class Diagram**

Figure 4.5

**ER Diagram**

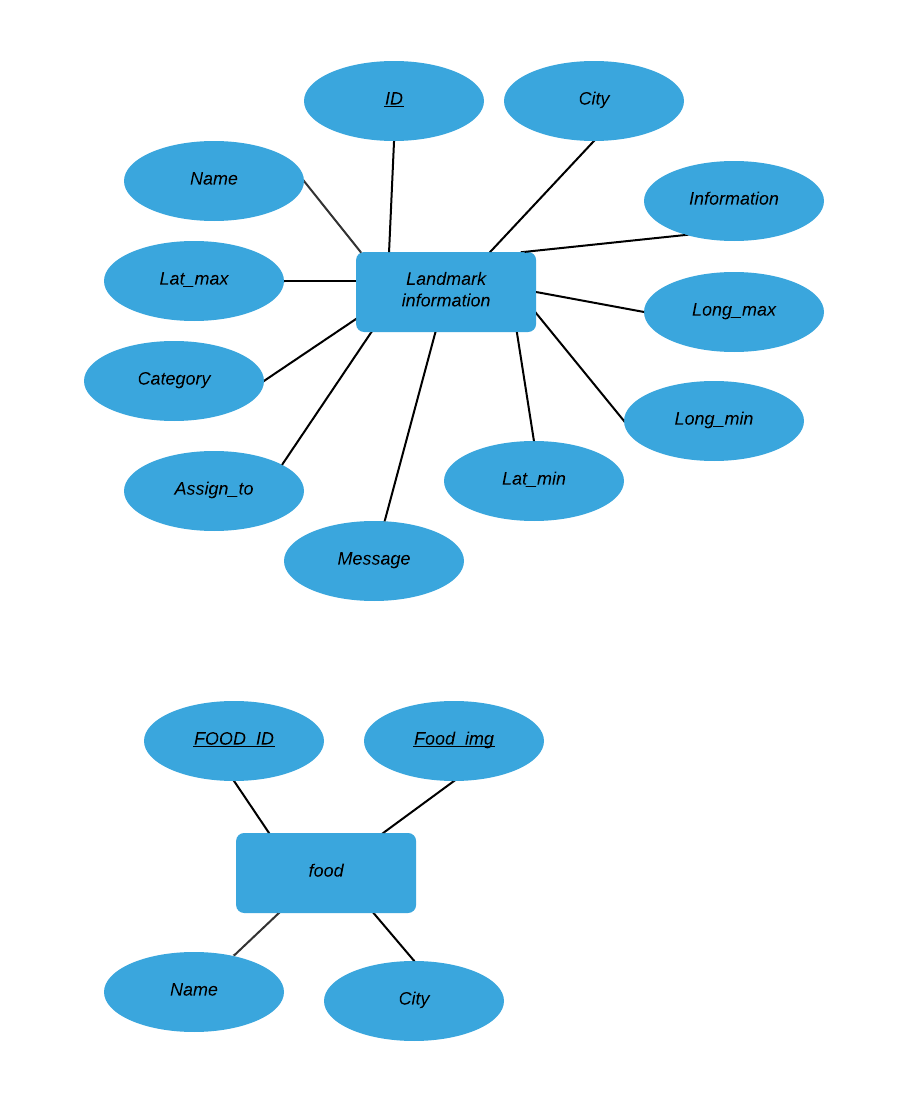


Figure 4.6

**Tables**

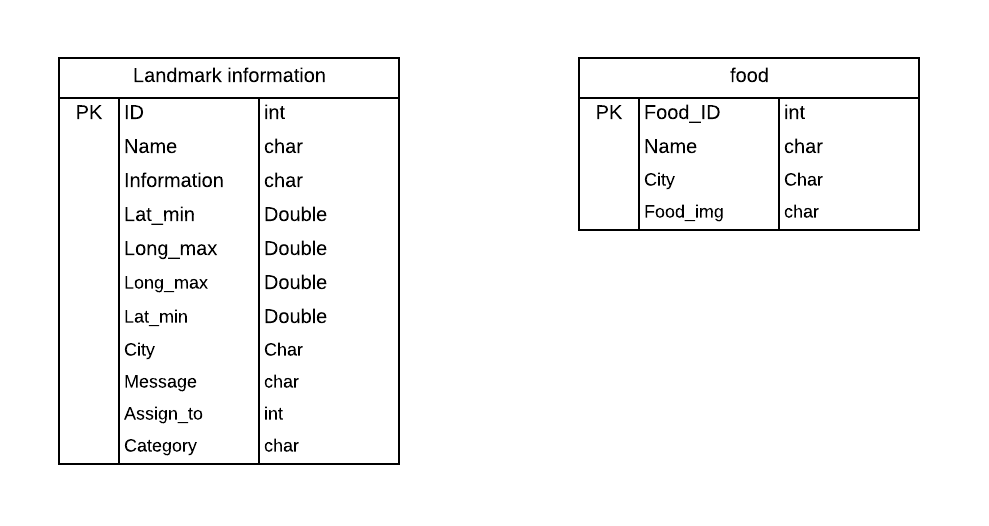


Figure 4.7

## Component Interactions and Collaborations

**Data Flow Diagram**

**Context level**

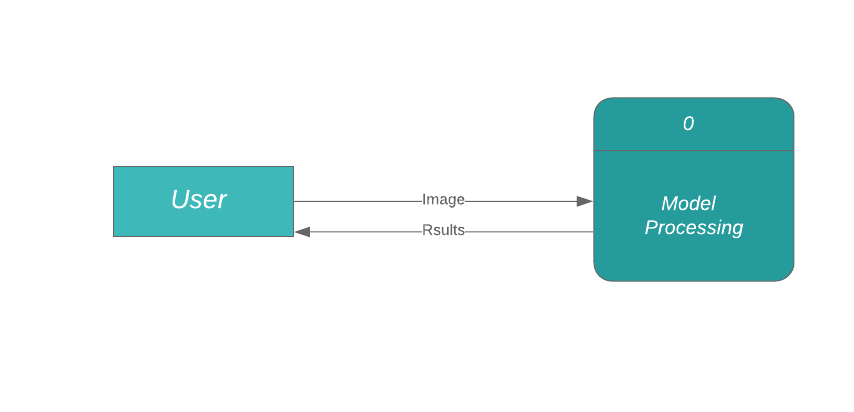


Figure 4.8

**Zero level**

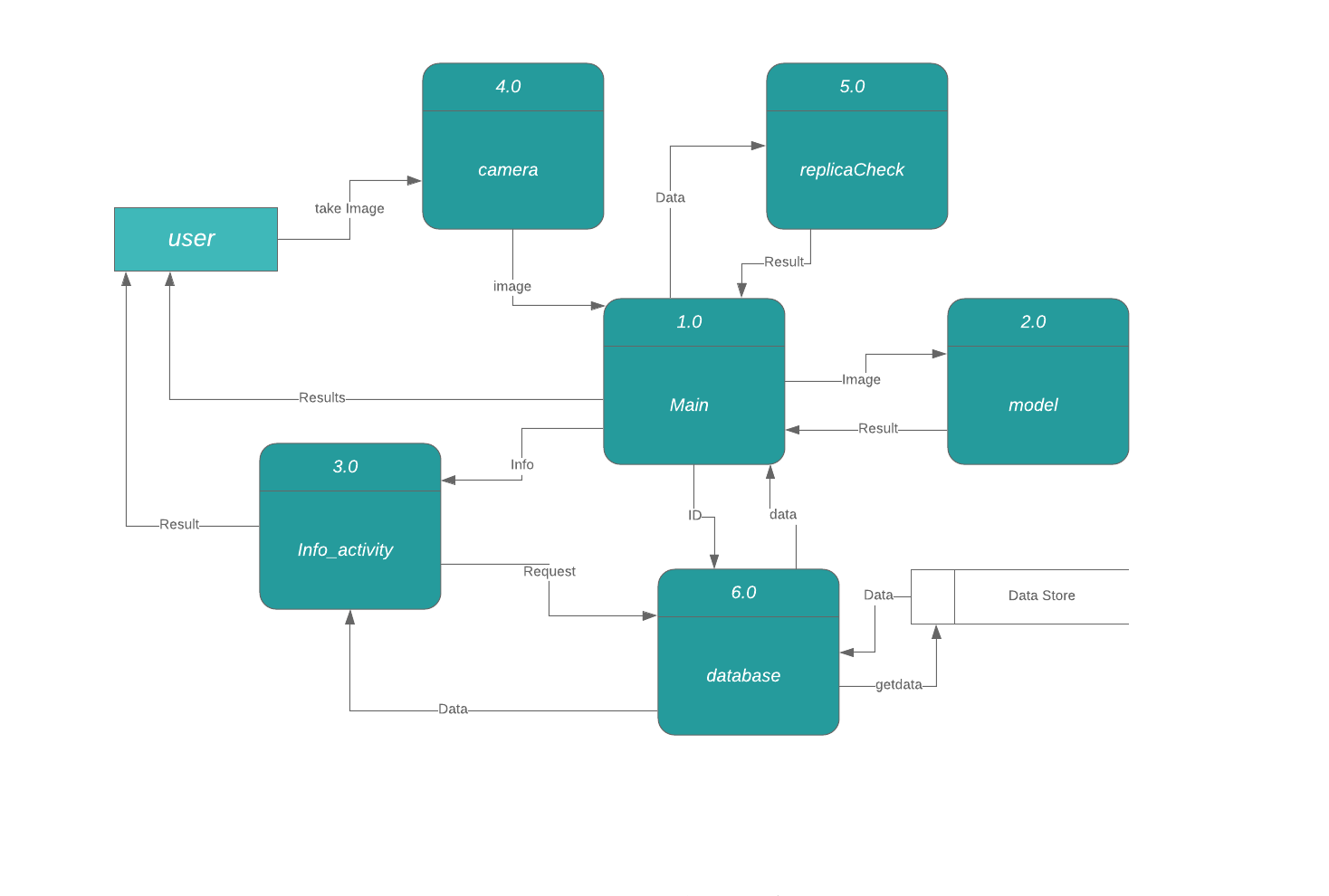


Figure 4.9

**Level one**

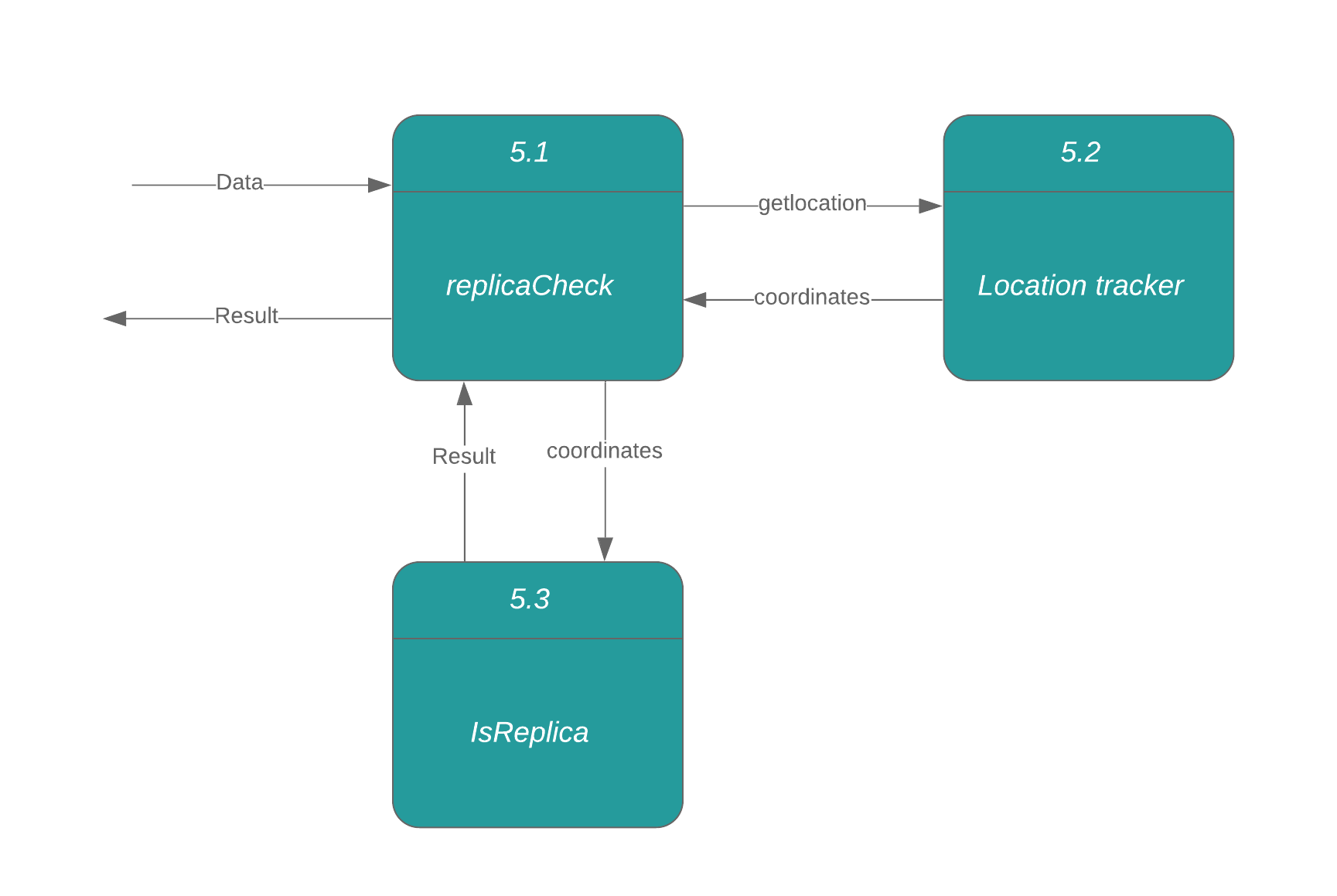


Figure 4.10 **Expanding Process 5**

**Expanding Process 2**

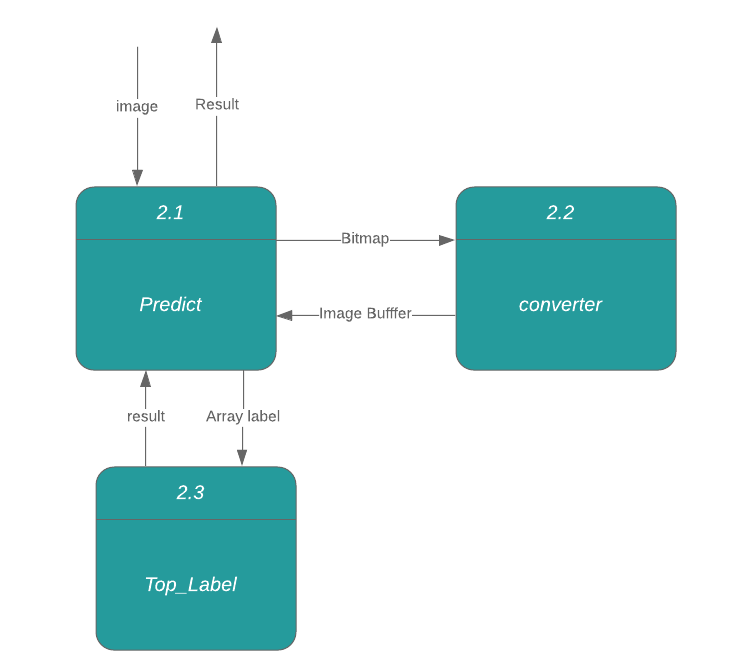


Figure 4.11

**Expanding Process 6**

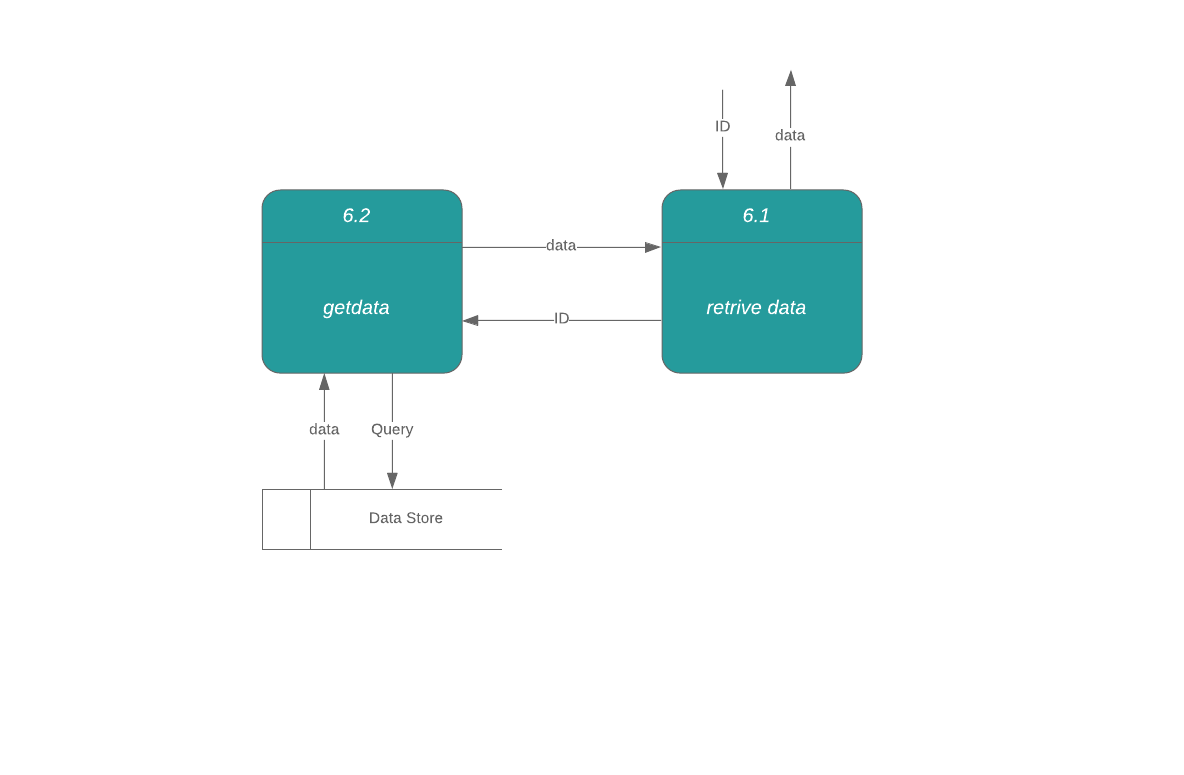


Figure 4.12

**Sequence Diagram**

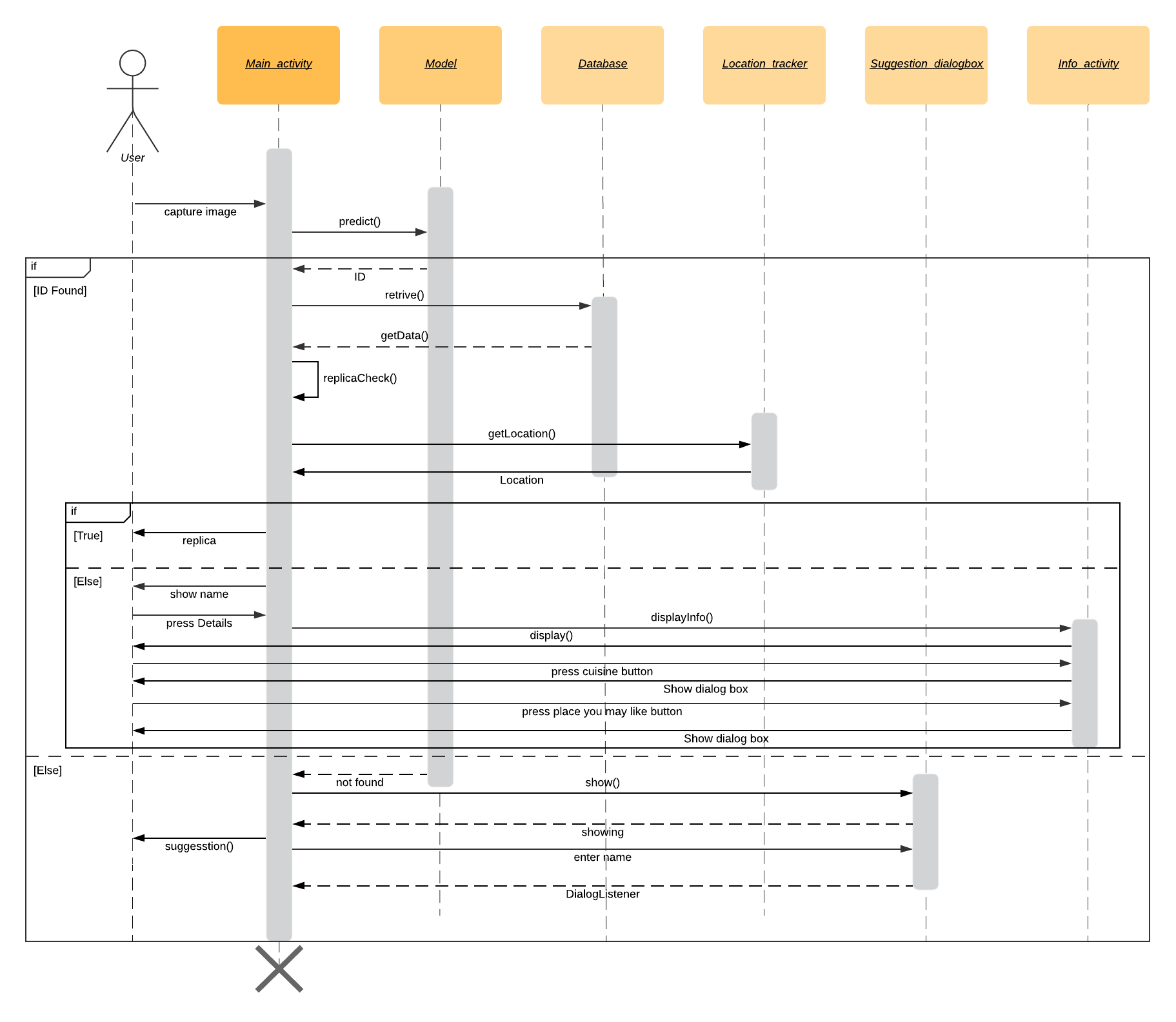


Figure 4.13

**Collaboration Diagram**

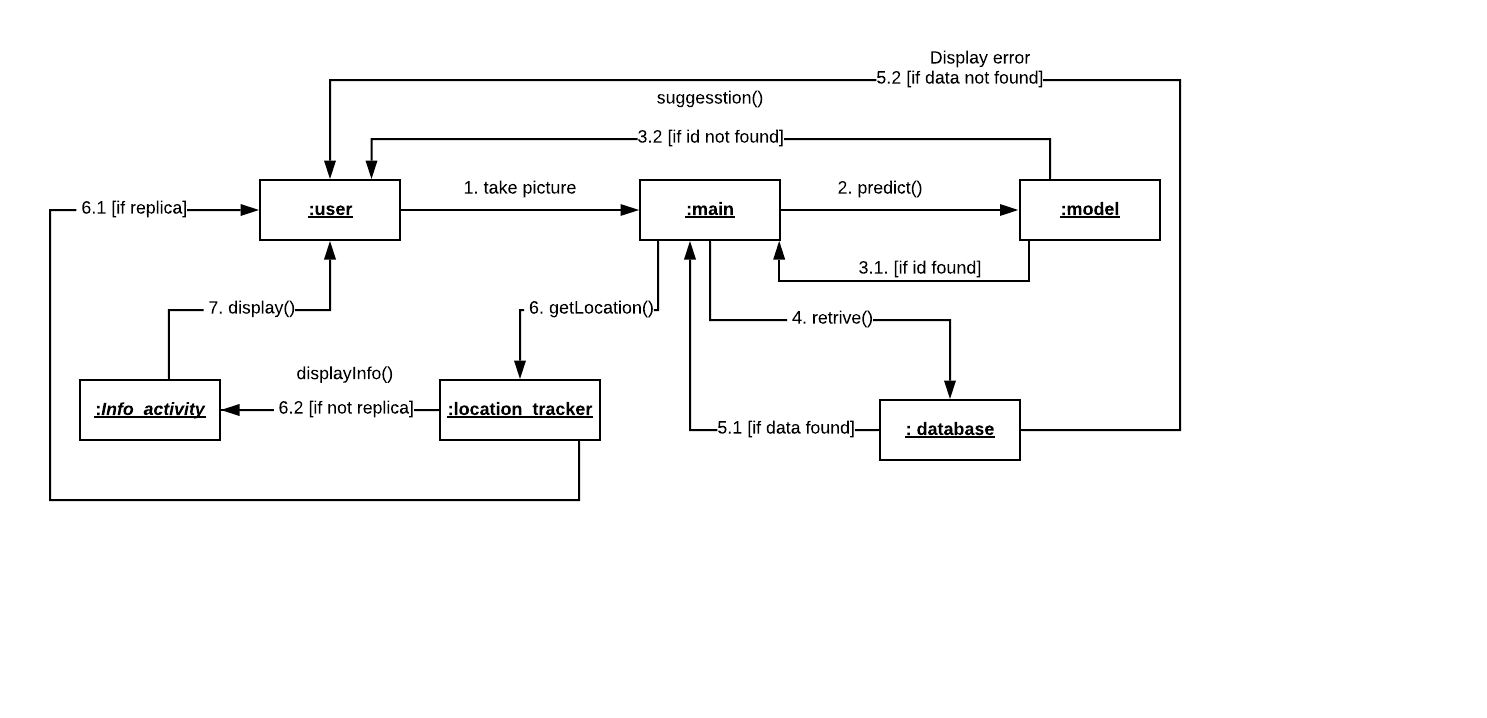


Figure 4.14

## Design Reuse and Design Patterns

Singleton Design Pattern is being used for the library that requests from the server as a singleton design pattern restricts the number of instances of a class and ensures that only one instance of the class exists. it is also used there to close the resource to avoid memory leakage.

