**Advance Image Downloader/Extractor**

**Revision Number: 1.0**

**Last date of revision: 02/11/2021**

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 29-08-2021 | 1 | Initial HLD-v1.0 | Nitesh Kumar |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

**Document Version Control…………………………………………………………………2**

**Abstract………………………………………………………………………………………...4**

1. **Introduction……………………………………………………………………..…….5**
   1. **Why this High-Level Design Document?..................................................5**
   2. **Scope……………………………………………………………………………...5**
   3. **Definitions………………………………………………………………………...6**
2. **General Description………………………………………………………………….6**
   1. **Product Perspective…………………………………………………………….6**
   2. **Problem Statement……………………………………………………………....6**
   3. **Proposed Solution……………………………………………………………….7**
   4. **Further Improvements…………………………………………………………..7**
   5. **Data Requirements………………………………………………………………7**
   6. **Constraints……………………………………………………………………......8**
   7. **Assumptions……………………………………………………………………...8**
3. **Design Details………………………………………………………………………….9**
   1. **Main Design Features…………………………………………………………..9**
   2. **Process Flows……………………………………………………………………9**
   3. **Database Design………………………………………………………………..10**
   4. **User Interface……………………………………………………………………10**
   5. **Event log…………………………………………………………………………11**
   6. **Error Handling…………………………………………………………………..11**
   7. **Performance……………………………………………………………………..12**
   8. **Reusability……………………………………………………………………….12**
   9. **Portability………………………………………………………………………...12**
   10. **Security………………………………………………………………………...12**
   11. **Resource Utilization…………………………………………………………13**
   12. **Deployment……………………………………………………………………13**
4. **Conclusion……………………………………………………………………………13**

**Abstract**

Even Images are also very important data source these day in the field of computer vision model training, Finding images for web site building and many more uses of daily life style. This app mainly use to download images from google and store it into database, This app can download images upto 100 of any kind as per user instruction, once user submit the required information ,Download link will be available on web page as well as in email .

1. **Introduction**
   1. **Why this High-Level Design Document?**

The motive of this excessive-level layout (HLD) file is to add the vital element to the current mission description to represent a appropriate version for coding. This report is also meant to assist discover contradictions previous to coding, and may be used as a reference manual for a way the modules engage at a high degree.

**The HLD will:**

* present all of the design factors and outline them in detail
* Describe the user interface being implemented
* Describe the Software interfaces
* Include design features and the architecture of the project
* List and describe the non-functional attributes like:
* Security
* Reliability
* Portability
* Reusability
* Resource Utilization
  1. **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 to mildly-technical term which should be understandable to the administrators of the system.

* 1. **Definitions**

|  |  |
| --- | --- |
| Term | Description |
| Database | Collection of all the information monitored by the system |
| IDE | Integrated Development Environment |
| Heroku | Heroku Cloud Service |

1. **General Description**
   1. **Product Perspective**

This project help us to download images from google using python tools It is completely python web based app, In which It is mainly scraping the images as per user request Once the job done ,user are notified in mail and get download links and This web app allow multiple user to submit their request simultaneously

* 1. **Problem Statement**

There are many times, we need a lot of pictures to work with. We can think of an example such as training Neural networks on house and furnitures images or going through hundreds of website design stuff. In those cases we need hundreds of photos immediately. This problem can be solved using the Advance Image Downloader application. The following scenarios can be used:

* To download the specified number of images of the particular query
* To notify with the downloadable link to the user over an email
* To let user, download the images through just single click on link
* To submit multiple job request by the single/multiple users

**2.3 Proposed Solution**

The proposed solution here is the web-based Advance Image Downloader / Extractor python web-based application that can be used to create application cases mentioned above. In the first case, the user must enter a search query, the time the user wants to work, the email and the number of photos the user wants. This operation will be performed on a date and time specified by the user. Once the task is completed, the user will receive a verification link that can be used to download the images.

**2.4 Further Improvements**

* Letting user input the Image type they want in result. E.g., JPEG, PNG etc.
* We can show the expected time required to finish the job.
* Size of the downloaded files can be sent over an email.
* To give the country input so that user can schedule the job at their

particular country time.

**2.5 Data Requirements**

* We will need user’s Email, Date and time, and search query for performing this job.

**2.6 Tools Used**

* Flask Framework is used as backend development
* UI is developed using HTML, CSS and JavaScript
* PyCharm is used as an IDE for coding
* Heroku is used for deployment of the model
* Cassandra is used as database to retrieve, insert, delete and update the databases
* Git is used as version control system







**2.7 Constraints**

It should have a design that is easy to use and functional. The user should not have to know how to download images in the background. The Internet connection is also a barrier to the application. As the app downloads data from the website and the internet, it is important to have an internet connection for the system to work. Since a user can make multiple requests at once, he may be forced to file offline requests and increase the time it takes to provide feedback.

