Software Requirements Specification

for

“photoRead: An App for Optical Character Recognization”

Version 1.0 approved

Prepared by

Group 6

Shivanshu Shrivastava TE COMP B 22

Anuja Somthankar TE COMP B 35

Vedant Pandya TE COMP B 49

TCET

21st August, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Vedant	21/08/20	First Draft	1.0

1. Introduction

1.1 Purpose

photoRead is a mobile application, which uses sophisticated Deep Learning models for performing Optical Character Recognition on user submitted images. Users are able to submit images to the pre-trained model, via the Mobile Application. The pre-trained model, then returns a captioned image, with the detected text. This project can be used for License Plate Detection, Sign Board Identification, Menu Card Reader and other application which require OCR. This document consists of the Software Requirements Specifications (SRS) of "photoRead: Optical Character Recognition using DL"

1.2 Intended Audience and Reading Suggestions

This document is intended for College Faculties, Project Team Members and Project Evaluators. This document contains the requirements of the project "photoRead: OCR using DL", starting with an Introduction of the Project, followed by Description of the Project, External Interface Requirements, System Features and Non-Functional Requirements. Document Readers are suggested to follow the inherent sequence, for better comprehension and easy understanding of this document.

1.3 Product Scope

photoRead is a Mobile Application, which allows for Text Detection in images submitted by user. This application can be used for only detecting handful of words inside the image, and cannot be used for detection of text in handwritten scanned documents. This application is intended for detecting text written in English language only. The deep learning model used in the application can detect only uppercase letters (A-Z), lowercase letters (a-z) and digits (0-9). It is unable to detect punctuation marks and other symbols. This application needs to be designed, developed and deployed in a span of 4 months.

1.4 References

This document refers http://vlabs.iitkgp.ac.in/se/1/ for identifying Requirements from Problem Statement.

2. Overall Description

2.1 Product Perspective

photoRead is a mobile application, written in Android Studio, which allows user to detect text in submitted images. It uses Deep Learning CNN's for performing OCR on the image and return the image, captioning the detected text and creating a bounding box highlighting the detected text. Users also have the option of translating the detected text in their native language, allowing users to better comprehend the text, in the image. Users can also receive search results about the detected text, along with the caption. photoRead can easily integrate with Cloud, to allow OCR on personal images, stored over Cloud.

2.2 Product Functions

- *Text Detection in an Image: Using Sophisticated Deep Learning Techniques, Bounding Box is drawn over the detected image.*
- *Text Recognition in an Image: Using OCR, submitted images are returned with captions, consisting of the detected text in the image.*
- *Secure Login/Logout Service: Each user has a unique username and an encrypted password for account Security, which can be used for logging in to the application, to avail features like integration with Cloud and Search Results.*
- *Easy integration with Cloud for storing images: Logged In Users can easily connect with Cloud for using personal images.*
- *Search Results for the Detected Text: Search Results about the text detected, is returned along with the captions, for images submitted by Logged In Users.*
- *Personalized OCR Model based on User Images: Depending on the type of images submitted by users, the OCR Model is personalized for each user.*
- *Translate the Detected Text in Local Language: Logged In Users can choose to translate the detected language in their native language.*

2.3 Design and Implementation Constraints

The Implementation process needs to be Agile in Nature. Front End Developers should provide a separate Module for UI of the app, which can be easily integrated with the OCR Model. Each function should have a Docstring, explaining its purpose, inputs and outputs. camelCasing should be practiced, while naming variables and functions. All the required modules and API's need to be included at the start of each module. Modules for OCR Model should be written in Python language. Modules for the mobile application, must be written in Java Language. Sensitive information like Google Account Credentials, should be stored in a secure file and should be imported, only wherever necessary. The Mobile Application needs to be ready in a span of 4 months.

