



MongoDB Physical Model

Schema for:

Model name: FightPandemics
Author:
Version: 2
File name: data_model.json
File path: C:\Users\nicol\Dev\fight_pandemics\DataModel\v2\data_model.json
Printed On: Wed Apr 22 2020 14:44:35 GMT-0300 (Horário Padrão de Brasília)
Created with: [Hackolade](#) - Visual data modeling for NoSQL and multimodel databases

1. Model

2. Databases

- 2.1 applicationDB
 - [2.1.2. Collections](#)
 - 2.1.2.1 comments
 - 2.1.2.2 feedbacks
 - 2.1.2.3 posts
 - 2.1.2.4 users

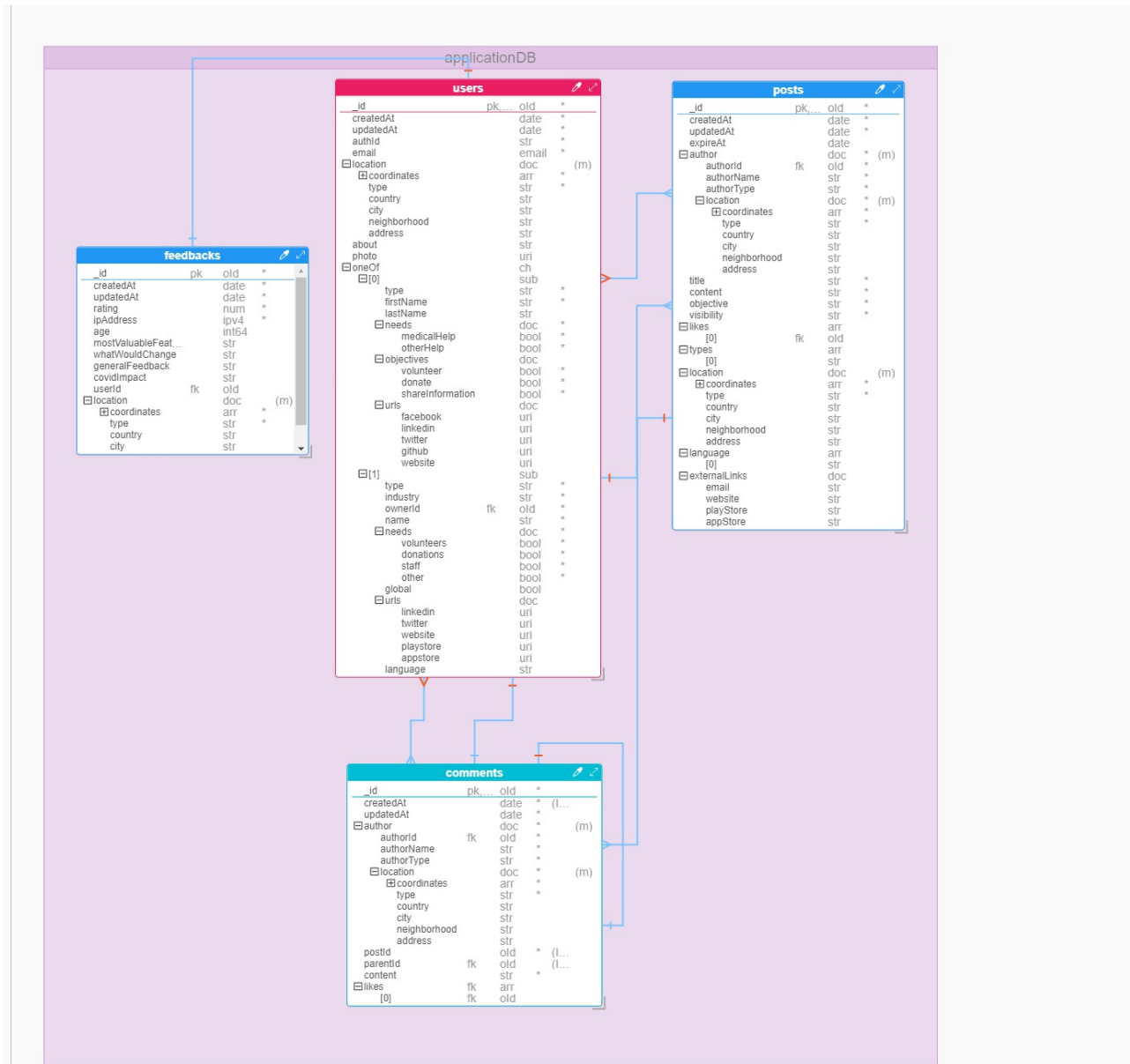
3. Relationships

- 3.1 PostHasComments
- 3.2 UserLikesComment
- 3.3 UserLikesPost
- 3.4 fk comments._id to comments.parentId
- 3.5 fk users._id to comments.likes
- 3.6 fk users._id to feedbacks.userId
- 3.7 fk users._id to organizations..organization.ownerId
- 3.8 fk users._id to posts..author.authorId

1. MODEL

1.1 Model FightPandemics

1.1.1 FightPandemics Entity Relationship Diagram



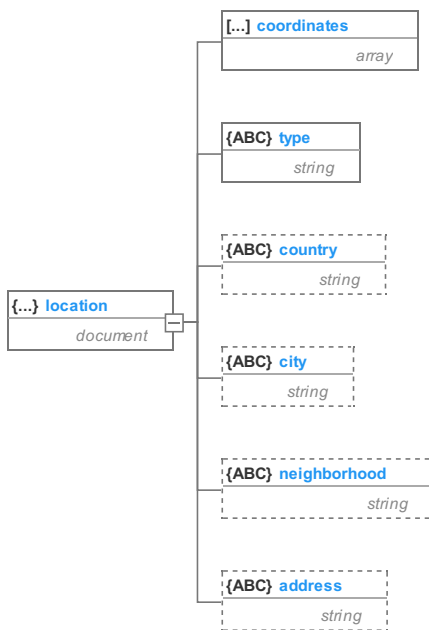
1.1.2 FightPandemics Properties

| PROPERTY | VALUE |
|----------------|----------------|
| Model name | FightPandemics |
| Technical name | |
| Description | |
| Author | |
| Version | 2 |
| DB vendor | MongoDB |
| DB version | v4.2 |
| Comments | |

1.1.3 FightPandemics DB Definitions

1.1.3.1 Field location

1.1.3.1.1 location Tree Diagram



1.1.3.1.2 location Hierarchy

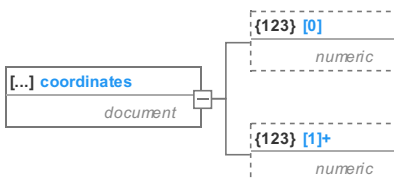
Parent field:

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|--------------|--------|-------|-----|---|----------|
| coordinates | array | true | | Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition. | |
| type | string | true | | Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point". | |
| country | string | false | | String holding the country's name. | |
| city | string | false | | String holding the city's name. | |
| neighborhood | string | false | | String holding the neighborhood's name. | |
| address | string | false | | String holding the address (street, number and other details). | |

1.1.3.1.3 location properties

| PROPERTY | VALUE |
|-----------------------|--|
| Name | location |
| Technical name | |
| Id | |
| Type | document |
| Description | Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| DBRef | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

1.1.3.2 Field coordinates**1.1.3.2.1 coordinates Tree Diagram****1.1.3.2.2 coordinates Hierarchy**Parent field: **location**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-------|---------|-------|-----|--------------------|----------|
| [0] | numeric | false | | Location longitude | |
| [1] | numeric | false | | Location latitude | |

1.1.3.2.3 coordinates properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | coordinates |
| Technical name | |
| Id | |
| Type | array |
| Description | Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Min items | 2 |
| Max items | 2 |
| Unique items | false |
| Additional items | true |
| Comments | |

1.1.3.3 Field [0]**1.1.3.3.1 [0] Tree Diagram**

| |
|---------------------------|
| {123} [0] |
| <i>document</i> |

1.1.3.3.2 [0] properties

| PROPERTY | VALUE |
|--------------------|--------------------|
| Id | lng |
| Type | numeric |
| Subtype | |
| Description | Location longitude |
| Dependencies | |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | 0 |
| Unit | degrees |
| Min value | -180 |
| Excl min | false |
| Max value | 180 |
| Excl max | false |
| Multiple of | |
| Divisible by | |
| Enum | |
| Sample | 35.26 |
| Comments | |

1.1.3.4 Field [1]

1.1.3.4.1 [1] Tree Diagram

| |
|-----------------|
| {123} [1] |
| <i>document</i> |

1.1.3.4.2 [1] properties

| PROPERTY | VALUE |
|--------------------|-------------------|
| Id | lat |
| Type | numeric |
| Subtype | |
| Description | Location latitude |
| Dependencies | |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | 0 |
| Unit | degrees |
| Min value | -90 |
| Excl min | false |
| Max value | 90 |
| Excl max | false |
| Multiple of | |
| Divisible by | |
| Enum | |
| Sample | -32.83 |
| Comments | |

1.1.3.5 Field type**1.1.3.5.1 type Tree Diagram**

| |
|-----------------|
| {ABC} type |
| <i>document</i> |

1.1.3.5.2 type properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | type |
| Technical name | |
| Id | |
| Type | string |
| Description | Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point". |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | Point |
| Sample | |
| Comments | |

1.1.3.6 Field country

1.1.3.6.1 country Tree Diagram

| |
|-------------------------------|
| {ABC} country |
| <i>document</i> |

1.1.3.6.2 country properties

| PROPERTY | VALUE |
|--------------------|------------------------------------|
| Name | country |
| Technical name | |
| Id | |
| Type | string |
| Description | String holding the country's name. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

1.1.3.7 Field city

1.1.3.7.1 city Tree Diagram

| |
|------------|
| {ABC} city |
| document |

1.1.3.7.2 city properties

| PROPERTY | VALUE |
|--------------------|---------------------------------|
| Name | city |
| Technical name | |
| Id | |
| Type | string |
| Description | String holding the city's name. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

1.1.3.8 Field neighborhood

1.1.3.8.1 neighborhood Tree Diagram

| |
|--------------------|
| {ABC} neighborhood |
| document |

1.1.3.8.2 neighborhood properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | neighborhood |
| Technical name | |
| Id | |
| Type | string |
| Description | String holding the neighborhood's name. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

1.1.3.9 Field address**1.1.3.9.1 address Tree Diagram**

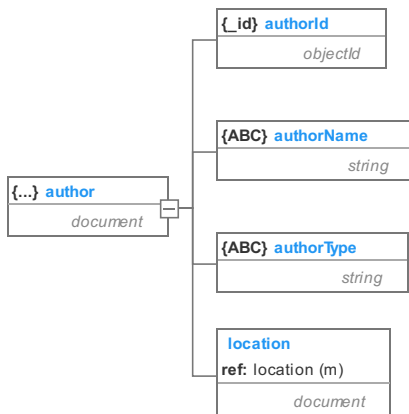
| |
|-------------------------------|
| {ABC} address |
| <i>document</i> |

1.1.3.9.2 address properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | address |
| Technical name | |
| Id | |
| Type | string |
| Description | String holding the address (street, number and other details). |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

1.1.3.10 Field author

1.1.3.10.1 author Tree Diagram



1.1.3.10.2 author Hierarchy

Parent field:

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|------------|-----------|------|-----|---|----------|
| authorId | objectId | true | fk | Foreign key to the user who created the post. | |
| authorName | string | true | | | |
| authorType | string | true | | | |
| location | reference | true | | | |

1.1.3.10.3 author properties

| PROPERTY | VALUE |
|-----------------------|----------|
| Name | author |
| Technical name | |
| Id | |
| Type | document |
| Description | |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| DBRef | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

1.1.3.11 Field authorId

1.1.3.11.1 authorId Tree Diagram



1.1.3.11.2 authorId properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | authorId |
| Technical name | |
| Id | |
| Type | objectId |
| Description | Foreign key to the user who created the post. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | users |
| Foreign field | _id |
| Relationship type | Foreign Key |
| Default | |
| Sample | |
| Comments | |

1.1.3.12 Field authorName**1.1.3.12.1 authorName Tree Diagram**

| |
|-------------------------|
| {ABC} authorName |
| <i>document</i> |

1.1.3.12.2 authorName properties

| PROPERTY | VALUE |
|--------------------|------------|
| Name | authorName |
| Technical name | |
| Id | |
| Type | string |
| Description | |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

1.1.3.13 Field authorType**1.1.3.13.1 authorType Tree Diagram**

| |
|-------------------------|
| {ABC} authorType |
| <i>document</i> |

1.1.3.13.2 authorType properties

| PROPERTY | VALUE |
|--------------------|------------|
| Name | authorType |
| Technical name | |
| Id | |
| Type | string |
| Description | |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

1.1.3.14 Field location

1.1.3.14.1 location Tree Diagram

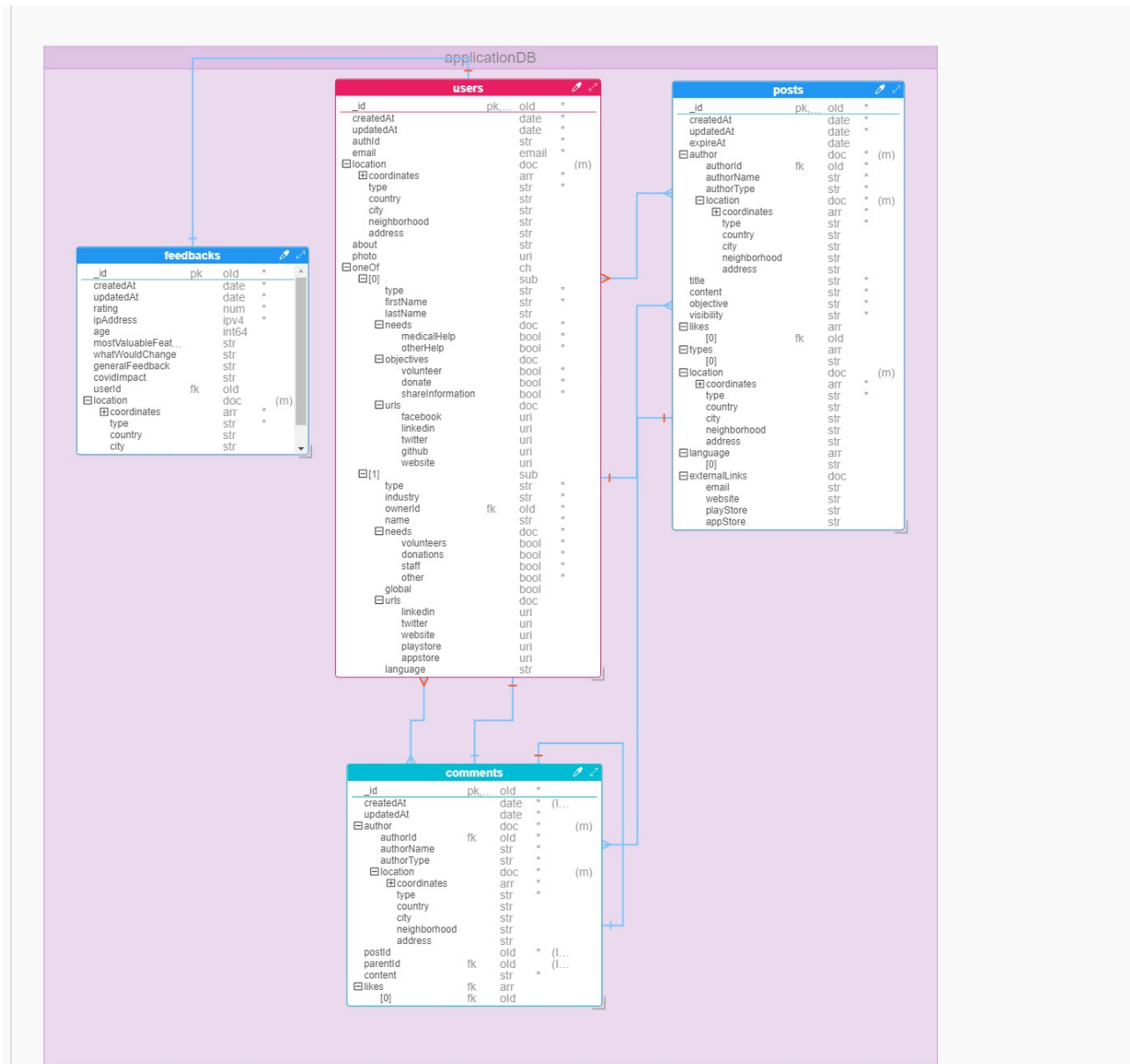
| |
|-------------------|
| location |
| ref: location (m) |
| <i>document</i> |

