

SE 3XA3: Software Requirements  
Specification  
google-images-downloader

Team 201, CAS Dream Team  
Sam Crawford, crawfs1, 400129435  
Joshua Guinness, guinnesj, 400134735  
Nicholas Mari, marin, 400132494

April 5, 2020

# Contents

<b>1</b>	<b>Project Drivers</b>	<b>1</b>
1.1	The Purpose of the Project . . . . .	1
1.2	The Stakeholders . . . . .	1
1.2.1	The Client . . . . .	1
1.2.2	The Customers . . . . .	1
1.2.3	Other Stakeholders . . . . .	2
1.3	Mandated Constraints . . . . .	2
1.3.1	Schedule Constraints . . . . .	2
1.3.2	Solution Constraints . . . . .	2
1.3.3	Budget Constraints . . . . .	3
1.3.4	Off-the-Shelf Software . . . . .	3
1.3.5	Partner or Collaborative Applications . . . . .	3
1.3.6	Implementation Environment of the Current System . . . . .	3
1.4	Naming Conventions and Terminology . . . . .	3
1.5	Relevant Facts and Assumptions . . . . .	3
<b>2</b>	<b>Functional Requirements</b>	<b>4</b>
2.1	The Scope of the Work and the Product . . . . .	4
2.1.1	The Context of the Work . . . . .	4
2.1.2	Work Partitioning . . . . .	4
2.1.3	Individual Product Use Cases . . . . .	5
2.2	Functional Requirements . . . . .	6
<b>3</b>	<b>Non-functional Requirements</b>	<b>9</b>
3.1	Look and Feel Requirements . . . . .	9
3.1.1	Appearance Requirements . . . . .	9
3.1.2	Style Requirements . . . . .	9
3.2	Usability and Humanity Requirements . . . . .	10
3.2.1	Ease of Use Requirements . . . . .	10
3.2.2	Personalization and Internationalization Requirements . . . . .	10
3.2.3	Learning Requirements . . . . .	10
3.2.4	Understandability and Politeness Requirements . . . . .	10
3.2.5	Accessibility Requirements . . . . .	10
3.3	Performance Requirements . . . . .	10
3.3.1	Speed and Latency Requirements . . . . .	10
3.3.2	Safety-Critical Requirements . . . . .	11

3.3.3	Precision or Accuracy Requirements . . . . .	11
3.3.4	Reliability and Availability Requirements . . . . .	11
3.3.5	Robustness Requirements . . . . .	11
3.3.6	Capacity Requirements . . . . .	11
3.3.7	Scalability or Extensibility Requirements . . . . .	11
3.3.8	Longevity Requirements . . . . .	11
3.4	Operational and Environmental Requirements . . . . .	11
3.4.1	Expected Physical Environment . . . . .	11
3.4.2	Requirements for Interacting with Adjacent Systems . . . . .	12
3.4.3	Productization Requirements . . . . .	12
3.4.4	Release Requirements . . . . .	12
3.5	Maintainability and Support Requirements . . . . .	12
3.5.1	Maintenance Requirements . . . . .	12
3.5.2	Supportability Requirements . . . . .	12
3.5.3	Adaptability Requirements . . . . .	13
3.6	Security Requirements . . . . .	13
3.6.1	Access Requirements . . . . .	13
3.6.2	Integrity Requirements . . . . .	13
3.6.3	Privacy Requirements . . . . .	13
3.6.4	Audit Requirements . . . . .	13
3.6.5	Immunity Requirements . . . . .	13
3.7	Cultural and Political Requirements . . . . .	13
3.7.1	Cultural Requirements . . . . .	13
3.7.2	Political Requirements . . . . .	14
3.8	Legal Requirements . . . . .	14
3.8.1	Compliance Requirements . . . . .	14
3.8.2	Standards Requirements . . . . .	14
3.9	Health and Safety Requirements . . . . .	14
<b>4</b>	<b>Project Issues</b>	<b>14</b>
4.1	Open Issues . . . . .	14
4.2	Off-the-Shelf Solutions . . . . .	15
4.3	New Problems . . . . .	15
4.4	Tasks . . . . .	15
4.5	Migration to the New Product . . . . .	16
4.6	Risks . . . . .	16
4.7	Costs . . . . .	16
4.8	User Documentation and Training . . . . .	16