2.4 Assumptions and Dependencies

It is assumed that the submitted image will have at least 1 character that can be detected and that all the text to be detected is in English language only. It is also assumed that the provided image will be a clear image, upon which OCR is possible. tensorFlow is used for creating the OCR model. photoRead uses Google Translate API for translation and Google API for providing Search Based results.

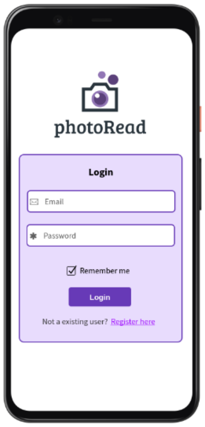
3. External Interface Requirements

3.1 User Interfaces

photoRead has four User Interfaces, namely:

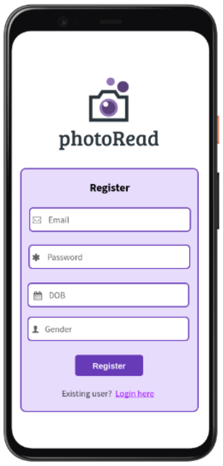
- *Login UI*
- *Registration UI*
- *Dashboard UI - I*
- *Dashboard UI – II*

3.1.1 Login UI



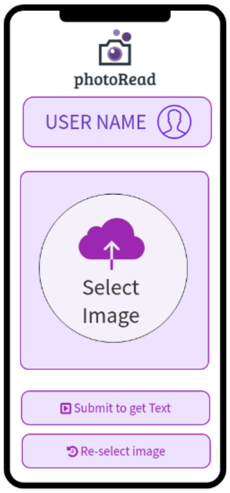
Login UI will be used by users to login. Each user has a unique username and an encrypted password, for secure Login/Logout. In case of invalid username, they will be redirected to Registration UI. In case of invalid password, they will be prompted to enter the password again.

3.1.2 Registration UI



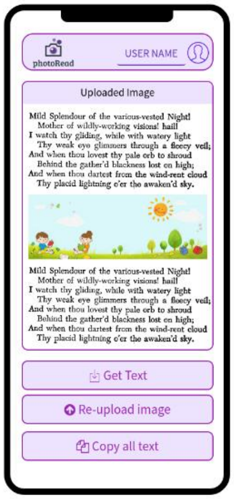
Registration UI will be used by users to register to photoRead, allowing them to access other functionalities like Search Results and Translation. In case of existing email, they will be prompted to enter a new email. In case of incorrect email format, they will be prompted to enter the email in the correct format. In case of weak Password, they will be prompted to enter a stronger password.

3.1.3 Dashboard UI-I



On successful Login, Users are redirected to their Dashboard. Dashboard allows the user to select the image, they would like to perform OCR on. It also provides users an option to re select the image, in case they would like to perform OCR on a different image. If the user is satisfied with the choice of uploaded image, they can submit the image for OCR, by clicking the "Submit" button.

3.1.4 Dashboard UI-II



Mild Splendour of the various-vested Night!
Mother of wildly-working visions! hail!
I watch thy gliding, while with watery light
Thy weak eye glimmers through a fleecy veil;
And when thou lovest thy pale orb to shroud
Behind the gather'd blackness lost on high;
And when thou dartest from the wind-rent cloud
Thy placid lightning o'er the awaken'd sky.

Mild Splendour of the various-vested Night!
Mother of wildly-working visions! hail!
I watch thy gliding, while with watery light
Thy weak eye glimmers through a fleecy veil;
And when thou lovest thy pale orb to shroud
Behind the gather'd blackness lost on high;
And when thou dartest from the wind-rent cloud
Thy placid lightning o'er the awaken'd sky.

Dashboard UI-II, displays the Submitted Image, along with bounding boxes over the detected text. The returned Image captions the text recognized in the image. Users are also provided an option for copying the detected text, to their clipboard. Logged In Users also receive Search Based Results, over the Detected Text and if selected, translated text, in the local language.

