Ref. Ares(2022)6741493 - 30/09/2022

D7.7 Communities' Engagement ICT Tool

Deliverable number	**D7.7 (First Version)**
Deliverable title	**Communities' Engagement ICT Tool**
Nature[1]	**Other**
Dissemination Level [2]	**Public**
Author (email) Institution	**Nikos Tousert (ICCS) , Katerina Georgiou (ICCS), Nikos Mitro (ICCS)**
Editor (email) Institution	**Antonis Kalis (ICCS)**
Leading partner	**ICCS**
Participating partners	
Official submission date:	**M40**
Actual submission date:	**M40**

[1] **R**=Document, report; **DEM**=Demonstrator, pilot, prototype; **DEC**=website, patent fillings, videos, etc.; **OTHER**=other

[2] **PU**=Public, **CO**=Confidential, only for members of the consortium (including the Commission Services), **CI**=Classified, as referred to in Commission Decision 2001/844/EC

Modifications Index	
Date	Version
08/01/2021	0.1 TOC
15/01/2021	0.2 First draft
20/01/2021	0.3 Chapter 1 populated
30/01/2021	0.4 Chapters 2 and 3 populated
14/02/2021	0.5 Chapter 4 populated
22/02/2021	0.6 Included security content in chapter 3
03/02/2021	0.7 Updated the APIs in chapter 4
11/03/2021	0.8 Updated chapter 2.2 after changing wizards in the app
22/03/2021	0.9 Final version, ready for review
24/03/2021	1.0 Updated version after review
15/09/2022	1.1 Updated executive summary and Introduction to include mobile app
16/09/2022	1.2 Chapter 5.1 & 5.2 populated
19/09/2022	1.3 Chapter 5.3 – 55 populated
20/09/2022	1.4 Final version, ready for review
29/09/2022	1.5 Updated version after review

This work is a part of the HYPERION project. HYPERION has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreementno 821054.

ACRONYMS AND ABBREVIATIONS

API	Application Program Interface
CH	Cultural Heritage
HRAP	Holistic Risk Assessment Platform
ICT	Information and Communications Technology
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
SDK	Software Development Kit
CRUD	Create – Read – Update - Delete
REST	Representational State Transfer
URL	Uniform Resource Locator

Table of Contents

Tables

Figures

Executive Summary

This deliverable is about the documentation of Hyperion Communities' Engagement ICT Tool. This tool is based on PLUGGY's Social platform, an open source platform specializing on the preservation and promotion of everyday all-around heritage, using crowd-sourced techniques. Hyperion has taken advantage of PLUGGY platform so as to engage citizens, local communities and business owners in reporting damages concerning cultural heritage sites.

More specifically, the Hyperion Communities' Engagement Tool enables citizens to create stories about the deterioration of CH sites and business interruptions. A user-friendly web–based interface has been developed for the innovative presentation of stories to users in order for them to experience the content and better understand the changesimposed by climate change and extreme events concerning the monuments and the operation of businesses. In addition to that, a dedicated mobile application, named Hyperion App and designed for Android devices, has been developed in order to provide a more direct and easy to use everyday tool that local communities and citizens can use on their smartphones.

The integration of the web-based tool and the Hyperion mobile application with the PLUGGY Authorization services, the APIs of the tool that expose the relevant content to HRAP, and the available functionalities are allincluded in this Deliverable. Moreover, the architecture, the relevant components andthe technology that has been used are described in a separate chapter so as to showcase the integration of the Communities' Engagement Tool with HRAP platform and PLUGGY.

1. Introduction

1.1 Background

Deliverable 7.7 is a report that contains the description and specifications of the Communities' Engagement ICT Tool and the Hyperion Mobile Application. These tool aims at incorporating active communities' participation by making use of the PLUGGY social platform. PLUGGY is asocial networking platform which gives voice to the citizens across Europe and enables them to safeguard and enrich the European cultural heritage landscape.

The Communities' Engagement ICT Tool and the Mobile Application are therefore a pluggable application which make use of already developed PLUGGY's APIs and database so as to support the cultural heritage participation and awareness through the reporting of monument damages and business interruptions.

As described in chapter 1.2, both tools have been developed by taking advantage of the already registered PLUGGY community and theflexible PLUGGY data model. To this end, it has been built upon PLUGGY and engagesits community so as for citizen to be able to give direct feedback to the relevant cultural authorities.

1.2 Purpose and Scope

The purpose was to create a pluggable application (i.e. Communities' Engagement ICT Tool) that will employ the PLUGGY social platform so as to engage citizens and local communities in reporting monument damages and business interruptions. Under the same scope the Mobile Application has been developed which focuses on usability and the more detailed description of the event.

PLUGGY provides the necessary architecture, structure and interfaces for the creation of pluggable applications, which allowed the Communities' Engagement Tool and the Hyperion Mobile Application to utilize PLUGGY's content and community for the purposes of Hyperion Project. To this end, with the integration of Engagement Tool, PLUGGY and HRAP, communities'participation is enhanced and citizens are able to participate in the procedure of damage assessment of cultural heritage sites in a user-friendly way.