1.1.3.14.2 location properties

| PROPERTY | VALUE |
|----------------|-----------------------------|
| Name | location |
| \$ref | #model/definitions/location |
| Reference type | model |

2. DATABASES

2.1 Database applicationDB



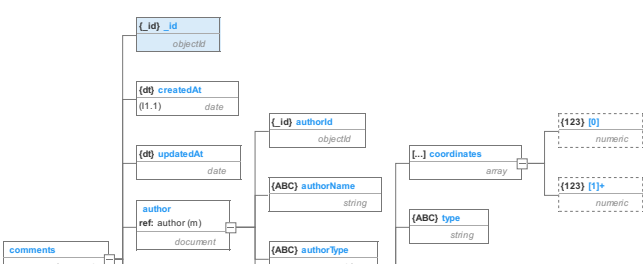
2.1.1 applicationDB Properties

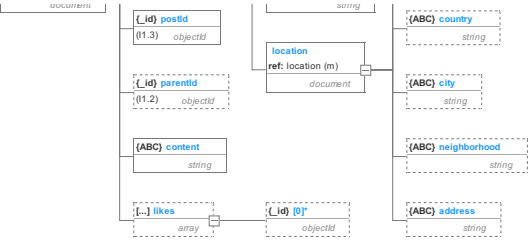
| PROPERTY | VALUE |
|-----------------|---------------|
| Database name | applicationDB |
| Technical name | |
| Enable sharding | |
| Description | |
| Comments | |

2.1.2 applicationDB Collections

2.1.2.1 Collection comments

2.1.2.1.1 comments Tree Diagram





2.1.2.1.2 comments Properties

| PROPERTY | VALUE |
|-----------------------|---------------|
| Collection name | comments |
| Technical name | |
| Id | |
| Description | |
| Database | applicationDB |
| Capped | false |
| Size | |
| Max | |
| Storage engine | WiredTiger |
| Config String | |
| Validation level | Off |
| Validation action | Warn |
| Additional properties | false |
| Comments | |

2.1.2.1.3 comments Fields

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|--------------|----------|-------|-----|--|----------|
| _id | objectId | true | | Automatically generated object_id as primary key. | |
| createdAt | date | true | | Timestamp with the comment's creation date | |
| updatedAt | date | true | | Timestamp with the comment's last update date | |
| author | document | true | | | |
| authorId | objectId | true | | Foreign key to the user who created the post. | |
| authorName | string | true | | | |
| authorType | string | true | | | |
| location | document | true | | Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing. | |
| coordinates | array | true | | Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition. | |
| [0] | numeric | false | | Location longitude | |
| [1] | numeric | false | | Location latitude | |
| type | string | true | | Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point". | |
| country | string | false | | String holding the country's name. | |
| city | string | false | | String holding the city's name. | |
| neighborhood | string | false | | String holding the neighborhood's name. | |
| address | string | false | | String holding the address (street, number and other details). | |
| postId | objectId | true | | | |
| parentId | objectId | false | | If the comment is nested (in another comment), this field will hold the _id of the parent comment. Optional field, where being blank means it is a root comment. | |
| content | string | true | | String holding the comment's content. | |
| likes | array | false | | Array holding the user_id of all the users who liked the comment. | |
| [0] | objectId | false | | _id of the user who liked the comment | |

2.1.2.1.3.1 Field _id

2.1.2.1.3.1.1 _id Tree Diagram

| |
|-----------|
| {_id} _id |
| document |

2.1.2.1.3.1.2 _id properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | _id |
| Technical name | |
| Id | |
| Type | objectId |
| Description | Automatically generated object_id as primary key. |
| Dependencies | |
| Required | true |
| Primary key | true |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.1.3.2 Field createdAt

2.1.2.1.3.2.1 createdAt Tree Diagram

| |
|-----------------|
| {dt} createdAt |
| (11.1) document |

2.1.2.1.3.2.2 createdAt properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | createdAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp with the comment's creation date |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.1.3.3 Field updatedAt

2.1.2.1.3.3.1 updatedAt Tree Diagram

| |
|----------------|
| {dt} updatedAt |
| document |

2.1.2.1.3.3.2 updatedAt properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | updatedAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp with the comment's last update date |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.1.3.4 Field author**2.1.2.1.3.4.1 author Tree Diagram**

| |
|-----------------|
| author |
| ref: author (m) |
| <i>document</i> |

2.1.2.1.3.4.2 author properties

| PROPERTY | VALUE |
|----------------|---------------------------|
| Name | author |
| \$ref | #model/definitions/author |
| Reference type | model |

2.1.2.1.3.5 Field postId**2.1.2.1.3.5.1 postId Tree Diagram**

| |
|-----------------------|
| {_id} postId |
| (1:3) <i>document</i> |

2.1.2.1.3.5.2 postId properties

| PROPERTY | VALUE |
|--------------------|----------|
| Name | postId |
| Technical name | |
| Id | |
| Type | objectId |
| Description | |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | posts |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.1.3.6 Field parentId

2.1.2.1.3.6.1 parentId Tree Diagram

| |
|-----------------------|
| {_id} parentId |
| (1:2) <i>document</i> |

2.1.2.1.3.6.2 parentId properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | parentId |
| Technical name | |
| Id | |
| Type | objectId |
| Description | If the comment is nested (in another comment), this field will hold the _id of the parent comment. Optional field, where being blank means it is a root comment. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | comments |
| Foreign field | _id |
| Relationship type | Foreign Key |
| Default | |
| Sample | |
| Comments | |

2.1.2.1.3.7 Field content

2.1.2.1.3.7.1 content Tree Diagram

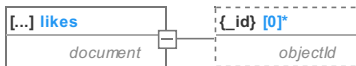
| |
|----------------------|
| {ABC} content |
| <i>document</i> |

2.1.2.1.3.7.2 content properties

| PROPERTY | VALUE |
|--------------------|---------------------------------------|
| Name | content |
| Technical name | |
| Id | |
| Type | string |
| Description | String holding the comment's content. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.1.3.8 Field likes

2.1.2.1.3.8.1 likes Tree Diagram



2.1.2.1.3.8.2 likes Hierarchy

Parent field: **author**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-------|----------|-------|-----|---------------------------------------|----------|
| [0] | objectId | false | fk | _id of the user who liked the comment | |

2.1.2.1.3.8.3 likes properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | likes |
| Technical name | |
| Id | |
| Type | array |
| Description | Array holding the user_id of all the users who liked the comment. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | users |
| Foreign field | _id |
| Relationship type | Foreign Key |
| Min items | |
| Max items | |
| Unique items | true |
| Additional items | true |
| Comments | |

2.1.2.1.3.9 Field coordinates

2.1.2.1.3.9.1 coordinates Tree Diagram



2.1.2.1.3.9.2 coordinates properties

| PROPERTY | VALUE |
|--------------------|---------------------------------------|
| Id | |
| Type | objectId |
| Description | _id of the user who liked the comment |
| Dependencies | |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | Foreign Key |
| Default | |
| Sample | |
| Comments | |

2.1.2.1.4 comments Indexes

| PROPERTY | NEW INDEX |
|------------------------|--|
| Name | New Index |
| Key | createdAt('ascending'), parentId('ascending'), postId('ascending') |
| Hashed | |
| Unique | |
| Drop duplicates | |
| Sparse | |
| Background indexing | |
| Partial filter exp | |
| Expire after (seconds) | |
| Storage engine | WiredTiger |
| Comments | |

2.1.2.1.5 comments JSON Schema

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "comments",
  "additionalProperties": false,
  "properties": {
    "_id": {
      "type": "string",
      "description": "Automatically generated object_id as primary key",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "createdAt": {
      "type": "string",
      "description": "Timestamp with the comment's creation date",
      "format": "date-time"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp with the comment's last update date",
      "format": "date-time"
    },
    "author": {
      "$ref": "#model/definitions/author"
    },
    "postId": {
      "type": "string",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "parentId": {
      "type": "string",
      "description": "If the comment is nested (in another comment), this field will hold the _id of the parent comment. Optional field, where being blank means it is a root comment.",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "content": {
      "type": "string",
      "description": "String holding the comment's content."
    },
    "likes": {
      "type": "array",
      "description": "Array holding the user_id of all the users who liked the comment.",
      "additionalItems": true,
      "uniqueItems": true,
      "items": {
        "type": "string",
        "description": "_id of the user who liked the comment",
        "pattern": "^[a-fA-F0-9]{24}$"
      }
    }
  },
  "required": [
    "_id",
    "createdAt",
    "updatedAt",
    "author",
    "postId",
    "content"
  ]
}

```

2.1.2.1.6 comments JSON data

```
{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "author": {
    "authorId": ObjectId("507f1f77bcf86cd799439011"),
    "authorName": "Lorem",
    "authorType": "Lorem",
    "location": {
      "coordinates": [
        35.26,
        -32.83
      ],
      "type": "Point",
      "country": "Lorem",
      "city": "Lorem",
      "neighborhood": "Lorem",
      "address": "Lorem"
    }
  },
  "postId": ObjectId("507f1f77bcf86cd799439011"),
  "parentId": ObjectId("507f1f77bcf86cd799439011"),
  "content": "Lorem",
  "likes": [
    ObjectId("507f1f77bcf86cd799439011")
  ]
}
```

2.1.2.1.7 comments Target Script

```
use applicationDB;

db.createCollection( "comments",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "comments",
      "additionalProperties": false,
      "properties": {
        "_id": {
          "bsonType": "objectId",
          "description": "Automatically generated object_id as primary key."
        },
        "createdAt": {
          "bsonType": "date",
          "description": "Timestamp with the comment's creation date"
        },
        "updatedAt": {
          "bsonType": "date",
          "description": "Timestamp with the comment's last update date"
        },
        "author": {
          "bsonType": "object",
          "properties": {
            "authorId": {
              "bsonType": "objectId",
              "description": "Foreign key to the user who created the post."
            },
            "authorName": {
              "bsonType": "string"
            }
          }
        }
      }
    }
  }
})
```

```

    },
    "authorType": {
      "bsonType": "string"
    },
    "location": {
      "bsonType": "object",
      "description": "Location document according to the norms of a [GeoJSON Object]
      (https://docs.mongodb.com/manual/reference/geojson/ ).\n\nThe fields address, neighborhood, city, state and country are added to store the
      result of the geoprocessing.",
      "properties": {
        "coordinates": {
          "bsonType": "array",
          "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON]
          (https://docs.mongodb.com/manual/reference/geojson/) definition.",
          "additionalItems": true,
          "minItems": 2,
          "maxItems": 2,
          "uniqueItems": false,
          "items": [
            {
              "bsonType": "number",
              "description": "Location longitude\n",
              "minimum": -180,
              "maximum": 180
            },
            {
              "bsonType": "number",
              "description": "Location latitude\n",
              "minimum": -90,
              "maximum": 90
            }
          ]
        },
        "type": {
          "bsonType": "string",
          "description": "Type of the GeoJSON Object.\n\nPlease refer to the [official documentation]
          (https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'.**",
          "enum": [
            "Point"
          ]
        },
        "country": {
          "bsonType": "string",
          "description": "String holding the country's name.\n"
        },
        "city": {
          "bsonType": "string",
          "description": "String holding the city's name."
        },
        "neighborhood": {
          "bsonType": "string",
          "description": "String holding the neighborhood's name."
        },
        "address": {
          "bsonType": "string",
          "description": "String holding the address (street, number and other details)."
        }
      },
      "additionalProperties": false,
      "required": [
        "coordinates",
        "type"
      ]
    },
    "authorId",
    "authorName",
    "authorType"
  ],
  "additionalProperties": false,
  "required": [
    "authorId",
    "authorName",
    "authorType"
  ]
}

```

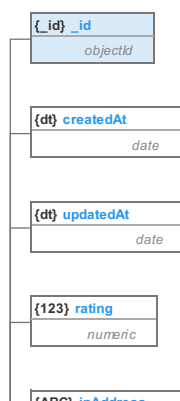
```

