MicroMappers

|  |  |
| --- | --- |
| REQ No.  1 | Project  MicroMappers |

Abstract

MicroMappers is a core applications what contains….

| **Rev** | **Date** | **Author** | **Status** |
| --- | --- | --- | --- |
| 1 | 12/02/2015 |  | start |
| 2 |  |  |  |
| 3 |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

|  |  |
| --- | --- |
| **Reviewers** | **FYI** |
| Ji Lucas  Patrick  Heather |  |

Table of Contents

1. Introduction 4

2. Related Documents/Links/People 4

3. Glossary 4

4. Applications 4

4.1. High Level Design 4

4.2. Modules 5

4.2.1. Pybossa-Service 5

4.2.2. MM-Pybossa-Scheduler 5

4.2.3. MM-API 6

4.2.4. MM-GIS 6

4.2.5. MM-Image 6

4.2.6. GDELT-PLUG-IN 6

4.2.7. MicroFilters 6

4.2.8. Aidr-trainer-api 7

4.2.9. Aidr-trainer-pybossa 7

4.3. Clickers 7

4.3.1. Text Clicker 7

4.3.2. Image Clicker 8

4.3.3. 3W Clicker 8

4.3.4. Text Geo Clicker 8

4.3.5. Image Geo Clicker 8

4.3.6. Aerial Clicker 8

4.3.7. Satelitte Clicker 8

4.4. MicroMaps 8

5. Database Design 9

5.1. Database 9

5.1.1. Pybossa 9

5.1.2. MM-Scheduler Schema 10

5.2. Application Details 11

5.2.1. MM-Pybossa-Scheduler 11

5.2.2. MM-API 11

5.2.3. MM-Image 11

5.2.4. MM-GIS 11

5.2.5. GDELT PLUGIN 11

5.3. Servers 11

5.3.1. Production 11

5.3.2. QA 11

5.3.3. Development 11

5.3.4. Basic Configuration 11

5.4. Non-functional 12

5.4.1. Security 12

5.4.2. Testing 12

Issues/Questions 12

6. Revision History 12

MicroMappers

# Introduction

The purpose of MicroMappers is to provide humanitarian organizations (and other social good initiatives) with a platform that can help them make sense of “Big Data”. This includes making sense of text-based data, pictures, videos, aerial imagery and satellite imagery. MicroMappers is primarily powered by volunteer-based crowdsourcing (or microtasking). The vision for MicroMappers is to combine this human computing approach with machine computing (artificial intelligence) in order to speed up the analysis of Big Data. This has already been achieved for text-based information (e.g., tweets) but still needs to be developed for pictures, videos and imagery. Ultimately, MicroMappers should enable anyone to rapidly make sense of Big Data by deploying the individual MicroMappers Clickers while fusing and mapping the results in near-real time.

# Related Documents/Links/People

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

| **Reference** | **Document/Link/Person/Application** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Glossary

| **Term** | **Definition** |
| --- | --- |
| Clickers |  |
| MicroMappers - Pybossa |  |
| Aerial |  |
| Satellite |  |
|  |  |

# Applications

## High Level Design



## Modules

### Pybossa-Service

* Customized Pybossa(pybossa.com) application for corwdsourcing. It enables to create and run clickers where digital volunteers help us with text, media, image, geo location and more.
* Technology : python, postgrsql, redis
* Servers :
  + TAMU-Q(Production)
  + aws(Production in progress)
  + Azure(QA)

### MM-Pybossa-Scheduler

* Scheduled service module that publishes and imports tasks to/from Pybossa-Service. The module is a stand alone application that is running based on data
* Technology : Spring MVC, mySQL
* Servers :
  + TAMU-Q(Production)
  + aws(Production in progress)
  + Azure(QA)

### MM-API

* restful api module that is interacting with internal or external services.
* Technology : Spring MVC, mySQL
* Servers :
  + TAMU-Q(Production)
  + aws(Production in progress)
  + Azure(QA)

### MM-GIS

* restful api module that is interacting with MicroMaps.
* Technology : Spring boots, Spring websocket, mySQL
* Servers :
  + TAMU-Q(Production)
  + aws(Production in progress)
  + Azure(QA)

### MM-Image

* restful api module that will process image.
* Image meta data extract, image slicing, image conversion, image duplication detect, image pattern anlysis
* Technology : SpringBatch, Java Geotools, metadata-extractor, mbtiles, mySQL
* Servers :
  + TAMU-Q(Production)
  + aws(Production in progress)
  + Azure(QA)

### GDELT-PLUG-IN

* Module to consume gdelt posting for Image Clicker & 3W Clicker.
* Technology : Spring boot, mySQL
* Servers :
  + TAMU-Q(Production)
  + aws(Production in progress)
  + Azure(QA)

### MicroFilters

* Module to consume AIDR output to generate image & video clicker’s data source.
* Technology : python, rebbitMQ
* Servers :
  + TAMU-Q(Production)
  + aws(Production in progress)
  + Azure(QA)