## Technology Architecture

The platform for this project is android smartphones. The infrastructure required for this project is a remote database. It will be used for storing data about landmarks and updating data will be easy without updating the application. This will give more control over the data that we presented to our users. The Internet will be required to access the data from the database. The machine learning model will run directly on the user’s Android device and it does not require any additional hardware. it will use the graphic processing unit of the device. The speed of processing will vary depending upon the device.

## Architecture Evaluation

In our application machine learning is being used for detection purposes. We are using a trained model that will run on the mobile and make predictions. The machine learning model is trained and tested using python APIs TensorFlow and Keras As our main goal for the model is to run on android and that is why Tensorflow is being used. TensorFlow has a lite variant that can run the model directly on an android device. TensorFlow Lite was the only API available for this purpose. To run the Tensorflow model on android, the Trained model needs to be converted in TensorFlow Lite format. It is better to use an already trained model than sending data over the internet for processing and waiting for the result. This is a faster way of doing it because it does not depend on the internet. The camera is updated to cameraX API as cameraX API is more stable than camera2 API and camera2 API.

# Detailed Design and Implementation

## Component-Component Interface

The main activity is the driver that can use the other classes depending upon the requirements. The main activity also runs the camera of the application. As the user takes the image, the Model class object will be created to predict its ID. If prediction becomes successful, then that ID will be sent to the database for retrieving data about the predicted landmark. After that location tracker object will be created to check the replica.

## Component-External Entities Interface

The main activity is the driver that can use the other classes depending upon the requirements. The main activity also runs the camera of the application. As the user takes the image, the Model class object will be created to predict its ID. If prediction becomes successful, then that ID will be sent to the database for retrieving data about the predicted landmark. After that location tracker object will be created to check the replica.

## Component-Human Interface

The first screen is a camera screen where users can capture image for prediction. while the second screen will show results in case of success, else ask for suggestion in case unsuccessful. The third screen will display details. The recommendation about places to visit next and try local cuisine is also given on this page.

user class is the general public who has access to an android phone. They can be of any age and education level if they understand the English language. Their location can be anywhere in Pakistan. They can also have a basic knowledge of using apps. Gestalt law is followed.