1.3 Approach

The development of the Communities' Engagement ICT Tool has been driven by the Deliverable 2.2 which defines Hyperion functional and non-functional requirements as well as the project's use cases. Moreover, the deliverable 2.3 which documents the Hyperion architecture has also been taken into consideration during the development of the tool and the preparation of this deliverable. Last but not least, PLUGGY's public deliverable "Guidelines and Instructions for PLUGGY Apps" has also played major role in order to integrate the Communities' Engagement Tool with PLUGGY's APIs and authorization services.

2. Functionalities and User Interface

2.1 General Description

The Communities Engagement ICT Tool will enhance communities' engagement in Hyperion. The tool is based on PLUGGY'S social platform, which is a web app specializing on the preservation and promotion of everyday all-around heritage, using crowdsourced techniques. The communities Engagement Tool makes use of PLUGGY social platform for the following reasons:

- To take advantage of the visitors already engaged in PLUGGY platform (https://pluggy.eu/). This means that a user already logged in in PLUGGY platform will be able to interact with the engagement ICT tool without leaving PLUGGY. This can be achieved through the plug-in mechanism of PLUGGY.
- To be able to retrieve content from PLUGGY platform (monuments information, used tags, photos, etc.) by making use of its APIs
- To take advantage of the number of PLUGGY's users. This is achieved with the integration between the Hyperion engagement tool and PLUGGY's authorization component. To this end, a user that has already created an account in PLUGGY, he is not obliged to create another account in Hyperion authorization component.
- All the content of the Hyperion engagement tool (monuments deterioration, business losses, etc.) can be viewed through PLUGGY platform.

The integration of the engagement tool with PLUGGY is not only seen as a technical integration that will just exploit already developed technical units of PLUGGY, but the objective is the semantic integration in a way that content related to cultural heritage stored in PLUGGY database and especially monuments geo-localized info, will be exploited by the engagement tool. Figure 1 shows the integration of the engagement tool with PLUGGY platform and its interaction with the user. The stories related to deterioration of CH sites or business damages can be retrieved by the HRAP through PLUGGY's RESTful services.

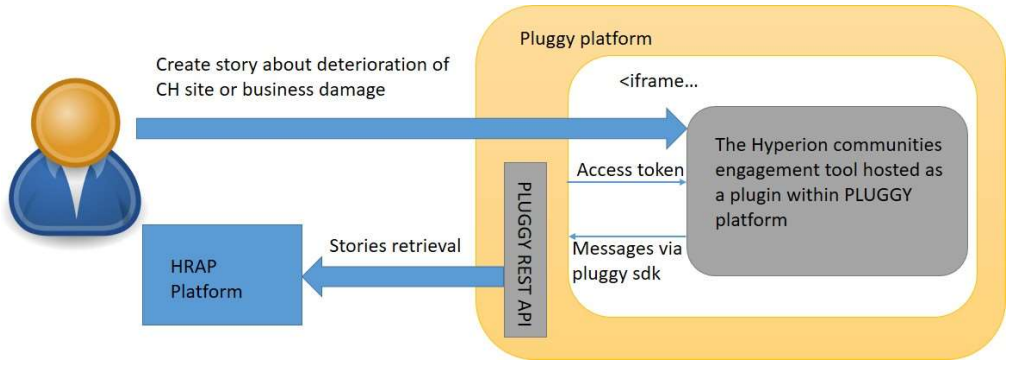

Figure 1: Integration between HRAP, PLUGGY platform and Communities' Engagement ICT Tool

2.2 Wizards for creating stories

There are two kind of stories that a user can store within the Engagement Tool through wizards:

- **Hyperion Citizen Stories**: Citizen are able to create stories about the deterioration of cultural heritage sites. In this case the user is able through a wizard to store the following info: title of the story, monument damage description**,** and relevant photo and tags.
- **Hyperion Business Owners Stories**: Local business owners are also able to report about their damages, since there is the need to promise business continuity model. More specifically, business owners are able to report through the tool the following: business damage, business risk assessment, info useful to local authorities, customers lost percentage, priority and criticality of the issue, relevant tags.

The two aforementioned kind of stories both can contain relevant images, location details and general info. Figures 2 and 3 depict the two steps of the wizard of the engagement tool for creating a citizen story concerning a monument damage.

