Software Requirements Specification for image-based web scraping software

Version 0.6

**Prepared by:**

* **Ahmed Mohammed Abd El-Ghani**
* **Ahmed Mohamed Agamy**
* **Hussein Muhammad El-Sayed**
* **Islam**

Contents

[I. Revision history 2](#_Toc517159169)

[1. Introduction 2](#_Toc517159170)

[1.1. Purpose 2](#_Toc517159171)

[1.2. Scope 2](#_Toc517159172)

[1.3. References 2](#_Toc517159173)

[1.4. Overview 3](#_Toc517159174)

[2. overall description 3](#_Toc517159175)

[2.1. Product perspective 3](#_Toc517159176)

[2.2. Product functionality 3](#_Toc517159177)

[2.2.1. Functionality as diagram 3](#_Toc517159178)

[2.3. User Characteristics 4](#_Toc517159179)

[2.3.1. Stakeholders 4](#_Toc517159180)

[2.3.2. Users objectives 4](#_Toc517159181)

[2.4. Constraints 4](#_Toc517159182)

[3. specific requirement 4](#_Toc517159183)

[3.1. External Interfaces 4](#_Toc517159184)

[3.1.1. External Interfaces requirement 4](#_Toc517159185)

[3.1.2. User interfaces 5](#_Toc517159186)

[3.2. System Features 6](#_Toc517159187)

[3.2.1. Use case: Select the scrapping type (UC ID1) 6](#_Toc517159188)

[3.2.2. Use case: Upload new picture (UC ID2) 7](#_Toc517159189)

[3.2.3. Use case: Select website (UC ID3) 7](#_Toc517159190)

[3.2.4. Use case: Get report (UC ID4) 8](#_Toc517159191)

[3.2.5. Use case: Save report (UC ID5) 8](#_Toc517159192)

[3.3. Non-Functional requirement 9](#_Toc517159193)

[3.3.1. Object recognition 9](#_Toc517159194)

[3.3.2. Report generation time 9](#_Toc517159195)

[3.3.3. Report csv file size 9](#_Toc517159196)

[3.4. Class diagram 9](#_Toc517159197)

[4. System evolution 9](#_Toc517159198)

[5. Glossary 10](#_Toc517159199)

# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 0.1 | Ahmed M. Abd El-Ghani | Initial | 6-6-2018 |
| 0.2 | Ahmed M. Abd El-Ghani | Introduction | 7-6-2018 |
| 0.3 | Ahmed M. Abd El-Ghani | overall description | 8-6-2018 |
| 0.4 | Ahmed Agamy | specific requirement | 13-6-2018 |
| 0.5 | Ahmed Agamy | specific requirement | 14-6-2018 |
| 0.6 | Ahmed M. Abd El-Ghani | specific requirement | 14-6-2018 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to present a detailed description of the web scraping system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers.

## Scope

This software system will be web scraper that uses image to search web sites, it will be designed to minimize the time taken to collect information about product category in an online e-commerce platform, it will only need a picture or text of that product (ex: mobile picture), And it will provide an easy to read report that contain links ,prices and names of that product from a list of online market websites, this report can be tremendously useful in market evaluation witch is an important part of visibility study, it also can be use full for searching and facilitating purchase from most of online e-commerce platforms.

## References

* *IEEE*. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## Overview

This document is written according to the standards for Software Specification Requirement explained in “IEEE Recommended Practice for Software Specification Requirement”.

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements that is used to establish a context for the technical requirements specification in the third chapter.

The third chapter, the Specific Requirement section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language

The fourth chapter, the Verification section, of this document gives information about the testing phase of development and describe the black and white and box tests performed and their results.

# overall description

## Product perspective

This System provides simple mechanism for users to get range of prices and number models/alternatives of specific product category, Call and return {Main program and subroutine} architecture style is used.