4.9	Waiting Room . . . . .	17
4.10	Ideas for Solutions . . . . .	17
<b>5</b>	<b>Appendix</b>	<b>18</b>
5.1	Symbolic Parameters . . . . .	18

## List of Tables

1	<b>Revision History</b> . . . . .	i
2	<b>Naming Conventions and Terminology</b> . . . . .	4
3	<b>Symbolic Constants</b> . . . . .	18

Table 1: **Revision History**

Date	Version	Notes
1/21/2020	1.0	Added to Repo
1/28/2020	1.1	Filled in name of project and team
2/08/2020	1.2	Initial draft of SRS
2/09/2020	1.2	Completed sections 3 and 4
4/05/2020	2.0	Changes made to address TA Comments for Revision 1

This document describes the requirements for google-images-downloader. The template for the Software Requirements Specification (SRS) is a subset of the ~~Volere template~~ [Volere template](#) (Robertson and Robertson, 2012).

# 1 Project Drivers

## 1.1 The Purpose of the Project

The purpose of this project is to re-create an open source software application following the software development life cycle, for the software engineering course 3XA3. This project will give the group exposure and experience in how to properly develop a software product from beginning to end.

The project chosen is a google images downloader command line tool that will allow end users to download a certain number of google images given specific keywords. ~~We want this product primarily to be able to help those involved in machine learning, and secondarily those involved in art~~ **This product will primarily help those involved in machine learning, and secondarily those involved in art .**

## 1.2 The Stakeholders

The following subsections will discuss various stakeholders of the product being developed.

### 1.2.1 The Client

**The clients of this project are** Dr. Asghar Bokhari, the professor of our course, and the TA responsible for marking all the deliverables, Andrew Lucentini.

### 1.2.2 The Customers

There are two main types of customers who would use the product, those involved in machine learning and those in art.

Individuals, or companies who are using machine learning algorithms or neural networks for image recognition purposes could benefit from this product as it would allow them to quickly obtain a large variety of images on a specific

keyword. This can help them make their models more accurate and robust.

Artists may use this tool for projects requiring many images based off a single keyword such as collages or reference work.

### **1.2.3 Other Stakeholders**

Other stakeholders include Google Images because their service is being used to scrape images off of it, **as well as the developers of this project.**

## **1.3 Mandated Constraints**

### **1.3.1 Schedule Constraints**

The project deliverables must be met on time according to the project description. The remaining ones are included below.

- Proof of Concept Demonstration - February 11, 2020
- Test Plan Revision 0 - February 28, 2020
- Design & Document Revision 0 - March 13, 2020
- Revision 0 Demonstration - March 17, 2020
- Final Demonstration (Revision 1) - March 31, 2020
- Final Documentation (Revision 1) - April 6, 2020

Further breakdown of these deliverables their respective group member assignments can be found in the [Gantt Chart](#).

### **1.3.2 Solution Constraints**

Description: The product shall be developed in Python 3.

Rationale: The existing implementation of the software is in Python 3, making it easier to re-develop if it is in the same language.

Fit criterion: The software is programmed in Python 3.

Description: The product shall get the image URLs from Google Images by scraping the HTML code returned.

Rationale: The existing implementation of the software designed its solution in this ~~was way. so since~~ **Since** we are re-developing it, that component of the solution will remain the same.

Fit criterion: The application works by scraping HTML code to get URLs

### **1.3.3 Budget Constraints**

The budget for this project is \$0 so the software must be able to built without spending any money.

