

# GeoStream - Exploiting User-Generated Geospatial Content Streams

---

FP7-SME-2012-315631

## UGCS authoring environments

### D4.1

Deliverable lead contractor: ATHENA

#### **Main contributors:**

George Papatheodorou	papatheodorou@imis.athena-innovation.gr
Nikolaos Papadakis	npapadakis@imis.athena-innovation.gr
Dimitris Skoutas	dskoutas@imis.athena-innovation.gr
George Lamprianidis	glampr@imis.athena-innovation.gr

*Due data: 31.07.2014*

---

#### Abstract

This deliverable describes the authoring tools that have been designed and developed for GeoStream. These tools are Web-based and the main idea is to combine map interfaces with word processing capabilities. In addition to the Web-based user interface, an API is also provided to facilitate reuse and integration with other applications.

---

*Copyright © 2013-2014 GeoStream – <http://www.geocontentstream.eu>*

Research Center "ATHENA," Greece  
FU BERLIN, Germany  
Fraunhofer, Germany  
Michael Mueller Verlag, Germany  
TALENT, Greece  
WIGEOGIS, Austria

## Table of Contents

<b>1.INTRODUCTION.....</b>	<b>4</b>
<b>2.RICH AUTHORIZING ENVIRONMENT.....</b>	<b>5</b>
2.1.MAIN CONCEPT.....	5
2.2.USER AUTHENTICATION.....	5
2.3.SEARCHING FOR POIS.....	6
2.4.CREATING TRIPS.....	8
<b>3.LIGHT AUTHORIZING ENVIRONMENT.....</b>	<b>16</b>
3.1.MAIN CONCEPT.....	16
3.2.EXTRACTING GEOSPATIAL ENTITIES FROM TEXT.....	16
<b>4.GEOCODING ENGINE.....</b>	<b>19</b>
<b>5.AUTHORIZING TOOLS API.....</b>	<b>21</b>
<b>6.CONCLUSIONS AND NEXT STEPS.....</b>	<b>24</b>
<b>APPENDIX.....</b>	<b>25</b>
<b>1.ENTITY DIAGRAM.....</b>	<b>25</b>
<b>2.EXAMPLE OF API CALL AND RESPONSE.....</b>	<b>25</b>

## Table of Figures

Figure 1: Login & sign up.....	6
Figure 2: The search interface.....	6
Figure 3: Search and results.....	7
Figure 4: Search: Item details.....	7
Figure 5: Search: link to original source.....	8
Figure 6: Creating trips from search results.....	8
Figure 7: Displaying a trip.....	9
Figure 8: Trip managment.....	9
Figure 9: Adding a trip.....	10
Figure 10: Displaying a trip.....	10
Figure 11: Trip overview.....	11
Figure 12: Modifying the default route of a trip.....	11
Figure 13: Changing the order of visits.....	12
Figure 14: Trip types, title and comments view.....	12
Figure 15: Updated trip view.....	13
Figure 16: "My visits" view.....	13
Figure 17: Editing a visit.....	14
Figure 18: Update a visit.....	14
Figure 19: Listing trips.....	15
Figure 20: Visits list.....	15
Figure 21: Navigating to the light authoring tool.....	16
Figure 22: Light authoring environment.....	17
Figure 23: Discovered locations as links.....	18
Figure 24: Editing the route.....	19
Figure 25: Processing workflow of the geocoding engine.....	20
Figure 26: Entity diagram of the main concepts Visits, Trips and POIs.....	25

## 1. Introduction

The main work conducted in the GeoStream project so far, in particular in WPs 1 and 2, as well as the ongoing work in WP3, have focused on how to collect, integrate, process and analyse user-generated geospatial content from various popular Web sources, which is widely recognised as a challenging and valuable task. Another important aspect of this process, which however receives usually much less attention, is to explore and address the opposite direction: how to enable and facilitate users to generate such content in the first place. The latter is the focus of the work presented in this deliverable.

In particular, we are interested in two main directions. The first is to provide a “rich” authoring tool, that is a Web-based application that combines a map interface and word processing capabilities to allow users to create and edit structured spatio-textual content in an easy way. This requires some modest effort from the user to edit the content, e.g. by filling forms and use drag-and-drop actions, but the created content is richer in terms of structure and semantics and hence can be more easily and accurately organized, search and exploited further.

The second direction is a “light” authoring tool, where the basic idea is to minimize, or completely eliminate, the user's involvement, by starting with a text document that is already available and trying to automatically identify locations described in it. Again, this is combined with a map interface, where automatically computed results are presented to the user in the form of suggestions for validation or selection. An essential part of this tool is a geocoding service.

Eventually, we combine both modules in an integrated Web application, so that users have a single point of access for content authoring, and can select, based on their needs and use cases, which steps of the process to perform. In addition to the described Web interface, we provide an API that can be used to search, create and edit content from other applications, so that the implemented functionalities can be extended or customized to fit the exact needs of the SME partners.

An evaluation of this work is being conducted in parallel and the results will be presented in Deliverable D4.2. Moreover, the developed tools that are described in this document have been integrated in the online demo presented in Deliverable D7.4.

The rest of this deliverable is structured as follows. Sections 2 and 3 describe, respectively, the rich and light authoring tools. Section 4 focuses on the geocoding service. Section 5 describes the API. Finally, Section 6 concludes the deliverable with a summary of the work and next steps.

## 2. Rich Authoring Environment

### 2.1. Main concept

In GeoStream, as well as in many other similar applications or research work in the literature, the main functionalities and services evolve most often around the notion of Points of Interest (POIs), such as monuments, restaurants, shops, train stations, museums, parks, etc. Numerous lists of such POIs exist, both free and commercial. However, these typically include some basic, “objective” (in the sense of not being user-dependent) information about a POI, such as its location, name, category, creation date, etc. What can add valuable information in these descriptions is augmenting them with content denoting user perspectives, experiences and opinions about these POIs, such as ratings, comments, recommendations about when to visit or how to get there, etc.

Our goal in designing and implementing the rich authoring environment was to allow users to augment POI descriptions with their own information. For this purpose, we use the concept *visit* to represent an augmented description of a POI from the perspective of a particular user. Furthermore, POIs may be related and appear together in a collection under some criterion, e.g. the POIs that a traveler visited during her trip. To model this aspect, we use the concept of *trip*, which constitutes a (potentially ordered) set of *visits* to particular POIs.

Consequently, the data model for the rich authoring environment comprises four main entities: *users*, *POIs*, *visits*, and *trips*. The corresponding entity-relationship diagram is shown in Appendix 1. POI descriptions are common and available to all users, while visits and trips are user-defined.

The authoring tool is a Web application developed using Ruby on Rails. In the following, we describe how the authoring is done, following a step-by-step approach and using screenshots to better illustrate each step of the process.

Note that the aforementioned concepts of visits and trips, and the description and examples in this document, are selected to suit a travel guide application, as this constitutes a major scenario for the partners in the GeoStream consortium. Nevertheless, the design and functionalities of the tools are not limited to this particular use case but can be adapted in a straightforward way to similar applications that manage collections of user-augmented location descriptions.

### 2.2. User authentication

As described above, visits and trips are user-defined; hence, the first step for a user before starting authoring is to login to her account. This is handled via a typical login screen, which provides also options for registering and changing/retrieving the password, as illustrated in Figure 1.

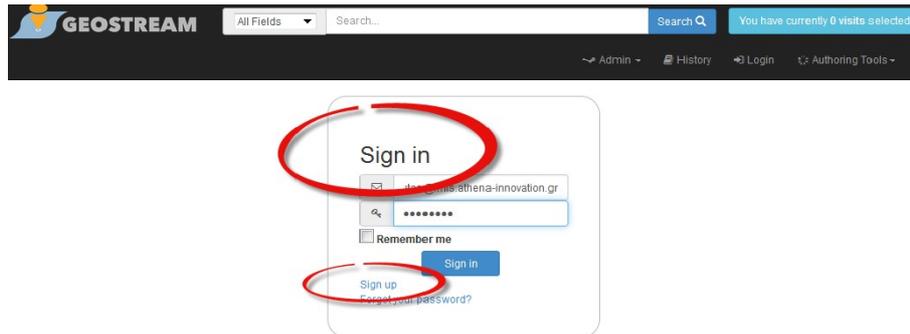


Figure 1: Login &amp; sign up

## 2.3. Searching for POIs

The next part is an iterative process, where the user searches for POIs, adds them as visits in an existing or new trip, and provides information about them. We describe these steps in more detail below.

