Software Requirements Specification

for

EasyDataLabeler

Version 1.0 approved

Prepared by : Dvij Kalaria
Roll no. : 18CS10018

IIT Kharagpur

12 March 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this software is to provide an interface on phone(android) to label raw data in XML format such that they can be used for training deep learning models. Also, this software shall provide a common portal for data labelers to easily put up their unlabeled data sets. The data shall be of the form of an image for which image labels are created or an audio file for which recognized speech is mentioned in the form of speech. This notated data shall be obtained in the form of an XML file or any format as required by the data set which can be extracted by that DL model as training data.

1.2 Document Conventions

As such, there is no particular convention followed in this document

1.3 Intended Audience

These tools shall be extremely helpful for data scientists to put up their unlabeled data set on a common portal. This app shall act as an interface between the data labellers who can easily annotate it through a touch screen interface.

1.4 Product Scope

As such there are no software available to label data on a phone interface like android. However this software is meant to run on android platform but once completed shall also be made available on IOS and other phone interfaces. There by it shall be convenient, easy and fasten up the process for data labellers as now instead of pointing the mouse pointer, the labeler can easily point out the locations by touching on the screen position.

1.5 References

https://github.com/wkentaro/labelme : Replicating the functions of this software meant for Windows, Ubuntu and Mac operating systems on Android

2. Overall Description

2.1 Product Perspective

This application is a replica of the features as provided by LabelMe software available for download from the github repository whose link is mentioned above. This is a first product in the series with base version (v 1.0.0). Further improvements, bug fixes, extending to other forms of data sets like annotating voice data etc. in the later versions.

2.2 Need

With the growing use of deep learning in various applications, it has been an utter need to get more annotated data to train these model to make them more accurate and robust. Currently data scientists hire data labellers to get their work done. This software will enable data scientists to put up their requirements on a common platform which can be addressed by the data labellers to label them in the required format.

2.3 Work plan

This project shall be completed in 2 months i.e. 8 weeks. The work flow for the project shall go in following manner:-

1. Week 1 & 2: In the first 2 weeks, I will try to learn the concepts required for the project and prepare the SRS document so that I can get a base reference for how to continue and proceed further. The concepts like Android Studio, socket programming to manage resources on a local network shall be used for my project.

2. Week 3 : Basic tools starting with image classification like polygon draw, rectangle draw, adding labels and convert them to XML in the correct way shall be implemented.

3. Week 4 : Work on the ineffective semi-working user interface (design) to navigate to a particular data set. Also add button for the data scientist to add a project on the portal, mark the completed ones.

4. Week 5 : Work on the socket programming to communication channel for enable data scientists to put up their work online via local server as a server and the users as clients by enabling download of this content on their offline devices for use.

5. Week 6 : Develop an undebugged working application for validation to be modified further and start removing bugs.

6. Week 7 and 8 : Validation and debugging to make the software as convenient as possible.

2.4 Product Functions

The functions provided by this software are as follows :-
1. The data scientists can upload their data sets on the server.
2. The data labellers can download this data on their offline devices and further label them.
3. Image annotation for polygon, rectangle, circle, line and point. (tutorial)
4. Image flag annotation for classification and cleaning. (#166)
5. Video annotation. (video annotation)
6. GUI customization (predefined labels / flags, auto-saving, label validation, etc). (#144)
7. Exporting VOC-format dataset for semantic/instance segmentation. (semantic segmentation, instance segmentation)

8. Exporting COCO-format dataset for instance segmentation. (instance segmentation)

2.5 User Classes and Characteristics

1. Log in : Separate accounts shall be maintained for data scientists to upload data sets and data labellers to download and further annotate the data.
2. Create account
3. Forgot password
4. For data scientist :-

- Start a new project (Define the kind of data label required i.e. segment image, identify a flag from image, track an object from video)
- Upload raw images or videos to a project to be labeled
- Download labeled images

5. For data labeler :-

- Open an unmasked data image by downloading it temporarily to be editted offline.
- Apply various tools to create labels of different shapes like polygon, circle, rectangle, spline etc. on the unmasked image.
- Upload the masked image in VOR or XML format back to the server and mark it as labeled.

2.6 User Interfaces

1) GUI along with meaningful Frames and buttons.
2) Toolbar for necessary tools as per requirements.
3) User Registration
4) Adding new projects.
5) Tab to evaluate progress of work done by data labeler.
6) Signing in by user.

2.7 Software Interfaces

1) Front End – Java (Software used to be android studio)
2) Back End – Basic server client communication in Java
3) Time Tracking (Example - Time Trex) as code is available for a short amount of time.
4) Captcha for preventing machine/robot usage

2.8 Hardware Interfaces

1) Hardware environment - Dual Core 2nd generation or higher
2) System Configuration - RAM - 512 MB HDD 4 GB or higher
3) Operating System - Android
4) Local server for Database storage (To be tested on ernet.iitkgp.ac.in server bu can be used in general for anything.

2.9 Design and Implementation Constraints

As per the initial framework some drawing tools like spline curve shall not be available so labeling to a certain precision shall not be possible also due to small screen size.

2.10 User Documentation

Refer to the user document that will be attached with the application and can be viewed from the GUI of the starting page.

2.11 Assumptions and Dependencies

The software is expected to run assuming the presence of a local server. Both the clients i.e. data labelers and data scientists have to be connected to the common local server to exchange data and use the software. So bare minimum requirements involve a local server and a wi-fi enabled android device.