### **1.3.4 Off-the-Shelf Software**

Since the purpose of the project is to re-implement an existing open source project, the existing version of the software will be referenced heavily when developing this version. The existing software can be found [here](#).

In addition to some common python libraries, urllib, urllib2, & httpplib will be used to manage and navigate URLs in the application.

### **1.3.5 Partner or Collaborative Applications**

The software product being developed will use the Google Images application to download images onto the local computer. The application will use the image URL links that are embedded in the HTML code for the image returned in a Google Images search query. These URLs are stored in 'data-iurl' under the 'img' tag.

### **1.3.6 Implementation Environment of the Current System**

The software product being developed will be installed on users local machines, or on servers, to function as a command line tool.

## **1.4 Naming Conventions and Terminology**

## **1.5 Relevant Facts and Assumptions**

It is assumed that users of the application have basic computer skills and are able to type commands on a command line given examples. It is also being

Table 2: Naming Conventions and Terminology

Name/Term	Definition
The System	The product defined in this document
HTML	Hypertext Markup Language
URL	The address of a website or file on the internet
Python 3	The third version of the python programming language
Keyword	The search term used to find images
Limit	The number of images the user wished to download
Google Images	The image search engine provided by Google
Search Constraints	restrictions images must follow such as size or file type
JPEG	A format for image files
config file	A file that contains configurations for the system
API	application programming interface
GUI	Graphical User Interface
git	A version control software used for co-working on code
GitLab	A product which stores projects using git

assumed that users have basic knowledge of ~~pictures~~ **picture related terms**, like what **such as** aspect ratio ~~means~~, or colour type ~~refers to~~.

## 2 Functional Requirements

### 2.1 The Scope of the Work and the Product

#### 2.1.1 The Context of the Work

In order for the product to function as intended, the system must interact with Google Images. As a result there is a need to understand how Google Images works. This is because the original implementation of this product scrapes the HTML off of Google Images and then finds the links to the original images in the scraped HTML. Therefore, the required background knowledge for the development of this product is how images are stored on the website so they can be effectively found in the scraped HTML.

#### 2.1.2 Work Partitioning

BE1: The User Requests an Image.



Inputs:

- Keyword
- Limit
- Search Constraints

Outputs:

- Images downloaded to computer

Summary:

- Download a number of images equal to the limit of the given keyword that also match any other given search constraints.

### **2.1.3 Individual Product Use Cases**

The main use case of this product is the user actor requesting an image. This use case can include the user specifying optional input arguments such as a limit of images, or other search constraints such as prefixes for searching or the size of images wanted.

The following scenarios will outline a general interaction between a user actor and the system for both the main use case, and sub use cases.

Scenario 1: The user requests an image without specifying a keyword. The system will return a message with the proper format for an image request and remind the user that a keyword is required.

Scenario 2: The user requests an image with a specified keyword. The system will retrieve images from Google Images that match the specified keyword. Since a limit on images was not specified the system will download the 100 images matching the search criteria. Since an output directory was not specified, the system will download the images into a downloads directory located in the same directory where the program was executed.

Scenario 3: The user requests a limit of 50 images with the keyword "Cat". The system will retrieve the first 50 images off of Google Images when Cat is used as the search term. Since an output directory was not specified, the system will download the images into a downloads directory located in the same directory where the program was executed.

- Scenario 4: The user requests a limit of 50 images with the keyword "Cat". The user specifies a download directory of "./Images/Cat/". The system will retrieve the first 50 images off of Google Images when Cat is used as the search term. Since an output directory was specified, the system will create a new directory named ~~images~~ **Images** then inside images create a new directory called ~~cats~~ **Cat**. The images will then be downloaded into the new cats directory. If the images and cats directory already exists, the system will use the existing directory rather than creating a new one.
- Scenario 5: The user requests a limit of 100 images with a specified keyword and download directory. The user also specifies specific search constraints. The system will search with the keyword and download 100 images that match the keyword and don't violate any of the search constraints. For example if the search constraint is a file type of JPEG, the system will only download JPEG images. The images will be downloaded into the specified directory, creating it if it does not already exist.

