MicroFilters Requirements

|  |  |
| --- | --- |
| REQ No.  2 | Project  MicroFilters 2.0 |

Abstract

This document covers requirements for MicroFilters Phase 2 that handles image extract and push.

| **Rev** | **Date** | **Author** | **Status** |
| --- | --- | --- | --- |
| 1 |  | Ji Lucas | First draft |
| 2 |  |  |  |
|  |  |  |  |

Distribution List

|  |  |
| --- | --- |
| **Reviewers** | **FYI** |
| Ji Lucas |  |

Table of Contents

1. Introduction 4

2. Related Documents/Links/People 4

3. Glossary 4

4. Enhancement Scope 5

4.1. Goals and Objectives 5

4.2. In Scope 5

4.3. Out of Scope 5

4.4. Assumptions 6

4.5. Constraints 6

4.6. Dependencies 6

4.7. Risks 6

5. Requirements 6

5.1. Overview 7

5.2. File Import Requirements 7

5.2.1. General Requirements 7

5.2.2. File Details 11

5.2.3. Sample Data 11

5.3. UI/UX functional 13

5.3.1. UI prototype 13

5.3.2. Color schema, outline info. 14

5.4. Non-functional 14

5.4.1. Security 14

5.4.2. QA/Testing 14

Issues/Questions 15

6. Revision History 15

# Introduction

# Related Documents/Links/People

References in the text throughout this document appear in square brackets (e.g., [1], [JS]).

| **Reference** | **Document/Link/Person/Application** |
| --- | --- |
|  | MicroMappers http://clickers.micromappers.org/ |
|  | AIDR http://aidr-dev.qcri.org/AIDRFetchManager/ |
|  | Digital Humanitarian volunteers coordinator |
|  | Image Clicker http://clickers.micromappers.org/app/MM\_ImageClicker/ |
|  | MicroFilters https://github.com/MicroMappers/MicroFilters |
|  |  |
|  |  |
|  |  |

# Glossary

| **Term** | **Definition** |
| --- | --- |
| AIDR | Artificial Intelligence for Disaster Response : <http://aidr-dev.qcri.org/AIDRFetchManager/>  To use the application, user has to have twitter account |
| MicroMappers | A cloned customized of Pybossa platform for tagging by Digital Humanitarians http://clickers.micromappers.org/ |
| MicroFilters | A web application that will generate data for MicroMappers.  URL WILL BE : <http://microfilters.micromappers.org/>  I will contact our network admin to configure DNS. |
| AIDR Collection Data | Twitter data that is collected by AIDR based on configuration. AIDR Collection output |
| Image Clicker | App that displays image only. Then, user selects one of options |
| GDELT Feed | GDELT generates image clicker source file every 15 minutes. The below link has 3 reference file locations.  [**http://data.gdeltproject.org/micromappers/lastupdate.txt**](http://data.gdeltproject.org/micromappers/lastupdate.txt)  For image clicker, the system should get “\*.mmic.txt”. The “\*.mmic.txt” will contains header. See the below.  **User-Name,Tweet,Time-stamp,Location,Latitude,Longitude,Image-Link,TweetID** |
| VM2 | MicroFilters hosting environment |
|  |  |
|  |  |
|  |  |

# Scope

## Goals and Objectives

1. Build a web based application that handles incoming data from AIDR.
2. Build a web based application that handles incoming data from GDELT feed.
3. Build a web based application that will generate data for Image Clicker feed.

## In Scope

1. Extract image link from AIDR tweet Data.
2. Extract image link from GDELT feed.
3. Save image link into database.
4. Save image size into database.
5. Save image md5 into database.
6. Remove duplicate image links.
7. Compare MD5 signature
   1. Issue a GET one with range (0-100000), i.e. up to 100K per image.
   2. Compute the MD5 of what was downloaded
   3. check MD5 already exists or not again stored data.
   4. store the MD5 if no same data exists.
8. Any duplicate or same MD5 link shouldn’t be saved.
9. Once data is exported to MicroMappers API, the system should mark exported images as exported flag.
10. Exported data should post to MicroMappers API.
11. Process should be triggered via restful api.
12. Proper logging
13. Running on tomcat
14. System failure notification to admin.
15. System failure restart job.

## Out of Scope

## Assumptions

1. Assumed that AIDR & AIDR-TRAINER-API is running without any interruption.
2. AIDR data output will have standardized format.
3. GDELT will have standardized format.
4. AIDR will generate output.
5. File export location should be reachable from configuration

## Constraints

1. MicroFilters should generate max.1500 records per file. If more than 1500 records exists, MicroFilters should generate separate files.
2. Accepted image extensions are .jpg, .png, .jpeg.

## Dependencies

1. AIDR
2. GDELT
3. MM-API

## Risks

1. Limited Bandwidth

# Requirements

The following subsections define software requirements. Each requirement is labeled as follows:

* **ID.** The unique identifier for the requirement.
* **Requirement.** A clear and concise description of the requirement.
* **Source.** A cross-reference to the source of the requirement.
* **Cat**egory**.** A classification for the requirement using the selections from Table 1.