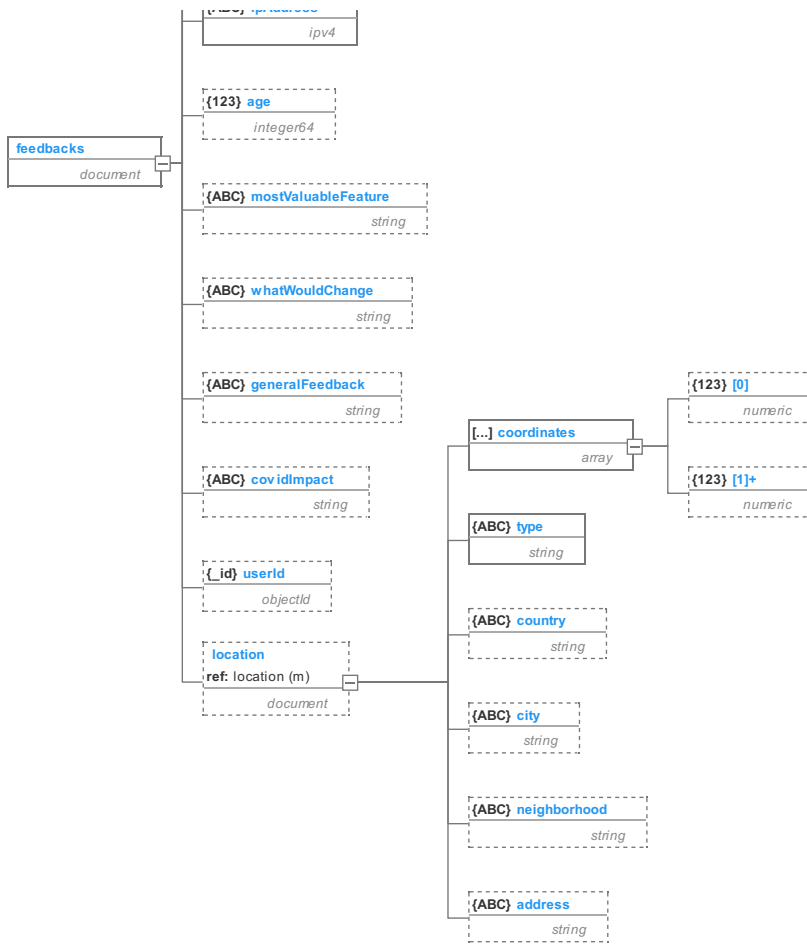
    ]
  },
  "postId": {
    "bsonType": "objectId"
  },
  "parentId": {
    "bsonType": "objectId",
    "description": "If the comment is nested (in another comment), this field will hold the _id of the parent comment. Optional field, where being blank means it is a root comment."
  },
  "content": {
    "bsonType": "string",
    "description": "String holding the comment's content."
  },
  "likes": {
    "bsonType": "array",
    "description": "Array holding the user_id of all the users who liked the comment.",
    "additionalItems": true,
    "uniqueItems": true,
    "items": {
      "bsonType": "objectId",
      "description": "_id of the user who liked the comment"
    }
  }
},
"required": [
  "_id",
  "createdAt",
  "updatedAt",
  "author",
  "postId",
  "content"
]
}
}
},
"validationLevel": "off",
"validationAction": "warn"
});
db.comments.createIndex(
{
  "createdAt": 1,
  "parentId": 1,
  "postId": 1
},
{
  "name": "New Index"
}
);

```

2.1.2.2 Collection feedbacks

2.1.2.2.1 feedbacks Tree Diagram





2.1.2.2.2 feedbacks Properties

| PROPERTY | VALUE |
|-----------------------|---|
| Collection name | feedbacks |
| Technical name | |
| Id | |
| Description | Collection to hold feedback on the application. This may be done with the user logged or not and, depending on that, we may or may not have a reference to the user in question at this model. |
| Database | applicationDB |
| Capped | false |
| Size | |
| Max | |
| Storage engine | WiredTiger |
| Config String | |
| Validation level | Off |
| Validation action | Warn |
| Additional properties | false |
| Comments | |

2.1.2.2.3 feedbacks Fields

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-------|------|-----|-----|-------------|----------|
| | | | | | |

| | | | |
|---------------------|-----------|-------|---|
| _id | objectId | true | Automatically generated object_id as primary key. |
| createdAt | date | true | Timestamp with the feedback's creation date |
| updatedAt | date | true | Timestamp with the feedback's last update date |
| rating | numeric | true | <p>A rating, in a scale of 1 to 5, of the application.</p> <p>Currently present as: "How well does FightPandemics meet your needs?".</p> <p>Present in the first part of the feedback and, therefore, is required.</p> |
| ipAddress | ipv4 | true | <p>Captured IP address in order to get user's location even if they are not logged.</p> <p>Captured at the first part of the feedback and, therefore, is required.</p> |
| age | integer64 | false | <p>User's age.</p> <p>Currently present as: "What is your age?".</p> <p>Present in the third part of the feedback and, therefore, not required.</p> |
| mostValuableFeature | string | false | <p>User's feedback into what is the features most valuable for him/her.</p> <p>Currently present as: "Which features are the most valuable for you?".</p> <p>Present in the second part of the feedback and, therefore, not required.</p> |
| whatWouldChange | string | false | <p>User's feedback into what he/she would change about the application.</p> <p>Currently present as: "If you could change one thing about FightPandemics, what would it be?".</p> <p>Present in the second part of the feedback and, therefore, not required.</p> |
| generalFeedback | string | false | <p>User's general feedback section.</p> <p>Currently present as: "Any other feedback for us?".</p> <p>Present in the second part of the feedback and, therefore, not required.</p> |
| covidImpact | string | false | <p>User's feedback into how the COVID impacted him.</p> <p>Currently present as a multiple choice question.</p> <p>Question: "How has COVID-19 impacted you?"</p> <p>Options:</p> <ol style="list-style-type: none"> 1. I go to work/school normally 2. I am healthy but in a stay-at-home quarantine 3. I have mild symptoms but haven't been tested 4. I am diagnosed with COVID-19 <p>Present in the second part of the feedback and, therefore, not required.</p> |

| | | | | | |
|--------------|----------|-------|--|--|--|
| | | | | required. | |
| userId | objectId | false | | Reference to the user's <code>_id</code> . This will be present only if the user was logged in when completing the feedback. | |
| location | document | false | | Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing. | |
| coordinates | array | true | | Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition. | |
| [0] | numeric | false | | Location longitude | |
| [1] | numeric | false | | Location latitude | |
| type | string | true | | Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point". | |
| country | string | false | | String holding the country's name. | |
| city | string | false | | String holding the city's name. | |
| neighborhood | string | false | | String holding the neighborhood's name. | |
| address | string | false | | String holding the address (street, number and other details). | |

2.1.2.2.3.1 Field createdAt

2.1.2.2.3.1.1 createdAt Tree Diagram

| |
|--------------------------------|
| {dt} createdAt |
| <i>document</i> |

2.1.2.2.3.1.2 createdAt properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | createdAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp with the feedback's creation date |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.2.3.2 Field _id**2.1.2.2.3.2.1 _id Tree Diagram**

| |
|-----------|
| {_id} _id |
| document |

2.1.2.2.3.2.2 _id properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | _id |
| Technical name | |
| Id | |
| Type | objectId |
| Description | Automatically generated object_id as primary key. |
| Dependencies | |
| Required | true |
| Primary key | true |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.2.3.3 Field updatedAt**2.1.2.2.3.3.1 updatedAt Tree Diagram**

| |
|----------------|
| {dt} updatedAt |
| document |

2.1.2.2.3.3.2 updatedAt properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | updatedAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp with the feedback's last update date |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.2.3.4 Field rating**2.1.2.2.3.4.1 rating Tree Diagram**

| |
|--------------|
| {123} rating |
| document |

2.1.2.2.3.4.2 rating properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | rating |
| Technical name | |
| Id | |
| Type | numeric |
| Subtype | |
| Description | A rating, in a scale of 1 to 5, of the application. Currently present as: "How well does FightPandemics meet your needs?". Present in the first part of the feedback and, therefore, is required. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Unit | |
| Min value | 1 |
| Excl min | false |
| Max value | 5 |
| Excl max | false |
| Multiple of | |
| Divisible by | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.2.3.5 Field ipAddress**2.1.2.2.3.5.1 ipAddress Tree Diagram**

| |
|------------------------|
| {ABC} ipAddress |
| <i>document</i> |

2.1.2.2.3.5.2 ipAddress properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | ipAddress |
| Technical name | |
| Id | |
| Type | string |
| Description | Caputred IP address in order to get user's location even if they are not logged. Captured at the first part of the feedback and, therefore, is required. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | ipv4 |
| Enum | |
| Sample | |
| Comments | |

2.1.2.2.3.6 Field age**2.1.2.2.3.6.1 age Tree Diagram**

| |
|---------------------------|
| {123} age |
| <i>document</i> |

2.1.2.2.3.6.2 age properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | age |
| Technical name | |
| Id | |
| Type | numeric |
| Subtype | integer64 |
| Description | User's age. Currently present as: "What is your age?". Present in the third part of the feedback and, therefore, not required. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Unit | |
| Min value | 18 |
| Excl min | false |
| Max value | |
| Excl max | false |
| Multiple of | |
| Divisible by | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.2.3.7 Field mostValuableFeature**2.1.2.2.3.7.1 mostValuableFeature Tree Diagram**

| |
|----------------------------------|
| {ABC} mostValuableFeature |
| <i>document</i> |

2.1.2.2.3.7.2 mostValuableFeature properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | mostValuableFeature |
| Technical name | |
| Id | |
| Type | string |
| Description | User's feedback into what is the features most valuable for him/her. Currently present as: "Which features are the most valuable for you?". Present in the second part of the feedback and, therefore, not required. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.2.3.8 Field whatWouldChange**2.1.2.2.3.8.1 whatWouldChange Tree Diagram**

| |
|---------------------------------------|
| {ABC} whatWouldChange |
| <i>document</i> |

2.1.2.2.3.8.2 whatWouldChange properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | whatWouldChange |
| Technical name | |
| Id | |
| Type | string |
| Description | User's feedback into what he/she would change about the application. Currently present as: "If you could change one thing about FightPandemics, what would it be?". Present in the second part of the feedback and, therefore, not required. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.2.3.9 Field generalFeedback**2.1.2.2.3.9.1 generalFeedback Tree Diagram**

| |
|---------------------------------------|
| {ABC} generalFeedback |
| <i>document</i> |

2.1.2.2.3.9.2 generalFeedback properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | generalFeedback |
| Technical name | |
| Id | |
| Type | string |
| Description | User's general feedback section. Currently present as: "Any other feedback for us?". Present in the second part of the feedback and, therefore, not required. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.2.3.10 Field covidImpact**2.1.2.2.3.10.1 covidImpact Tree Diagram**

| |
|-------------------|
| {ABC} covidImpact |
| document |

2.1.2.2.3.10.2 covidImpact properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | covidImpact |
| Technical name | |
| Id | |
| Type | string |
| Description | <p>User's feedback into how the COVID impacted him.</p> <p>Currently present as a multiple choice question.</p> <p>Question: "How has COVID-19 impacted you?"</p> <p>Options:</p> <ol style="list-style-type: none"> 1. I go to work/school normally 2. I am healthy but in a stay-at-home quarantine 3. I have mild symptoms but haven't been tested 4. I am diagnosed with COVID-19 <p>Present in the second part of the feedback and, therefore, not required.</p> |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.2.3.11 Field userId**2.1.2.2.3.11.1 userId Tree Diagram**

| |
|---------------------|
| {_id} userId |
| document |

2.1.2.2.3.11.2 userId properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | userId |
| Technical name | |
| Id | |
| Type | objectId |
| Description | Reference to the user's _id. This will be present only if the user was logged in when completing the feedback. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | users |
| Foreign field | _id |
| Relationship type | Foreign Key |
| Default | |
| Sample | |
| Comments | |

2.1.2.2.3.12 Field location**2.1.2.2.3.12.1 location Tree Diagram**

| |
|-------------------|
| location |
| ref: location (m) |
| document |

2.1.2.2.3.12.2 location properties

| PROPERTY | VALUE |
|----------------|-----------------------------|
| Name | location |
| \$ref | #model/definitions/location |
| Reference type | model |

2.1.2.2.4 feedbacks JSON Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "feedbacks",
  "description": "Collection to hold feedback on the application.\n\nThis may be done with the user logged or not and, depending on that, we may or may not have a reference to the user in question at this model.",
  "additionalProperties": false,
  "properties": {
    "createdAt": {
      "type": "string",
      "description": "Timestamp with the feedback's creation date",
      "format": "date-time"
    },
    "_id": {
      "type": "string",
      "description": "Automatically generated object_id as primary key",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp with the feedback's last update date",
      "format": "date-time"
    },
    "rating": {
```

```

    "type": "number",
    "description": "Rating, in a scale of 1 to 5, of the application.\n\nCurrently present as: \"How well does FightPandemics meet your needs?\".\n\nPresent in the first part of the feedback and, therefore, is required.",
    "minimum": 1,
    "maximum": 5
  },
  "ipAddress": {
    "type": "string",
    "description": "Captured IP address in order to get user's location even if they are not logged.\n\nCaptured at the first part of the feedback and, therefore, is required.",
    "format": "ipv4"
  },
  "age": {
    "type": "long",
    "description": "User's age.\n\nCurrently present as: \"What is your age?\".\n\nPresent in the third part of the feedback and, therefore, not required.",
    "minimum": 18
  },
  "mostValuableFeature": {
    "type": "string",
    "description": "User's feedback into what is the features most valuable for him/her.\n\nCurrently present as: \"Which features are the most valuable for you?\".\n\nPresent in the second part of the feedback and, therefore, not required."
  },
  "whatWouldChange": {
    "type": "string",
    "description": "User's feedback into what he/she would change about the application.\n\nCurrently present as: \"If you could change one thing about FightPandemics, what would it be?\".\n\nPresent in the second part of the feedback and, therefore, not required."
  },
  "generalFeedback": {
    "type": "string",
    "description": "User's general feedback section.\n\nCurrently present as: \"Any other feedback for us?\".\n\nPresent in the second part of the feedback and, therefore, not required."
  },
  "covidImpact": {
    "type": "string",
    "description": "User's feedback into how the COVID impacted him.\n\nCurrently present as a multiple choice question.\n\n**Question:** \"How has COVID-19 impacted you?\"\n\n**Options:** \n1. I go to work/school normally\n2. I am healthy but in a stay-at-home quarantine\n3. I have mild symptoms but haven't been tested\n4. I am diagnosed with COVID-19\n\nPresent in the second part of the feedback and, therefore, not required."
  },
  "userId": {
    "type": "string",
    "description": "Reference to the user's _id.\n\nThis will be present only if the user was logged in when completing the feedback.",
    "pattern": "^a-fA-F0-9{24}$"
  },
  "location": {
    "$ref": "#model/definitions/location"
  }
},
"required": [
  "createdAt",
  "_id",
  "updatedAt",
  "rating",
  "ipAddress"
]
}

```

2.1.2.2.5 feedbacks JSON data

```
{
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "rating": 1,
  "ipAddress": "192.168.0.1",
  "age": 8528994625872795000,
  "mostValuableFeature": "Lorem",
  "whatWouldChange": "Lorem",
  "generalFeedback": "Lorem",
  "covidImpact": "Lorem",
  "userId": ObjectId("507f1f77bcf86cd799439011"),
  "location": {
    "coordinates": [
      35.26,
      -32.83
    ],
    "type": "Point",
    "country": "Lorem",
    "city": "Lorem",
    "neighborhood": "Lorem",
    "address": "Lorem"
  }
}
```

2.1.2.2.6 feedbacks Target Script

```
use applicationDB;

db.createCollection( "feedbacks",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "feedbacks",
      "description": "Collection to hold feedback on the application.\n\nThis may be done with the user logged or not and, depending on that, we may or may not have a reference to the user in question at this model.",
      "additionalProperties": false,
      "properties": {
        "_id": {
          "bsonType": "objectId",
          "description": "Automatically generated object_id as primary key."
        },
        "createdAt": {
          "bsonType": "date",
          "description": "Timestamp with the feedback's creation date"
        },
        "updatedAt": {
          "bsonType": "date",
          "description": "Timestamp with the feedback's last update date"
        },
        "rating": {
          "bsonType": "number",
          "description": "A rating, in a scale of 1 to 5, of the application.\n\nCurrently present as: \"How well does FlightPandemics meet your needs?\".\n\nPresent in the first part of the feedback and, therefore, is required.",
          "minimum": 1,
          "maximum": 5
        },
        "ipAddress": {
          "bsonType": "string",
          "description": "Captured IP address in order to get user's location even if they are not logged.\n\nCaptured at the first part of the"
        }
      }
    }
  }
})
```

```

feedback and, therefore, is required."
    },
    "age": {
      "bsonType": "long",
      "description": "User's age.\n\nCurrently present as: \"What is your age?\".\n\nPresent in the third part of the feedback and, therefore,
not required.",
      "minimum": 18
    },
    "mostValuableFeature": {
      "bsonType": "string",
      "description": "User's feedback into what is the features most valuable for him/her.\n\nCurrently present as: \"Which features are the
most valuable for you?\".\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "whatWouldChange": {
      "bsonType": "string",
      "description": "User's feedback into what he/she would change about the application.\n\nCurrently present as: \"If you could change
one thing about FightPandemics, what would it be?\".\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "generalFeedback": {
      "bsonType": "string",
      "description": "User's general feedback section.\n\nCurrently present as: \"Any other feedback for us?\".\n\nPresent in the second
part of the feedback and, therefore, not required."
    },
    "covidImpact": {
      "bsonType": "string",
      "description": "User's feedback into how the COVID impacted him.\n\nCurrently present as a multiple choice
question.\n\n**Question:** \"How has COVID-19 impacted you?\"\n\n**Options:** \n1. I go to work/school normally\n2. I am healthy but in a
stay-at-home quarantine\n3. I have mild symptoms but haven't been tested\n4. I am diagnosed with COVID-19\n\nPresent in the second part
of the feedback and, therefore, not required."
    },
    "userId": {
      "bsonType": "objectId",
      "description": "Reference to the user's _id.\n\nThis will be present only if the user was logged in when completing the feedback."
    },
    "location": {
      "bsonType": "object",
      "description": "Location document according to the norms of a [GeoJSON Object]
(https://docs.mongodb.com/manual/reference/geojson/).\n\nThe fields address, neighborhood, city, state and country are added to store the
result of the geoprocessing.",
      "properties": {
        "coordinates": {
          "bsonType": "array",
          "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON]
(https://docs.mongodb.com/manual/reference/geojson/) definition.",
          "additionalItems": true,
          "minItems": 2,
          "maxItems": 2,
          "uniqueItems": false,
          "items": [
            {
              "bsonType": "number",
              "description": "Location longitude\n",
              "minimum": -180,
              "maximum": 180
            },
            {
              "bsonType": "number",
              "description": "Location latitude\n",
              "minimum": -90,
              "maximum": 90
            }
          ]
        },
        "type": {
          "bsonType": "string",
          "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation]
(https://docs.mongodb.com/manual/reference/geojson/) for more information.\n\n**In our case, the value will always be of the type \"Point\".**",
          "enum": [
            "Point"
          ]
        }
      }
    }
  }
}