3.2 Software Interfaces

The Mobile Application sends the image, submitted by the user, to a hosted Python API, whose job is to return a JSON string, having the following fields: i) detectedText ii) boundingBoxCoOrdinates. This interface, contains the OCR model, responsible for text detection and recognition in the image. The Python API will be hosted on a Flask App, running as an API endpoint.

3.3 Communication Interfaces

FTP protocol will be used for transferring image from User to Python API. JSON string is returned to the User UI using HTTP Protocol. -TBD

4. System Features

4.1 Text Detection in an Image

4.1.1 Description and Priority

Using sophisticated Deep learning, an OCR(optical character recognition) model is built which is able to detect letters. A bounding box is drawn over the detected model. The priority of this feature is high.

4.1.2 Stimulus/Response Sequences

The user must first open the application. The user must login. They may then upload the image which they want to convert or detect text on. After submission of the image, the algorithm will be applied on the image and text will be detected. If there is no text, such a message will be given to the user to inform them.

4.1.3 Functional Requirements

REQ-1: User must provide a clear image, otherwise the UI must prompt the user to resubmit the image.

REQ-2: User must provide an image consisting text, otherwise the UI must prompt the user to resubmit the image.

4.2 Text Recognition in an Image

4.1.1 Description and Priority

Through the OCR model API written in Python, the uploaded image will be processed and text will be detected and the text will be recognized. Using OCR, submitted images are returned with captions, consisting of the detected text in the image.

The priority is high because the recognition of text is a basic functionality for the rest of the application.

4.1.2 Stimulus/Response Sequences

The user must first open the application. The user must login. They may then upload the image which they want to convert or detect text on. After submission of the image, the algorithm will be applied on the image and text will be detected. Using OCR, submitted images are returned with captions, consisting of the detected text in the image.

4.1.3 Functional Requirements

REQ-1: The recognized text, must be captioned, below the image.

REQ-2: The recognized text, must be enclosed in a bounding box, inside the text, to highlight the detected text.

REQ-3: In case of no text detected, the UI must prompt user to submit a clearer image or a new image.

4.3 Secure Login/Logout Service

4.1.1 Description and Priority

The user must login if they want to save their document and have access to the database. Only after that will they be able to translate the document and connect to the Cloud to save the results. The priority for this service is high because there is always a severe need of security.

4.1.2 Stimulus/Response Sequences

When the user opens the application, the login window or register window will be opened where the user must enter his or her details to use the application. If the details for logging are incorrect, the pop up message stating the same will be shown. If the details are correct, the user will be logged in.

4.1.3 Functional Requirements

REQ-1: In case of correct Username and Password, the user must be redirected to Dashboard.

REQ-2: In case of incorrect Username, the user should be prompted to re-enter the username or the registration button needs to be highlighted.

REQ-3: In case of incorrect Password, the user should be prompted to re-enter the Password or the Forget Password Button should be highlighted.

4.4 Easy integration with Cloud for storing images

4.1.1 Description and Priority

If the user is logged in, the integration with Cloud is done. Logged In Users can easily connect with Cloud for using personal images. The priority for this functionality is medium because although this is not required for the core functionality of detection and recognizing text, it is required to work further on that text, like translating it, storing it etc.

4.1.2 Stimulus/Response Sequences

The user will first have to log in for the use of this feature. Once logged in, he or she can pick out an image stored on their cloud and upload it easily for processing. Once the processing is done, the digital text can be saved in the form of a document on their personal cloud. Similarly, if they want to translate the text detected, they can do so through one click presented.

4.1.3 Functional Requirements

REQ-1: Only Logged In Users are allowed for integration with Cloud.

REQ-2: In case non Logged In Users try connecting to Cloud, UI should prompt the users to login.

4.5 Search Results for the Detected Text

4.1.1 Description and Priority

If the user is logged in, the integration with Cloud is done. Because of this, the user can easily produce search results for the text that is recognized. Search Results about the text detected, is returned along with the captions, for images submitted by Logged In Users. The priority for this functionality is low because it is not a core functionality of the application and it is an independent feature upon which no other feature is dependent on.