### Aidr-trainer-api

* restful api module that is interacting with aidr only to create text based clicker. And, populate task data for tagging. Also, import data for classification.
* Its handles only AIDR specific data
* Technology : Spring MVC, MySQL, ejb
* Servers :
  + Azure VM1 – AIDR PRODUCTION(Production)
  + Azure VM2(QA)

### Aidr-trainer-pybossa

* Scheduled service module that creates apps, publishes and imports tasks to/from Pybossa-Service. The module is a stand alone application that is running based on data
* Its handles only AIDR specific data and handles only text clicker
* Technology : Spring MVC, MySQL,
* Servers :
  + Azure VM1 – AIDR PRODUCTION(Production)
  + Azure VM2(QA)

## Clickers



### Text Clicker

Classify text based data

### Image Clicker

Classify image based data

### Video Clicker

Classify video based data

### 3W Clicker

Classify news media data

### Text Geo Clicker

geotag text based data that is promoted from Text Clicker

### Image Geo Clicker

geotag Image based data that is promoted from Image Clicker

### Video Geo Clicker

geotag video based data that is promoted from video Clicker

### Aerial Clicker

Feature tracing on aerial imagery

### Satelitte Clicker

Feature tracing on satellite imagery

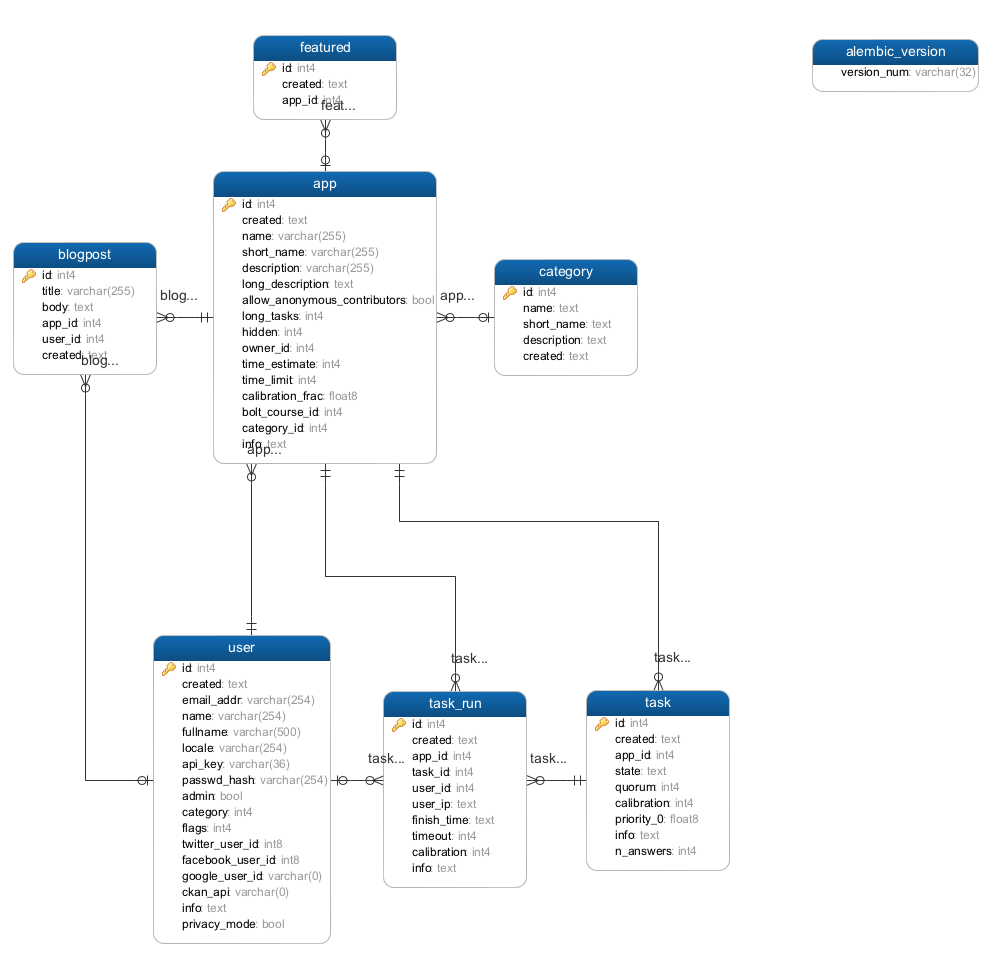
## MicroMaps

Centralized Map portal where clicker’s output will be displayed based on clicker type & crisis.