Through the search and browsing interface, the user can perform keyword search to retrieve POIs collected from various Web sources and stored in the GeoStream database. To further filter the results, a set of facets is also provided, from which the user can select to narrow down the search.

Figure 2 illustrates a keyword search for “Acropolis”. The results corresponding to the query term “Acropolis” are retrieved and displayed in a list as well as on the map.

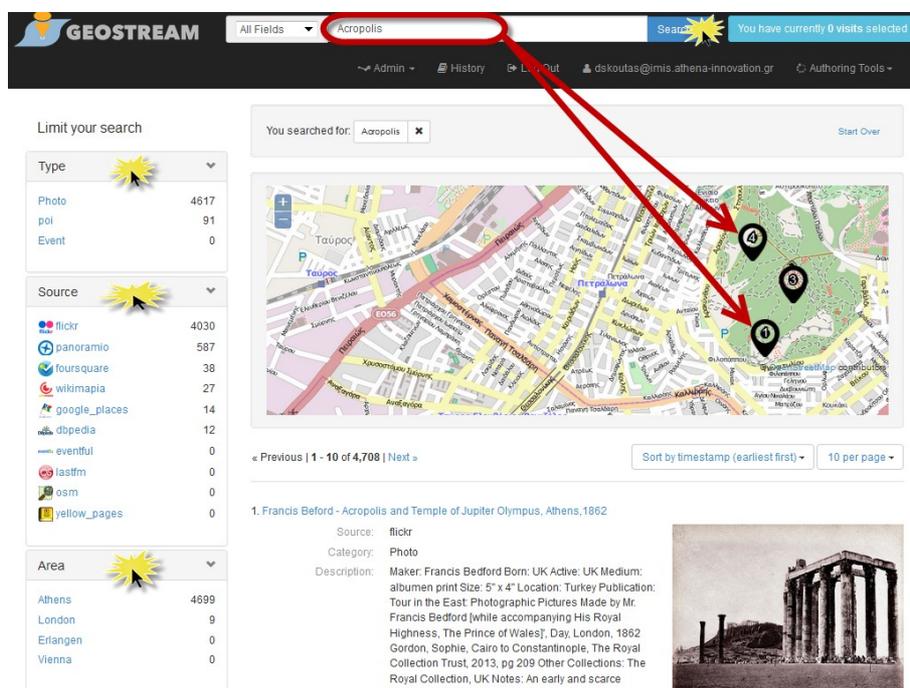


Figure 2: The search interface

From the left panel, the user can select more filtering options for the term “Acropolis”. For example, by selecting Type = “poi”, the search results include only POIs (as opposed to photos or events) that contain the term “Acropolis” (see Figure 3).

By clicking on a search result, e.g. "Hilton Athens", the details of this particular item are displayed. These include also a hyperlink to navigate to the original source (e.g. Foursquare) from which this particular item was collected (see Figures 4 and 5).

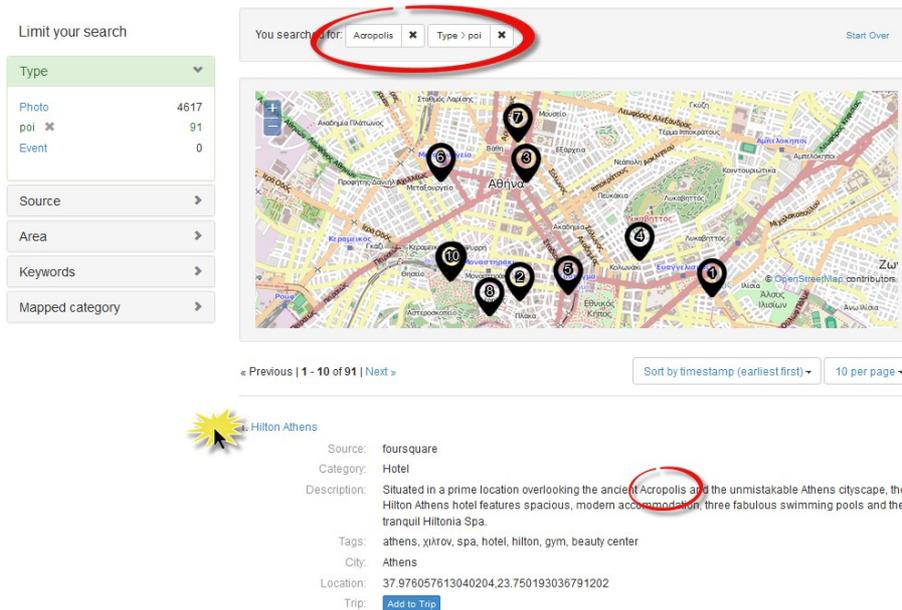


Figure 3: Search and results

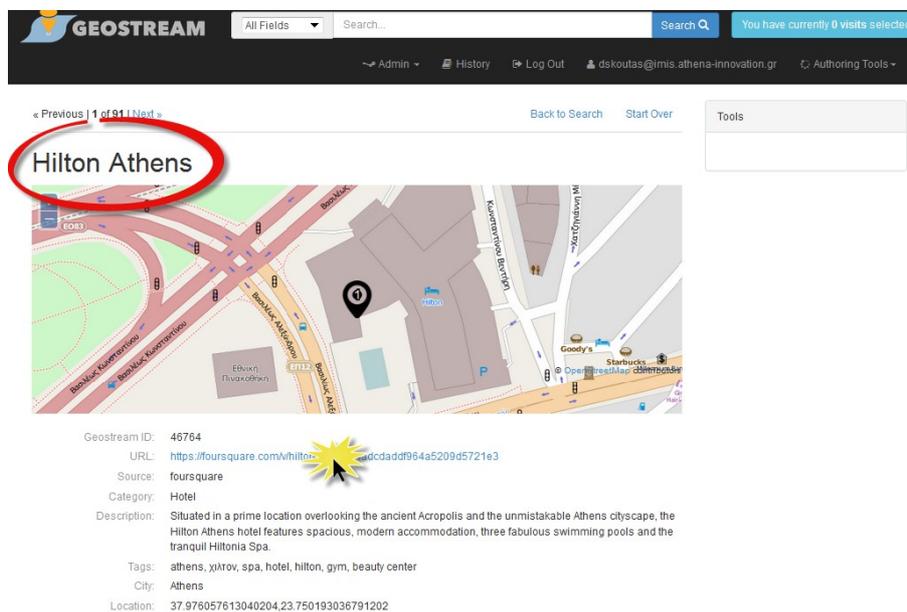


Figure 4: Search: Item details

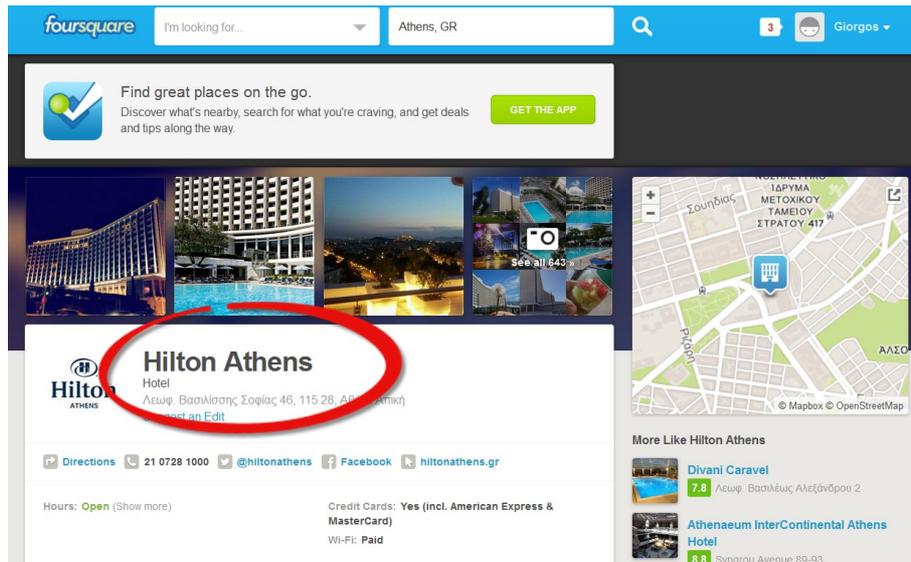


Figure 5: Search: link to original source

## 2.4. Creating trips

We next describe how to create a trip from search results. As an example, we show the steps for creating a trip based on locations of different kinds of music venues and then we show how to add or modify information of the trip and the locations visited.

