**CodeZillas**

# Look It
# Software Requirements Specification
# For Classify the Face Emotions

**Version 1.0**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 12-Feb-19 | 1.0 | Document created using initial plans | S.Sabesan |
| | | | |
| | | | |
| | | | |

# Table of Contents

# Software Requirements Specification

## 1. Introduction

The introduction of the Software Requirement Specification (SRS) provides an **overview** of the entire SRS. It includes the **purpose, scope, definitions, acronyms, abbreviations, references, and overview** of the SRS. This document explains whole software scenario for **Look It**, an Android application to detect the face emotion of the user by defining software **requirements for functionalities and non- functionalities**. Moreover, the **technologies used, software design constraints and assumptions** for above topics are also included in this document.

### 1.1 Purpose

The purpose of this SRS document is to analyze all the functional and non-functional requirements that define the system for this Android application. For further development, more requirements, technologies and software design architecture are sorted out so that goals which should be achieved during the system development will be identified easily.

### 1.2 Scope

The scope of this project is to develop an Android application which **pick an real time image** of a user face and **give the text output** of the emotion. The system should receive the results data from emotion API, send them to the user when the user's system is online.

### 1.3 Definitions, Acronyms, and Abbreviations

- Emotion API – The API which gives results by getting real time pictures from users as input and giving **probability** of an emotion to the given input as output
- REST API - A RESTful API is an application program interface (API) that uses **HTTP requests to GET, PUT, POST and DELETE data**.
- User – Person who **interacts** with the system

### 1.4 References


- ei.yale.edu, [Online]. Available: http://ei.yale.edu/mood-meter-app/
- itune.apple.com, [Online]. Available: https://itunes.apple.com/us/app/touch-and-learn-emotions/id451685022?mt=8


### 1.5 Overview

Overall description explains the overview of the system. The **basic functionalities** of the system will be described with the general matters regarding the system and the requirements **without stating the specific requirements**. Moreover, the background of the requirements and such details are also discussed here. Finally, **the assumptions and constraints regarding the project are described**. Also the technologies which have been **planned** to be used are also provided in this document.

# 2. Overall Description

## 2.1 Product Perspective

The system is an Android application which **handles the results data of the Emotion API** and **hands them over to the users** when their systems are online. When a classification is done by an Emotion API using a **certain user images**, the user will be **notified through text reply** informing that the classification processing has been finished. Thereafter the user has to **decide whether he/she want to see a video or not**.

But when the user fails to collect the results within a certain period, the input image as well as the results text will be **erased** from the application. Thus, a REST API has to be developed so as to receive the results data from Emotion API, send them to the user when the user's system is online. It will make sure that the user will **receive the results data without loss**.

## 2.2 Product Functions

There are **no user accounts** because of this app get image from users and **send through REST API** and give output as text **Emotion API is secure** so we don't need user Accounts and It will send the **accurate** results to the **respective user**.
Moreover, the system has to receive and send all the result data to the respective users without any loss.

## 2.3 Product Characteristics

Users are the people to **interact** with the system by having this application in the system. They are expected to have just the **basic technological knowledge** to use Android application to handle the camera. The system will be an online system with a **user friendly interface** to meet those requirements.

## 2.4 Constraints

The main constraint of the system is the **internet access** because it is an **online system**. Since the **online database** is used, it is necessary to have a **stable and fast internet connection**. Moreover, the size of the server is **not a limitation** because the results data is a **small size** and system **doesn't keep the result**.

## 2.5 Assumptions and dependencies

The Emotion API should accept the REST API to send the **respective results** data to the system. Moreover, the system should be able to **access the camera** in Android mobile of users in the Android application to ensure that the **results are sent to the respective users**.

# 3. Specific Requirements

## 3.1 Functionality

### 3.1.1 Capture an image from User

The system should be able to **access the camera** from the user's Android mobile and capture the real time image of the user.

### 3.1.2 View received result details

The user should be able to see all the **results data as text** obtained from the Emotion API which belong to him/her.