## Product functionality

### Functionality as diagram

A close up of a logo

Description generated with high confidence

Figure 1. System functionality

## User Characteristics

### Stakeholders

#### System engineer

* Responsible for requirements gathering
* Responsible for development
* Responsible for deployment and support
* Responsible for testing

#### Users

* Search for the product category

### Users objectives

#### System engineer

* Gain Experience in software engineering and development

#### Users

* Get report contain list of prices, links and names of the product he/she want

## Constraints

The system is designed to be built using python with already made TensorFlow API object recognition model and selenium for web-scrapping are used to generate the report

# specific requirement

## External Interfaces

### External Interfaces requirement

#### Input

* Image file of extension JPG or PNG with minimum quality 300x300 pixels.
* Text contain the name of product.

#### Output

* Csv file (report) contain all links, names and prices of the product category.

### User interfaces

#### Screen image

A screenshot of a cell phone

Description generated with very high confidence

Figure 2. software interface

#### Description

First the user selects the scrapping type ether by text entered in the “scrap by text input filed” or by image selected using the button open, when selecting scrap by image the selected image appears in the space below “scrap by image input” and the path is displayed in it.

Then select the desired website (at least one option must be selected).

At last by clicking on start scrapping the software starts and generate the report files which is displayed in output field.

## System Features

A close up of text on a white background

Description generated with high confidence

Figure 3. Use case diagram

### Use case: Select the scrapping type (UC ID1)

#### **Participating actors:** User

#### **Entry condition:** Open the program

#### **Exit condition:** Select one option

#### **Typical flow of event:**

1. User select ether text or image scrapping
   1. Select the method of software operation

### Use case: Upload new picture (UC ID2)

#### **Participating actors:** User

#### **Entry condition:** Click open button in the software GUI

#### **Exit condition:** Successful upload the picture OR cancel

#### **Quality requirement:** minimum image quality is 300 X 300 pixels, image extensions JPG, PNG.

#### **Related requirement:** object recognition

#### **Typical flow of event:**

1. User upload an image
   1. The system will send the image to the TensorFlow API model to recognize the objects in the image.
   2. The API will send the name of the object/s recognized back to system.

#### **Exceptions:**

1. User upload an image
2. The API send error massage if low quality image or couldn’t recognize the object/s in the image

### Use case: Select website (UC ID3)

#### **Participating actors:** User

#### **Entry condition:** Upload image (UC ID2)

#### **Exit condition:** Select one or more website

#### **Related requirement:** Get report

#### **Typical flow of event:**

1. User select website
   1. Set the selected website/s URL to start web scrapping

### Use case: Get report (UC ID4)

#### **Participating actors:** User

#### **Entry condition:** Upload image (UC ID2), Select website (UC ID3) and click start scrapping button

#### **Exit condition:** Report successfully generated OR error getting data

#### **Quality requirement:** Response time on 1MBit internet speed for 1 report 40 ± 20 seconds

#### **Related requirement:** Save report

#### **Typical flow of event:**

1. User click on start scrapping
   1. System send recognized object name to web scrapper (selenium).
   2. Web scrapper search the website for the object’s name.
   3. Extract the links, names and prices of products (objects) from the website..
   4. Put the links, names and prices in a csv file (a file for each object on a website)

#### **Exceptions:**

1. User click on start scrapping
   1. Website doesn’t allow to download XML code

### Use case: Save report (UC ID5)

#### **Participating actors:** Get report (UC ID4)

#### **Entry condition:** Report successfully generated

#### **Exit condition:** Exit automatically after save

#### **Quality requirement:** The report/s are saved at the same location that contain the image file size 18 ± 2KB

#### **Typical flow of event:**

1. Save the file/s
   1. File/s are saved to default location

#### **Exceptions:**

1. Save the file/s
   1. No authorization to save in the default location
   2. Location doesn’t exist

## Non-Functional requirement

### Object recognition

The software can recognize 3 in one image of minimum quality of 300x300 pixel.

### Report generation time

Reports are generated and saved in 40 ± 20 seconds on internet of 1MBit speed and processor core i7 with 8GB ram.

### Report csv file size

Report are generated in csv file form and its size is 18KB for one file (object in one website).

## Class diagram

A screenshot of a cell phone

Description generated with very high confidence

Figure 4. class diagram

# System evolution

* Make our own TensorFlow API model to be able to recognize specific object not a whole category
* Increase the range of supported image extensions to include TIFF and GIF
* Give the software the ability to generate report in Microsoft word extension.

# Glossary