# Database Design

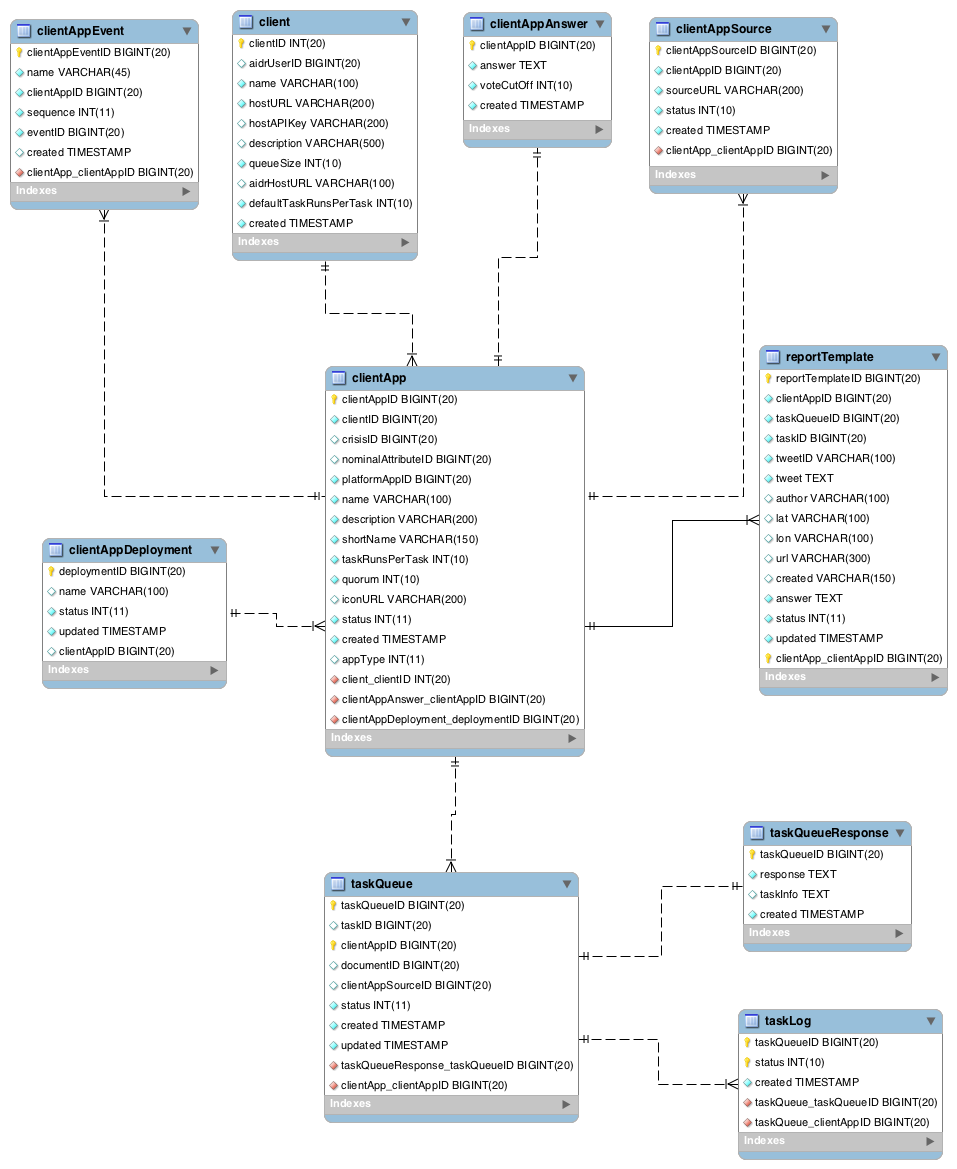
## Database

### Pybossa



|  |  |  |
| --- | --- | --- |
| **TABLE NAME** | **DESCRIPTION** | **NOTE** |
| user | User info/token |  |
| app | Clicker info |  |
| task | Task info |  |
| task\_run | User’s responses of each task |  |
| category | Category of apps |  |
| featured | Apps for front page |  |

### MM-Scheduler Schema



|  |  |  |
| --- | --- | --- |
| **TABLE NAME** | **DESCRIPTION** | **NOTE** |
| client | Aidr, micromappers high level configuration. It holds pybossa api configuration, default clientApp configuration, aidr server configuration |  |
| clientApp | holds each clicker configuration |  |
| taskQueue | holds all available & completed tasks |  |
| taskQueueResponse | holds completed task responses, user’s responses is stored |  |
| reportTemplate | holds task info with answer that is the source data to generate geo clicker’s data feed. During task import process, only filtered based on business rule will be stored in the table. |  |
| clientAppAnswer | Holds question/answers/promotion rule for tagging & for filter task responses. | Need to re-structure json |
| clientAppSource | Clientapp’s task source will be stored and, will be used to publish task |  |
| taskLog | taskQueue history/logging |  |
| clientAppEvent | If clientApp requires sequence promotion, for example, from text clicker to text geoclicker, the rule has to be stored in the table |  |

## Application Details

### MM-Pybossa-Scheduler

### MM-API

### MM-Image

### MM-GIS

### GDELT PLUGIN

## Servers

### Production

### QA

### Development

### Basic Configuration

## Non-functional

### Security

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

### Testing

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

# Issues/Questions

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

# Revision History

| **Date** | **Change Description** |
| --- | --- |
|  | Initial draft. |
|  |  |
|  |  |
|  |  |