### 3.1.3 Receive results from Emotion API

When a **classification is finished at the Emotion API**, the system should get those results from the Emotion API and sent it to the REST API.

### 3.1.4 Suggest Videos

Suggest videos to the users **according the Results** Which is get from the Emotion API. Suggest videos for negative emotions.

### 3.1.5 Sent Image to the Emotion API

Send the Image which **captured** from the User's mobile to Emotion API.

### 3.1.6 Share results

When the user requests to share the received results, the system should let him share the result to his/her **social media**.

## 3.2 Usability

### 3.2.1 Friendly and attractive user interface

Users of the system should be able to **interact** with the system easily and the **interaction** should be **easily understandable** without any ambiguities.

### 3.2.2 Ease to use

The users will **not need any extra skills** or **high technical knowledge** to use the system. Basic knowledge about using the Android Mobile will be enough to use the system.

### 3.3 Reliability

#### 3.3.1 Availability

The **Look It** Android Application will be download through online and **available anytime** for any users. Unless there are **any server side problems**, the system will have **no limitations** for the users.

#### 3.3.2 Accuracy

The system will not **lose any results** from the Emotion API and will make sure that the results are sent to the **respective users**.

### 3.4 Performance and Security

#### 3.4.1 Secured User Images

System should be able to **provide secured** for user's images as input of the system. The results data may be **containing personal or confidential information** for some user. Thus, it is necessary to have **secure for user images**.

#### 3.4.2 Injection secured system

There are possibilities to be **attacked by injections**. System should be able to **protect its users from the injections**. Thus it should provide a secured way to input and output data to and from the **database sub-system**.

### 3.5 Supportability

#### 3.5.1 Adaptability

As it is an Android Application system, any users can **access** this system using their Android Mobile easily. System is developed so as to respond to **all popular Android Mobiles**. So system may be **adapted to the new environment**. Moreover, the system should be able to all sizes of mobile screens such as Huawei, Samsung, OPPO, etc mobile screens.

#### 3.5.2 Configurability

There is a one configuration to **access camera** needed to reach the system in the user's perspective. Anyways, user need to have Internet connection to reach the Online System.

### 3.6 Design Constraints

#### 3.6.1 Software languages

The REST API will be developed using **Java** and the Interface will be designed using **Android Studio** which are the **popular and highly used languages** to develop Android based applications.

#### 3.6.2 Development tools

The system will be using **MySQL** as the database. **Android Studio IDE** will be used to develop the system.

### 3.7 Purchased Components

As this is an Android based system, the system deployment may need some of the following purchased components.

- Huge Web Server for the online storage because it may have to **deal with large number of videos**.

### 3.8 Interfaces

#### 3.8.1 User Interfaces

- **Interface to take image**
  This will be the first Interface that the users will see when entering the system. It will have a button to access camera.

- **Received Text Result display Interface.**
  The details of the received results text will be shown in a **Text format**. It will have option to **share** the respective results.

- **Video suggestion Interface**
  If the user have negative emotions then the system will suggest the videos according the emotions from the MySQL database.

#### 3.8.2 Hardware Interfaces

The Look It application requires **Android Mobiles**. Users can use the system using their existing Android devices such as Huawei, Samsung, OPPO. But they need **active network connection** to connect to the Emotion API. So they need some hardware such as **mobile hotspot or Wi-Fi Routers** to connect to the Internet from their Mobile.

#### 3.8.3 Software Interfaces

The platform does **not need any specific software interface** for the client side usage. Users can use the application using their **existing mobile.**

#### 3.8.4 Communications Interfaces

For the Communication, the system will use the **HTTP/HTTPS protocol**.

### 3.9 Database Requirements

Look It application will be using MySQL database as the centralized data storage. It is a free and open-source cross-platform document-oriented database program. Classified as a SQL database program, MySQL uses JSON-like documents with schemas.
Users can not delete ant data from the database. They can ignore the videos from the Database. System is needed to back up the database for safety.

## 4. Supporting Information

Table of contents is given on page 3 of the document.