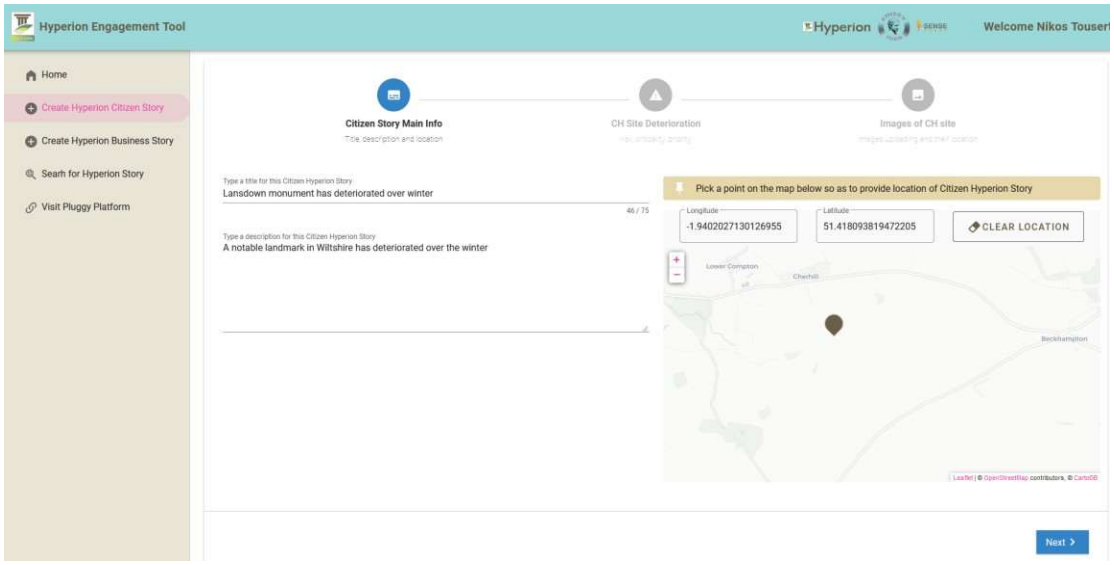

Figure 2: In the first step of Hyperion Citizen Story Wizard, the user inserts information concerning title and description of the story (e.g. Lansdown monument has deteriorated over winter).

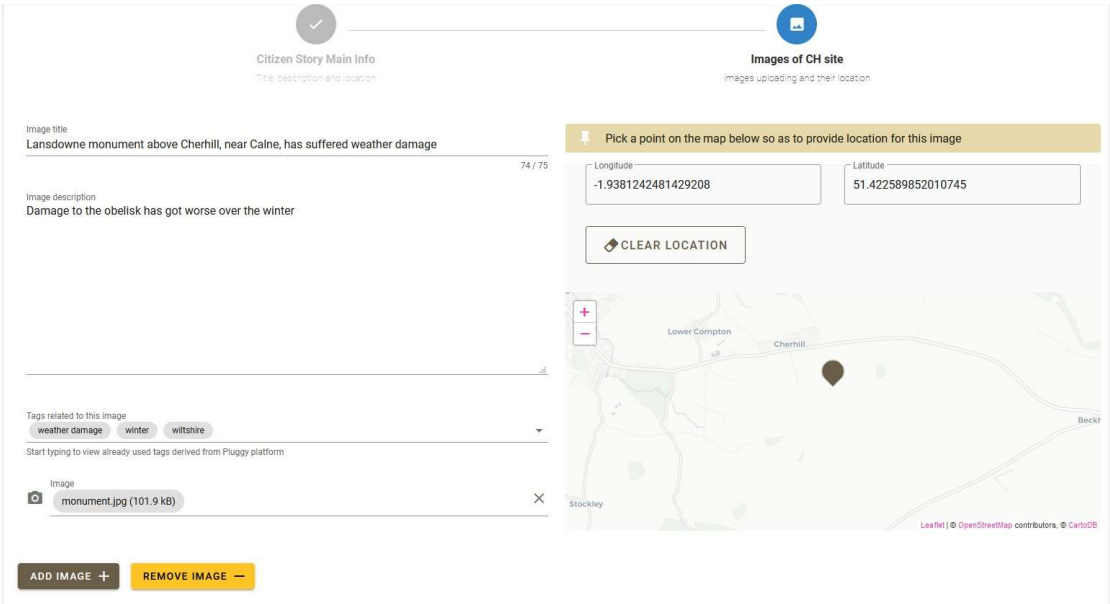

Figure 3: In the second step of Hyperion Citizen Story Wizard, the user inserts relevant photos along with their location, their title, description and tags. The user is able to upload as many photos as they like.

The wizard for creating a Hyperion Business Story is pretty much the same. The only difference lies in the fact that there is one more step (second step) of the wizard, where the user has to insert information relevant to business damages (i.e. damage of the business, risk assessment for the business, info useful to local authorities, customers lost percentage, priority of the issue, criticality of the issue and relevant tags). Figure 4 depicts the second step of the wizard related to business owners' stories.

