# High Level Design (HLD)

## Advanced Image extractor/Downloader

Revision Number: 1.0
Last date of revision: 10/11/2021

# Document Version Control

| Date Issued | Version | Description | Author |
|---|---|---|---|
| 09/11/2021 | 1.0 | HLD-Version 1.0 | Naveen Pujar |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Contents

# Abstract

Image scrapping is automated gathering of images from the Internet that involves some amount of data parsing in order to obtain only the required information. This project will be helpful for the users who want to download a collection of images of a particular category, as Image scrapping techniques are extensively used in the industry today for collecting a huge number of images that are used as inputs for training the object detection/classification/identification models. The images are being downloaded using Python and web scraping techniques.

# 1 Introduction

## 1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
  - Security
  - Reliability
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Resource utilization
  - Serviceability

## 1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical and mildly-technical terms which should be understandable to the administrators of the system.

## 1.3 Definitions

| *Term* | *Description* |
|---|---|
| *DB/db* | Database, an organized collection of structured information, or data, typically stored electronically in a computer system.The data can then be easily accessed, managed, modified, updated, controlled, and organized. |
| Job | Advance Image Downloader |
| API | API stands for Application Programming Interface. It allows two applications to communicate with one another to access data. |

# 2 General Description

## 2.1 Product Perspective

Advance Image Downloader/Extractor is a python-based project which will help to extract desired quantity of images from web and send it to a client's email ID on a scheduled time.

## 2.2 Problem statement

To train a model, you need images. You can most certainly download them manually by clicking on each image or possibly even somewhere in batch, but there are ways to download them as much we want in one go. Let's use Python and some web scraping techniques to download images in this project.

The problem statement here is to develop a web application to download thousands of images from the internet for the below given requirements (eg: Cat, Dog) using Python and web scraping techniques.

- ➢ User should able to specify the search string of any kind of image that they want to bulk download.
- ➢ User can choose to get an email with the downloadable link at any point of a time as per their wish.
- ➢ User can able to download the required amount of images in one click.

## 2.3 Proposed Solution

The solution proposed here is an Image Extractor job which can be implemented to perform the tasks given in the section 2.2 (Problem statement). Firstly, the user specifies the inputs through the front end and the back end begins to scrap images in bulk from web using selenium. Scraped images which are stored in a folder are zipped and uploaded to the S3 bucket. The downloadable link is displayed back over to the front end as well as an email will be sent to the user’s email address with the downloadable link at the time which they wished to get the email.

## 2.4 Further Improvements

Advanced image extractor/downloader can be improved to accept few more keywords from the requestor (eg: white cat, brown dog) to get little more specific information regarding the kind of images that they are interested in downloading.

## 2.5 Technical Requirements

Web scraping is a technique using which the webpages from the internet are fetched and parsed to understand and extract specific information similar to a human being. Web scrapping consists of two parts:

- HTML Parsing: Parsing the HTML content of the webpages obtained through web crawling and then extracting specific information from it.
- Request: Requests allow you to send HTTP requests extremely easily. This module also does not come built-in with Python. To install this type the below command in the terminal.
- Urllib: It is a python module that allows you to access, and interacts with, websites with their URL.