```

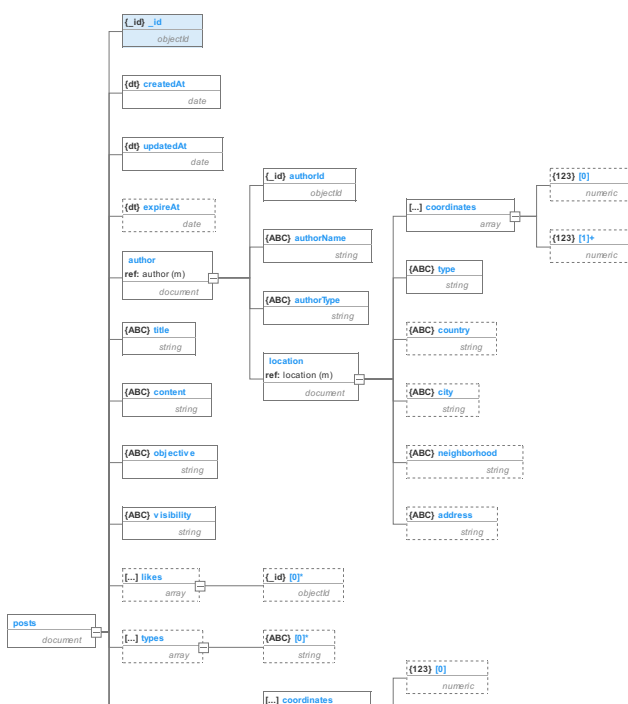
```

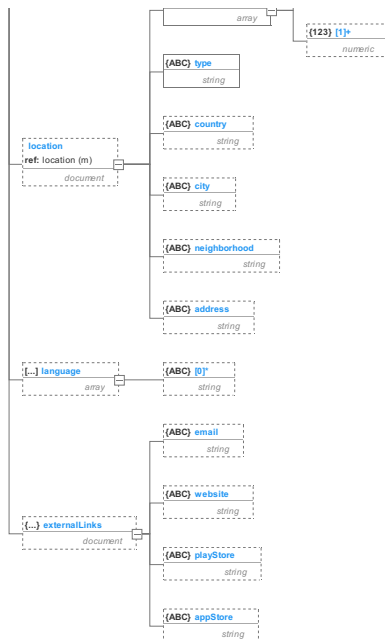
    },
    "country": {
      "bsonType": "string",
      "description": "String holding the country's name.\n"
    },
    "city": {
      "bsonType": "string",
      "description": "String holding the city's name."
    },
    "neighborhood": {
      "bsonType": "string",
      "description": "String holding the neighborhood's name."
    },
    "address": {
      "bsonType": "string",
      "description": "String holding the address (street, number and other details).\"
    }
  },
  "additionalProperties": false,
  "required": [
    "coordinates",
    "type"
  ]
}
},
"required": [
  "createdAt",
  "_id",
  "updatedAt",
  "rating",
  "ipAddress"
]
}
},
"validationLevel": "off",
"validationAction": "warn"
});

```

2.1.2.3 Collection posts

2.1.2.3.1 posts Tree Diagram





2.1.2.3.2 posts Properties

| PROPERTY | VALUE |
|-----------------------|---|
| Collection name | posts |
| Technical name | |
| Id | |
| Description | Document to represent a post, asking or offering help to be displayed on the feed |
| Database | applicationDB |
| Capped | false |
| Size | |
| Max | |
| Storage engine | WiredTiger |
| Config String | |
| Validation level | Off |
| Validation action | Warn |
| Additional properties | false |
| Comments | |

2.1.2.3.3 posts Fields

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|------------|----------|-------|-----|---|----------|
| _id | objectId | true | | | |
| createdAt | date | true | | Timestamp with the post's creation date | |
| updatedAt | date | true | | Timestamp with the post's last update date | |
| expireAt | date | false | | Timestamp with the date the post will expire. If no expiration date is present, the post does not expire. | |
| author | document | true | | | |
| authorId | objectId | true | | Foreign key to the user who created the post. | |
| authorName | string | true | | | |
| authorType | string | true | | | |
| | | | | Location document according to the norms of a GeoJSON Object . | |

| | | | | | |
|--------------|----------|-------|--|--|--|
| location | document | true | | The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing. | |
| coordinates | array | true | | Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition. | |
| [0] | numeric | false | | Location longitude | |
| [1] | numeric | false | | Location latitude | |
| type | string | true | | Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point". | |
| country | string | false | | String holding the country's name. | |
| city | string | false | | String holding the city's name. | |
| neighborhood | string | false | | String holding the neighborhood's name. | |
| address | string | false | | String holding the address (street, number and other details). | |
| title | string | true | | The post title. | |
| content | string | true | | String for the post content. | |
| objective | string | true | | Field to indicate the post's objective. Currently only two values allowed: "offering help" and "looking for help". | |
| visibility | string | true | | Field to specify the post's visibility. It can be: worldwide, country, state and zipcode. | |
| likes | array | false | | Array holding the user_id of all the users who liked the post. | |
| [0] | objectId | false | | _id of the user who liked the post | |
| types | array | false | | Array with the post's tags. Can be: Medical Supplies, Groceries/Food, Business, Education, Legal, Wellbeing/Mental, Entertainment, Information, Funding, R&D and Others | |
| [0] | string | false | | | |
| location | document | false | | Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing. | |
| coordinates | array | true | | Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition. | |
| [0] | numeric | false | | Location longitude | |
| [1] | numeric | false | | Location latitude | |
| | | | | | |

| | | | | | |
|---------------|----------|-------|--|---|--|
| type | string | true | | Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point". | |
| country | string | false | | String holding the country's name. | |
| city | string | false | | String holding the city's name. | |
| neighborhood | string | false | | String holding the neighborhood's name. | |
| address | string | false | | String holding the address (street, number and other details). | |
| language | array | false | | | |
| [0] | string | false | | | |
| externalLinks | document | false | | | |
| email | string | false | | An email related to the post's content. | |
| website | string | false | | Link for a website related to the post's content. | |
| playStore | string | false | | Link for a Play Store application related to the post's content. | |
| appStore | string | false | | Link for a AppStore application related to the post's content. | |

2.1.2.3.3.1 Field _id

2.1.2.3.3.1.1 _id Tree Diagram

| |
|-----------|
| {_id} _id |
| document |

2.1.2.3.3.1.2 _id properties

| PROPERTY | VALUE |
|--------------------|----------|
| Name | _id |
| Technical name | |
| Id | |
| Type | objectId |
| Description | |
| Dependencies | |
| Required | true |
| Primary key | true |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.3.3.2 Field createdAt

2.1.2.3.3.2.1 createdAt Tree Diagram

| |
|----------------|
| {dt} createdAt |
| document |

2.1.2.3.3.2 createdAt properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | createdAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp with the post's creation date |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.3.3.3 Field updatedAt**2.1.2.3.3.3.1 updatedAt Tree Diagram**

| |
|----------------|
| {dt} updatedAt |
| document |

2.1.2.3.3.2 updatedAt properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | updatedAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp with the post's last update date |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.3.3.4 Field expireAt**2.1.2.3.3.4.1 expireAt Tree Diagram**

| |
|---------------|
| {dt} expireAt |
| document |

2.1.2.3.3.4.2 expireAt properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | expireAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp with the date the post will expire. If no expiration date is present, the post does not expire. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.3.3.5 Field author**2.1.2.3.3.5.1 author Tree Diagram**

| |
|-----------------|
| author |
| ref: author (m) |
| document |

2.1.2.3.3.5.2 author properties

| PROPERTY | VALUE |
|----------------|---------------------------|
| Name | author |
| \$ref | #model/definitions/author |
| Reference type | model |

2.1.2.3.3.6 Field title**2.1.2.3.3.6.1 title Tree Diagram**

| |
|--------------------|
| {ABC} title |
| document |

2.1.2.3.3.6.2 title properties

| PROPERTY | VALUE |
|--------------------|-----------------|
| Name | title |
| Technical name | |
| Id | |
| Type | string |
| Description | The post title. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.3.3.7 Field content**2.1.2.3.3.7.1 content Tree Diagram**

| |
|----------------------|
| {ABC} content |
| <i>document</i> |

2.1.2.3.3.7.2 content properties

| PROPERTY | VALUE |
|--------------------|------------------------------|
| Name | content |
| Technical name | |
| Id | |
| Type | string |
| Description | String for the post content. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.3.3.8 Field objective

2.1.2.3.3.8.1 objective Tree Diagram

| |
|------------------------|
| {ABC} objective |
| <i>document</i> |

2.1.2.3.3.8.2 objective properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | objective |
| Technical name | |
| Id | |
| Type | string |
| Description | Field to indicate the post's objective. Currently only two values allowed: "offering help" and "looking for help". |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | request,offer |
| Sample | offer |
| Comments | |

2.1.2.3.3.9 Field visibility**2.1.2.3.3.9.1 visibility Tree Diagram**

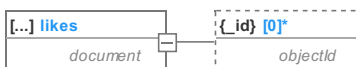
| |
|-------------------------|
| {ABC} visibility |
| <i>document</i> |

2.1.2.3.3.9.2 visibility properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | visibility |
| Technical name | |
| Id | |
| Type | string |
| Description | Field to specify the post's visibility. It can be: worldwide, country, state and zipcode. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | zipcode,state,country,worldwide |
| Sample | |
| Comments | |

2.1.2.3.3.10 Field likes

2.1.2.3.3.10.1 likes Tree Diagram



2.1.2.3.3.10.2 likes Hierarchy

Parent field: **location**

Child field(s):

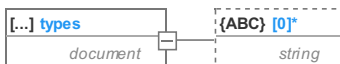
| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-------|----------|-------|-----|------------------------------------|----------|
| [0] | objectId | false | fk | _id of the user who liked the post | |

2.1.2.3.3.10.3 likes properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | likes |
| Technical name | |
| Id | |
| Type | array |
| Description | Array holding the user_id of all the users who liked the post. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Min items | |
| Max items | |
| Unique items | true |
| Additional items | true |
| Comments | |

2.1.2.3.3.11 Field [0]**2.1.2.3.3.11.1 [0] Tree Diagram****2.1.2.3.3.11.2 [0] properties**

| PROPERTY | VALUE |
|--------------------|------------------------------------|
| Id | |
| Type | objectId |
| Description | _id of the user who liked the post |
| Dependencies | |
| Primary key | false |
| Foreign collection | users |
| Foreign field | _id |
| Relationship type | Foreign Key |
| Default | |
| Sample | |
| Comments | |

2.1.2.3.3.12 Field types**2.1.2.3.3.12.1 types Tree Diagram****2.1.2.3.3.12.2 types Hierarchy**Parent field: **coordinates**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-------|--------|-------|-----|-------------|----------|
| [0] | string | false | | | |

2.1.2.3.3.12.3 types properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | types |
| Technical name | |
| Id | |
| Type | array |
| Description | Array with the post's tags. Can be: Medical Supplies, Groceries/Food, Business, Education, Legal, Wellbeing/Mental, Entertainment, Information, Funding, R&D and Others |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Min items | |
| Max items | |
| Unique items | false |
| Additional items | true |
| Comments | |

2.1.2.3.3.13 Field type**2.1.2.3.3.13.1 type Tree Diagram**

| |
|------------|
| {ABC} type |
| document |

2.1.2.3.3.13.2 type properties

| PROPERTY | VALUE |
|--------------------|--|
| Id | |
| Type | string |
| Description | |
| Dependencies | |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | Medical Supplies,Groceries/Food,Business,Education,Legal,Wellbeing/Mental,Entertainment,Information,Funding,R&D,Others |
| Sample | Groceries/Food |
| Comments | |

2.1.2.3.3.14 Field location

2.1.2.3.3.14.1 location Tree Diagram

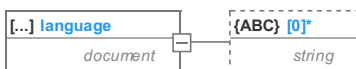


2.1.2.3.3.14.2 location properties

| PROPERTY | VALUE |
|----------------|-----------------------------|
| Name | location |
| \$ref | #model/definitions/location |
| Reference type | model |

2.1.2.3.3.15 Field language

2.1.2.3.3.15.1 language Tree Diagram



2.1.2.3.3.15.2 language Hierarchy

Parent field: **location**

Child field(s):

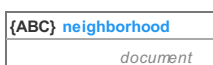
| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-------|--------|-------|-----|-------------|----------|
| [0] | string | false | | | |

2.1.2.3.3.15.3 language properties

| PROPERTY | VALUE |
|--------------------|----------|
| Name | language |
| Technical name | |
| Id | |
| Type | array |
| Description | |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Min items | |
| Max items | |
| Unique items | false |
| Additional items | true |
| Comments | |

2.1.2.3.3.16 Field neighborhood

2.1.2.3.3.16.1 neighborhood Tree Diagram

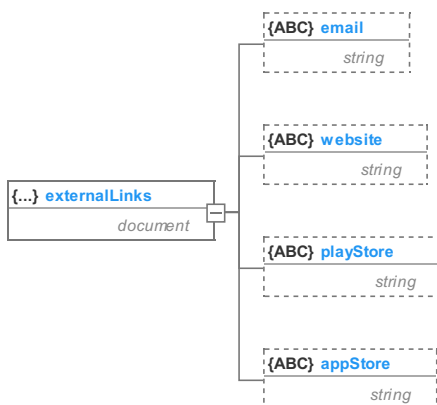


2.1.2.3.3.16.2 neighborhood properties

| PROPERTY | VALUE |
|--------------------|--------|
| Id | |
| Type | string |
| Description | |
| Dependencies | |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.3.3.17 Field externalLinks

2.1.2.3.3.17.1 externalLinks Tree Diagram



2.1.2.3.3.17.2 externalLinks Hierarchy

Parent field: **location**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-----------|--------|-------|-----|--|----------|
| email | string | false | | An email related to the post's content. | |
| website | string | false | | Link for a website related to the post's content. | |
| playStore | string | false | | Link for a Play Store application related to the post's content. | |
| appStore | string | false | | Link for a AppStore application related to the post's content. | |

2.1.2.3.3.17.3 externalLinks properties

| PROPERTY | VALUE |
|-----------------------|---------------|
| Name | externalLinks |
| Technical name | |
| Id | |
| Type | document |
| Description | |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| DBRef | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

2.1.2.3.3.18 Field email**2.1.2.3.3.18.1 email Tree Diagram**

| |
|--------------------|
| {ABC} email |
| <i>document</i> |

2.1.2.3.3.18.2 email properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | email |
| Technical name | |
| Id | |
| Type | string |
| Description | An email related to the post's content. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.3.3.19 Field website**2.1.2.3.3.19.1 website Tree Diagram**

| |
|----------------------|
| {ABC} website |
| <i>document</i> |

2.1.2.3.3.19.2 website properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | website |
| Technical name | |
| Id | |
| Type | string |
| Description | Link for a website related to the post's content. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.3.3.20 Field playStore**2.1.2.3.3.20.1 playStore Tree Diagram**

| |
|------------------------|
| {ABC} playStore |
| <i>document</i> |

2.1.2.3.3.20.2 playStore properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | playStore |
| Technical name | |
| Id | |
| Type | string |
| Description | Link for a Play Store application related to the post's content. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.3.3.21 Field appStore

