Software Requirements Specification for EyeDentify – An Android app for Object Identification

Version 1.0

Prepared by:

Sachi Alagiya

Naga Durga SriTejitha Kandarpa

Naga Ramya Vajja

Mahmood Quadri Syed

Shivansh Saurabh

Table of Contents

I. Preface

1. Introduction

This is a Software Requirements Specification (SRS) document for EyeDentify - An Android app for Object Identification. The document is intended as a reference for the software and general use.

2. Purpose

EyeDentify - An object identification app is used to identify and detect objects and retrieve the description of those objects. The user can use the app by taking the picture using camera or by choosing an image from the gallery.

3. Document Conventions

This document uses the dotted – numbered outline format to organize information and it contains different sections to describe the functional and Non- functional requirements and scope of this app.

4. Intended audience

This requirements document is intended for end-users and programmers of this software system.

5. Project Scope of Document

This requirement document provides information about the EyeDentify (Object Identification) app.

The application has the following functions:

i. This app can be downloaded from the Android Play store in any android mobile devices.
ii. User can take picture from camera / select an image from the gallery to get the description of the objects in that image.
iii. Allows user to take pictures using camera to get description of the objects.
iv. Can detect and display description of more than one objects in that picture.
v. Search the images from the gallery based on the given search word.

6. Definitions

i. EyeDentify – The name of the android mobile application.
ii. Gallery – The image gallery where all the pictures are stored in the android mobile phone.
iii. Microsoft cognitive service - Cognitive Services are a set of machine learning algorithms that Microsoft has developed to solve problems in the field of Artificial Intelligence (AI).

7. References

i. Microsoft cognitive Service

https://azure.microsoft.com/en-us/services/cognitive-services/

ii. Object Detection

https://www.edureka.co/blog/tensorflow-object-detection-tutorial/
https://arxiv.org/pdf/1506.02640.pdf
https://www.cis.upenn.edu/~jshi/papers/obj_det_liming_accv07.pdf
http://cs.brown.edu/people/pfelzens/papers/detection.pdf

iii. Gradle

https://guides.gradle.org/building-java-web-applications/
https://gradle.org/guides/

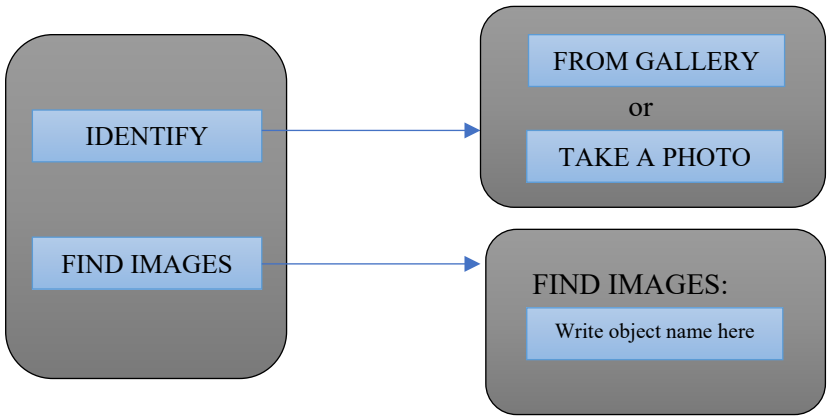
II. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to introduce the basic functionality of it.

1. Product Perspective

EyeDentify application will be used to detect the object. This application will communicate to Microsoft cognitive service which in turn provides the object description. The functionality provided by the Microsoft will be embedded into the application for the user to be able to use the functions in the application in a seamlessly manner.

2. Product Functions



With this mobile application, the users will be able to identify the object and search images in the gallery based on the search word. The result will be based on the criteria the user inputs.

2.1 Identify

The user will have the option to either select the picture from the gallery or take the picture using the camera to identify the object in that picture. The user will have to take a picture using camera and then this application will identify the object in that picture. For selecting the image from the gallery, user will be allowed to select one picture from the gallery at a time.