3. External Interface Requirements

3.1 User Interfaces

User interface similar to labelme github tool as available as a github repository.

3.2 Software Interfaces

The server must have decent processing unit (6 cores at 3 GHz average clock speed), also sufficient storage and a good graphical processing unit to enable efficient and fast communication of visual data. The user side must have an active working wi-fi hardware to connect to the local network and retrieve or upload data on it.

3.3 Communications Interfaces

Communication shall be carried with the help of a local server. All types of communication including the visual image data and it’s corresponding HTML file containing the data labels shall be used.

4. System Features

4.1 Creating an account

4.1.1 Description and Priority
This is the first step for any user to start using these services. The user needs to create an account by signing up to these services. Registration is necessary to avoid any false, unauthorized users or machine codes to make large no. of accounts to block server space from making account.

4.1.2 Stimulus/Response Sequences
User has to input his personal, educational and professional details after specifying the account type.

1) If the entered details are proper and acceptable i.e. The user-name is unique, email-id is correct, captcha matches with the required one. The system will create space on the server to store the details for the same.
2) If the entered details are incorrect, dialog box with the corresponding explanation to in-approval of the requested details which maybe due to incorrect email-id or any blank necessary field, already existent user-id, poor password, re entered password not matching with the first one

4.1.3 Functional Requirements
● Ask for account type – data scientist or labeler
● Ask for user details (Same for both types).
● Allocate space in the server memory with necessary details.
● A function to provide a secure user-name (unique) and password to the user.
● A function to provide an extension to keep the user logged in.

4.2 Logging in to the account

4.2.1 Description and Priority
After creating the account, the user has to login to his/her account before using any services.
This is necessary for authorizing the user.

4.2.2 Stimulus/Response Sequences
User has to fill his/her account authorization account details i.e. user-name or email-id, password and the captcha.
1) If the entered details are proper and acceptable i.e. the user-name or email-id matches with it's corresponding password stored on the server, captcha matches with the required one. The system will create space on the server to store the details for the same.
2) If the entered details are incorrect, dialog box with the corresponding explanation to in-approval of the entered details which maybe due to incorrect email-id or any blank necessary field, wrong password.

4.2.3 Functional Requirements
● Ask for account type – data scientist or labeler
● Ask for user-name and password until it is correct.
● A function to provide an extension to keep the user logged in.

4.3 Start a new project (For data scientists)

4.3.1 Description and Priority
To start a new project to upload a similar data-set on the server.
This is necessary for authorizing the user.

4.3.2 Stimulus/Response Sequences
User has to fill the project name, the type of data, supported data formats as input.
1) If the project name is unique and and all the details are filled correctly, the corresponding project is opened.

2) If the entered details are incorrect, dialog box with the corresponding explanation to in-approval of the entered details which maybe due to unacceptable project name or data type, is shown and the user is asked to re-enter the details.

4.3.3 Functional Requirements
● Ask for the project name and other details until they are acceptable.

4.4 Uploading data images

4.4.1 Description and Priority
User (data scientist) can then upload some test images on the current image. The maximum total size of all the files that are uploaded is 500 Megabytes.

4.4.2 Functional Requirements
● The user navigates to the required directory to upload the files to the system in the necessary format.
● System stores the files on a common space accessible to all the users.
● User can access these files anytime and even download them.

4.5 Label and upload labeled data images

4.5.1 Description and Priority
User (data labeler) can then label the image from any project. After labeling the labeled data is then put back on the server and marked labeled.

4.5.2 Functional Requirements
● The user navigates to the required project.
● The user downloads the required image to be labeled from the project,
● The user adds labels to the downloaded image. The tools available for labeling shall be rectangle, polygon, circle etc.
● The corresponding HTML file is formed to account for all the labels.
● The HTML file is uploaded to the common server space in a separate folder to account for labeled images and the corresponding image is marked as labeled.

4.6 Download data images

4.6.1 Description and Priority
User (data scientist) can then download the set of labeled images from any project into his/her device.

4.6.2 Functional Requirements
● The user navigates to the required project.
● The user can either select specific images whose HTML files along with the labeled images are to be downloaded or download all the images marked as labeled.

5. Other Nonfunctional Requirements

5.1 Performance Requirements
The response of server should be fast (within 2 seconds) when coupled with high speed Internet. The photos uploaded to the system must be compressed to 3 mb limit before being uploaded to maintain system space and improve performance.

5.2 Safety Requirements
There is no risk of any threat that can be issued by the system. Since it is a server based application, there will definitely be chances of network based threats. These need to be minimized as much as possible. A virus scanner will scan all the documents and photos before a user can upload it.

5.3 Security Requirements
All the personal data stored for a particular user will be encrypted by using the standard encryption algorithm. This needs to be protected from any possible data theft.

5.4 Software Quality Attributes
The software must work with at least the IIT Kharagpur local network on which it shall be tested. User must have a fast wi-fi connection. Software must operate on android devices.

5.5 Business Rules
This system is going to be rolled out free. Though to sustain our project we will be endorsing paid ads on side panes.

6. Goals of Implementation

6.1 Extension to operate on global server
As an additional feature, in the future versions, the software shall also work for global server i.e. Internet.

6.2 Data privacy for data uploaded by different firms
In the future, aim is to enable different channels for uploading data to the server by different firms which shall have different rules for pay based on the data labeled. Data protection for the uploadeed data shall be planned by making limited part of data available to be used by the users.