2.1.2.3.3.21.1 appStore Tree Diagram

| |
|-----------------------|
| {ABC} appStore |
| <i>document</i> |

2.1.2.3.3.21.2 appStore properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | appStore |
| Technical name | |
| Id | |
| Type | string |
| Description | Link for a AppStore application related to the post's content. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.3.4 posts JSON Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "posts",
  "description": "Document to represent a post, asking or offering help to be displayed on the feed",
  "additionalProperties": false,
  "properties": {
    "_id": {
      "type": "string",
      "pattern": "^[a-zA-F0-9]{24}$"
    },
    "createdAt": {
      "type": "string",
      "description": "Timestamp with the post's creation date",
      "format": "date-time"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp with the post's last update date",
      "format": "date-time"
    },
    "expireAt": {
      "type": "string",
      "description": "Timestamp with the date the post will expire.\nIf no expiration date is present, the post does not expire.",
      "format": "date-time"
    },
    "author": {
      "$ref": "#model/definitions/author"
    },
    "title": {
```

```

    "type": "string",
    "description": "The post title."
  },
  "content": {
    "type": "string",
    "description": "String for the post content."
  },
  "objective": {
    "type": "string",
    "description": "Field to indicate the post's objective.\nCurrently only two values allowed: \"offering help\" and \"looking for help\".",
    "enum": [
      "request",
      "offer"
    ]
  },
  "visibility": {
    "type": "string",
    "description": "Field to specify the post's visibility. It can be: worldwide, country, state and zipcode.",
    "enum": [
      "zipcode",
      "state",
      "country",
      "worldwide"
    ]
  },
  "likes": {
    "type": "array",
    "description": "Array holding the user_id of all the users who liked the post.",
    "additionalItems": true,
    "uniqueItems": true,
    "items": {
      "type": "string",
      "description": " _id of the user who liked the post",
      "pattern": "^[a-fA-F0-9]{24}$"
    }
  },
  "types": {
    "type": "array",
    "description": "Array with the post's tags.\nCan be: Medical Supplies, Groceries/Food, Business, Education, Legal, Wellbeing/Mental, Entertainment, Information, Funding, R&D and Others",
    "additionalItems": true,
    "uniqueItems": false,
    "items": {
      "type": "string",
      "enum": [
        "Medical Supplies",
        "Groceries/Food",
        "Business",
        "Education",
        "Legal",
        "Wellbeing/Mental",
        "Entertainment",
        "Information",
        "Funding",
        "R&D",
        "Others"
      ]
    }
  },
  "location": {
    "$ref": "#model/definitions/location"
  },
  "language": {
    "type": "array",
    "additionalItems": true,
    "uniqueItems": false,
    "items": {
      "type": "string"
    }
  },

```



```
"externalLinks": {
  "type": "object",
  "properties": {
    "email": {
      "type": "string",
      "description": "An email related to the post's content."
    },
    "website": {
      "type": "string",
      "description": "Link for a website related to the post's content."
    },
    "playStore": {
      "type": "string",
      "description": "Link for a Play Store application related to the post's content."
    },
    "appStore": {
      "type": "string",
      "description": "Link for a AppStore application related to the post's content."
    }
  },
  "additionalProperties": false
},
"required": [
  "_id",
  "createdAt",
  "updatedAt",
  "author",
  "title",
  "content",
  "objective",
  "visibility"
]
}
```

2.1.2.3.5 posts JSON data

```
{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "expireAt": ISODate("2016-04-08T15:06:21.595Z"),
  "author": {
    "authorId": ObjectId("507f1f77bcf86cd799439011"),
    "authorName": "Lorem",
    "authorType": "Lorem",
    "location": {
      "coordinates": [
        35.26,
        -32.83
      ],
      "type": "Point",
      "country": "Lorem",
      "city": "Lorem",
      "neighborhood": "Lorem",
      "address": "Lorem"
    }
  },
  "title": "Lorem",
  "content": "Lorem",
  "objective": "offer",
  "visibility": "worldwide",
  "likes": [
    ObjectId("507f1f77bcf86cd799439011")
  ],
  "types": [
    "Groceries/Food"
  ],
  "location": {
    "coordinates": [
      35.26,
      -32.83
    ],
    "type": "Point",
    "country": "Lorem",
    "city": "Lorem",
    "neighborhood": "Lorem",
    "address": "Lorem"
  },
  "language": [
    "Lorem"
  ],
  "externalLinks": {
    "email": "Lorem",
    "website": "Lorem",
    "playStore": "Lorem",
    "appStore": "Lorem"
  }
}
```

2.1.2.3.6 posts Target Script

```
use applicationDB;

db.createCollection( "posts",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
```

```

"bsonType": "object",
"title": "posts",
"description": "Document to represent a post, asking or offering help to be displayed on the feed",
"additionalProperties": false,
"properties": {
  "_id": {
    "bsonType": "objectId"
  },
  "createdAt": {
    "bsonType": "date",
    "description": "Timestamp with the post's creation date"
  },
  "updatedAt": {
    "bsonType": "date",
    "description": "Timestamp with the post's last update date"
  },
  "expireAt": {
    "bsonType": "date",
    "description": "Timestamp with the date the post will expire.\nIf no expiration date is present, the post does not expire."
  },
  "author": {
    "bsonType": "object",
    "properties": {
      "authorId": {
        "bsonType": "objectId",
        "description": "Foreign key to the user who created the post."
      },
      "authorName": {
        "bsonType": "string"
      },
      "authorType": {
        "bsonType": "string"
      }
    },
    "location": {
      "bsonType": "object",
      "description": "Location document according to the norms of a [GeoJSON Object]
      (https://docs.mongodb.com/manual/reference/geojson/ ).\nThe fields address, neighborhood, city, state and country are added to store the
      result of the geoprocessing.",
      "properties": {
        "coordinates": {
          "bsonType": "array",
          "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON]
          (https://docs.mongodb.com/manual/reference/geojson/) definition.",
          "additionalItems": true,
          "minItems": 2,
          "maxItems": 2,
          "uniqueItems": false,
          "items": [
            {
              "bsonType": "number",
              "description": "Location longitude\n",
              "minimum": -180,
              "maximum": 180
            },
            {
              "bsonType": "number",
              "description": "Location latitude\n",
              "minimum": -90,
              "maximum": 90
            }
          ]
        },
        "type": {
          "bsonType": "string",
          "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation]
          (https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'**.***",
          "enum": [
            "Point"
          ]
        }
      }
    }
  }
}

```

```

    "country": {
      "bsonType": "string",
      "description": "String holding the country's name.\n"
    },
    "city": {
      "bsonType": "string",
      "description": "String holding the city's name."
    },
    "neighborhood": {
      "bsonType": "string",
      "description": "String holding the neighborhood's name."
    },
    "address": {
      "bsonType": "string",
      "description": "String holding the address (street, number and other details).\"
    }
  },
  "additionalProperties": false,
  "required": [
    "coordinates",
    "type"
  ]
},
"additionalProperties": false,
"required": [
  "authorId",
  "authorName",
  "authorType"
]
},
"title": {
  "bsonType": "string",
  "description": "The post title."
},
"content": {
  "bsonType": "string",
  "description": "String for the post content."
},
"objective": {
  "bsonType": "string",
  "description": "Field to indicate the post's objective.\nCurrently only two values allowed: \"offering help\" and \"looking for help\".",
  "enum": [
    "request",
    "offer"
  ]
},
"visibility": {
  "bsonType": "string",
  "description": "Field to specify the post's visibility. It can be: worldwide, country, state and zipcode.",
  "enum": [
    "zipcode",
    "state",
    "country",
    "worldwide"
  ]
},
"likes": {
  "bsonType": "array",
  "description": "Array holding the user_id of all the users who liked the post.",
  "additionalItems": true,
  "uniqueItems": true,
  "items": {
    "bsonType": "objectId",
    "description": "\"_id of the user who liked the post\""
  }
},
"types": {
  "bsonType": "array",
  "description": "Array with the post's tags.\nCan be: Medical Supplies, Groceries/Food, Business, Education, Legal, Wellbeing/Mental,

```

```

Entertainment, Information, Funding, R&D and Others",
  "additionalItems": true,
  "uniqueItems": false,
  "items": {
    "bsonType": "string",
    "enum": [
      "Medical Supplies",
      "Groceries/Food",
      "Business",
      "Education",
      "Legal",
      "Wellbeing/Mental",
      "Entertainment",
      "Information",
      "Funding",
      "R&D",
      "Others"
    ]
  }
},
"location": {
  "bsonType": "object",
  "description": "Location document according to the norms of a [GeoJSON Object]
(https://docs.mongodb.com/manual/reference/geojson/).\n\nThe fields address, neighborhood, city, state and country are added to store the
result of the geoprocessing.",
  "properties": {
    "coordinates": {
      "bsonType": "array",
      "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON]
(https://docs.mongodb.com/manual/reference/geojson/) definition.",
      "additionalItems": true,
      "minItems": 2,
      "maxItems": 2,
      "uniqueItems": false,
      "items": [
        {
          "bsonType": "number",
          "description": "Location longitude\n",
          "minimum": -180,
          "maximum": 180
        },
        {
          "bsonType": "number",
          "description": "Location latitude\n",
          "minimum": -90,
          "maximum": 90
        }
      ]
    },
    "type": {
      "bsonType": "string",
      "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation]
(https://docs.mongodb.com/manual/reference/geojson/) for more information.\n\n**In our case, the value will always be of the type 'Point'**.***",
      "enum": [
        "Point"
      ]
    },
    "country": {
      "bsonType": "string",
      "description": "String holding the country's name.\n"
    },
    "city": {
      "bsonType": "string",
      "description": "String holding the city's name."
    },
    "neighborhood": {
      "bsonType": "string",
      "description": "String holding the neighborhood's name."
    },
    "address": {

```

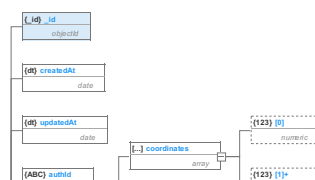
```