In the search and browsing interface, searching for “rock music” (including quotation marks), yields a set of results, as shown in Figure 6. The user can select any of these results and click the “Add to Trip” button.

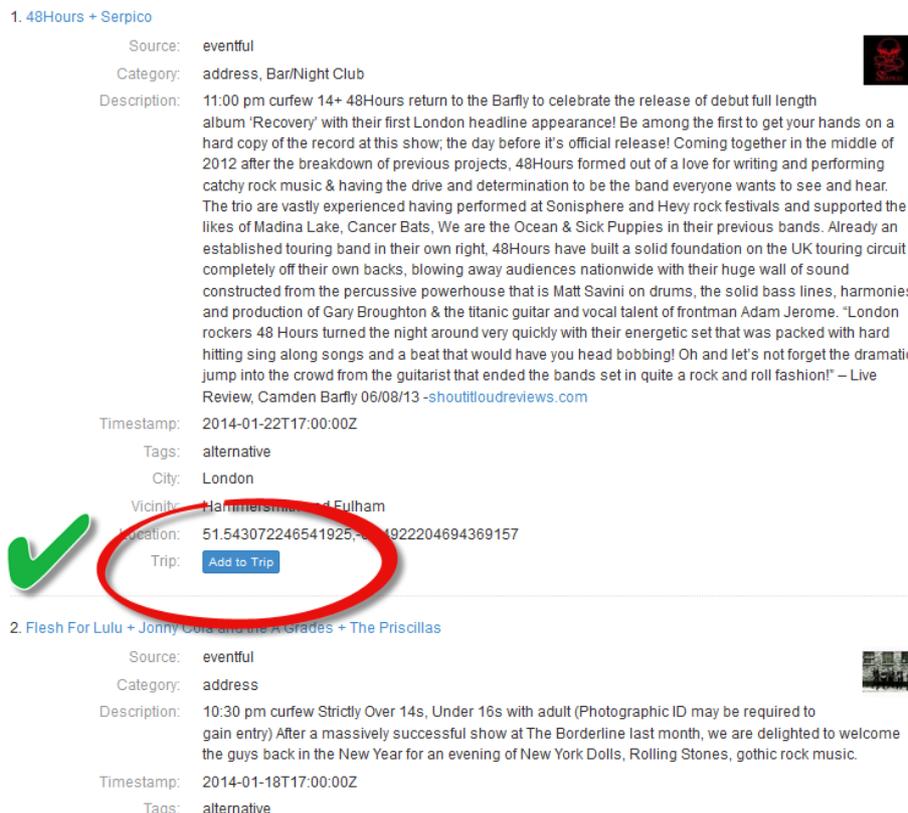


Figure 6: Creating trips from search results

A new trip is automatically created, containing a new visit that refers to the selected search result. This can be verified by observing that in section “My Visits” the selected visit is displayed. Also, the map displays the location of the visit and the message “You have 1 visits in your trip” is shown (see Figures 7 and 8).

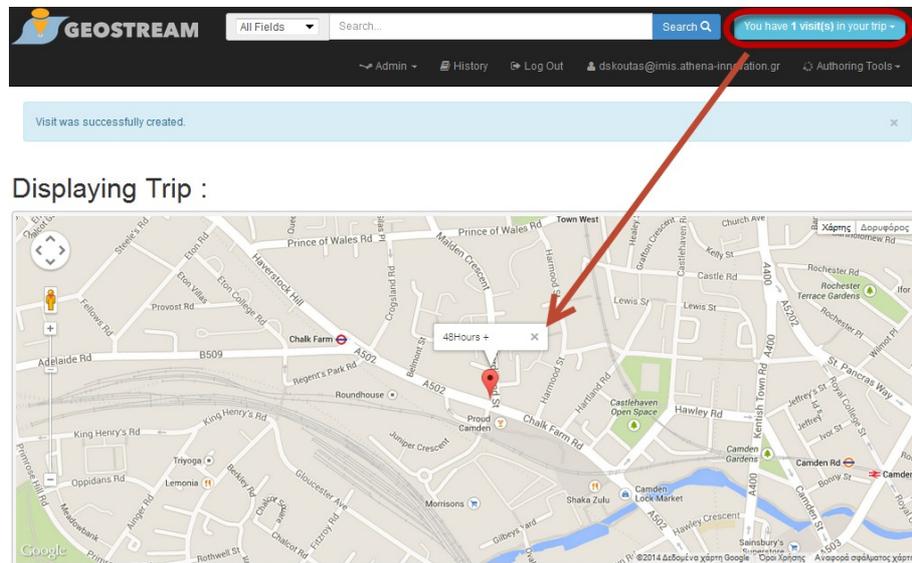


Figure 7: Displaying a trip

**Trip Details**

Title  
 Comment  
 Rating ★★☆☆☆  
 Type  
 Time spent  
 Transportation  
 Total price(€) 0

Click “Edit” to manage your Trip

**My Visits**

No	Image	Title	Category	Comment	Rating	Price(€)	Arrival	Departure
1		48Hours + Serpico	address,BarNight Club		★★★☆☆	0		

Figure 8: Trip management

We continue the example by repeating the search with the term “Mozart” and clicking “Add to trip” on the result “Vienna Mozart Concerts” (see Figure 9).

Now that the trip contains two visits, the user can see a default route connecting them, as illustrated in Figure 10. On the right of the map there is also a panel providing directions. The entries are clickable and give more detailed information, such as which turn to take, on which road, and for how many kilometers. The user can repeat the process to add more visits to her trip.

1. Midori Komachi (violin) and Ian Brown (piano) play Mozart, Walton, Brahms, Ravel

Source: eventful  
 Category: address  
 Timestamp: 2014-10-21T16:00:00Z  
 Tags: concert-diary.com  
 City: London  
 Vicinity: Greater London  
 Location: 51.501,-0.124  
 Trip: [Add to Trip](#)

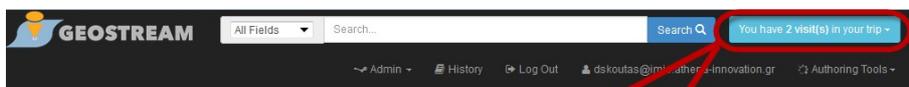


2. Vienna Mozart Concerts

Source: eventful  
 Category: address  
 Description: **The Vienna Mozart Orchestra performs with internationally renowned vocalists and instrumentalists in Vienna's most extraordinary concert halls: the Vienna State Opera, Musikverein Golden Hall, Hofburg (Imperial Palace) and the Vienna Konzerthaus. As the name Vienna Mozart Orchestra implies, the 30 musicians of this ensemble have devoted themselves to the work of Wolfgang Amadeus Mozart, the most accomplished representative of the period who, although born in Salzburg, found his home in Vienna. The Vienna Mozart Orchestra plays in magnificent historical costumes and wigs to create a special atmosphere. In keeping with the tradition of the 'musical academies', as Vienna concerts were known in Mozart's time, the audience enjoys single movements from symphonies and solo concertos, as well as operatic overtures, arias and duets from especially popular and well known works.**  
 Timestamp: 2014-07-09T17:15:00Z  
 Tags: classical  
 City: Vienna  
 Vicinity: Wien  
 Location: 48.2033952,11.3685573  
 Trip: [Add to Trip](#)



Figure 9: Adding a trip



Visit was successfully created.