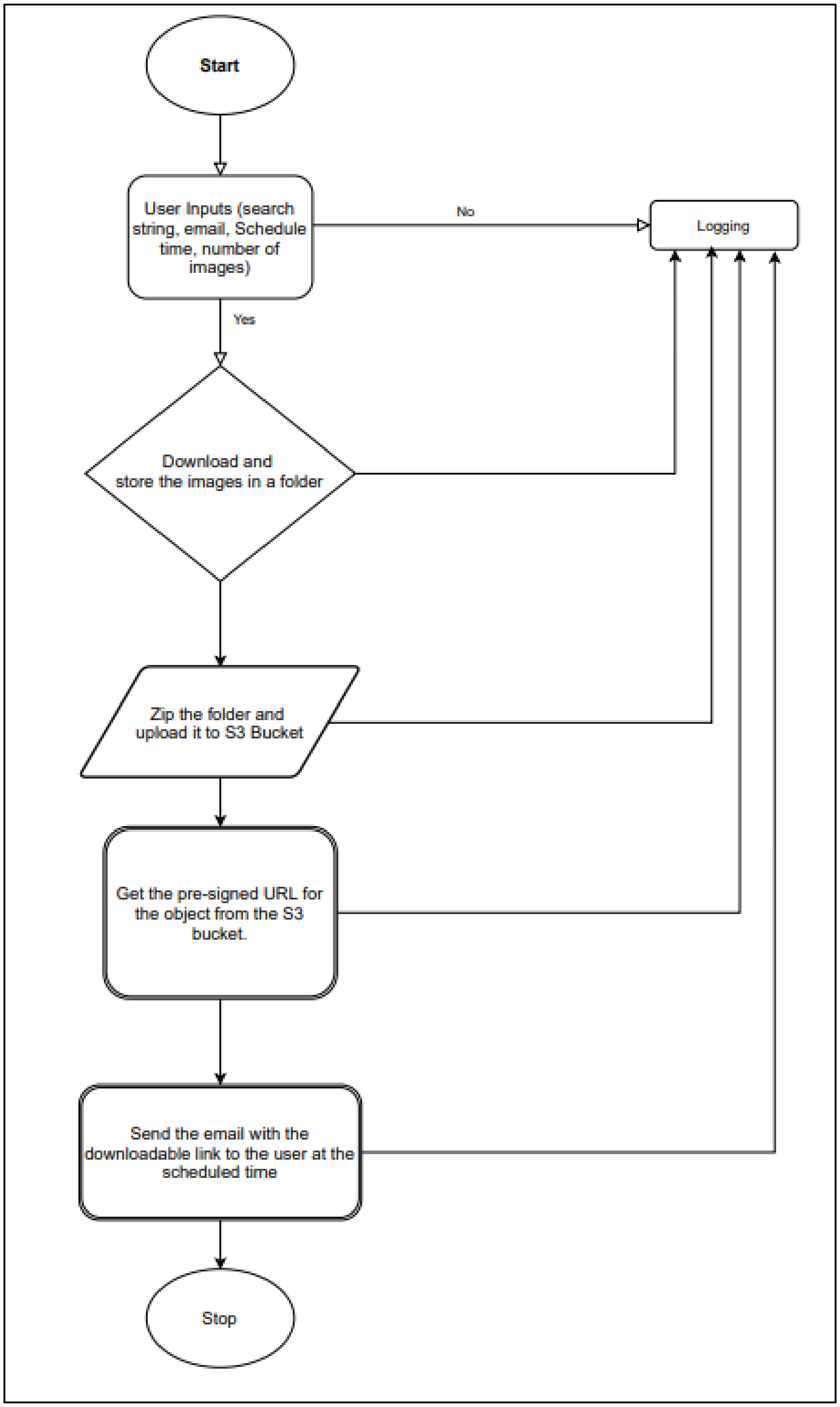
## 2.6 Tools used

Python programming language and frameworks such as flask, Selenium, requests and Urllib. Cassandra database and AWS for deployment are used to build the whole application.

- VS Code is used as IDE.
- AWS is used for deployment of the model.
- Cassandra is used to save inputs from requestor and logging data.
- Front end development is done using HTML/CSS
- Python Flask is used for backend development.
- GitHub is used as version control system.
- AWS Lambda is used to trigger email at the scheduled time.
- S3 to store the images
- Selenium to scrape the image

# 3 Design Details

## 3.1 Process Flow


Start
User Inputs (search string, email, Schedule time, number of images)
No
Logging
Yes
Download and store the images in a folder
Zip the folder and upload it to S3 Bucket
Get the pre-signed URL for the object from the S3 bucket.
Send the email with the downloadable link to the user at the scheduled time
Stop

## 3.2 Event log

The system should log every event so that the user will know what process is running internally.

**Initial Step-By-Step Description:**

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

## 3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

# 4 Performance

The web scrapping application which automatically sends requests to web and then extracts specific information from it. This project comes in handy for any object detection/classification model. Proper implantation of code is required in order to enhance the performance of the solution. Try-catch method has been included to catch any incorrect key words inserted by the user.

## 4.1 Reusability

The code written and the components used should have the ability to be reused with no problems. Should time allow, and detailed instructions are written on how to create this project, everything will be completely reusable to anyone.

## 4.2 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

## 4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

## 4.4 Deployment

The Python app that we have developed is residing on our local machine. But to make it available to end-users, we need to deploy it to either an on premise server or to a cloud service. Below deployment platform can be used for making the UI accessible to end users.

## 4.5 KPls (Key Performance Indicators)

1. Ability to download required number of images as per requestors wish.
2. How fast the images are going to download without system crashing.
3. The requestor should able get the URL at the scheduled time.

# 5 Conclusion

The designed project will provide the user with thousands of images via an URL which he/she will be receiving at the scheduled time.