## Screenshots/Prototype

### Workflow

**Swimlane diagram**

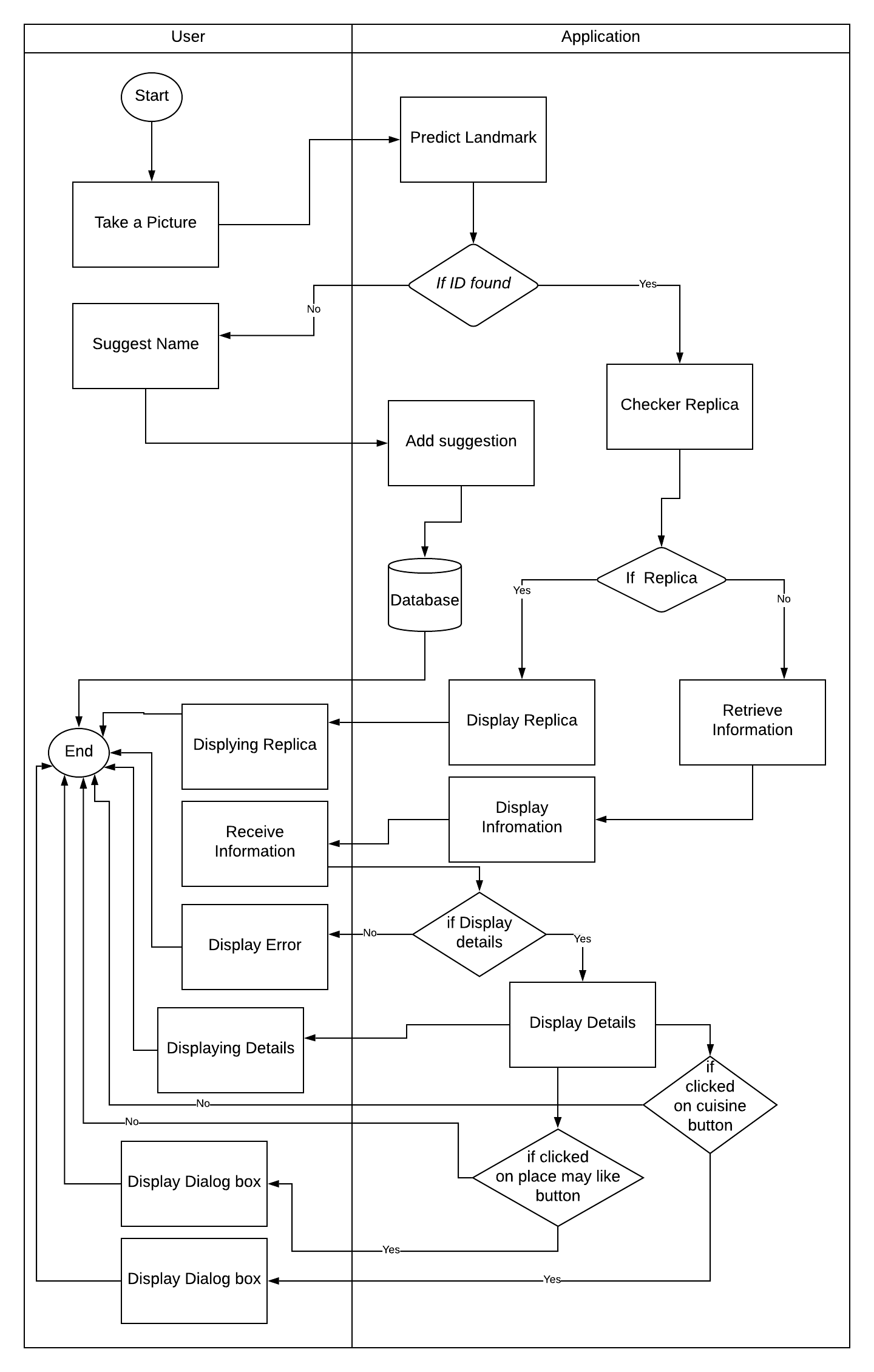
**

Figure 5.1

### Screens

1. Camera Screen



Figure 5.2

2 a. Detected successfully

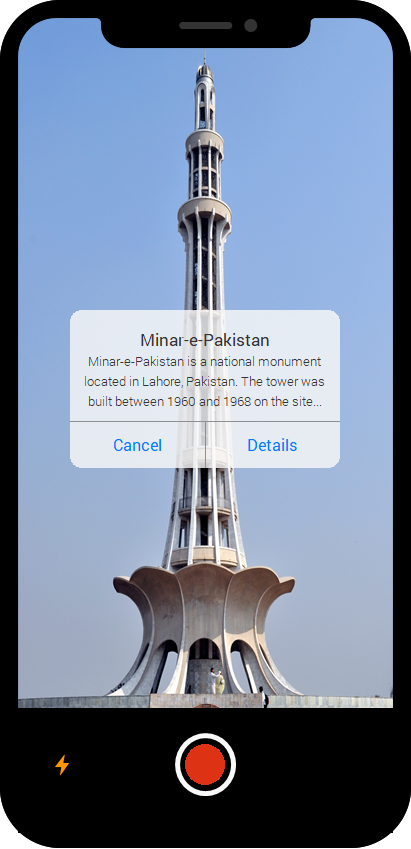


Figure 5.3

3. If we click the “Details” button

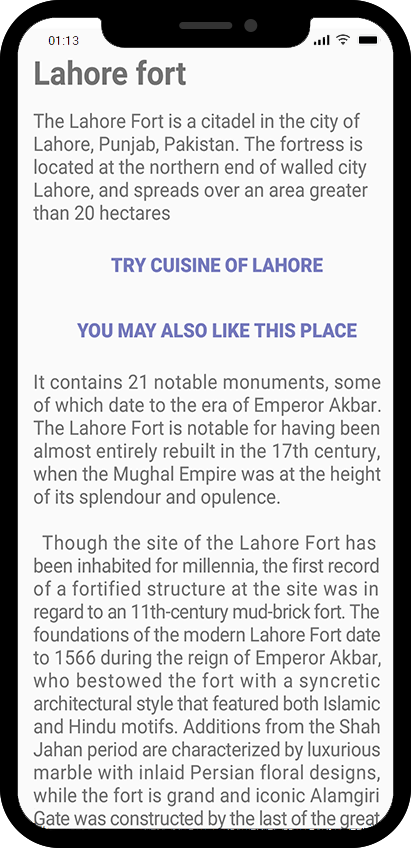


Figure 5.4

2 b. If the scan was unsuccessful

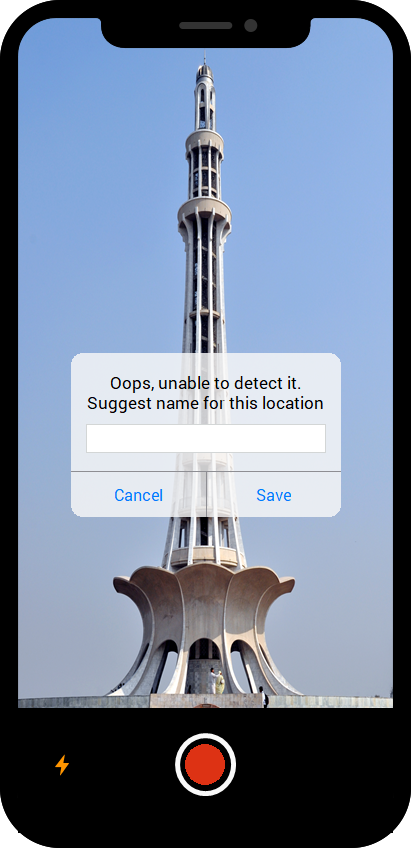


Figure 5.5

4. Recommending food

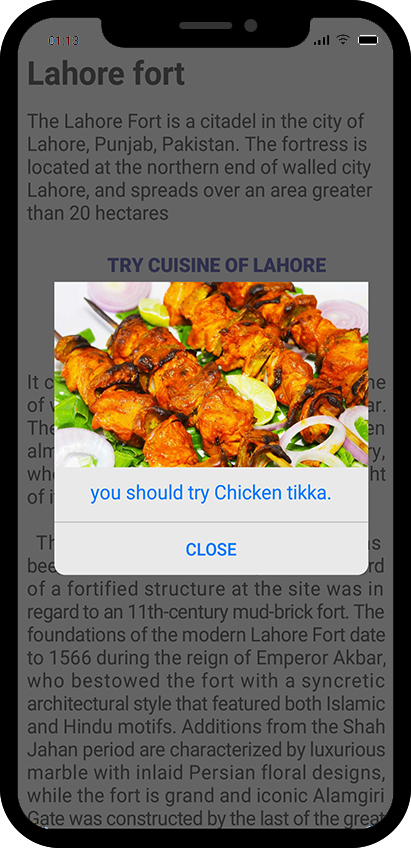


Figure 5.6

5. Recommending the nearest place to visit

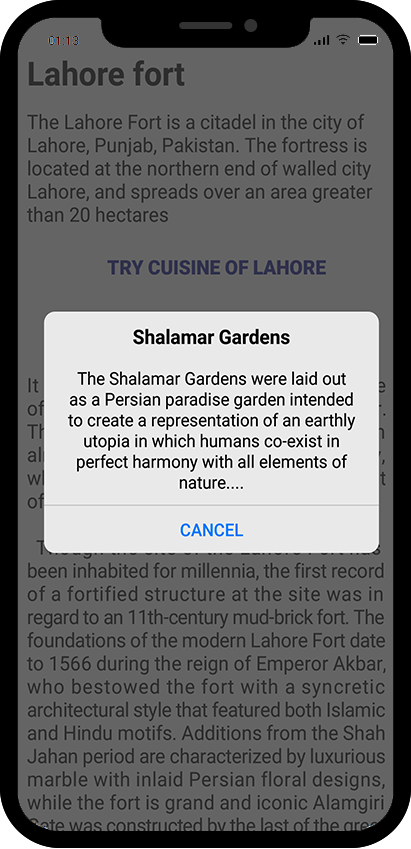


Figure 5.7

## Other Design Details

In this phase of the project. we move to transfer learning. First, we use VGG16 with image net weights and fine-tune it according to our needs but it is producing unacceptable results. small data was considered as the root cause of the problem so more data gathered, but still, there was no considerable improvement. VGG19 was considered as the next model to fine-tune for training. With the addition of 3 more convolutional layers in VGG19, it performs much better than VGG16. but still, accuracy was less than 80 percent. to improve the accuracy data augmentation was used. With that above 80 percent of accuracy was achieved. The trained model was converted to TensorFlow lite but the available converter decreased the accuracy.