Displaying Trip :

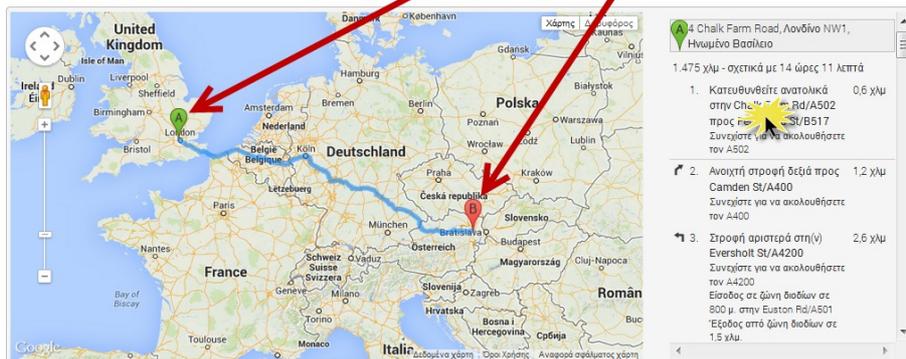


Figure 10: Displaying a trip

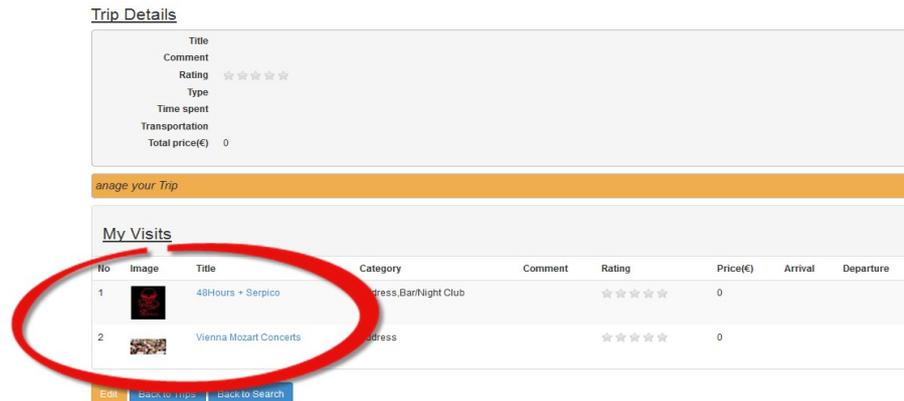


Figure 11: Trip overview

If the user wishes to add to or modify an existing trip, she can return to the trip details screen and click the “Edit” button for the desired trip on the bottom of page, as depicted in Figure 11.

Once in edit mode, the user can modify the trip details. She can also modify the default route of the trip, e.g. to indicate that the path should include Hamburg as an intermediate point. This is done via drag-and-drop functionality directly on the path displayed on the map. A “reset map” button is provided to undo any modifications (see Figure 12).

#### Editing route

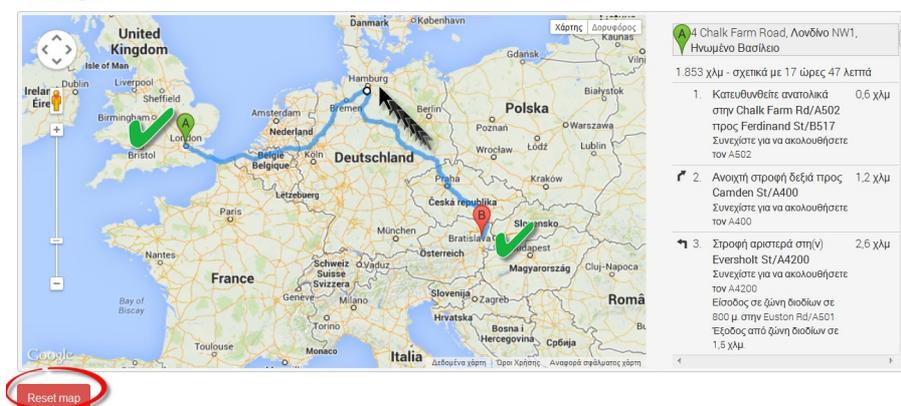


Figure 12: Modifying the default route of a trip

Visits can be added to a trip in random order. Then, the user can re-arrange them to determine the correct order by using drag-and-drop to move each visit further up or down the list (see Figure 13).

Various other details of the trip can be edited. In the current version, attributes include title, comments, rating, type and mode of transportation, as depicted in Figure 14.

Editing route

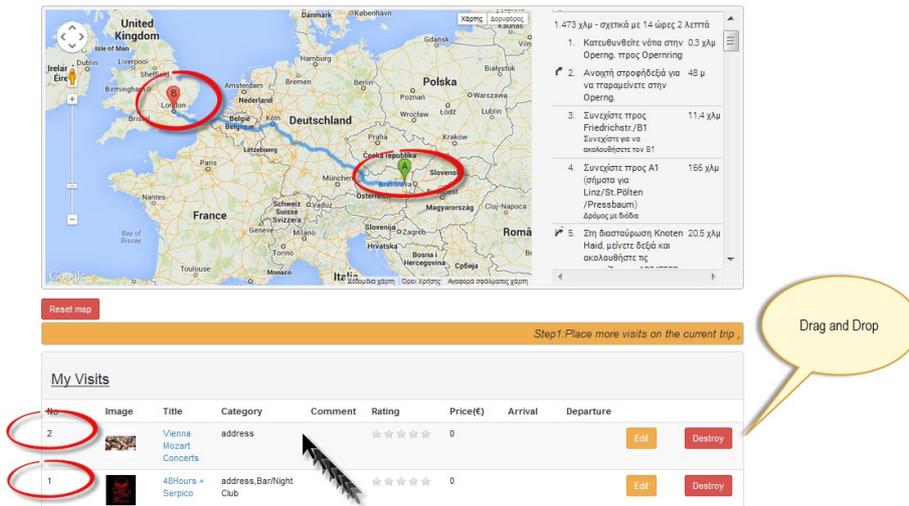


Figure 13: Changing the order of visits

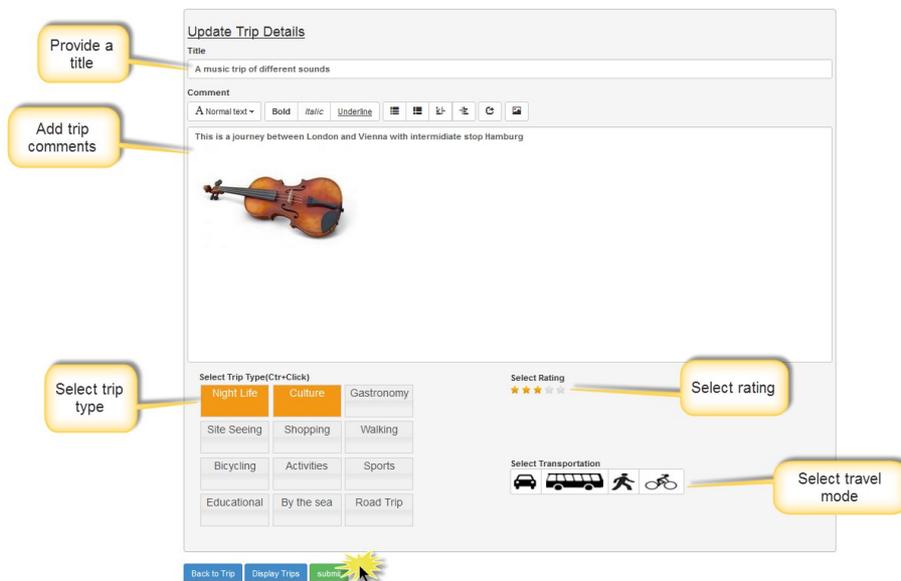


Figure 14: Trip types, title and comments view

Once finished editing, the user can save the changes. The updated trip information is then displayed on the initial page (Figure 15).