## 2.2 Functional Requirements

FR1: ~~The user shall be able to download images that match a given keyword~~  
**The system shall be able to download a set of images that match a user given keyword or phrase.**

Rationale: This is the main functionality of the product. With this, users will be able to download multiple images that all match the keyword given with just one command.

FR2: The user shall be able to specify the number of images they want **to download**.

Rationale: This allows the user to select how many images they want and will be a useful quality of life feature when the user knows approximately how many images they want.

FR3: The user shall be able to specify a directory to download the images into.

Rationale: This allows the user to have better control of the organization of the directories on their computers and will allow the user to have better control over how they use this product.

FR4: The user shall be able to specify the file type of images they desire.

Rationale: This will give the user greater freedom when using the system as they can have a set of images with a uniform file type which may help in certain situations.

FR5: The user shall be able to prevent the system from downloading images from specified websites.

Rationale: Allowing the user to blacklist websites will give the user greater control over the system and prevent images from inappropriate sources from being downloaded in professional settings.

FR6: The user shall be able to download images only from specified websites.

Rationale: Allowing the user to white-list websites will give the user greater control over the system and prevent images from inappropriate sources from being downloaded in professional settings.

FR7: ~~The user shall be able to add prefixes and suffixes to their image searches.~~ The user shall be able to specify the colour of Images they want to download.

Rationale: ~~Adding prefixes and suffixes will allow users to find the exact images they want as descriptive searches will narrow the search for the images they want. For example "Red Car" will only return red cars whereas "Car" would return images with cars of multiple different colours~~ Adding a colour search query will allow the user to filter their images even further by allowing them to only download images that contain a desired colour. This gives the user greater freedom and choice over the images that are downloaded for them.

FR8: The user shall be able to specify the size of images they want.

Rationale: Allowing the user to specify size would be useful to artists aiming to make collages which depend on the placement and size of images used.

FR9: The user shall be able to pass all search constraints, image limits and keywords through a config file.

Rationale: Using a config file would allow groups of users to use the same settings which would be a useful features for businesses or teams using the product.

FR10: The user shall be able to download images directly to a server address specified by the user.

Rationale: This will allow users who work on remoter servers or are hosting a training set on a server to reduce the amount of workrequired for the images to be sent to the server they desire

FR11: The system shall be able to download images of a specified aspect ratio given by the user.

Rationale: This will allow the user greater controll over the system and allow them to focus on only the images that satisfy their needs.

FR12: The user shall be able to specify an age of the image so that the system shall only download images that match the user specified age

Rationale: With an image age search constraint, users of the product will be able to get only the newest images or filter new images out so they have older images only. This will allow users of the product to have greater freedom over the images they want downloaded.

FR13: The user shall be able to specify a specific useage rights license that images downloaded by the system shall have.

Rationale: This feature will allow users to have greater freedom over the images they download. Additionally, specifying the usage rights the user wants for their images will be useful for artists looking for non copyrighted images.

FR14: The user should be able to specify if they want the system to do a safe search while downloading images.

Rationale: This will ensure that if the user desires images that are safe for work, the system will only download images that are safe for work

FR15: the user should be able to specify what region or country they would like images they request to have been uploaded in.

Rationale: This feature will allow users to only download images that were uploaded in a specified country allowing users who are out of their home country to still download images from their preferred geographical location.

## **3 Non-functional Requirements**

### **3.1 Look and Feel Requirements**

#### **3.1.1 Appearance Requirements**

AR1: Output messages will be formatted to fit the console.

Fit Criterion: 80% of users find that the output messages fit the console nicely.

#### **3.1.2 Style Requirements**

STR1: N/A

## **3.2 Usability and Humanity Requirements**

### **3.2.1 Ease of Use Requirements**

EUR1: The product must be easy to use.