2.2 Find Images

This functionality will allow user to search all the images from the gallery. The search criteria will be based on the object name. The result of the search will be all the images from the gallery containing that object.

3. User characteristics

The users of the EyeDentify application will be the users that interact with the system. The EyeDentify application users can only use the application to detect an object or find the images having that object. In order for the users to get a relevant search result, there should be a precise keyword of the object. The user is expected to have knowledge about how to navigate a mobile application.

4. Constrains

The mobile application is constrained by the object description to the Microsoft Cognitive service. Since we are using the Microsoft cognitive service to detect the object description, the result will most likely not be the same for every one of the pictures. The system must also be able to display the description that Microsoft cognitive service will return.

The application will require the permission to access the gallery and camera in that android phone to perform the functionalities. The Internet connection is also a constraint for the application since the application uses the Microsoft API over the internet to fetch the object description.

5. Assumptions and dependencies

One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the phone does not have camera, this application will not work at all. This application will also require the permission to access the camera and the gallery in the mobile phone where the user will try to use the application.

III. Functional Requirements

i. Summary

This android application will use Microsoft cognitive web service to identify description for an image and will find all related images with matching description in image gallery. This application uses the following tools:

- Android Studio as integrated development environment.
- Git for version control
- Junit for unit testing
- Microsoft Cognitive service
- Gradle for build automation system to build android packages

ii. Limitations

- Objects are generally not detected if they are very small (less than 5% of the image).
- Objects are not differentiated by brand or product names (different types of sodas on a store shelf, for example).

1. Connectivity

This application uses real time web services. Hence connection to Internet is basic non-functional requirement for this. Following are the cases the user may encounter while trying to access this app.

1.1 Use Case 1

Name	: Connectivity Ok
Priority	: Essential
Trigger	: App and user interaction
Precondition	: App should be installed in the android device

Basic Path:

1. User starts the android application on mobile device.
2. Android application searches for internet connection to search for an image.
3. As connectivity is there app gives user two options i.e. to use gallery and to use camera.
    a. Requirements: This test case is the base test case or the default state of the app when opens.

1.2Use Case 2

Name	: No Internet connectivity
Priority	: Essential
Trigger	: App and User interaction
Precondition	: App should be installed in the android device

Basic Path:

1.User starts the android application on mobile device.
2.Android application searches for internet connection to search for an image.
3.As connectivity is not there i.e. internet connection is not there App shows either "Retry" or "Blank Screen"

2. Image selection

For this application an image will be provided as input, this image can be sourced from gallery or a new picture can be taken from camera. Following are the different test cases user may encounter when trying to select an image.

2.1Use Case 1

Name	: Access to camera if source is selected as camera
Priority	: Essential
Trigger	: User interaction with mobile device
Precondition	: Android device should have internet connectivity and app should be installed in mobile.

Basic Path:

1.User starts the android application on mobile device.
2.Android application starts with options of Camera or gallery.
3.User selects Camera.
4.As access to camera is already given by user app opens camera in mobile device.

2.2Use Case 2

Name	: Access to storage if source is selected as gallery
Priority	: Essential
Trigger	: User interaction with mobile device

Precondition : Android device should have internet connectivity and app should be installed in mobile.

Basic Path:

1. User starts the android application on mobile device.
2. Android application starts with options of Camera or gallery.
3. User selects Camera.
4. As access to camera is not given by user app pops out a window with asking for camera access.

2.3 Use Case 3

Name : Image type supported (JPEG, JPG, PNG) etc
Priority : Essential
Trigger : User interaction with mobile device
Precondition : Android device should have internet connectivity and app should be installed in mobile.

Basic Path:

1. User starts the android application on mobile device.
2. Android application starts with options of Camera or gallery.
3. User selects gallery.
4. If image type in gallery is not in the list of supported types, image cannot be loaded from app.