**GEOSTREAM** All Fields Search... Search Q You have 2 visit(s) in your trip

Admin History Log Out dskoutas@imis.athena-innovation.gr Authoring Tools

Trip was successfully updated.

### Displaying Trip A music trip of different sounds

962 χλμ - σχετικά με 8 ώρες 51 λεπτά

1. Κατευθυνθείτε νότια στην Οπερνγκ. προς Οπερνγκ 0.3 χλμ
2. Ανοική στροφή αριστερά για να παραμείνετε στην Οπερνγκ. 0.1 χλμ
3. Στροφή αριστερά στη(n)ν Karlspl./B1 60 μ
4. Ανοική στροφή δεξιά για να παραμείνετε στην Karlspl./B1 Συνεχίστε για να ακολουθήσετε τον B1 0.6 χλμ
5. Στροφή δεξιά στη(n)ν Schwarzenbergpl. 64 μ

**Trip Details**

Title: A music trip of different sounds  
 Comment: This is a journey between London and Vienna with intermediate stop Hamburg

Rating: ★★★★★  
 Type: Night Life Culture  
 Time spent:  
 Transportation: Public Transportation  
 Total price(€): 0

**"Edit" to manage your Trip**

**My Visits**

No	Image	Title	Category	Comment	Rating	Price(€)	Arrival	Departure
1		Vienna Mozart Concerts	address		★★★★★	0		
2		48Hours + Serpico	address,Bar/Night Club		★★★★★	0		

Edit Back to Trips Back to Search

Figure 15: Updated trip view

In a similar manner, the user can select a visit contained in a trip in order to modify it or add new information (see Figure 16).

**My Visits**

No	Image	Title	Category	Comment	Rating	Price(€)	Arrival	Departure
1		Vienna Mozart Concerts	address		★★★★★	0		
2		48Hours + Serpico	address,Bar/Night Club		★★★★★	0		

Edit Back to Trips Back to Search

Figure 16: "My visits" view

This is done by clicking the "Edit" button at the bottom of the page displaying the information about a visit (see Figure 17).

### Visit Details

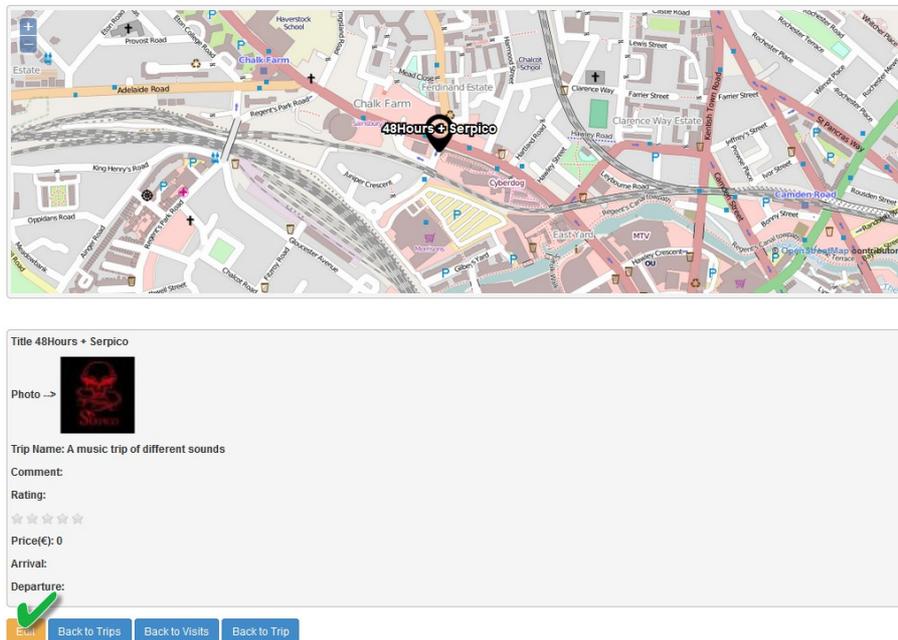


Figure 17: Editing a visit

The attributes used to describe a visit include free-text comments, rating, price, arrival and departure time, as depicted in Figure 18.

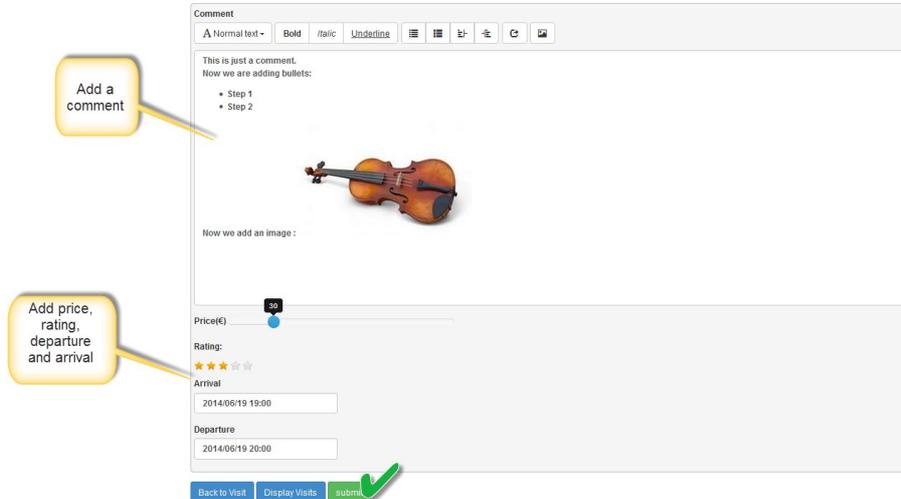


Figure 18: Update a visit

In the beginning of this example, we assumed that the user does not have any already created trips, hence a new trip (and an initial visit) was created automatically when a POI was selected and added to it. Of course, it is possible to add visits to existing trips. This is done by navigating to the page displaying the list of all the trips created by the user and then selecting a trip as "current trip" (see Figure 19). This means that visits will now be added to this trip.

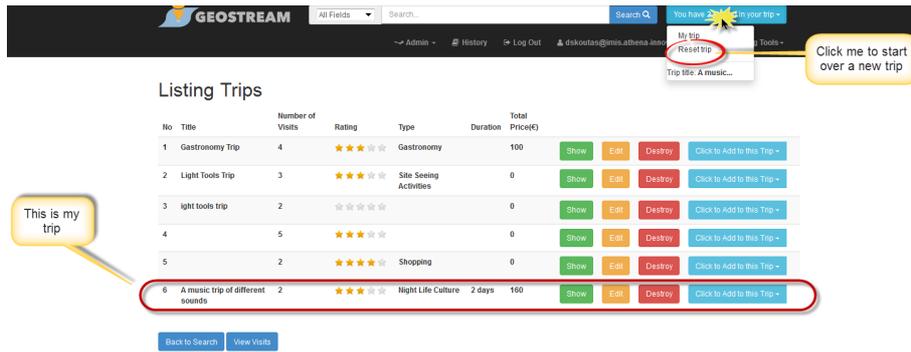


Figure 19: Listing trips

Furthermore, an overview of the visits included in a user's trips can be displayed by going to "Authoring Tools → My Visits" as shown in Figure 20.