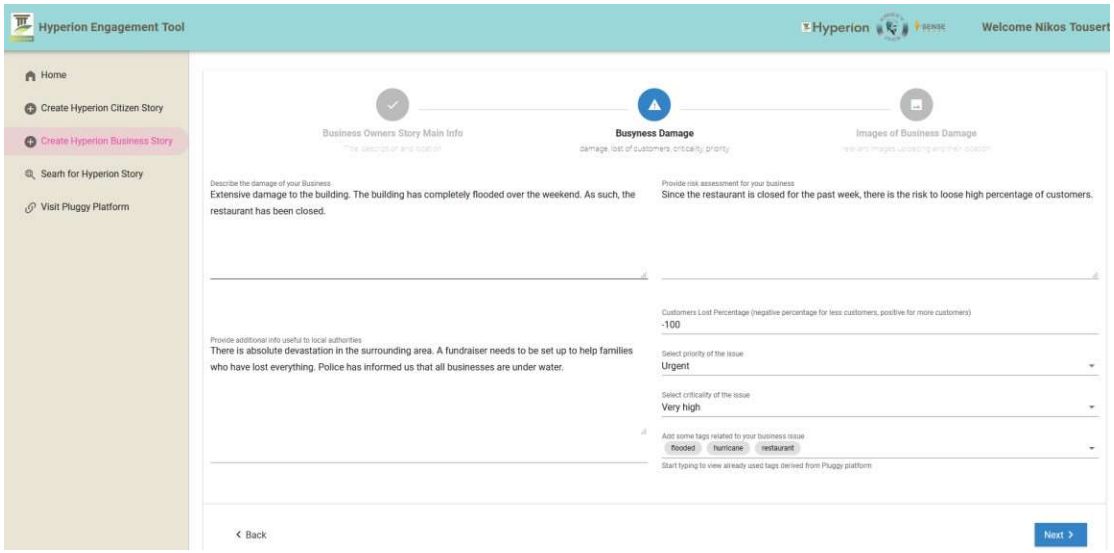

Figure 4: In the second step of Hyperion Business Story Wizard, the user inserts more business damage detailed info (e.g. damage of the business, risk assessment for the business, additional info useful to local authorities, customers lost, etc.).

After the completion of each wizard, all relevant info concerning the stories is stored in the PLUGGY back-end and can be retrieved through its APIs. Thereafter, both the Communities' Engagement ICT tool and HRAP will be able to access the content and present to the users the stories information.

2.3 Presentation of created stories to users and editing functionalities

In addition to uploading these two kind of stories (monument and business damage stories), the users of the Engagement Tool are also able to view already stored stories. The content of a Hyperion Citizen story is shown in figure 5. As shown in the figure, there are two tabs, one for displaying the story text (title, description, tags) and one for displaying the relevant photos. Figure 6 shows the page when clicking the photos tab. In both tabs, the user is able to view on the right of the screen the location of the deteriorated monument.

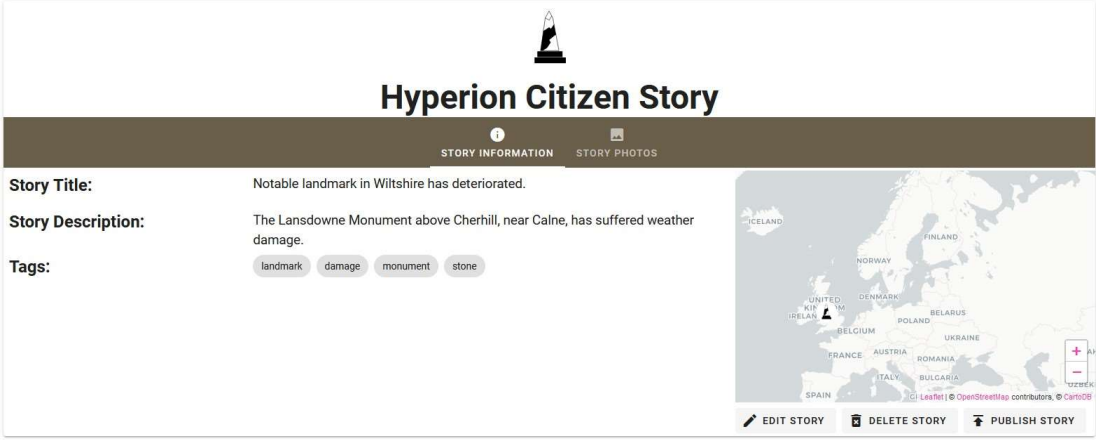

Figure 5: Users are able to view stored Hyperion stories. This figure is about a citizen story and specifically it depicts information (title, description, tags) concerning a damage of Lansdowne monument. The location of the monument is also depicted on the right.

.

Figure 6: A grid that contains all the thumbnails (photos with reduced file size) of a citizen story is depicted in this figure. The button "ADD PHOTO" allows the user to add one more photo to an already stored story. The location of the deteriorated monument is shown on the right.

The user is also able to view the full size photo along with its metadata when clicking at an image on the grid. Figure 7, depicts the page of the Engagement Tool which has been designed so as to render image specific metadata (title, description, tags), image location on the map and the full size of the image.