2.4 Use Case 4

Name : Image size to be less than 10 MB
Priority : Essential
Trigger : User interaction with mobile device
Precondition : Android device should have internet connectivity and app should be installed in mobile.

Basic Path:

1. User starts the android application on mobile device.
2. Android application starts with options of Camera or gallery.
3. User selects gallery.
4. If image selected from gallery is more than 10MB then image cannot be loaded.

3. Image Identification

For the application an image will be provided as input, this image can be sourced from gallery or a new picture can be taken from camera. Once image is uploaded into app it detects the items in the image and gives the description of the object as output. Following are the different test cases user may encounter when app tries to detect the objects in an image.

3.1 Use Case 1

Name	: One object in image
Priority	: Essential
Trigger	: App interaction with image
Precondition	: Android device should have internet connectivity and app should be installed in mobile and image should be selected/captured.

Basic Path:

1. User starts the android application on mobile device.
2. Android application starts with options of Camera or gallery.
3. User selects/captures an image.
4. If image selected from gallery has one object, app should be able to identify that single object and return the description of object as output in a bounded box.

3.2 Use Case 2

Name	: Multiple objects in image
Priority	: Essential
Trigger	: App interaction with image
Precondition	: Android device should have internet connectivity and app should be installed in mobile and image should be selected/captured.

Basic Path:

1. User starts the android application on mobile device.
2. Android application starts with options of Camera or gallery.
3. User selects/captures an image.
4. If image selected from gallery has multiple objects, app should be able to identify those objects and return the description of all the objects in image as output in a bounded box.

3.3 Use Case 3

Name	: Human and Animal images
Priority	: Essential
Trigger	: App interaction with image
Precondition	: Android device should have internet connectivity and app should be installed in mobile and image should be be selected/captured.

Basic Path:

1.User starts the android application on mobile device.
2.Android application starts with options of Camera or gallery.
3.User selects/captures an image.
4.If image selected from gallery has human or animal images, app should return the description of human as either “A Man” or “A woman” or animal name as output in the bounded box.

3.4 Use Case 4

Name	: Portrait images
Priority	: Essential
Trigger	: App interaction with image
Precondition	: Android device should have internet connectivity and app should be installed in mobile and image should be selected/captured.

Basic Path:

1. User starts the android application on mobile device.
2. Android application starts with options of Camera or gallery.
3. User selects/captures an image.
4. If image selected from gallery has portrait image, app should output description of the object in the bounded box.

3.5 Use Case 5

Name	: negative case, not clear image
Priority	: Essential
Trigger	: App interaction with image
Precondition	: Android device should have internet connectivity and app should be installed in mobile and image should be selected/captured.

Basic Path:

1. User starts the android application on mobile device.
2. Android application starts with options of Camera or gallery.
3. User selects/captures an image.
4. If image selected from gallery has blurred or unclear image, app may or may not output description of the object in the bounded box.

3.6 Use Case 6

Name	: Should be able to find and show all images from gallery where description matches
Priority	: Essential
Trigger	: App interaction with image
Precondition	: Android device should have internet connectivity and app should be installed in mobile and image should be selected/captured.

Basic Path:

1.User starts the android application on mobile device.
2.Android application starts with options of Camera or gallery.
3.User selects/captures an image.
4.Once app detects the object in the image, all images with matching description in the gallery should be given as output.

IV. Non-functional Requirements

The application will have background images to match the current weather status.

1. Reliability

The application should be able to identify the objects in Images using the Microsoft Cognitive API and should also pull out corresponding similar other images in the gallery.

2. Robustness

The application shall display errors in the event of no internet connection, no API available, or if the API is unable to identify objects in image.

3. Maintainability

The application shall be periodically tested to ensure maintained connectivity to the API, and be updated any time the API structure or link is updated.

4. Security

The application shall not store nor share any other information from the mobile device to other sources.