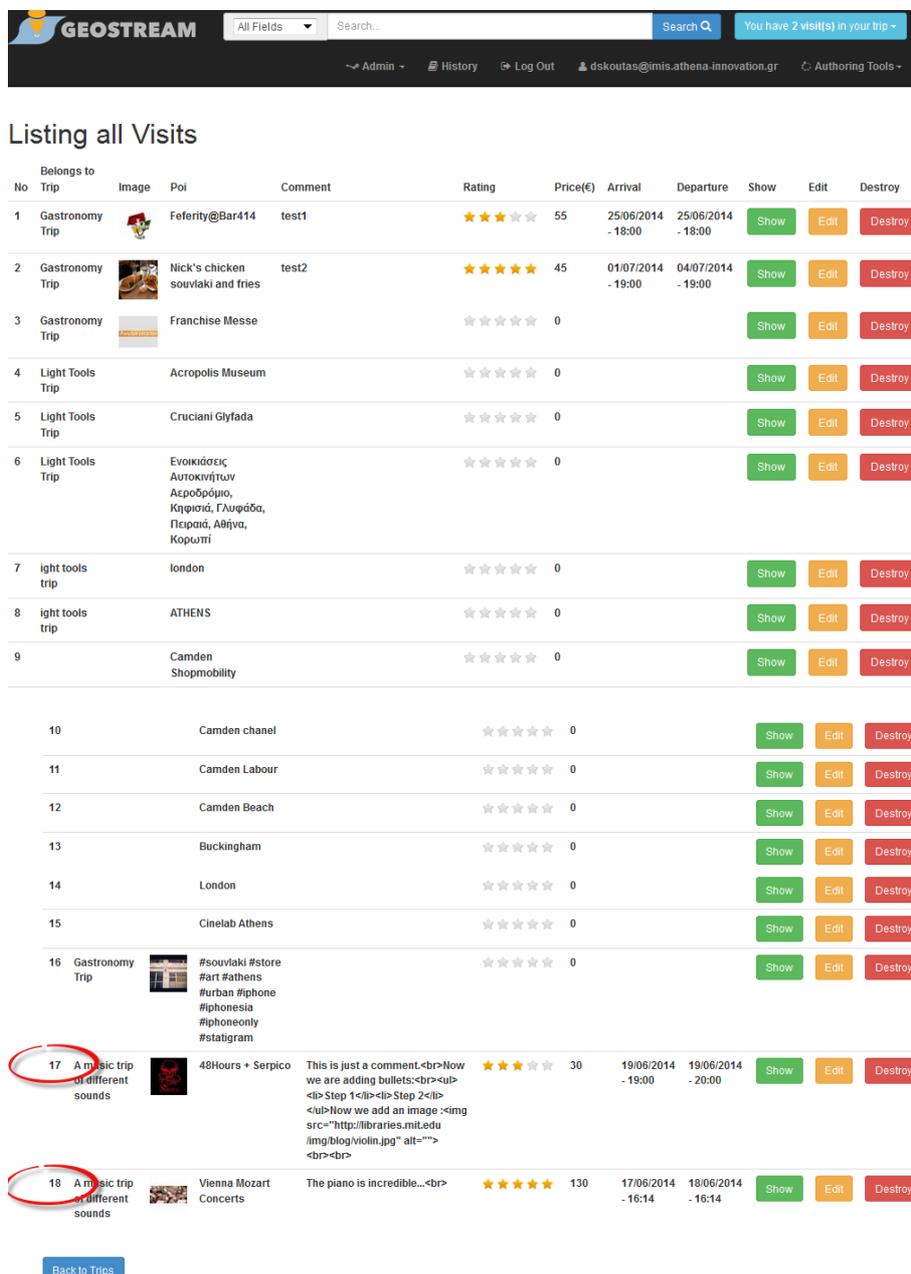


Figure 20: Visits list

## 3. Light Authoring Environment

### 3.1. Main concept

As explained in Section 1, we consider also an alternative method for obtaining user content that involves minimum involvement by the user, extracting instead geospatial information automatically from text documents. This approach is aimed, for example, at authors writing a travel guide using some word processing software or travelers writing on a blog to express and share their experiences from their trip.

However, as we wish to view these functionalities as complementary rather than as alternative, mutually exclusive ways of how a user can create content, we combine both parts in a single workflow and Web application, where the user can choose which steps to perform.

In a nutshell, the process is as follows. The user starts with an existing text document that contains mentions of one or more locations (among other). This is given as input to the application, which automatically identifies and extracts the mentioned locations, mapping them to POIs found in the GeoStream database (or other external sources that have been indicated). The details of the geocoding service are described in Section 4. Then, instead of simply annotating the original document with a specific markup indicating the geocoded words or phrases, it uses these extracted POIs as input for the rich authoring tool, i.e. allowing the user to proceed in the next stage to add more information regarding them.

Next, we describe the functionality of the light authoring tool in more detail, use a step-by-step example to better illustrate the process.

### 3.2. Extracting geospatial entities from text

The following steps show how the light authoring tool can be used to extract geospatial entities from a text document, and how the results can then be used as input to create a trip and its visits, to allow for further editing, if desired.

The user can access the tool by selecting “Light Tools” in the navigation menu of the Web application, as shown in Figure 21.

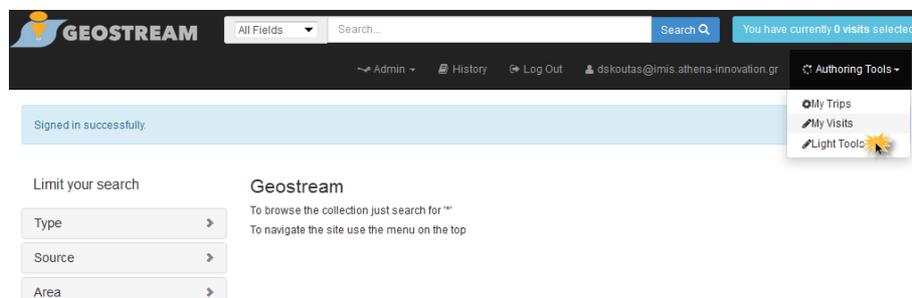


Figure 21: Navigating to the light authoring tool

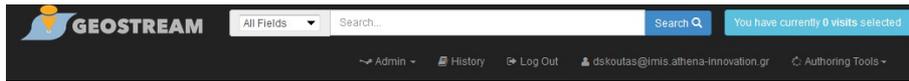
At the main page of the tool, the user can specify a title for the trip that will eventually be created, and use the provided text area to write (or paste) the text describing, for example, her trip. As an illustrating example, consider that the

user inputs the phrase "I went to London and then to Berlin." This is entered in the left text area shown in Figure 22.

Upon clicking the "Geocode" button, the same text is presented on the right box. Processing may take a few seconds, as the geocoding service is invoked to process the given text. As a result, in the newly presented text on the right, the city names "London" and "Berlin" have been identified as locations by the geocoding service and are presented as hyperlinks. Briefly, what happens in the background is that when the geocoding service identifies a token (a word or a sequence of words) as a possible location, it hyperlinks them with relevant geodata from the back-end database. The result can be seen again in Figure 22.

Figure 22: Light authoring environment

The user can then click on the "London" hyperlink to display on the map locations of places containing the word "London" that have been identified by the geocoding service. Left click can be used to view the information of such a place, while right click can be used to select the desired location among the provided alternatives (selected marker becomes green). A "Reset selection" button is also provided to unselect the location(s), if needed to go back to a previous state. An example is depicted in Figure 23.