Table - Requirement Categories

| **Category** | **Description** |
| --- | --- |
| **M**andatory | Required minimum functionality. |
| **O**ptional | Nice to have and will be implemented if time permits. |
| **F**uture | Will not be implemented now but should be considered for a future enhancement. |
| **D**ropped | Requirement determined to be completely out of scope **after** baseline. |
| **E**xisting | Functionality already present in the software (for documentation purposes). |
| **Op**erational | Will not be addressed in the software (documented for training purposes). This requirement should have a related item in the issue list in order to identify an owner. |

## Overview

## Requirements

### General Requirements

| **ID** | **Requirement** | **Source** | **Cat** |
| --- | --- | --- | --- |
| .1 | System should have restful api to start process |  | M |
| .2 | System should have restful api to stop process |  | M |
| .3 | System should have restful api to one time process |  | M |
| .4 | System should post export file location to mm-api |  | M |
| .5 | System should use libraries RQ and RQScheduler or similar library, e.g Spring batch to allow slow or computationally-heavy tasks to be run in the background in an asynchronous way |  | M |
|  | System should read AIDR output |  | M |
|  | AIDR output will have hyper link per row or record. The system should crawl image link that is located in tweetpic or instagram only. |  |  |
|  | System should read GDELT Image Clicker file |  | M |
|  | AIDR OUTPUT should be reachable via api based on parameters. |  | M |
|  | GDELT Image Clicker file will be generated every 15min.  The below link has 3 reference file locations.  [**http://data.gdeltproject.org/micromappers/lastupdate.txt**](http://data.gdeltproject.org/micromappers/lastupdate.txt)  For image clicker, the system should get “\*.mmic.txt”. The “\*.mmic.txt” will contains header. See the below.  **User-Name,Tweet,Time-stamp,Location,Latitude,Longitude,Image-Link,TweetID** |  | M |
|  | Once system starts, the process should run until a stop request invoked |  | M |
|  |  |  |  |
| .6 | System should remove duplicate image URL |  | M |
| .7 | System should compare md5 signature.   * 1. Issue a GET one with range (0-100000), i.e. up to 100K per image.   2. Compute the MD5 of what was downloaded   3. check MD5 already exists or not again stored data.   4. store the MD5 if no same data exists. |  | M |
| .8 | System should save all incoming image URL |  | M |
| .9 | System should save all incoming image SIZE |  | M |
| .10 | System should save all incoming image MD5 |  | M |
| .11 | Exported images should be flagged |  | M |
| .12 | System should generate a file(s) with a header  **See the below:**  "User-Name","Tweet","Time-stamp","Location","Latitude","Longitude","Image-link","TweetID" |  | M |
| .13 | A file should have max.1500 records. |  | M |
| .14 | No empty file should be generated. |  | M |
| .15 | File name should be unique. |  | M |
| .16 | System will have its own database |  | M |
| .17 | Once a file is generated, system post info to mm-api  url : http://api.mm.clickers.org/MMAPI/rest/source/save  json format :  [  {  "fileURL": "http://PUBLIC-URL /mm201406150956.csv",  "appID": 73  }  ] |  | M |
| .18 | System failure should be monitored and notified |  | M |
| .19 | System should have proper logging |  | M |
| .20 | System should restart if fails. |  |  |
| .21 |  |  |  |
|  |  |  |  |
|  |  |  |  |

### File Details

|  |  |
| --- | --- |
| **Filename:** | {**APPNAME**}yyyymmddHHmmss.csv |
| **Zipped?** | No |
| **Zip filename:** | n/a |
| **Field Delimiter:** | Comma |
| **Record Delimiter:** | Carriage Return |
| **Character set:** | UTF-8 |
| **Import Frequency:** | On demand |
| **Header row?** | Yes |
| **Footer row?** | No |
| **Retrieval location:** | User’s local or URL |
| **Import database:** | NO |

Header:

**"User-Name","Tweet","Time-stamp","Location","Latitude","Longitude","Image-link","TweetID"**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Column Name** | **Type** | **Len** | **M/O** | **Comments/Data Mapping** |
| 1 | User-Name |  |  | O | Tweet user name/ tweet handler |
| 2 | Tweet |  |  | M | Tweet text |
| 3 | Time-stamp |  |  | M | created date/time |
| 4 | Location |  |  |  | Image url |
| 5 | Latitude |  |  | O | latitude |
| 6 | Longitude |  |  | O | longitude |
| 7 | Image-link |  |  | O | Image url |
| 8 | TweetID |  |  | M | TweetID |

### Sample Data

## UI/UX functional

## Non-functional

### Security

| **ID** | **Requirement** | **Source** | **Cat** |
| --- | --- | --- | --- |
| .1 |  |  |  |

### QA/Testing

| **ID** | **Use Case** | **Date/Status** |
| --- | --- | --- |
| .1 |  |  |

# Issues/Questions

| **Issue #** | **Issue/Resolution Description** | **Date/Status** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Revision History

Changes to the text of this document are indicated by bars in the outside margin adjacent to the affected text.

| **Date** | **Change Description** |
| --- | --- |
| June 09, 2015 | Initial draft. |
|  |  |
|  |  |
|  |  |