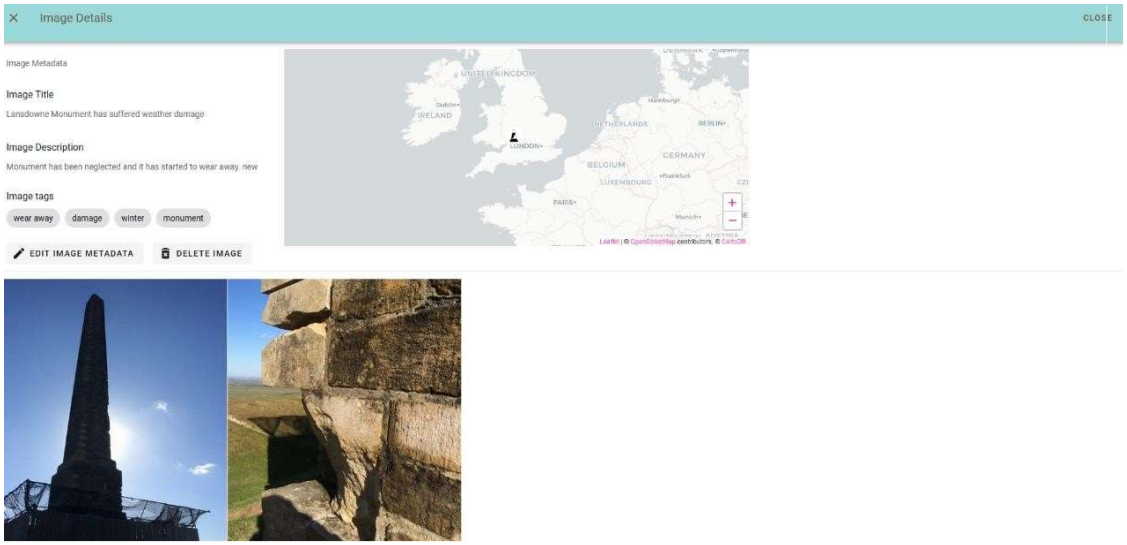

Figure 7: Full size image along with its metadata (title, description, tags and location) is rendered to the user. There are also buttons for editing or deleting the photo.

Regarding the presentation of the content of Business Owners stories, the relevant page is shown in figure 8.

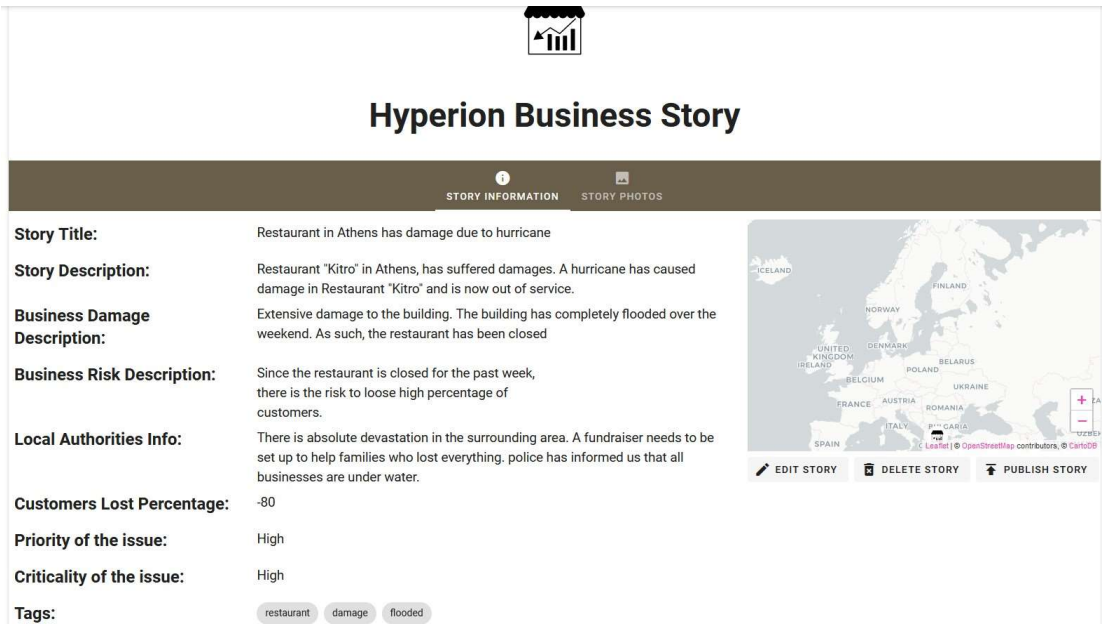

Figure 8: The content of a Business Owners story is presented to the user. The information rendered to the screen includes the following: story title, story description, business damage description, business risk description, information useful to local authorities, customers lost percentage, priority of the issue, criticality of the issue and relevant tags.