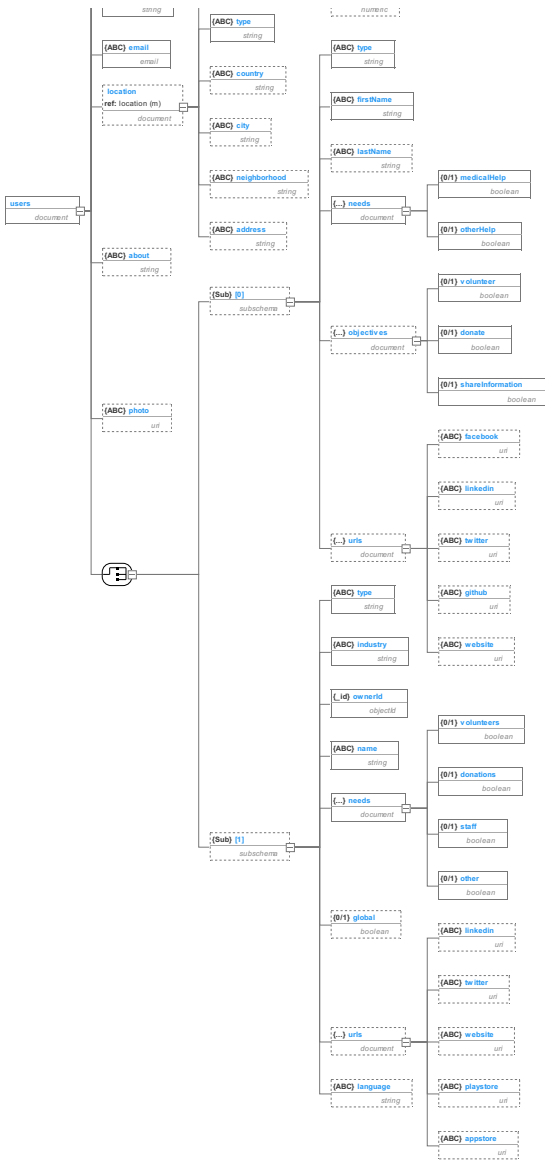
        "bsonType": "string",
        "description": "String holding the address (street, number and other details)."

```

2.1.2.4 Collection users

2.1.2.4.1 users Tree Diagram





2.1.2.4.2 users Properties

| PROPERTY | VALUE |
|-----------------------|---|
| Collection name | users |
| Technical name | |
| Id | |
| Description | <p>Collection that holds all the data on system's users.</p> <p>The users can be of two types: (1) Individual, representing a private user account; and (2) Organization, representing an organization of various types.</p> <p>The field 'type' is used to differentiate between the two subschemas, with 'individual' representing private user accounts and "Community", "Government", "Health care provider", "Non-profit", "Other", "R&D", "Startup", "Traditional Company" and "University".</p> |
| Database | applicationDB |
| Capped | false |
| Size | |
| Max | |
| Storage engine | WiredTiger |
| Config String | |
| Validation level | Off |
| Validation action | Warn |
| Additional properties | true |
| Comments | |

2.1.2.4.3 users Fields

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-------------|----------|-------|-----|--|--|
| _id | objectId | true | | Automatically generated _id as primary key. | |
| createdAt | date | true | | Timestamp with the user's creation date | Thu Apr 16 2020 04:59:24 GMT-0300 (Horário Padrão de Brasília): changed the name to registeredAt |
| updatedAt | date | true | | Timestamp indicating last update on the database. | |
| authId | string | true | | String holding the authentication service id (Auth0). This field is a foreign key to the user in the Auth0 database , connecting through the user_id field. | |
| email | email | true | | User's email validated by a regex pattern. | |
| location | document | false | | Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing. | |
| coordinates | array | true | | Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition. | |

| | | | | | |
|--------------|-----------|-------|--|---|--|
| [0] | numeric | false | | Location longitude | |
| [1] | numeric | false | | Location latitude | |
| type | string | true | | Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point". | |
| country | string | false | | String holding the country's name. | |
| city | string | false | | String holding the city's name. | |
| neighborhood | string | false | | String holding the neighborhood's name. | |
| address | string | false | | String holding the address (street, number and other details). | |
| about | string | false | | Small self-introduction, limited to 100 char | |
| photo | uri | false | | URI with the location of the profile's photo. | |
| oneOf | choice | false | | | |
| subschema | subschema | false | | Subschema representing a individual account. | |
| type | string | true | | | |
| firstName | string | true | | The first name of the user | |
| lastName | string | false | | The last name of the user. Optional field. | |
| needs | document | true | | User's need in the app, representing what kind of help he/she needs | |
| medicalHelp | boolean | true | | Flag to indicate wheter the user needs medical help. Marking this indicates he shows symptoms of COVID-19. | |
| otherHelp | boolean | true | | Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc.. | |
| objectives | document | false | | Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information. | |
| volunteer | boolean | true | | Flag to indicate wheter the user wants to volunteer for iniciatives through the app. | |
| donate | boolean | true | | Flag to indicate wheter the user wants to donate for iniciatives through the app. | |
| | | | | | |

| | | | | | |
|------------------|-----------|-------|--|--|--|
| shareInformation | boolean | true | | Flag to indicate wheter the user wants to share information through the app. | |
| urls | document | false | | Document holding the user urls to external websites and social networks | |
| facebook | uri | false | | URL to user's Facebook page. | |
| linkedin | uri | false | | URL to user's Linkedin page. | |
| twitter | uri | false | | URL to user's Twitter page. | |
| github | uri | false | | URL to user's Github page. | |
| website | uri | false | | URL to user's personal website. | |
| subschema | subschema | false | | Subschema representing an organization (colective) account. | |
| type | string | true | | The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only. | |
| industry | string | true | | Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries. | |
| ownerId | objectId | true | | _id of the user who owns the organization. | |
| name | string | true | | Organization's name. | |
| needs | document | true | | Document with what are the objectives of the organization through the application, such as finding volunteers, staff, investors and donors. | |
| volunteers | boolean | true | | Flag to indicate wheter the organization is seeking volunteers through the app. | |
| donations | boolean | true | | Flag to indicate wheter the organization is seeking donations through the app. | |
| staff | boolean | true | | Flag to indicate wheter the organization is seeking staff through the app. | |
| other | boolean | true | | Flag to indicate wheter the organization is seeking other needs through the app. | |
| global | boolean | false | | Flag to indicate wheter this organization is global. | |
| urls | document | false | | Document holding external urls and refs to the organization such as social media profiles. | |

| | | | | | |
|-----------|--------|-------|--|--|--|
| linkedin | uri | false | | URL to organization's Linkedin page. | |
| twitter | uri | false | | URL to organization's Twitter page. | |
| website | uri | false | | URL to organization's website. | |
| playstore | uri | false | | URL to organization's page at Google Play Store. | |
| appstore | uri | false | | URL to organization's page at Apple App Store | |
| language | string | false | | Field to indicate the language used by the organization. | |

2.1.2.4.3.1 Field createdAt

2.1.2.4.3.1.1 createdAt Tree Diagram

| |
|----------------|
| {dt} createdAt |
| document |

2.1.2.4.3.1.2 createdAt properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | createdAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp with the user's creation date |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | Thu Apr 16 2020 04:59:24 GMT-0300 (Horário Padrão de Brasília): changed the name to registeredAt |

2.1.2.4.3.2 Field _id

2.1.2.4.3.2.1 _id Tree Diagram

| |
|-----------|
| {_id} _id |
| document |

2.1.2.4.3.2.2 _id properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | _id |
| Technical name | |
| Id | |
| Type | objectId |
| Description | Automatically generated _id as primary key. |
| Dependencies | |
| Required | true |
| Primary key | true |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.4.3.3 Field updatedAt**2.1.2.4.3.3.1 updatedAt Tree Diagram**

| |
|-----------------------|
| {dt} updatedAt |
| <i>document</i> |

2.1.2.4.3.3.2 updatedAt properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | updatedAt |
| Technical name | |
| Id | |
| Type | date |
| Description | Timestamp indicating last update on the database. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Sample | |
| Comments | |

2.1.2.4.3.4 Field authId**2.1.2.4.3.4.1 authId Tree Diagram**

| |
|---------------------|
| {ABC} authId |
| <i>document</i> |

2.1.2.4.3.4.2 authId properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | authId |
| Technical name | |
| Id | |
| Type | string |
| Description | String holding the authentication service id (Auth0). This field is a foreign key to the user in the Auth0 database , connecting through the user_id field. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.4.3.5 Field email

2.1.2.4.3.5.1 email Tree Diagram

| |
|-------------|
| {ABC} email |
| document |

2.1.2.4.3.5.2 email properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | email |
| Technical name | |
| Id | |
| Type | string |
| Description | User's email validated by a regex pattern. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | email |
| Enum | |
| Sample | test_user@fightpandemics.com |
| Comments | |

2.1.2.4.3.6 Field location

2.1.2.4.3.6.1 location Tree Diagram

| |
|-------------------|
| location |
| ref: location (m) |
| document |

2.1.2.4.3.6.2 location properties

| PROPERTY | VALUE |
|----------------|-----------------------------|
| Name | location |
| \$ref | #model/definitions/location |
| Reference type | model |

2.1.2.4.3.7 Field about

2.1.2.4.3.7.1 about Tree Diagram

| |
|--------------------|
| {ABC} about |
| document |

2.1.2.4.3.7.2 about properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | about |
| Technical name | |
| Id | |
| Type | string |
| Description | Small self-introduction, limited to 100 char |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

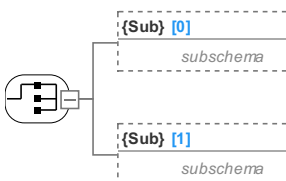
2.1.2.4.3.8 Field photo

2.1.2.4.3.8.1 photo Tree Diagram

| |
|--------------------|
| {ABC} photo |
| document |

2.1.2.4.3.8.2 photo properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | photo |
| Technical name | |
| Id | |
| Type | string |
| Description | URI with the location of the profile's photo. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | |
| Comments | |

2.1.2.4.3.9 Field oneOf**2.1.2.4.3.9.1 oneOf Tree Diagram****2.1.2.4.3.9.2 oneOf Hierarchy**Parent field: **coordinates**

Child field(s):

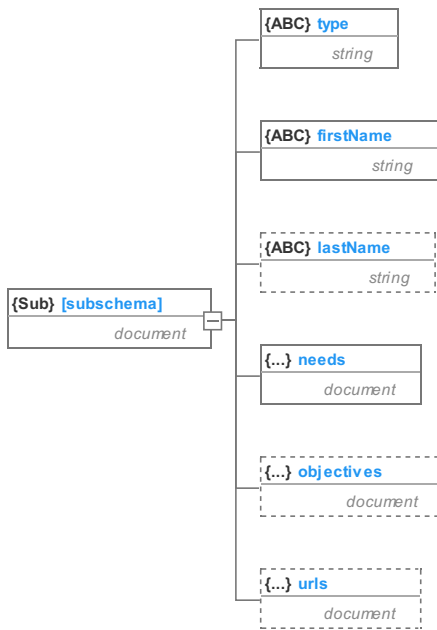
| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-----------|-----------|-------|-----|---|----------|
| subschema | subschema | false | | Subschema representing a individual account. | |
| subschema | subschema | false | | Subschema representing an organization (colective) account. | |

2.1.2.4.3.9.3 oneOf properties

| PROPERTY | VALUE |
|----------|-------|
| Choice | oneOf |

2.1.2.4.3.10 Field [1]

2.1.2.4.3.10.1 [1] Tree Diagram



2.1.2.4.3.10.2 [1] Hierarchy

Parent field: **location**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|------------|----------|-------|-----|---|----------|
| type | string | true | | | |
| firstName | string | true | | The first name of the user | |
| lastName | string | false | | The last name of the user. Optional field. | |
| needs | document | true | | User's need in the app, representing what kind of help he/she needs | |
| objectives | document | false | | Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information. | |
| urls | document | false | | Document holding the user urls to external websites and social networks | |

2.1.2.4.3.10.3 [1] properties

| PROPERTY | VALUE |
|-----------------------|--|
| Id | individual |
| Description | Subschema representing a individual account. |
| Required | |
| Min Properties | |
| Max Properties | |
| Additional properties | true |
| Comments | |

2.1.2.4.3.11 Field type

2.1.2.4.3.11.1 type Tree Diagram

| |
|------------|
| {ABC} type |
| document |

2.1.2.4.3.11.2 type properties

| PROPERTY | VALUE |
|--------------------|------------|
| Name | type |
| Technical name | |
| Id | |
| Type | string |
| Description | |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | individual |
| Sample | |
| Comments | |

2.1.2.4.3.12 Field firstName

2.1.2.4.3.12.1 firstName Tree Diagram

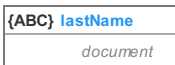
| |
|-----------------|
| {ABC} firstName |
| document |

2.1.2.4.3.12.2 firstName properties

| PROPERTY | VALUE |
|--------------------|----------------------------|
| Name | firstName |
| Technical name | |
| Id | |
| Type | string |
| Description | The first name of the user |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.4.3.13 Field lastName

2.1.2.4.3.13.1 lastName Tree Diagram

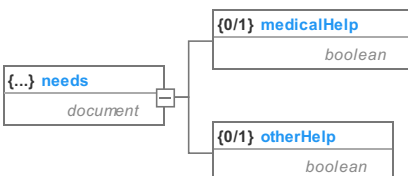


2.1.2.4.3.13.2 lastName properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | lastName |
| Technical name | |
| Id | |
| Type | string |
| Description | The last name of the user. Optional field. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.4.3.14 Field needs

2.1.2.4.3.14.1 needs Tree Diagram



2.1.2.4.3.14.2 needs Hierarchy

Parent field: **location**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-------------|---------|------|-----|--|----------|
| medicalHelp | boolean | true | | Flag to indicate wheter the user needs medical help. Marking this indicates he shows symptoms of COVID-19. | |
| otherHelp | boolean | true | | Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc.. | |

2.1.2.4.3.14.3 needs properties

| PROPERTY | VALUE |
|-----------------------|---|
| Name | needs |
| Technical name | |
| Id | |
| Type | document |
| Description | User's need in the app, representing what kind of help he/she needs |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| DBRef | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

2.1.2.4.3.15 Field medicalHelp**2.1.2.4.3.15.1 medicalHelp Tree Diagram**

| |
|--------------------------|
| {0/1} medicalHelp |
| <i>document</i> |

2.1.2.4.3.15.2 medicalHelp properties

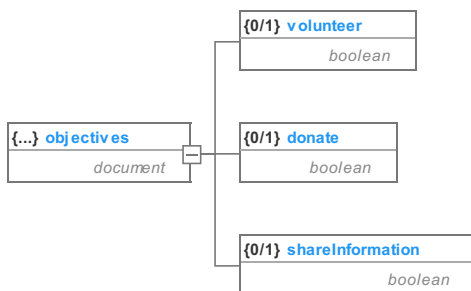
| PROPERTY | VALUE |
|--------------------|--|
| Name | medicalHelp |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the user needs medical help. Marking this indicates he shows symptoms of COVID-19. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | false |
| Comments | |

2.1.2.4.3.16 Field otherHelp**2.1.2.4.3.16.1 otherHelp Tree Diagram**

| |
|------------------------|
| {0/1} otherHelp |
| <i>document</i> |

2.1.2.4.3.16.2 otherHelp properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | otherHelp |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc.. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | false |
| Comments | |

2.1.2.4.3.17 Field objectives**2.1.2.4.3.17.1 objectives Tree Diagram****2.1.2.4.3.17.2 objectives Hierarchy**Parent field: **users**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|------------------|---------|------|-----|--|----------|
| volunteer | boolean | true | | Flag to indicate wheter the user wants to volunteer for iniciatives through the app. | |
| donate | boolean | true | | Flag to indicate wheter the user wants to donate for iniciatives through the app. | |
| shareInformation | boolean | true | | Flag to indicate wheter the user wants to share information through the app. | |

2.1.2.4.3.17.3 objectives properties

| PROPERTY | VALUE |
|-----------------------|---|
| Name | objectives |
| Technical name | |
| Id | |
| Type | document |
| Description | Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| DBRef | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

2.1.2.4.3.18 Field volunteer**2.1.2.4.3.18.1 volunteer Tree Diagram**

| |
|------------------------|
| {0/1} volunteer |
| <i>document</i> |

2.1.2.4.3.18.2 volunteer properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | volunteer |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the user wants to volunteer for iniciatives through the app. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | false |
| Comments | |

2.1.2.4.3.19 Field donate**2.1.2.4.3.19.1 donate Tree Diagram**

| |
|---------------------|
| {0/1} donate |
| <i>document</i> |

2.1.2.4.3.19.2 donate properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | donate |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the user wants to donate for iniciatives through the app. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | false |
| Comments | |

2.1.2.4.3.20 Field shareInformation**2.1.2.4.3.20.1 shareInformation Tree Diagram**

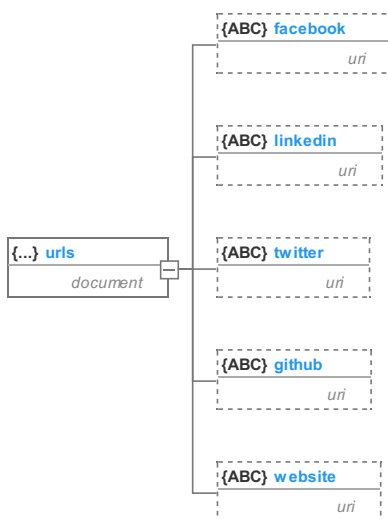
| |
|--|
| {0/1} shareInformation |
| <i>document</i> |

2.1.2.4.3.20.2 shareInformation properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | shareInformation |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the user wants to share information through the app. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | true |
| Sample | true |
| Comments | |

2.1.2.4.3.21 Field urls

2.1.2.4.3.21.1 urls Tree Diagram



2.1.2.4.3.21.2 urls Hierarchy

Parent field: **subschema**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|----------|--------|-------|-----|---------------------------------|----------|
| facebook | string | false | | URL to user's Facebook page. | |
| linkedin | string | false | | URL to user's Linkedin page. | |
| twitter | string | false | | URL to user's Twitter page. | |
| github | string | false | | URL to user's Github page. | |
| website | string | false | | URL to user's personal website. | |

2.1.2.4.3.21.3 urls properties

| PROPERTY | VALUE |
|-----------------------|---|
| Name | urls |
| Technical name | |
| Id | |
| Type | document |
| Description | Document holding the user urls to external websites and social networks |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| DBRef | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

2.1.2.4.3.22 Field facebook

2.1.2.4.3.22.1 facebook Tree Diagram

| |
|-----------------------|
| {ABC} facebook |
| <i>document</i> |

2.1.2.4.3.22.2 facebook properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | facebook |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to user's Facebook page. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://facebook.com/myProfile |
| Comments | |

2.1.2.4.3.23 Field linkedin

2.1.2.4.3.23.1 linkedin Tree Diagram

| |
|-----------------------|
| {ABC} linkedin |
| <i>document</i> |

2.1.2.4.3.23.2 linkedin properties

| PROPERTY | VALUE |
|--------------------|----------------------------------|
| Name | linkedin |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to user's Linkedin page. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://linkedin.com/in/myProfile |
| Comments | |

2.1.2.4.3.24 Field twitter**2.1.2.4.3.24.1 twitter Tree Diagram**

| |
|----------------------|
| {ABC} twitter |
| <i>document</i> |

2.1.2.4.3.24.2 twitter properties

| PROPERTY | VALUE |
|--------------------|------------------------------|
| Name | twitter |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to user's Twitter page. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://twitter.com/myProfile |
| Comments | |

2.1.2.4.3.25 Field github

2.1.2.4.3.25.1 github Tree Diagram

| |
|---------------------|
| {ABC} github |
| <i>document</i> |

2.1.2.4.3.25.2 github properties

| PROPERTY | VALUE |
|--------------------|-----------------------------|
| Name | github |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to user's Github page. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://github.com/myProfile |
| Comments | |

2.1.2.4.3.26 Field website**2.1.2.4.3.26.1 website Tree Diagram**

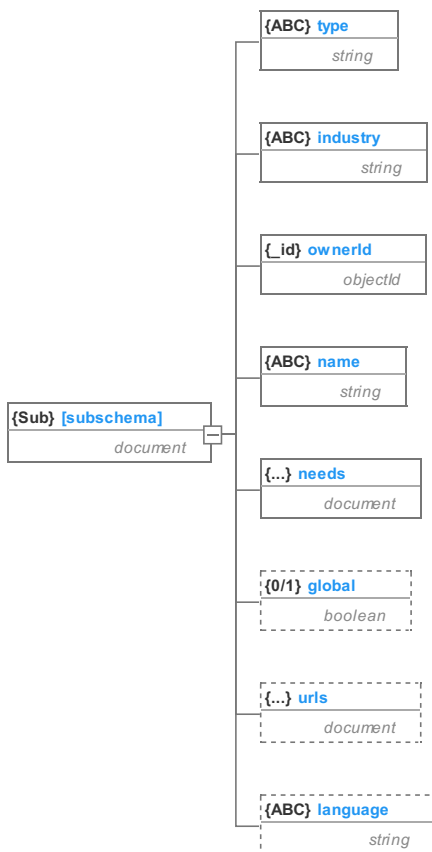
| |
|----------------------|
| {ABC} website |
| <i>document</i> |

2.1.2.4.3.26.2 website properties

| PROPERTY | VALUE |
|--------------------|---------------------------------|
| Name | website |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to user's personal website. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://mypersonalwebsite.com |
| Comments | |

2.1.2.4.3.27 Field [1]

2.1.2.4.3.27.1 [1] Tree Diagram



2.1.2.4.3.27.2 [1] Hierarchy

Parent field: **objectives**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|----------|----------|-------|-----|--|----------|
| type | string | true | | The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only. | |
| industry | string | true | | Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries. | |
| ownerId | objectId | true | fk | _id of the user who owns the organization. | |
| name | string | true | | Organization's name. | |
| needs | document | true | | Document with what are the objectives of the organization through the application, such as finding volunteers, staff, investors and donors. | |
| global | boolean | false | | Flag to indicate wheter this organization is global. | |
| urls | document | false | | Document holding external urls and refs to the organization such as social media profiles. | |
| language | string | false | | Field to inidicate the language used by the organization. | |

2.1.2.4.3.27.3 [1] properties

| PROPERTY | VALUE |
|-----------------------|--|
| Id | organization |
| Description | Subschema representing an organization (collective) account. |
| Required | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

2.1.2.4.3.28 Field type**2.1.2.4.3.28.1 type Tree Diagram**

| |
|-------------------|
| {ABC} type |
| <i>document</i> |

2.1.2.4.3.28.2 type properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | type |
| Technical name | |
| Id | |
| Type | string |
| Description | The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | startup,traditional company,community,government,r&d,non-profit,university,health care provider,other |
| Sample | |
| Comments | |

2.1.2.4.3.29 Field industry**2.1.2.4.3.29.1 industry Tree Diagram**

| |
|-----------------------|
| {ABC} industry |
| <i>document</i> |

2.1.2.4.3.29.2 industry properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | industry |
| Technical name | |
| Id | |
| Type | string |
| Description | Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.4.3.30 Field ownerId**2.1.2.4.3.30.1 ownerId Tree Diagram**

| |
|----------------------|
| {_id} ownerId |
| document |

2.1.2.4.3.30.2 ownerId properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | ownerId |
| Technical name | |
| Id | |
| Type | objectId |
| Description | _id of the user who owns the organization. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | users |
| Foreign field | _id |
| Relationship type | Foreign Key |
| Default | |
| Sample | |
| Comments | |

2.1.2.4.3.31 Field name

2.1.2.4.3.31.1 name Tree Diagram

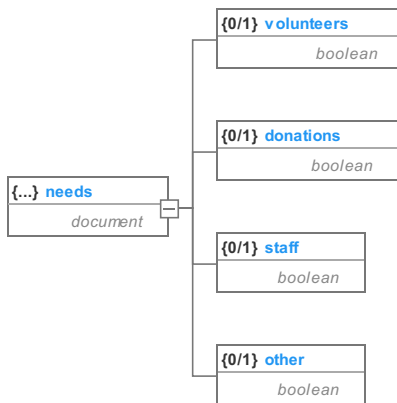
| |
|-------------------|
| {ABC} name |
| <i>document</i> |

2.1.2.4.3.31.2 name properties

| PROPERTY | VALUE |
|--------------------|----------------------|
| Name | name |
| Technical name | |
| Id | |
| Type | string |
| Description | Organization's name. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.4.3.32 Field needs

2.1.2.4.3.32.1 needs Tree Diagram



2.1.2.4.3.32.2 needs Hierarchy

Parent field: **urls**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|------------|---------|------|-----|--|----------|
| volunteers | boolean | true | | Flag to indicate wheter the organization is seeking volunteers through the app. | |
| donations | boolean | true | | Flag to indicate wheter the organization is seeking donations through the app. | |
| staff | boolean | true | | Flag to indicate wheter the organization is seeking staff through the app. | |
| other | boolean | true | | Flag to indicate wheter the organization is seeking other needs through the app. | |

2.1.2.4.3.32.3 needs properties

| PROPERTY | VALUE |
|-----------------------|---|
| Name | needs |
| Technical name | |
| Id | |
| Type | document |
| Description | Document with what are the objectives of the organization through the application, such as finding volunteers, staff, investors and donors. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| DBRef | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

2.1.2.4.3.33 Field volunteers

2.1.2.4.3.33.1 volunteers Tree Diagram

| |
|-------------------------|
| {0/1} volunteers |
| <i>document</i> |

2.1.2.4.3.33.2 volunteers properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | volunteers |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the organization is seeking volunteers through the app. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | true |
| Comments | |

2.1.2.4.3.34 Field donations**2.1.2.4.3.34.1 donations Tree Diagram**

| |
|------------------------|
| {0/1} donations |
| <i>document</i> |

2.1.2.4.3.34.2 donations properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | donations |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the organization is seeking donations through the app. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | false |
| Comments | |

2.1.2.4.3.35 Field staff**2.1.2.4.3.35.1 staff Tree Diagram**

| |
|--------------------|
| {0/1} staff |
| <i>document</i> |

2.1.2.4.3.35.2 staff properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | staff |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the organization is seeking staff through the app. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | false |
| Comments | |

2.1.2.4.3.36 Field other**2.1.2.4.3.36.1 other Tree Diagram**

| |
|-------------|
| {0/1} other |
| document |

2.1.2.4.3.36.2 other properties

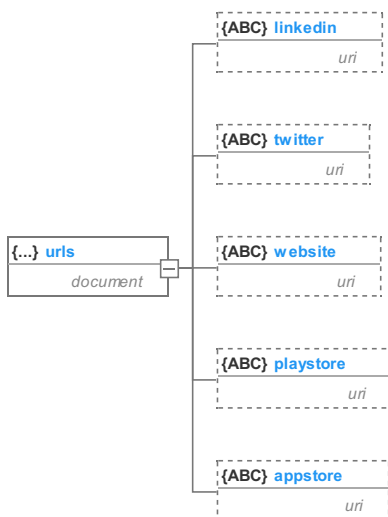
| PROPERTY | VALUE |
|--------------------|--|
| Name | other |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter the organization is seeking other needs through the app. |
| Dependencies | |
| Required | true |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | false |
| Comments | |

2.1.2.4.3.37 Field global**2.1.2.4.3.37.1 global Tree Diagram**

| |
|--------------|
| {0/1} global |
| document |

2.1.2.4.3.37.2 global properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | global |
| Technical name | |
| Id | |
| Type | boolean |
| Description | Flag to indicate wheter this organization is global. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | false |
| Sample | |
| Comments | |

2.1.2.4.3.38 Field urls**2.1.2.4.3.38.1 urls Tree Diagram****2.1.2.4.3.38.2 urls Hierarchy**Parent field: **subschema**

Child field(s):

| FIELD | TYPE | REQ | KEY | DESCRIPTION | COMMENTS |
|-----------|--------|-------|-----|--|----------|
| linkedin | string | false | | URL to organization's Linkedin page. | |
| twitter | string | false | | URL to organization's Twitter page. | |
| website | string | false | | URL to organization's website. | |
| playstore | string | false | | URL to organization's page at Google Play Store. | |
| appstore | string | false | | URL to organization's page at Apple App Store | |

2.1.2.4.3.38.3 urls properties

| PROPERTY | VALUE |
|-----------------------|--|
| Name | urls |
| Technical name | |
| Id | |
| Type | document |
| Description | Document holding external urls and refs to the organization such as social media profiles. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| DBRef | |
| Min Properties | |
| Max Properties | |
| Additional properties | false |
| Comments | |

2.1.2.4.3.39 Field linkedin**2.1.2.4.3.39.1 linkedin Tree Diagram**

| |
|-----------------------|
| {ABC} linkedin |
| document |

2.1.2.4.3.39.2 linkedin properties

| PROPERTY | VALUE |
|--------------------|--------------------------------------|
| Name | linkedin |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to organization's LinkedIn page. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://linkedin.com/in/myProfile |
| Comments | |

2.1.2.4.3.40 Field twitter**2.1.2.4.3.40.1 twitter Tree Diagram**

| |
|----------------------|
| {ABC} twitter |
| document |

2.1.2.4.3.40.2 twitter properties

| PROPERTY | VALUE |
|--------------------|-------------------------------------|
| Name | twitter |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to organization's Twitter page. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://twitter.com/myProfile |
| Comments | |

2.1.2.4.3.41 Field website**2.1.2.4.3.41.1 website Tree Diagram**

| |
|-------------------------------|
| {ABC} website |
| <i>document</i> |

2.1.2.4.3.41.2 website properties

| PROPERTY | VALUE |
|--------------------|--------------------------------|
| Name | website |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to organization's website. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://mypersonalwebsite.com |
| Comments | |

2.1.2.4.3.42 Field playstore

2.1.2.4.3.42.1 playstore Tree Diagram

| |
|------------------------|
| {ABC} playstore |
| <i>document</i> |

2.1.2.4.3.42.2 playstore properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | playstore |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to organization's page at Google Play Store. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://playstore.com/myOrg |
| Comments | |

2.1.2.4.3.43 Field appstore

2.1.2.4.3.43.1 appstore Tree Diagram

| |
|-----------------------|
| {ABC} appstore |
| <i>document</i> |

2.1.2.4.3.43.2 appstore properties

| PROPERTY | VALUE |
|--------------------|---|
| Name | appstore |
| Technical name | |
| Id | |
| Type | string |
| Description | URL to organization's page at Apple App Store |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | uri |
| Enum | |
| Sample | http://appstore.com/myOrg |
| Comments | |

2.1.2.4.3.44 Field language**2.1.2.4.3.44.1 language Tree Diagram**

| |
|-----------------------|
| {ABC} language |
| <i>document</i> |

2.1.2.4.3.44.2 language properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | language |
| Technical name | |
| Id | |
| Type | string |
| Description | Field to indicate the language used by the organization. |
| Dependencies | |
| Required | false |
| Primary key | false |
| Foreign collection | |
| Foreign field | |
| Relationship type | |
| Default | |
| Min length | |
| Max length | |
| Pattern | |
| Format | |
| Enum | |
| Sample | |
| Comments | |

2.1.2.4.4 users JSON Schema

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "users",
  "description": "Collection that holds all the data on system's users. \n\nThe users can be of **two types**: **(1) Individual**, representing a private user account; and **(2) Organization**, representing an organization of various types.\n\nThe field 'type' is used to differentiate between the two subschemas, with 'individual' representing private user accounts and 'Community', 'Government', 'Health care provider', 'Non-profit', 'Other', 'R&D', 'Startup', 'Traditional Company' and 'University'.",
  "additionalProperties": true,
  "properties": {
    "createdAt": {
      "type": "string",
      "description": "Timestamp with the user's creation date",
      "format": "date-time"
    },
    "_id": {
      "type": "string",
      "description": "Automatically generated _id as primary key.",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp indicating last update on the database.",
      "format": "date-time"
    },
    "authId": {
      "type": "string",
      "description": "String holding the authentication service id (Auth0).\n\nThis field is a **foreign key to the user in the Auth0 database**, connecting through the user_id field."
    },
    "email": {
      "type": "string",
      "description": "User's email validated by a regex pattern.",
      "format": "email"
    },
    "location": {
      "$ref": "#model/definitions/location"
    },
    "about": {
      "type": "string",
      "description": "Small self-introduction, limited to 100 char"
    },
    "photo": {
      "type": "string",
      "description": "URI with the location of the profile's photo.",
      "format": "uri"
    }
  },
  "oneOf": [
    {
      "id": "individual",
      "type": "object",
      "description": "Subschema representing a individual account.",
      "properties": {
        "type": {
          "type": "string",
          "enum": [
            "individual"
          ]
        },
        "firstName": {
          "type": "string",
          "description": "The first name of the user"
        },
        "lastName": {
          "type": "string",
          "description": "The last name of the user. Optional field."
        },
        "needs": {