**Training Accuracy vs Validation Accuracy**

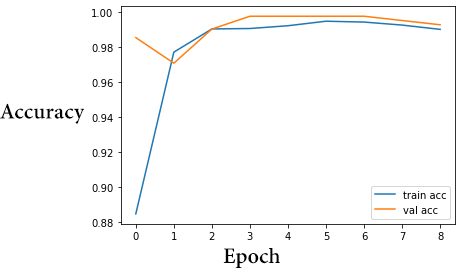


Figure 5.8

**Training Loss vs Validation Loss**

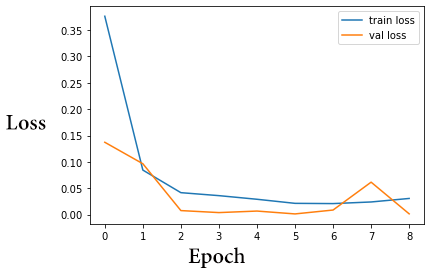
****

Figure 5.9

**Confusion matrix**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Actual** | | | | | | | | | | |
| **PREDICTED** | N = 90 | **Naulakha Pavilion** | **Lahore Fort** | **Minar-e-**  **Pakistan** | **Badshahi mosque** | **Diwan-i-Am** | **guest house** | **Sheesh Mahal** | **other** | **Tomb of Iqbal** |
| **Naulakha Pavilion** | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Lahore Fort** | 0 | 10 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| **Minar-e-**  **Pakistan** | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Badshahi mosque** | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 1 |
| **Diwan-i-Am** | 0 | 0 | 0 | 0 | 9 | 1 | 1 | 0 | 0 |
| **guest house** | 0 | 0 | 0 | 0 | 0 | 9 | 0 | 0 | 0 |
| **Sheesh Mahal** | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 |
| **other** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 8 | 0 |
| **Tomb of Iqbal** | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 9 |

Table 5.1

Accuracy = (TP+TN)/(TP+TN+FP+FN) = 80/90 = 89%  
  
Precision = TP/(TP+FP) = 9/10 = 90%

# Test Specification and Results

## Test Case Specification

|  |  |
| --- | --- |
| Identifier | TC-1 |
| Short description | Successful Prediction |
| Pre-condition(s) | open the app. |
| Input data | capture the landmark’s image |
| Detailed steps | User will capture the landmark’s image, then the system will predict id against that image which will be verified in database and information will return to the user |
| Expected result(s) | A dialog box will appear to the user, there will be short detail. Users can click the ‘Detail’ button for more information. |
| Post-condition(s) | Users can press the cancel or detail button on the dialogue box. |
| Actual result(s) | Same as expected results. |
| Test Case Result | Successful |

## Table 6.1: TC-1

## 

|  |  |
| --- | --- |
| **Identifier** | TC-2 |
| **Short description** | Predict The Replica Image |
| **Pre-condition(s)** | open the app. |
| **Input data** | Capture the fake landmark’s image |
| **Detailed steps** | The user will capture the fake landmark’s image, then the system will predict id against that image. retrieval of data will happen against that id, by using GPS replica of the landmark will be detected |
| **Expected result(s)** | A dialog box will appear to the user, And the user will be informed that it is a replica. |
| **Post-condition(s)** | Users can press the cancel button on the dialogue box. |
| **Actual result(s)** | Same as expected results. |
| **Test Case Result** | Successful |

Table 6.2: TC-2

|  |  |
| --- | --- |
| **Identifier** | TC-3 |
| **Short description** | Predict The Image (Prediction Failed) |
| **Pre-condition(s)** | open the app. |
| **Input data** | Capture the unknown landmark’s image |
| **Detailed steps** | The user will capture the landmark’s image, then the system will be unable to predict id against that image. |
| **Expected result(s)** | A dialogue box will appear to the user that the system is unable to predict and asks the user to suggest the name of the location. |
| **Post-condition(s)** | Users can press the cancel button or add the suggestion. |
| **Actual result(s)** | Same as expected results. |
| **Test Case Result** | Successful |

Table 6.3: TC-3

|  |  |
| --- | --- |
| **Identifier** | TC-4 |
| **Short description** | Get details of the landmark |
| **Pre-condition(s)** | open the app. |
| **Input data** | Capture the known landmark’s image |
| **Detailed steps** | The user will capture the landmark’s image, then a dialogue box will appear. the user will press the detail button |
| **Expected result(s)** | A new screen will appear to the user that will show all the detail of that landmark |
| **Post-condition(s)** | Users can press the back button to go back |
| **Actual result(s)** | Same as expected results. |
| **Test Case Result** | Successful |

Table 6.4: TC-4

|  |  |
| --- | --- |
| **Identifier** | TC-5 |
| **Short description** | Recommendation |
| **Pre-condition(s)** | On detail page. |
| **Input data** | click food Recommendation button |
| **Detailed steps** | By using the id of the predicted landmark, its city is extracted, and the recommendation of food will take place depending on the city. |
| **Expected result(s)** | A dialogue box will appear to the user recommending food. |
| **Post-condition(s)** | Users can press the back button to go back or cancel button |
| **Actual result(s)** | Same as expected results. |
| **Test Case Result** | Successful |

Table 6.5: TC-5

## Summary of Test Results

**Summary of All Test Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module Name** | Test cases run | Number of defects found | Number of defects corrected so far | Number of defects still need to be corrected |
| **Model(for Prediction)** | TC1, TC2, TC3, TC4 | 7 | 7 | 0 |
| **camera** | TC1, TC2, TC3, TC4 | 1 | 1 | 0 |
| **location\_tracker** | TC1, TC2, TC4 | 2 | 2 | 0 |
| **info\_activity** | TC4 | 0 | 0 | 0 |
| **Recommendation** | TC4, TC5 | 0 | 0 | 0 |
| **suggestion** | TC3 | 0 | 0 | 0 |
| **database** | TC1, TC2, TC4, TC5 | 0 | 0 | 0 |
| **Complete System** | 16 | 10 | 10 | 0 |

Table 6.6

# Project Completion Status/Conclusion

The recommendation system for the nearest landmark and food to try out are not included in the proposal and are added on the suggestion of the jury.

**Project Completion Status**

|  |  |
| --- | --- |
| **Module Name** | **Status**  (Complete, Partially Implemented, Not Implemented) |
| **Camera** | complete |
| **location\_tracker** | complete |
| **info\_activity** | complete |
| **suggestion** | complete |
| **Model(for Prediction)** | complete |
| **database** | complete |
| **Recommendation** | complete |
| **Complete System** | complete |

Table 7.1

**Objective(s)/Target(s) Status**

|  |  |  |
| --- | --- | --- |
| **Target/Objective** | **Status**  (Completed,  Partially Completed,  Not Completed) | **Reason(s)** |
| **Build an application** | completed |  |
| **Achieve above 80% Accuracy** | completed |  |
| **Data gathering** | completed |  |
| **Model training** | completed |  |
| **Model testing** | completed |  |
| **Integration of model in android** | completed |  |
| **Built database** | completed |  |
| **Number of Targets Completed** | 7 |  |
| **Number of Targets Partially Completed** | 0 |  |
| **Number of Targets Not Completed** | 0 |  |

Table 7.2

# References

1. K.Ozaki and S.Yokoo. “Recognition under a Noisy and Diverse Dataset for landmark detection”, *arXiv preprint arXiv:1906.04087v2*, Jun. 2019.
2. “International Tourist Arrivals Reach 1.4 Billion Two Years Ahead of Forecasts.” World Tourism Organization.  [<https://www.unwto.org/global/press-release/2019-01-21/international-tourist-arrivals-reach-14-billion-two-years-ahead-forecasts> /](http://www2.unwto.org/) (accessed Nov. 2, 2019).
3. C. McNabb and his team. “Google Landmark Recognition using Transfer Learning.” towardsdatascience.com.<https://towardsdatascience.com/google-landmark-recognition-using-transfer-learning-dde35cc760e1> (accessed Aug. 20, 2019).
4. “Google image search”, Google, <http://www.google.co.in/insidesearch/features/images/searchbyimage.html>, (accessed Aug. 20, 2019).
5. M. Pablo Garcia and I. Frida, “Visual positioning system,” WIPO Patent Application WO/2014/170758, October. 23, 2014.