As shown in figure 8, the user is able to view info relevant to the given business interruption and also the relevant photos which should depict business damages (e.g. flooded restaurants). In addition to the presentation of both citizen and business owners stories, the owners of the stories are also able to edit or delete them. Figure 9 depicts the Hyperion Engagement Tool page which is used for editing a citizen story.

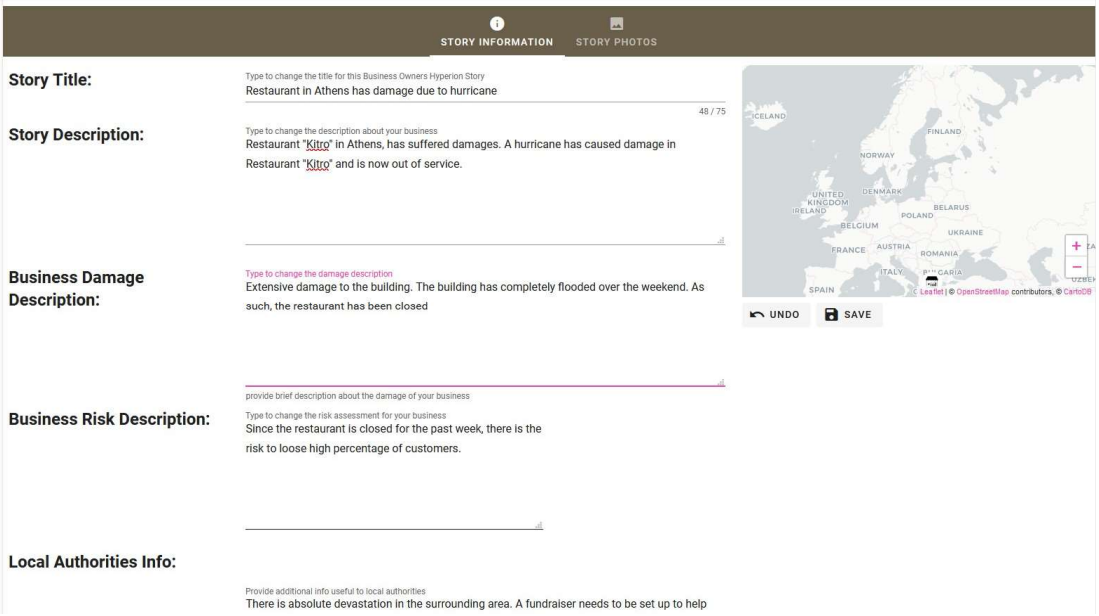

Figure 9: The fields of a business story are editable so as for the owner of the story to change the content.

2.4 Responsive Design of Hyperion Communities' Engagement Tool

Responsive web design (RWD) is an approach to web design that makes web pages render well on a variety of devices and window or screen sizes. Responsive web design has become more important as the amount of mobile traffic has come to account for more than half of total internet traffic.

Thereafter, content, design and performance are necessary across all devices to ensure usability and satisfaction. The Hyperion Communities' Engagement Tool adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries so as for the user to be able to use the tool and view the content of citizen and business stories from any device independent on its screen size. Figure 10 depicts the content of a citizen story if it was to be viewed in a screen 320 * 510 and figure 11 depicts the same content in the same screen if rotated (landscape mode). Finally, figure 12 depicts a part of the first step concerning the citizen story wizard as rendered in a small mobile screen.