**2.8 Assumptions**

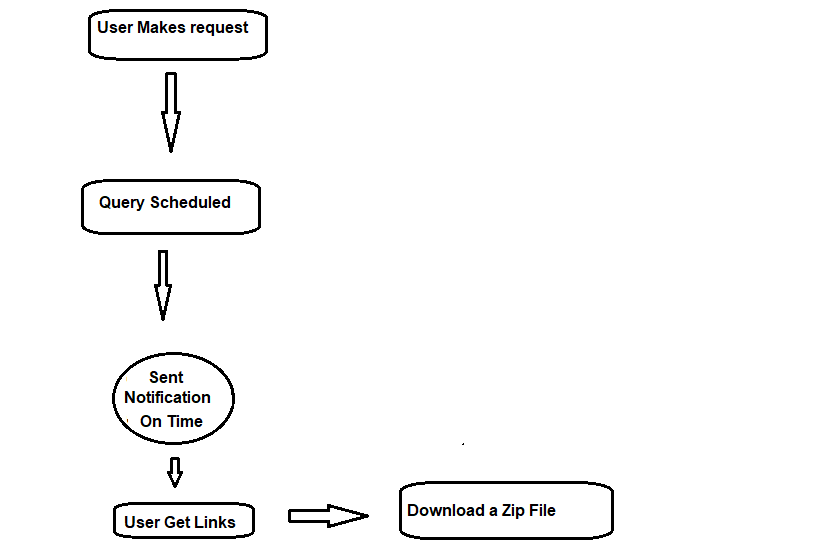
Assumption is that, the user must have enough space in its hardware where the downloaded images can be stored. If the space inside the hardware is not sufficient enough then the application may not work as intended.

1. **Design Details**
   1. **Main Design Features**

Key design features include components such as Architecture, User Interface Design, Database, Process Relationships. To make these designs easier to understand, the design is attached with drawings such as Flow Diagram.

* 1. **Process Flow**

Advance Image Downloader app follows the following architecture:

****

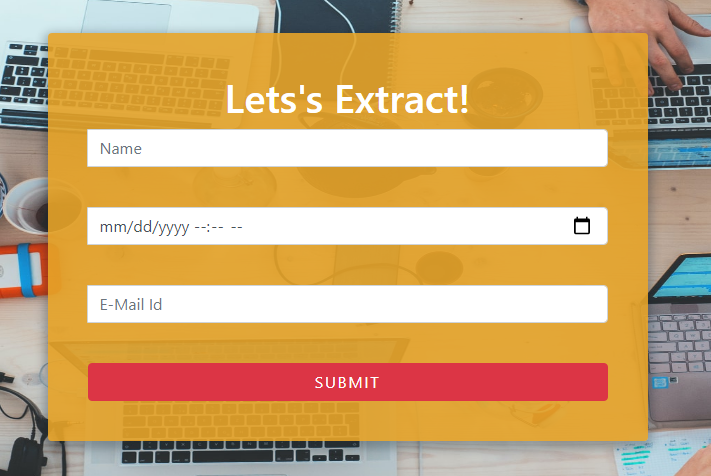
* 1. **Database Design**

The Database Schema Consists of req\_id(uuid datatype),email(String datatype),url(string datatype)

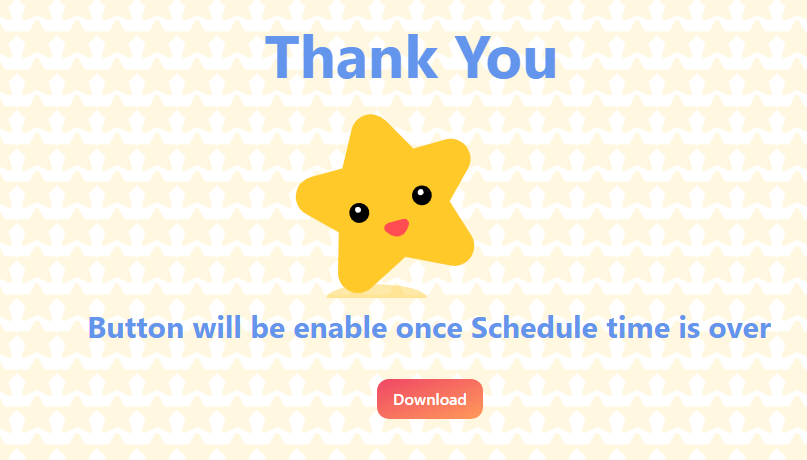
|  |
| --- |
| AstraImage |
| req\_id:uuid |
| email:string |
| url:string |

* 1. **User Interface**

1. **Home Page**

****

1. **Data Submitted**

****

* 1. **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 logging file as well.
4. System must be able to handle logging at greater scale because it helps debugging the issue and hence it is mandatory to do.
   1. **Error Handling**

Error handling is done in two ways:

1. **UI part** - If user performs some incorrect action on UI, then the error page will be shown to the user which will have the appropriate error message.
2. **Email Part** - If the error comes while handling the user request at the backend, then user will receive an email regarding the same and repeating the job submission again.
   1. **Performance**

For everything to run smoothly the app must be able to handle multiple requests at similar time to give the best user experience as possible. It should also be able to give the results back in the earliest time possible without making user to wait for result longer amount of time. User should get appropriate result from the app about what he/she is expecting to improve the performance of search in the app itself. The database server must be able to need to keep up with the database requests without any failure and data losses.

* 1. **Reusability**

The coded code and module should have the ability to be reused without problems.

* 1. **Portability**

Since this is a web app, it can be accessed via any OS system until and unless that OS is connected to the internet.

* 1. **Security**

Since we are capturing the user’s email address, we have added the functionality by which the user request which consist of user’s email, search query etc. will get deleted from the database after certain interval of time (usually after 30 minutes).

* 1. **Resource Utilization**

For each user request, cloud will use multiple threads to simulate the multithreading environment until that process is finished. Databases has to perform retrieving, inserting and deletion operation until that process gets removed from the database.

* 1. **Deployment**

Posting of this web application will be done on the Heroku cloud.

Deployment steps are as follows:

1. **Conclusion**

The Advance Image Downloader/Extractor (Job) will help the user download the hundreds of images in single click. To simply the process where at time we require bunch of images to work with.