**Appendix A Glossary**

CNN Convolutional neural network.

VGG Visual Geometry Group.

GPS Global Positioning System.

CPU Central Processing Unit.

# Deployment/Installation Guide

* From your Android device, open play store.
* connect your device with an internet connection.
* Search app by name (Landmark detection).
* Press the Install button.

# User Manual

* Tap on the application icon to launch it.
* Allow all the permissions to be the application.
* Focus on the landmark and capture the image.

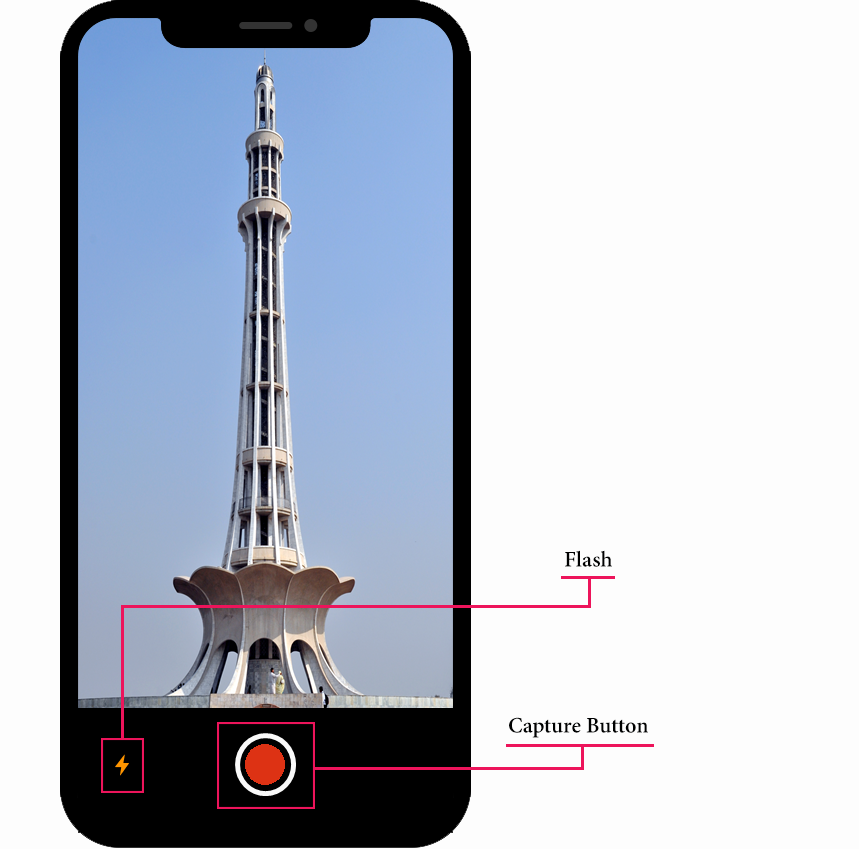


Figure 8.1

* If you require more information, then tap to the “Detail” button and “Cancel” button to exit.

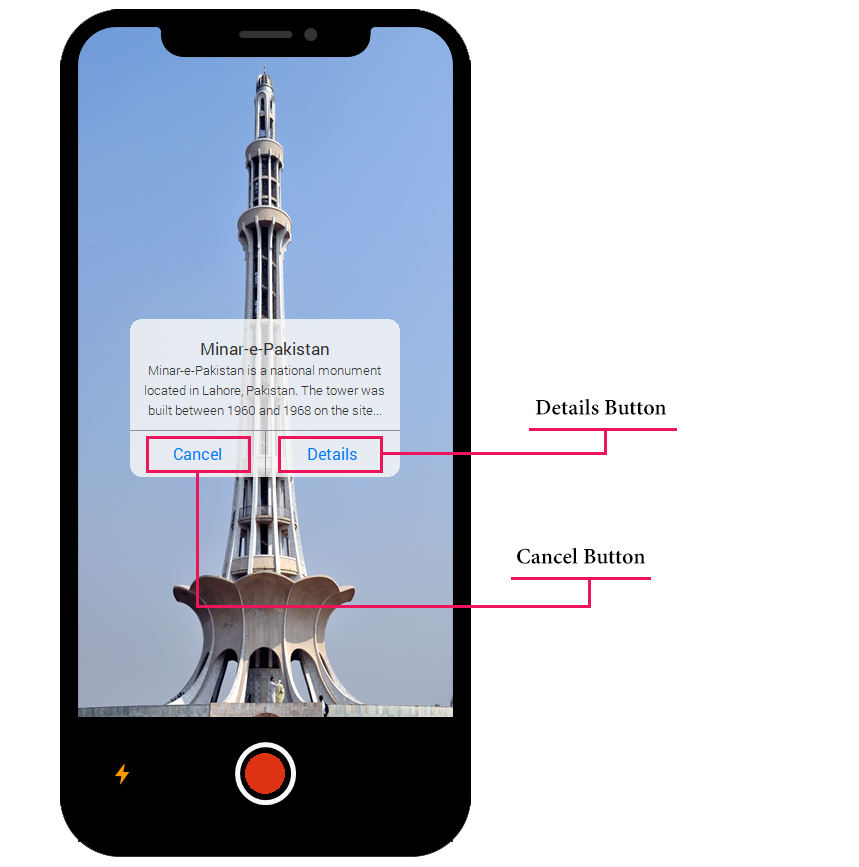


Figure 8.2

* In case of unsuccessful prediction, you can suggest the location name and tap to the “Save” button or you can simply tap to the “Cancel” button.