```



```

    "type": "object",
    "description": "User's need in the app, representing what kind of help he/she needs",
    "properties": {
      "medicalHelp": {
        "type": "boolean",
        "description": "Flag to indicate wheter the user needs medical help. \nMarking this indicates he shows symptoms of COVID-19.",
        "default": false
      },
      "otherHelp": {
        "type": "boolean",
        "description": "Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc..",
        "default": false
      }
    },
    "additionalProperties": false,
    "required": [
      "medicalHelp",
      "otherHelp"
    ]
  },
  "objectives": {
    "type": "object",
    "description": "Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information.",
    "properties": {
      "volunteer": {
        "type": "boolean",
        "description": "Flag to indicate wheter the user wants to volunteer for iniciatives through the app.",
        "default": false
      },
      "donate": {
        "type": "boolean",
        "description": "Flag to indicate wheter the user wants to donate for iniciatives through the app.",
        "default": false
      },
      "shareInformation": {
        "type": "boolean",
        "description": "Flag to indicate wheter the user wants to share information through the app.",
        "default": true
      }
    },
    "additionalProperties": false,
    "required": [
      "volunteer",
      "donate",
      "shareInformation"
    ]
  },
  "urls": {
    "type": "object",
    "description": "Document holding the user urls to external websites and social networks",
    "properties": {
      "facebook": {
        "type": "string",
        "description": "URL to user's Facebook page.",
        "format": "uri"
      },
      "linkedin": {
        "type": "string",
        "description": "URL to user's Linkedin page.",
        "format": "uri"
      },
      "twitter": {
        "type": "string",
        "description": "URL to user's Twitter page.",
        "format": "uri"
      },
      "github": {
        "type": "string",
        "description": "URL to user's Github page.",

```

```

        "format": "uri"
      },
      "website": {
        "type": "string",
        "description": "URL to user's personal website.",
        "format": "uri"
      }
    },
    "additionalProperties": false
  }
},
"additionalProperties": true,
"required": [
  "type",
  "firstName",
  "needs"
]
},
{
  "id": "organization",
  "type": "object",
  "description": "Subschema representing an organization (colective) account.",
  "properties": {
    "type": {
      "type": "string",
      "description": "The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only.",
      "enum": [
        "startup",
        "traditional company",
        "community",
        "government",
        "r&d",
        "non-profit",
        "university",
        "health care provider",
        "other"
      ]
    },
    "industry": {
      "type": "string",
      "description": "Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries."
    },
    "ownerId": {
      "type": "string",
      "description": "_id of the user who owns the organization.",
      "pattern": "^[a-zA-F0-9]{24}$"
    },
    "name": {
      "type": "string",
      "description": "Organization's name.\n"
    },
    "needs": {
      "type": "object",
      "description": "Document with what are the objectives of the organization through the application, such as finding volunteers, staff, investors and donors.",
      "properties": {
        "volunteers": {
          "type": "boolean",
          "description": "Flag to indicate wheter the organization is seeking volunteers through the app.",
          "default": false
        },
        "donations": {
          "type": "boolean",
          "description": "Flag to indicate wheter the organization is seeking donations through the app.",
          "default": false
        },
        "staff": {
          "type": "boolean",
          "description": "Flag to indicate wheter the organization is seeking staff through the app."
        }
      }
    }
  }
}

```

```

        "default": false
    },
    "other": {
        "type": "boolean",
        "description": "Flag to indicate wheter the organization is seeking other needs through the app.",
        "default": false
    }
},
"additionalProperties": false,
"required": [
    "volunteers",
    "donations",
    "staff",
    "other"
]
},
"global": {
    "type": "boolean",
    "description": "Flag to indicate wheter this organization is global.",
    "default": false
},
"urls": {
    "type": "object",
    "description": "Document holding external urls and refs to the organization such as social media profiles.",
    "properties": {
        "linkedin": {
            "type": "string",
            "description": "URL to organization's Linkedin page.",
            "format": "uri"
        },
        "twitter": {
            "type": "string",
            "description": "URL to organization's Twitter page.",
            "format": "uri"
        },
        "website": {
            "type": "string",
            "description": "URL to organization's website.",
            "format": "uri"
        },
        "playstore": {
            "type": "string",
            "description": "URL to organization's page at Google Play Store.",
            "format": "uri"
        },
        "appstore": {
            "type": "string",
            "description": "URL to organization's page at Apple App Store",
            "format": "uri"
        }
    }
},
"additionalProperties": false
},
"language": {
    "type": "string",
    "description": "Field to inidicate the language used by the organization."
}
},
"additionalProperties": false,
"required": [
    "type",
    "industry",
    "ownerId",
    "name",
    "needs"
]
}
],
"required": [
    "createdAt",