Fit Criterion: 90% of users should successfully download images after reading the sample inputs and explanations for the first time.

### **3.2.2 Personalization and Internationalization Requirements**

PIR1: N/A

### **3.2.3 Learning Requirements**

LR1: ~~The product must be easy to learn and find information about.~~

Fit Criterion: ~~80% of users should be able to find the information they need within two minutes of searching.~~

LR2: The product must be easy to download for individuals familiar with using the command line.

Fit Criterion: 90% of users are able to have the product downloaded and functioning within five minutes.

### **3.2.4 Understandability and Politeness Requirements**

UPR1: N/A

### **3.2.5 Accessibility Requirements**

ASR1: N/A

## **3.3 Performance Requirements**

### **3.3.1 Speed and Latency Requirements**

SLR1: N/A

### **3.3.2 Safety-Critical Requirements**

SCR1: N/A

### **3.3.3 Precision or Accuracy Requirements**

PAR1: N/A

### **3.3.4 Reliability and Availability Requirements**

RAR1: N/A

### **3.3.5 Robustness Requirements**

RR1: The product must provide a helpful error message after improper input parameters.

Fit Criterion: 75% of users are able to identity their mistake from the error message returned.

### **3.3.6 Capacity Requirements**

CR1: N/A

### **3.3.7 Scalability or Extensibility Requirements**

SER1: N/A

### **3.3.8 Longevity Requirements**

LOR1: N/A

## **3.4 Operational and Environmental Requirements**

### **3.4.1 Expected Physical Environment**

EVE1: N/A

### **3.4.2 Requirements for Interacting with Adjacent Systems**

IAR1: The product must work on the most recent version of the google images HTML return code.

Fit Criterion: The application downloads all the images that use the newest format.

### **3.4.3 Productization Requirements**

PDR1: The product must be available to download on a public GitLab repository.

Fit Criterion: The git repository is public and includes download instructions.

### **3.4.4 Release Requirements**

RER1: N/A

## **3.5 Maintainability and Support Requirements**

### **3.5.1 Maintenance Requirements**

MSR1: The application will be maintained by the developers and the open source community.

Fit Criterion: The git repository is public.

MSR2: The application will require approved pull requests to be modified.

Fit Criterion: The application guidelines for the modification of the system clearly state that pull requests are required.

### **3.5.2 Supportability Requirements**

SUR1: N/A



### **3.5.3 Adaptability Requirements**

ADR1: The product must run under Linux, Windows, and Mac platforms.

Fit Criterion: All test cases run successfully on each platform.

## **3.6 Security Requirements**

### **3.6.1 Access Requirements**

ACR1: The application will be open source for anyone to access.

Fit Criterion: The git repository is public.

### **3.6.2 Integrity Requirements**

IR1: N/A

### **3.6.3 Privacy Requirements**

PRR1: N/A

### **3.6.4 Audit Requirements**

ADR1: N/A

### **3.6.5 Immunity Requirements**

IMR1: N/A

## **3.7 Cultural and Political Requirements**

### **3.7.1 Cultural Requirements**

CR1: The naming conventions of the product does not offend any groups or cultures.

Fit Criterion: Users questioned in anonymous surveys do not find the product offensive.

### 3.7.2 Political Requirements

PR1: N/A

## 3.8 Legal Requirements

### 3.8.1 Compliance Requirements

CPR1: The application must state explicitly in a disclaimer in the README.md that the application does not take responsibility for any copyright violations by the users.

Fit Criterion: The disclaimer is in the README.md.

### 3.8.2 Standards Requirements

SR1: The application must be developed using Google's python coding style and either flake8 or pylint as a linter.

Fit Criterion: The code matches the guidelines set out in the Google coding style and the linter returns no errors.