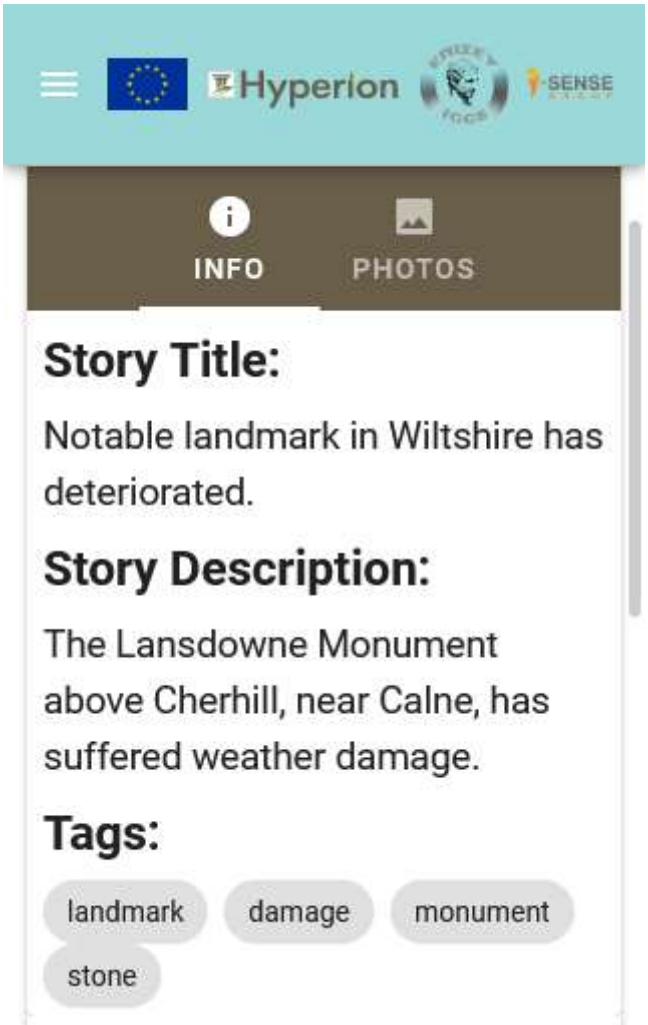

*Figure 10: The content of a citizen story depicted in a mobile device with screen 320 * 510.*

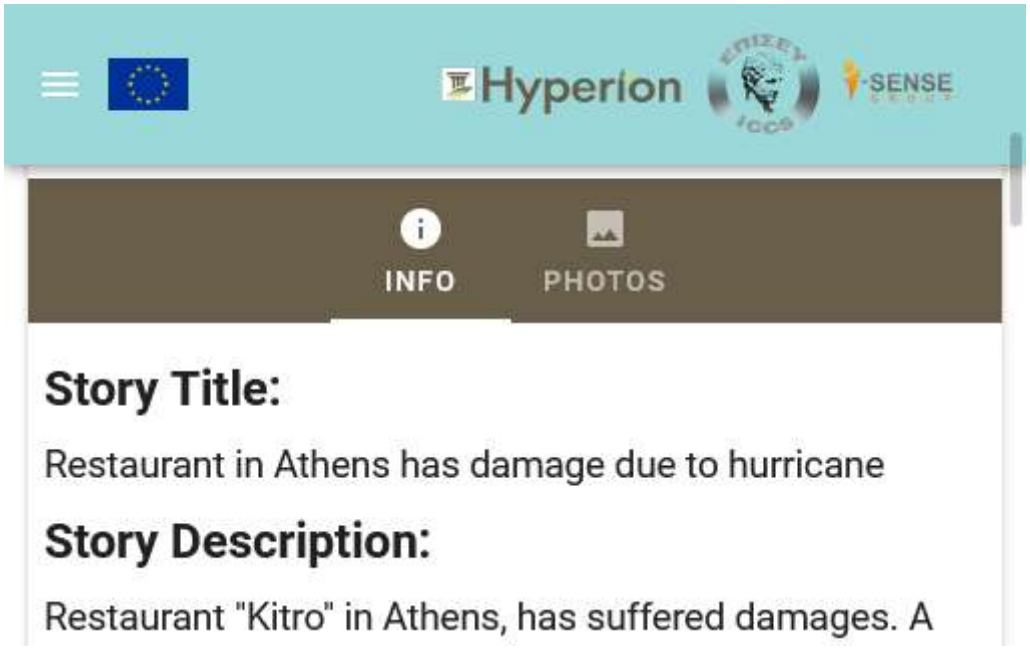

*Figure 11: The content of a citizen story as depicted in a screen 510 * 320 (landscape mode).*

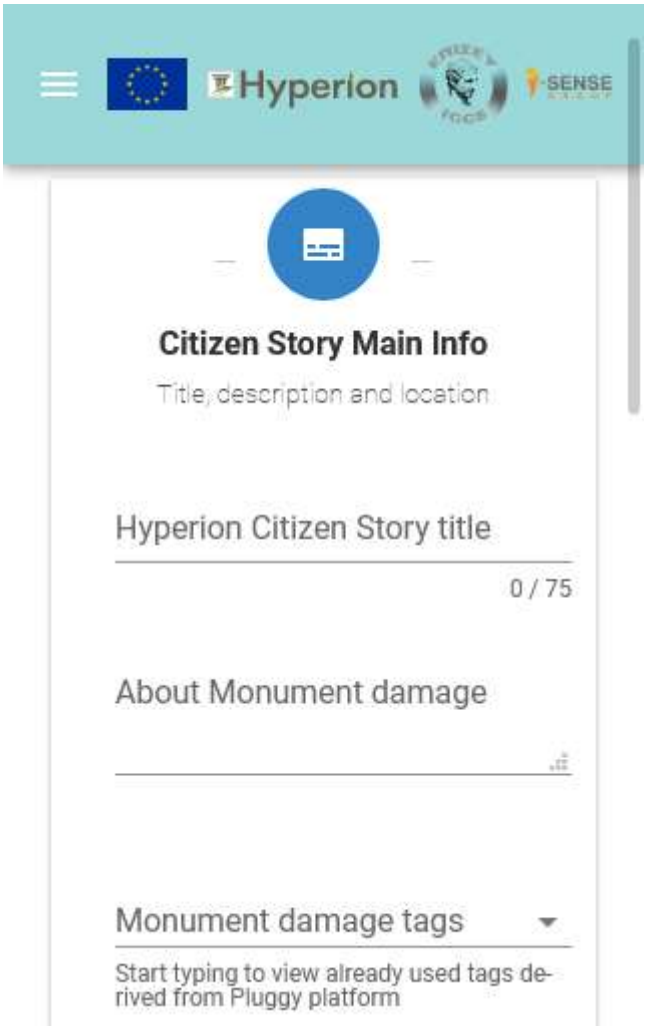

*Figure 12: A part of the first step of the citizen story wizard as rendered in a screen 320 * 510.*