4.1.2 Stimulus/Response Sequences

The user will first have to log in for the use of this feature. Once logged in, he or she can pick out an image stored on their cloud and upload it easily for processing. Once the processing is done, the digital text is shown to the user after which they can easily search the text through one-click.

4.1.3 Functional Requirements

REQ-1: Only Logged In Users are allowed to have search results accompanying the captioned text.

REQ-2: In case non Logged In Users want Search Results, UI should prompt the users to login.

4.6 Personalized OCR Model based on User Images

4.1.1 Description and Priority

If the user is logged in, the integration with Cloud is done, which is required for this function. Depending on the type of images submitted by users, the OCR Model is personalized for each user. The priority for this functionality is low because it is not a core functionality of the application and it is an independent feature upon which no other feature is dependent on.

4.1.2 Stimulus/Response Sequences

The user will first have to log in for the use of this feature. Once logged in, he or she can pick out an image stored on their cloud and upload it easily for processing. Depending on the type of images submitted by users, the OCR Model is personalized for each user if they choose that option.

4.1.3 Functional Requirements

REQ-1: TBD

4.7 Translate the Detected Text in Local Language

4.1.1 Description and Priority

If the user is logged in, the integration with Cloud is done, which is required for this function. Once the text is recognized, using the Google API, we can recognize the language in which the text is written and translate it into the language understood by the user. The priority for this functionality is low because it is not a core functionality of the application and it is an independent feature upon which no other feature is dependent on.

4.1.2 Stimulus/Response Sequences

The user will first have to log in for the use of this feature. Once logged in, he or she can pick out an image stored on their cloud and upload it easily for processing. After the text is recognized, the user will have the option to translate the text into any language they want.

4.1.3 Functional Requirements

REQ-1: Only Logged In Users are allowed for translation.

REQ-2: In case non Logged In Users try for translation, UI should prompt the users to login.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

photoRead application should have a loading screen, which stays on the screen for at least 3000-3500 ms. In this duration, all the necessary resources required for the app should be loaded. Every submitted image must be returned in less than 4000 ms. Text Recognition must be completed in less than 1500 ms. Every Mobile UI should be scalable and should fit to Landscape and Portrait modes of Mobile and Tablet Screens. Uniform UI Design must be followed for every UI. Logos, images, etc. used should be .svg's wherever necessary for easier scaling.

5.2 Safety Requirements

photoRead must not be used for extracting personal information for public images. Possible safeguards include a filter before showing the detected text, filtering any personal information about the user who submitted the image. Actions like performing OCR of confidential documents like Passport/Credit-Card/Debit-Card, etc. should be prevented.

5.3 Security Requirements

All passwords of users must be encrypted before storing them in databases. Only authorized users should have access to databases storing personal user information. User data used for personalization of the OCR model, should be obtained only after the permission is granted by the user. Users should be made aware, that their User Data will be used, if they choose Personalized OCR Model.

5.4 Software Quality Attributes

photoRead should be available on all Android Devices. It should be scalable, allowing for low and medium traffic. It should be flexible, allowing developers to add new updates, without having to change prior code. It should be maintained and should be checked for errors, weekly. It should be robust and easily portable. All modules should be testable and reusable.

Appendix A: Glossary

- *OCR: Optical Character Recognition*
- *DL: Deep Learning*
- *CNN: Convolutional Neural Network*
- *UI: User Interface*
- *API: Application Program Interface*
- *JSON: Javascript Object Notation*
- *FTP: File Transfer Protocol*
- *HTTP: Hyper Text Transfer Protocol*

Appendix B: To Be Determined List

- *User Interfaces*
- *User Classes and Characteristics*
- *Functional Requirements of Personalized OCR Based Model for Users feature.*
- *Business Rules*