```

```

    "_id",
    "updatedAt",
    "authId",
    "email"
  ]
}
```

2.1.2.4.5 users JSON data

```

{
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "authId": "Lorem",
  "email": "test_user@fightpandemics.com",
  "location": {
    "coordinates": [
      35.26,
      -32.83
    ],
    "type": "Point",
    "country": "Lorem",
    "city": "Lorem",
    "neighborhood": "Lorem",
    "address": "Lorem"
  },
  "about": "Lorem",
  "photo": "./resource.txt#frag01",
  "type": "other",
  "industry": "Lorem",
  "ownerId": ObjectId("507f1f77bcf86cd799439011"),
  "name": "Lorem",
  "needs": {
    "volunteers": true,
    "donations": false,
    "staff": false,
    "other": false
  },
  "global": false,
  "urls": {
    "linkedin": "http://linkedin.com/in/myProfile",
    "twitter": "http://twitter.com/myProfile",
    "website": "http://mypersonalwebsite.com",
    "playstore": "http://playstore.com/myOrg",
    "appstore": "http://appstore.com/myOrg"
  },
  "language": "Lorem"
}
```

2.1.2.4.6 users Target Script

```

use applicationDB;

db.createCollection( "users",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
```

```

"title": "users",
"description": "Collection that holds all the data on system's users. \n\nThe users can be of **two types**: **(1) Individual**, representing a private user account; and **(2) Organization**, representing an organization of various types.\n\nThe field 'type' is used to differentiate between the two subschemas, with 'individual' representing private user accounts and 'Community', 'Government', 'Health care provider', 'Non-profit', 'Other', 'R&D', 'Startup', 'Traditional Company' and 'University'.",
"additionalProperties": true,
"properties": {
  "_id": {
    "bsonType": "objectId",
    "description": "Automatically generated _id as primary key."
  },
  "createdAt": {
    "bsonType": "date",
    "description": "Timestamp with the user's creation date"
  },
  "updatedAt": {
    "bsonType": "date",
    "description": "Timestamp indicating last update on the database."
  },
  "authId": {
    "bsonType": "string",
    "description": "String holding the authentication service id (Auth0).\n\nThis field is a **foreign key to the user in the Auth0 database**, connecting through the user_id field."
  },
  "email": {
    "bsonType": "string",
    "description": "User's email validated by a regex pattern."
  },
  "location": {
    "bsonType": "object",
    "description": "Location document according to the norms of a [GeoJSON Object] (https://docs.mongodb.com/manual/reference/geojson/ ).\n\nThe fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.",
    "properties": {
      "coordinates": {
        "bsonType": "array",
        "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON] (https://docs.mongodb.com/manual/reference/geojson/) definition.",
        "additionalItems": true,
        "minItems": 2,
        "maxItems": 2,
        "uniqueItems": false,
        "items": [
          {
            "bsonType": "number",
            "description": "Location longitude\n",
            "minimum": -180,
            "maximum": 180
          },
          {
            "bsonType": "number",
            "description": "Location latitude\n",
            "minimum": -90,
            "maximum": 90
          }
        ]
      },
      "type": {
        "bsonType": "string",
        "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation] (https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'**.**",
        "enum": [
          "Point"
        ]
      },
      "country": {
        "bsonType": "string",
        "description": "String holding the country's name.\n"
      },
      "city": {

```

```

        "bsonType": "string",
        "description": "String holding the city's name."
    },
    "neighborhood": {
        "bsonType": "string",
        "description": "String holding the neighborhood's name."
    },
    "address": {
        "bsonType": "string",
        "description": "String holding the address (street, number and other details)."
    }
},
"additionalProperties": false,
"required": [
    "coordinates",
    "type"
]
},
"about": {
    "bsonType": "string",
    "description": "Small self-introduction, limited to 100 char"
},
"photo": {
    "bsonType": "string",
    "description": "URI with the location of the profile's photo."
}
},
"oneOf": [
    {
        "bsonType": "object",
        "description": "Subschema representing a individual account.",
        "properties": {
            "type": {
                "bsonType": "string",
                "enum": [
                    "individual"
                ]
            },
            "firstName": {
                "bsonType": "string",
                "description": "The first name of the user"
            },
            "lastName": {
                "bsonType": "string",
                "description": "The last name of the user. Optional field."
            },
            "needs": {
                "bsonType": "object",
                "description": "User's need in the app, representing what kind of help he/she needs",
                "properties": {
                    "medicalHelp": {
                        "bsonType": "bool",
                        "description": "Flag to indicate wheter the user needs medical help. \nMarking this indicates he shows symptoms of COVID-19."
                    },
                    "otherHelp": {
                        "bsonType": "bool",
                        "description": "Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc.."
                    }
                }
            },
            "additionalProperties": false,
            "required": [
                "medicalHelp",
                "otherHelp"
            ]
        }
    },
    {
        "bsonType": "object",
        "description": "Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information."
    }
]

```

```

    "properties": {
      "volunteer": {
        "bsonType": "bool",
        "description": "Flag to indicate wheter the user wants to volunteer for iniciatives through the app."
      },
      "donate": {
        "bsonType": "bool",
        "description": "Flag to indicate wheter the user wants to donate for iniciatives through the app."
      },
      "shareInformation": {
        "bsonType": "bool",
        "description": "Flag to indicate wheter the user wants to share information through the app."
      }
    },
    "additionalProperties": false,
    "required": [
      "volunteer",
      "donate",
      "shareInformation"
    ]
  },
  "urls": {
    "bsonType": "object",
    "description": "Document holding the user urls to external websites and social networks",
    "properties": {
      "facebook": {
        "bsonType": "string",
        "description": "URL to user's Facebook page."
      },
      "linkedin": {
        "bsonType": "string",
        "description": "URL to user's Linkedin page."
      },
      "twitter": {
        "bsonType": "string",
        "description": "URL to user's Twitter page."
      },
      "github": {
        "bsonType": "string",
        "description": "URL to user's Github page."
      },
      "website": {
        "bsonType": "string",
        "description": "URL to user's personal website."
      }
    }
  },
  "additionalProperties": false
}
},
"additionalProperties": true,
"required": [
  "type",
  "firstName",
  "needs"
]
},
{
  "bsonType": "object",
  "description": "Subschema representing an organization (collective) account.",
  "properties": {
    "type": {
      "bsonType": "string",
      "description": "The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only.",
      "enum": [
        "startup",
        "traditional company",
        "community",
        "government",
        "r&d",

```

```

        "non-profit",
        "university",
        "health care provider",
        "other"
    ]
},
"industry": {
    "bsonType": "string",
    "description": "Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries."
},
"ownerId": {
    "bsonType": "objectId",
    "description": "._id of the user who owns the organization."
},
"name": {
    "bsonType": "string",
    "description": "Organization's name.\n"
},
"needs": {
    "bsonType": "object",
    "description": "Document with what are the objectives of the organization through the application, such as finding volunteers,
staff, investors and donors.",
    "properties": {
        "volunteers": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the organization is seeking volunteers through the app."
        },
        "donations": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the organization is seeking donations through the app."
        },
        "staff": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the organization is seeking staff through the app."
        },
        "other": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the organization is seeking other needs through the app."
        }
    }
},
"additionalProperties": false,
"required": [
    "volunteers",
    "donations",
    "staff",
    "other"
]
},
"global": {
    "bsonType": "bool",
    "description": "Flag to indicate wheter this organization is global."
},
"urls": {
    "bsonType": "object",
    "description": "Document holding external urls and refs to the organization such as social media profiles.",
    "properties": {
        "linkedin": {
            "bsonType": "string",
            "description": "URL to organization's Linkedin page."
        },
        "twitter": {
            "bsonType": "string",
            "description": "URL to organization's Twitter page."
        },
        "website": {
            "bsonType": "string",
            "description": "URL to organization's website."
        },
        "playstore": {
            "bsonType": "string",

```

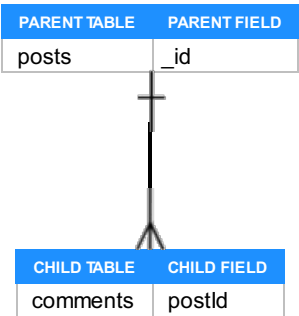


```
        "description": "URL to organization's page at Google Play Store."
      },
      "appstore": {
        "bsonType": "string",
        "description": "URL to organization's page at Apple App Store"
      }
    },
    "additionalProperties": false
  },
  "language": {
    "bsonType": "string",
    "description": "Field to indicate the language used by the organization."
  }
},
"additionalProperties": false,
"required": [
  "type",
  "industry",
  "ownerId",
  "name",
  "needs"
]
}
],
"required": [
  "createdAt",
  "_id",
  "updatedAt",
  "authId",
  "email"
]
}
},
"validationLevel": "off",
"validationAction": "warn"
});
```

3. RELATIONSHIPS

3.1 Relationship PostHasComments

3.1.1 PostHasComments Diagram

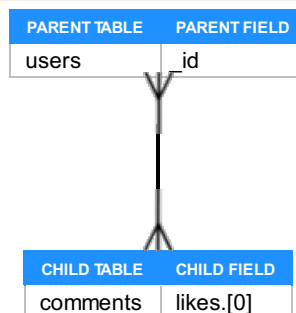


3.1.2 PostHasComments Properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | PostHasComments |
| Description | Relationship between a Post and it's respective Comments made at it. |
| Parent Collection | posts |
| Parent field | _id |
| Parent Cardinality | 1 |
| Child Collection | comments |
| Child field | postId |
| Child Cardinality | n |
| Comments | |

3.2 Relationship UserLikesComment

3.2.1 UserLikesComment Diagram

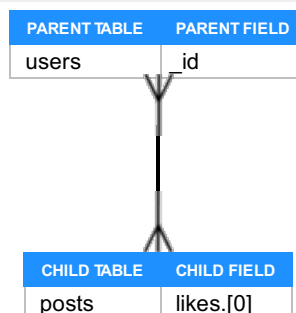


3.2.2 UserLikesComment Properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | UserLikesComment |
| Description | Relation to represent the User's who liked a Comment. Even though this is a N-N relation, we opt to store the ids only at the Comment side as an array of _id (with the _id of the users who liked it). |
| Parent Collection | users |
| Parent field | _id |
| Parent Cardinality | n |
| Child Collection | comments |
| Child field | |
| Child Cardinality | n |
| Comments | |

3.3 Relationship UserLikesPost

3.3.1 UserLikesPost Diagram

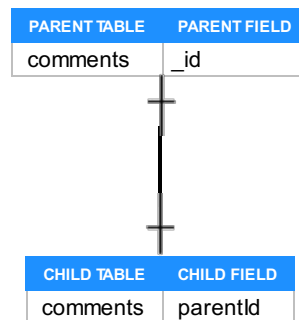


3.3.2 UserLikesPost Properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | UserLikesPost |
| Description | Relation to represent the User's who liked a Post. Even though this is a N-N relation, we opt to store the ids only at the Post side as an array of <code>_id</code> (with the <code>_id</code> of the users who liked it). |
| Parent Collection | users |
| Parent field | <code>_id</code> |
| Parent Cardinality | n |
| Child Collection | posts |
| Child field | |
| Child Cardinality | n |
| Comments | |

3.4 Relationship fk comments._id to comments.parentId

3.4.1 fk comments._id to comments.parentId Diagram

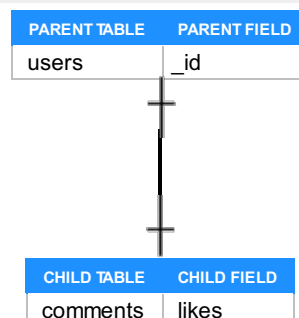


3.4.2 fk comments._id to comments.parentId Properties

| PROPERTY | VALUE |
|--------------------|--------------------------------------|
| Name | fk comments._id to comments.parentId |
| Description | |
| Parent Collection | comments |
| Parent field | <code>_id</code> |
| Parent Cardinality | 1 |
| Child Collection | comments |
| Child field | <code>parentId</code> |
| Child Cardinality | 1 |
| Comments | |

3.5 Relationship fk users._id to comments.likes

3.5.1 fk users._id to comments.likes Diagram

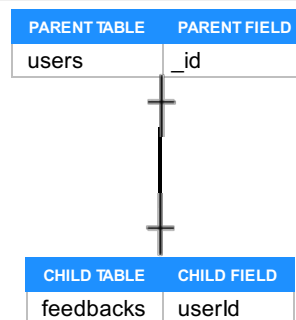


3.5.2 fk users._id to comments.likes Properties

| PROPERTY | VALUE |
|--------------------|--------------------------------|
| Name | fk users._id to comments.likes |
| Description | |
| Parent Collection | users |
| Parent field | _id |
| Parent Cardinality | 1 |
| Child Collection | comments |
| Child field | likes |
| Child Cardinality | 1 |
| Comments | |

3.6 Relationship fk users._id to feedbacks.userId

3.6.1 fk users._id to feedbacks.userId Diagram

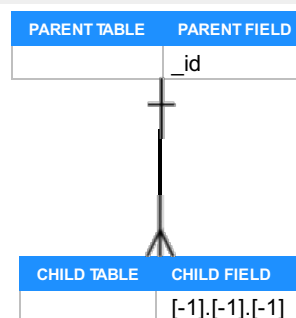


3.6.2 fk users._id to feedbacks.userId Properties

| PROPERTY | VALUE |
|--------------------|----------------------------------|
| Name | fk users._id to feedbacks.userId |
| Description | |
| Parent Collection | users |
| Parent field | _id |
| Parent Cardinality | 1 |
| Child Collection | feedbacks |
| Child field | userId |
| Child Cardinality | 1 |
| Comments | |

3.7 Relationship fk users._id to organizations..organization.ownerId

3.7.1 fk users._id to organizations..organization.ownerId Diagram

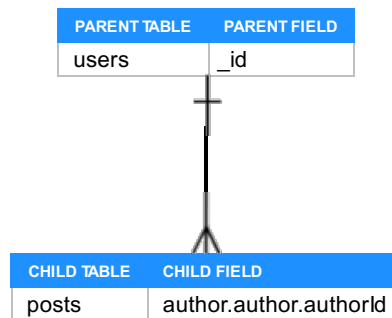


3.7.2 fk users._id to organizations..organization.ownerId Properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | fk users._id to organizations..organization.ownerId |
| Description | Relation between an User and an Organization created by him/her. Initially projected as 1-N, with only one User owning the Organization, it is though to later on be evolved into N-N, with many Users being able to manage the Organization with different roles. |
| Parent Collection | |
| Parent field | _id |
| Parent Cardinality | 1 |
| Child Collection | |
| Child field | |
| Child Cardinality | n |
| Comments | |

3.8 Relationship fk users._id to posts..author.authorId

3.8.1 fk users._id to posts..author.authorId Diagram



3.8.2 fk users._id to posts..author.authorId Properties

| PROPERTY | VALUE |
|--------------------|--|
| Name | fk users._id to posts..author.authorId |
| Description | Relation between a User and the Post created by him/her. |
| Parent Collection | users |
| Parent field | _id |
| Parent Cardinality | 1 |
| Child Collection | posts |
| Child field | |
| Child Cardinality | n |
| Comments | |