Light Authoring Tools (Version 3.0)

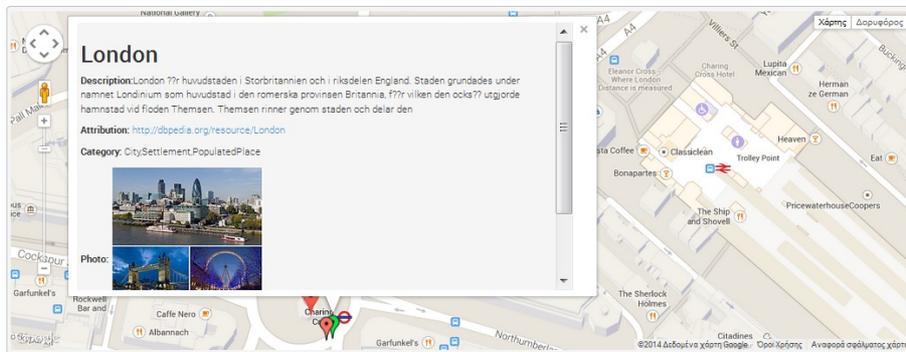
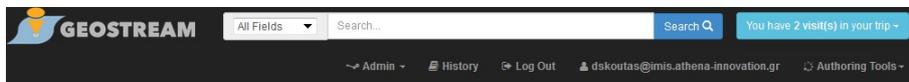


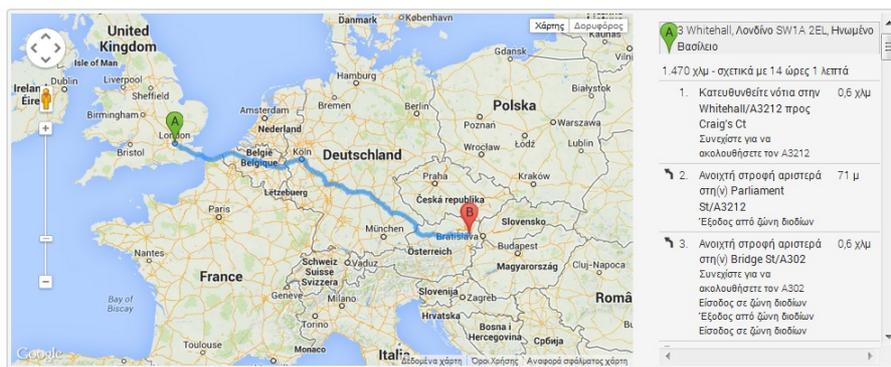
Figure 23: Discovered locations as links

This process can then be repeated for other identified locations, e.g. for the hyperlink “Berlin” in our example.

Once all identified locations have been verified by the user, she can click the “Generate Trip” button. A trip is then created with the selected locations, and the user can proceed in the Edit mode to specify further information, if needed (see Figure 24).



Editing route



Reset map  
 trip by dragging it. Step4: Add more information on each visit.

My Visits

No	Image	Title	Category	Comment	Rating	Price(€)	Arrival	Departure
1		London	City,Settlement,PopulatedPlace		☆☆☆☆☆	0		
2		Esoteric Fairs Vienna	named place		☆☆☆☆☆	0		

**Update Trip Details**

Title  
Trip from London to Berlin

Comment  
A Normal text - Bold Italic Underline [List] [List] [List] [List] [List] [List] [List] [List]

I went to London and then to Berlin.

Select Trip Type(Ctr+Click)

Night Life	Culture	Gastronomy
Site Seeing	Shopping	Walking
Bicycling	Activities	Sports
Educational	By the sea	Road Trip

Select Rating  
☆☆☆☆☆

Select Transportation

Back to Trip Display Trips submit

Figure 24: Editing the route

## 4. Geocoding engine

As described in the previous section, the central processing component of the light authoring environment is the geocoding engine, which identifies locations in text documents appearing in the GeoStream database (or other external data sources configured for use). We describe this component separately, as it constitutes an independent module that can further extended, configured and used by other components.

The geocoding engine employs machine learning to perform its task, i.e. to detect and geocode spatial entities in natural language text. More specifically, it includes a sentence detector, a tokenizer, a location finder, a parts-of-speech (POS) tagger, a chunker, and a parser.

For its implementation, the Apache OpenNLP java library has been used (for more information see: <http://opennlp.apache.org>). All OpenNLP functions require a training model to identify names (locations in our case) and for the tagging. At this stage, our model for identifying locations has been based on the default (initial) model provided by the OpenNLP community; however, to potentially increase the accuracy for our scenarios, we plan to additionally train and test it with corpora provided by our partners, in particular text from travel guides provided by MMV. Moreover, the English natural language model has been used so far, but it is possible to use or train models for other languages as well, following the same process.

The geocoding engine, which is developed in Java, has been integrated into the light authoring environment, which is implemented in Ruby on Rails, using Ruby runtime calls to a Java wrapper class that returns the positions of the locations found in an arbitrary input text.

Regarding its functionality, the geocoding engine processes natural language text input by the user by performing the following main operations:

1. *Sentence detection*: identifies the sentence boundaries of the input text by determining the punctuation characters that mark the end of a sentence.
2. *Sentence splitting*: each sentence is further subdivided into words (tokens) using a tokenizer specific for the particular natural language.
3. *Tagging*: each token of a sentence is tagged with part-of-speech tags.
4. *Spatial entity detection*: entity detection is performed and locations are extracted from the "name finder" component, which looks up the examined token in the GeoStream database (or other provided source of POIs)
5. *Entity annotation*: the location(s) found in the text are hyperlinked with relevant geodata from the back-end database.

The resulted rich text includes, for each discovered location within the input, a set of GIS positions. The following diagram illustrates the processing workflow of the geocoding engine.

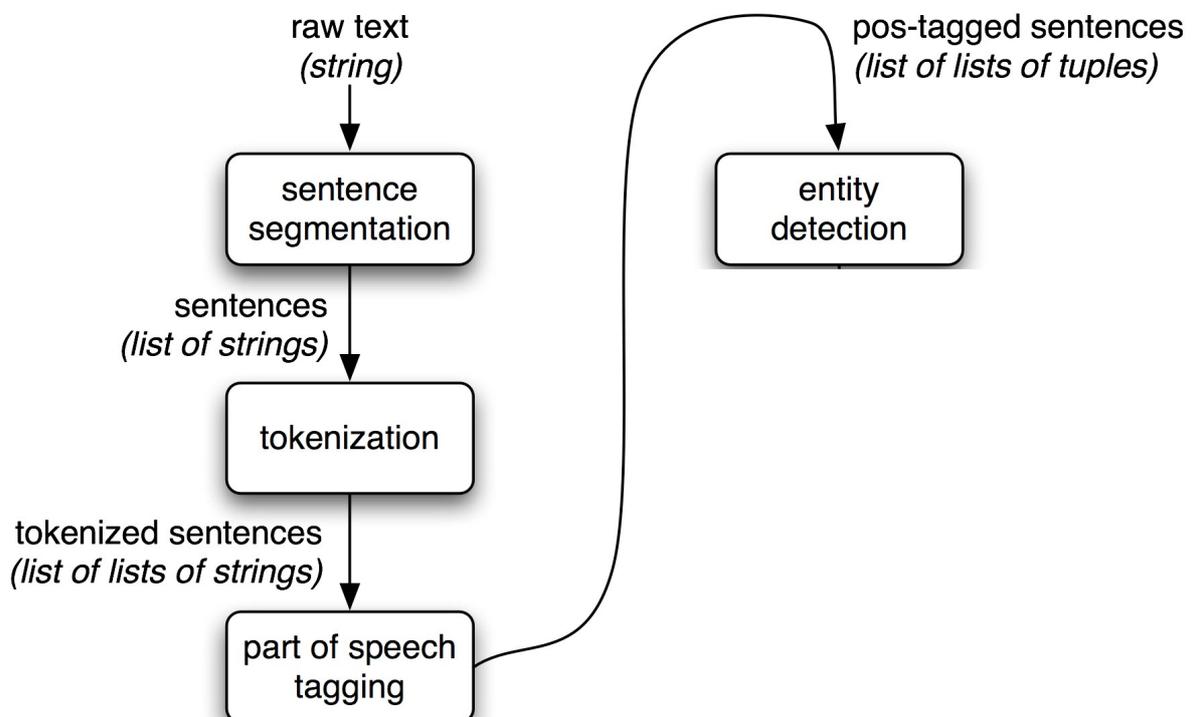


Figure 25: Processing workflow of the geocoding engine

## 5. Authoring tools API

In the previous sections, we have illustrated the functionalities and features of the authoring tools by showing their usage via the Web application and user interface. However, an application programming interface (API) has been also developed and is provided to make these tools reusable and extensible, e.g. for making it easier for the partners to reuse and integrate them in their applications and front-ends.