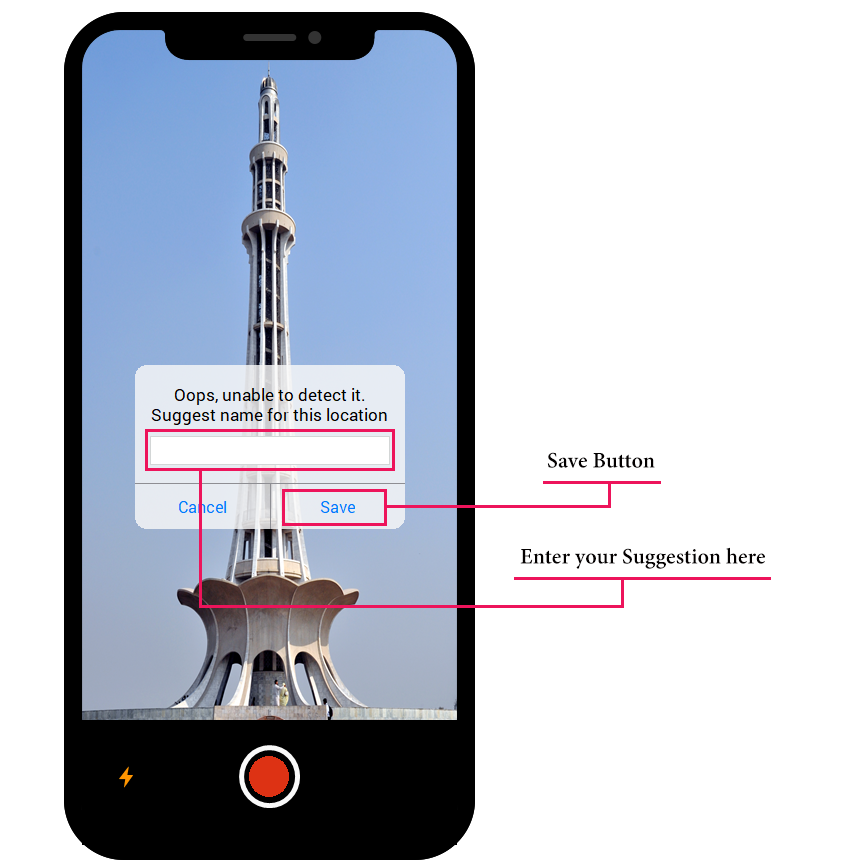


Figure 8.3

* Cuisine recommendation

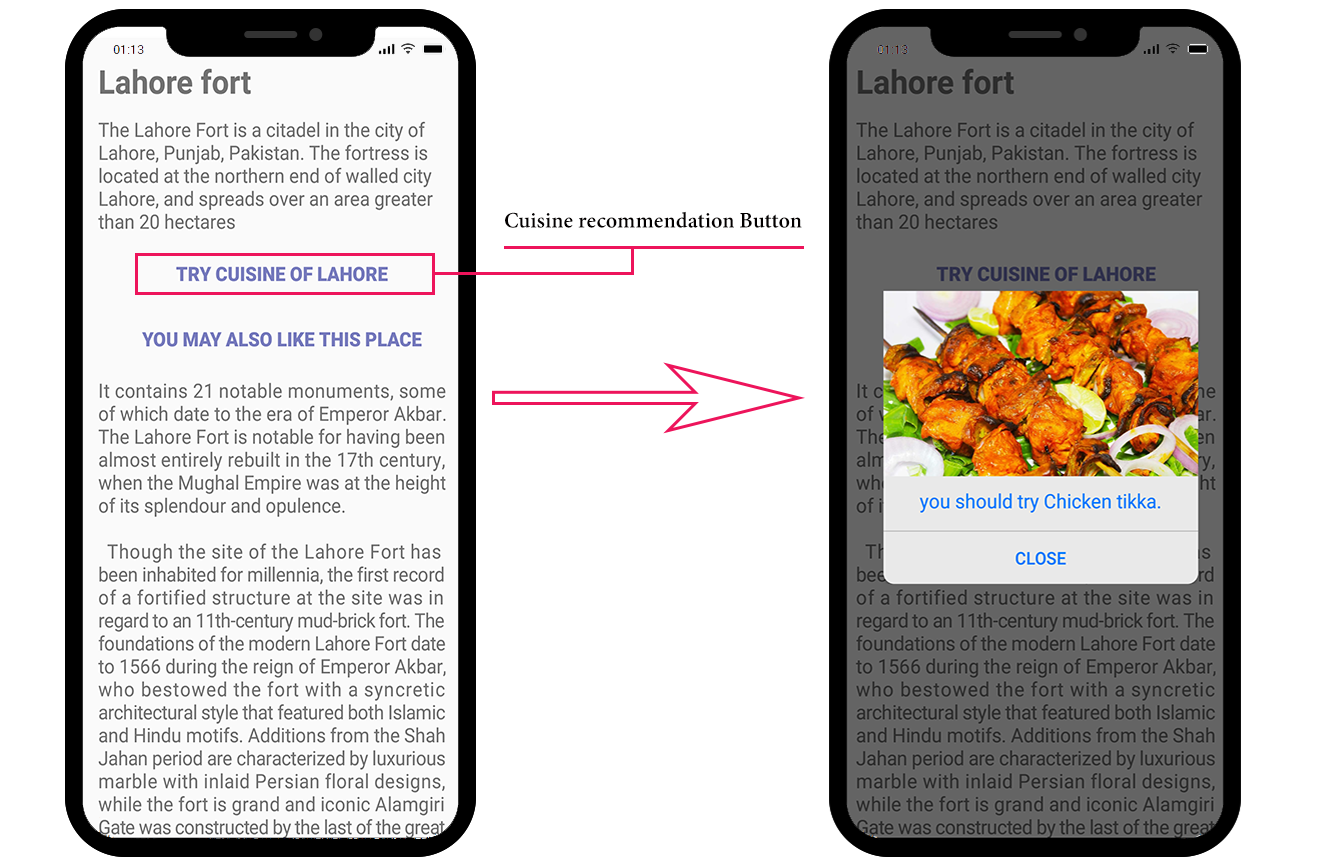


Figure 8.4

* Place recommendation.

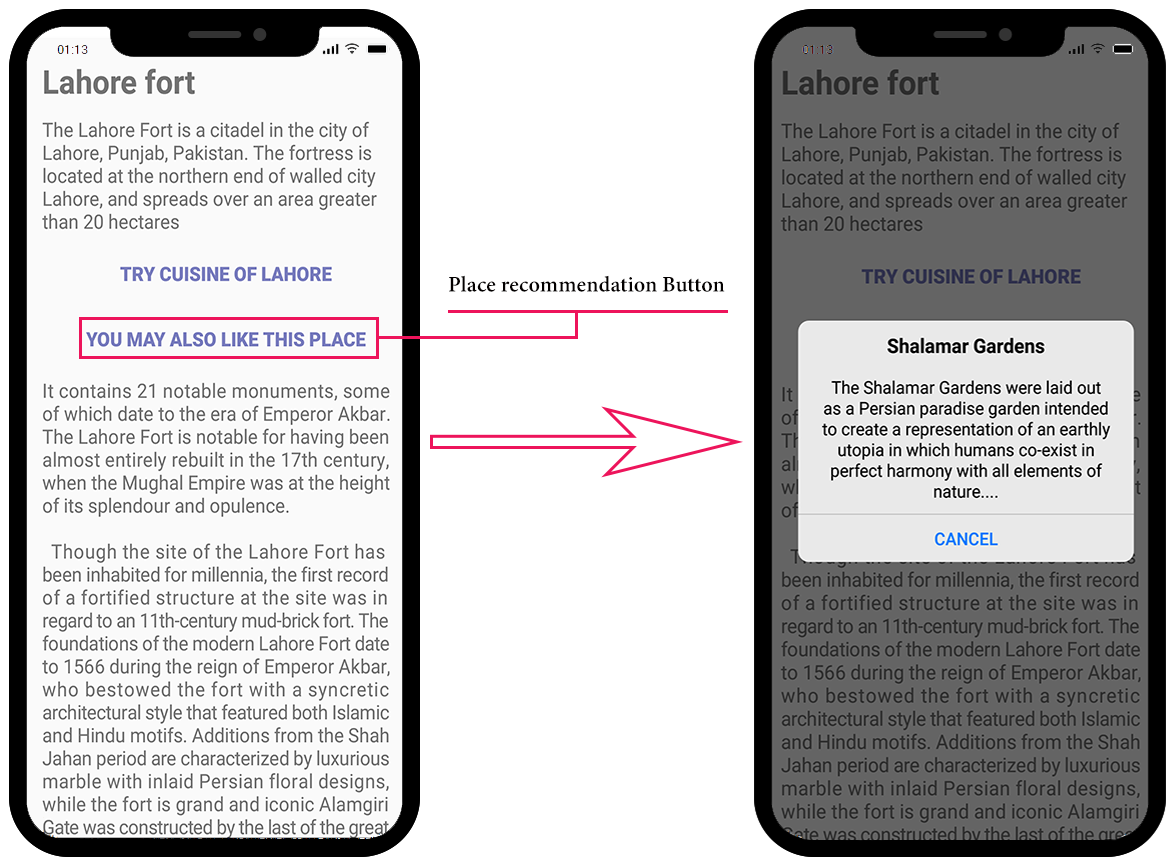


Figure 8.5