## 3.9 Health and Safety Requirements

~~There are no health and safety requirements that need to be considered for this project.~~

HSR1: The application will allow the filtering of not safe for work images

## 4 Project Issues

### 4.1 Open Issues

As the original implementation of this product relies on scraping the HTML of Google Images to download the images, there is uncertainty if this implementation is sustainable. This is because if Google Images changes how the images are stored in the HTML, a new version must be pushed for the product to retain functionality. As Google Images has recently changed how the

images are stored in the HTML, the current implementation of this project is not functional. As a result this is an issue that must be kept in mind throughout the development process.

## 4.2 Off-the-Shelf Solutions

Since the purpose of the project is to re-implement an open source project, the existing version of the software will be reference heavily when developing this version. The existing software can be found [here](#).

Additionally, The implementation of this project could use an existing API or Python Library such as urllib, urllib2, httplib or SerpApi to query and scrape Google Images.

## 4.3 New Problems

A potential problem that could arise due to this product is the unauthorized use of copyrighted materials. As the product only gets images off of Google Images, there is the possibility that a user could use a copyrighted image in a way that goes against the copyright holders wishes. This could cause undesired legal issues, and should be kept in mind when a licence is chosen for the product.

## 4.4 Tasks

The tasks for this project are the project deliverables that must be completed by their respective due dates. Detailed breakdowns of timing and due dates for the project deliverables can be found on the Gantt Chart found [here](#).

For a quick summary, the upcoming deliverables and their due dates are as follows:

- Proof of Concept Demonstration - February 11, 2020
- Test Plan Revision 0 - February 28, 2020
- Design & Document Revision 0 - March 13, 2020
- Revision 0 Demonstration - March 17, 2020

- Final Demonstration (Revision 1) - March 31, 2020
- Final Documentation (Revision 1) - April 6, 2020

## **4.5 Migration to the New Product**

In order for users to migrate to the new product, they must first uninstall the previous version of the product. This is to ensure that no errors occur during the install process. Once the previous version has been uninstalled, the users can then install the new product and use it as intended.

## **4.6 Risks**

A big risk for this project is the reliance on a third party platform. This is because due to the current implementation of the original product if Google Images changes how the images are stored in the HTML, the entire product loses functionality and a new version must be created and pushed before the product can be used as intended. This is a high priority risk as it has a high probability of becoming a problem during the life of the product.

## **4.7 Costs**

For this project, there are two types of cost, monetary cost and cost of effort. As one of the budget constraints for this product is that we do not spend any money to produce the product, the monetary cost for this project is \$0. For the cost of effort, based on the low number of business events and functional requirements, the cost of effort is expected to be low. This could however change as requirements are refined and more work is required to make a functional product.

## **4.8 User Documentation and Training**

The product will have the following documentation provided to the user to help them learn the software and guide them as they use the product.

- Installation Guide to setup the product
- Sample Inputs with explanations of each input argument

## 4.9 Waiting Room

The creation of the GUI and functional requirements surrounding the GUI have been put on hold due to time constraints surrounding the release of this project.

Ideally these features will be implemented in a future release. A GUI would allow users that are less comfortable with using the command line still have access to the product and all its features.

## 4.10 Ideas for Solutions

As the purpose of this project is to re-implement an open source project, our solutions will be inspired by the original material. One such solution provided by the original material is scraping the HTML from google images and using the image links found through the HTML to download the images. However, due to the changes made to how images are stored on Google Images, this solution would have to be updated to accommodate for this issue.

## References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

## 5 Appendix

This section has been added to the Volere template.

### 5.1 Symbolic Parameters

Table 3: **Symbolic Constants**

<b>Constant</b>	<b>Definition</b>
FR	Functional Requirements
AR	Appearance Requirements
EUR	Ease of Use Requirements
LR	Learning Requirements
RR	Robustness Requirements
IAR	Requirements for Interacting with Adjacent Systems
PDR	Productization Requirements
MSR	Maintainability and Support Requirements
ADR	Adaptability Requirements
ACR	Access Requirements
CR	Cultural Requirements
CPR	Compliance Requirements
SR	Standards Requirements