The implemented API provides several search methods that can be used to retrieve information from the database, and also it implements CRUD (i.e. create, read, update and delete) actions for the entities *vistis* and *trips*, so that these can also be managed programmatically. In the following, we list the documentation for the calls related to the search capabilities. For all calls, the returned result is formatted in json. An indicative example of such a call and the corresponding response provided by the system are also shown in Appendix 2.

search		
<b>Parameters:</b>	service	From which service to get results (Value: Must be one of: geostream, google)
	query	A query string.
	[page]	The starting page. It is an optional string field.
	[rows]	The maximum number of rows to return. It is an optional string field.
<b>HTTP method:</b>	GET	
<b>Path:</b>	/api/v1/places/search	
<b>Example:</b>	/api/v1/places/search?service=geostream&query=acropolis	
<b>Returns:</b>	Results found using the Geostream or Google service for a given set of terms.	

reverse		
<b>Parameters:</b>	service	From which service to get results (Value: Must be one of: geostream, google)

<b>reverse</b>		
	type	The type of results to return. Value: Must be one of: poi, Photo, Event.
	pt	The coordinates for the search, as a string, in the form 'lat, lon'
	d	The search radius. Must be string.
	[page]	The starting page. It is an optional string field.
	[rows]	The maximum number of rows to return. It is an optional string field.
<b>HTTP method:</b>	GET	
<b>Path:</b>	/api/v1/places/reverse	
<b>Example:</b>	/api/v1/places/reverse? service=geostream&type=Photo&pt=37.98933792114258,23.729211807250977&d=0.1	
<b>Returns:</b>	Reverse Geocoding service. Returns POIs for a specific Geolocation.	

<b>bbox</b>		
<b>Parameters:</b>	service	From which service to get results (Value: Must be one of: geostream, google)
	type	The type of results to return. Value: Must be one of: poi, Photo, Event.
	[q]	query parameter, as a string
	lat_min	The minimum lat
	lat_max	The maximum lat
	lon_min	The minimum lon
	lon_max	The maximum lon

<b>bbox</b>		
	[page]	The starting page. It is an optional string field.
	[rows]	The maximum number of rows to return. It is an optional string field.
<b>HTTP method:</b>	GET	
<b>Path:</b>	/api/v1/places/bbox	
<b>Returns:</b>	Reverse Geocoding service. Returns POIs included in a geography bounding box with specified upper and lower corners.	

<b>clusters</b>		
<b>Parameters:</b>	city	The city
	category	The geostream cluster categories. Value: Must be one of: Photo, Tweet, Athletics Sports, Culture, Education, Entertainment, Food, Professional, Services, Shops, Travel Transport.
<b>HTTP method:</b>	GET	
<b>Path:</b>	/api/v1/places/clusters	
<b>Returns:</b>	Json result containing areas (Geo-polygons) of interest.	

<b>cities</b>	
<b>Parameters:</b>	No specific parameters are required
<b>HTTP method:</b>	GET
<b>Path:</b>	/api/v1/places/cities
<b>Returns:</b>	Json formatted result containing the available in the geostream server cities

## 6. Conclusions and next steps

This deliverable describe the prototype that has been developed for the GeoStream authoring environment. This aims at facilitating users in creating annotated geospatial content, and comes in two forms, a rich and a light authoring tool. The former focuses on a more structured way to create and manipulate content, around the notions of visits and trips, which group together POIs and enhance them with user-defined descriptions, such as comments and ratings. The latter requires less user involvement; instead, its purpose is to detect and annotate geospatial entities found in text documents. Both tools are integrated in a Web application, while a corresponding API is also provided. The described functionalities have been integrated in the GeoStream online demo (available here: <http://dataminer.geocontentstream.eu/>).

In parallel, an evaluation of the authoring environment has been taking place, and the results of the survey will be reported in Deliverable D4.2. These findings, which actually cover not only the authoring environment but the GeoStream demo overall, will be analysed to identify features and functionalities that need to be added or improved.

Some points that are already identified as parts of ongoing and future work include: (a) allowing user-defined attributes for describing visits and trips, (b) additional training of the geocoding engine and improving the ranking of the results, and (c) adaptations of the API based on feedback by the SME partners.

## Appendix

### 1. Entity diagram

The main entities involved in the authoring tools, from a database point of view, include the concepts *Visits*, *Trips* and *POIs*, and are represented by the tables *line\_items*, *routes* and *venues* respectively. The figure below shows their relationships as well as their main attributes.

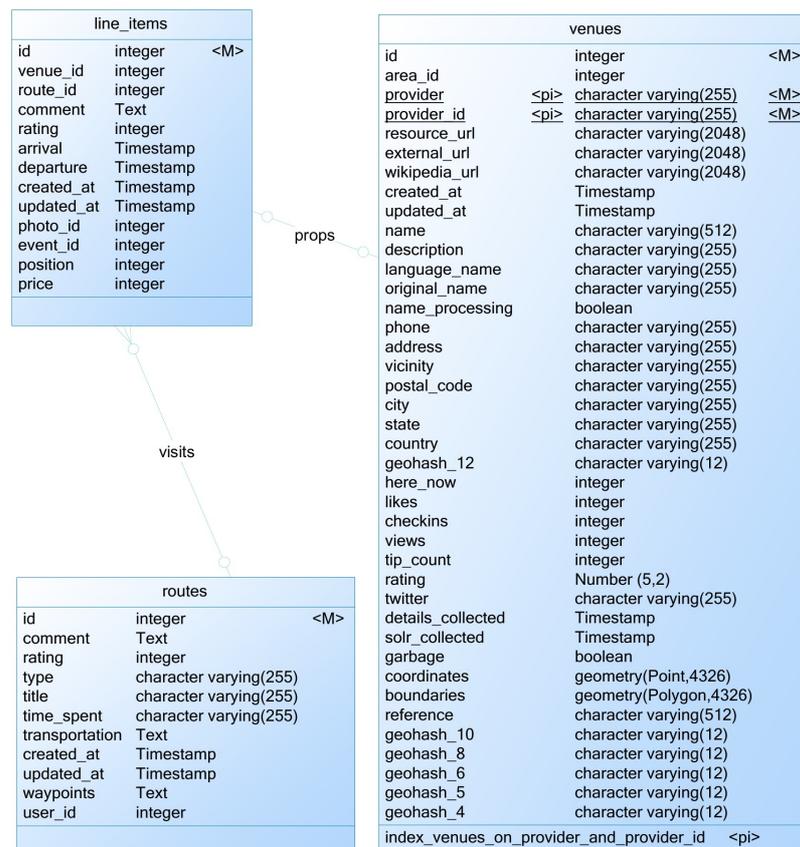


Figure 26: Entity diagram of the main concepts Visits, Trips and POIs

### 2. Example of API call and response

**call: search with parameters service=geostream and query=myconos**

```

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          23.721035710814675,
          37.92127434771073
        ]
      },
      "properties": {
        "label": "Paranga Myconos",
      }
    }
  ]
}
  
```

```
"description": null,
"sourceURL": "https://foursquare.com/v/paranga-myconos/4e2684ccb61ce3fb054f644c",
"externalURL": null,
"category": [
  "place"
],
"tag": null,
"address": null,
"city": "Athens",
"state": "Attica",
"country": "Greece",
"postalcode": null,
"phone": null,
"thumbnail": null,
"photo": null,
"database_id": 96682
},
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      23.697187900543213,
      37.9404238570229
    ]
  },
  "properties": {
    "label": "Metropolitan Hotel Myconos Conference Room",
    "description": null,
    "sourceURL": "https://foursquare.com/v/metropolitan-hotel-myconos-conference-room/50b4acb119a981e9b8dd3c7b",
    "externalURL": null,
    "category": [
      "Conference Room"
    ],
    "tag": null,
    "address": "Syngrou Avenue 385",
    "city": "Athens",
    "state": null,
    "country": "Greece",
    "postalcode": "17564",
    "phone": [
      "+302109471000"
    ],
    "thumbnail": null,
    "photo": null,
    "database_id": 90538
  }
}
]
```