3. Integration with the PLUGGY Platform

The Hyperion Communities' Engagement tool heavily relies on PLUGGY's Social Platform as regards the following:

- It stores its data in PLUGGY's database
- It has been integrated with PLUGGY Identity Manager
- It has been integrated as a plugin within the PLUGGY social platform.

Chapter 3.1 deals with PLUGGY's database and its association with Hyperion and chapter 3.2 addresses the integration with PLUGGY's Authorization services which is essential for HRAP in order to get protected resources from Hyperion tool.

3.1 How PLUGGY's Database is associated with Hyperion Data Model

The Hyperion Communities' Engagement Tool makes use of PLUGGY's Platform database which is depicted in the following figure:

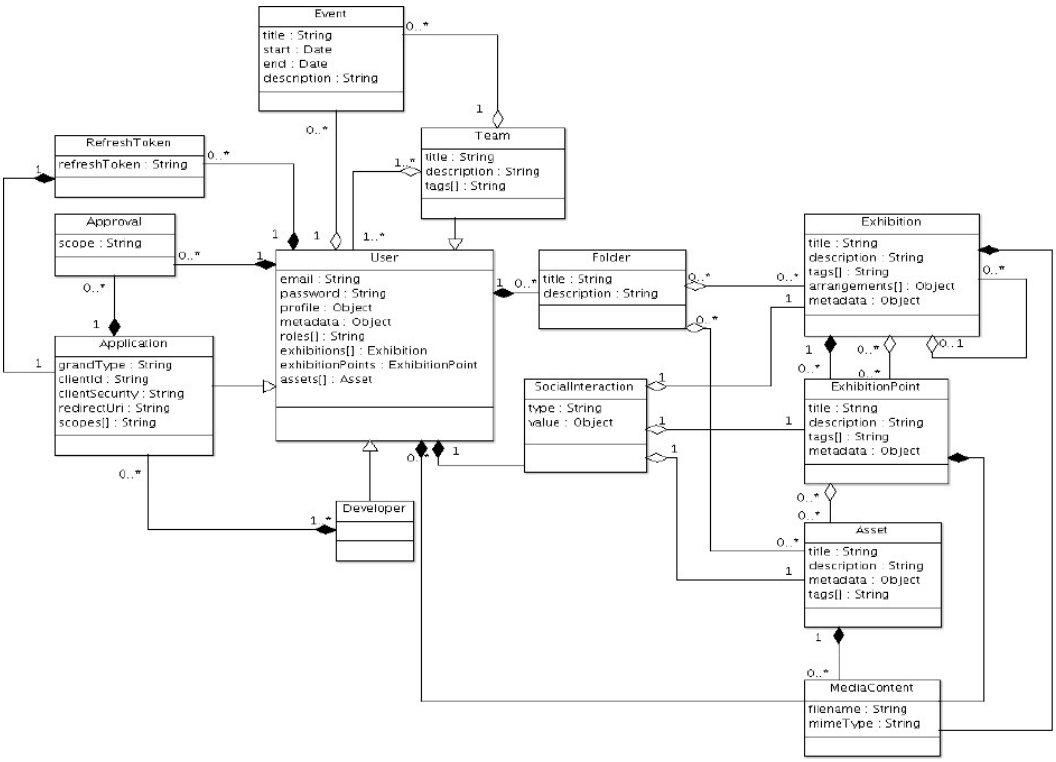

Figure 13: Data Model of PLUGGY Platform. This is also the data model that Hyperion Communities' Engagement Tool makes use of.

The PLUGGY's data model is generic enough to be used by other apps and most specifically by PLUGGY's plugins. This is why the Hyperion Engagement ICT tool (being a PLUGGY plugin) makes use of both PLUGGY's APIs and database. The main entities used in PLUGGY data model along with their interpretation are the following:

- **Asset (https://pluggy.eu/api/doc/#/Asset_CRUD)**: It is the elementary unit of content in PLUGGY. An asset is a media file with an identified owner, a title, a description, a set of tags and a license, which specifies how this file can be reused. The Asset can be used as a digital representation of cultural heritage artifact, tangible or intangible. The media file can be text, image audio, 3d model or any type of binary data.
- **Exhibition (https://pluggy.eu/api/doc/#/Exhibition_CRUD)**: Cultural heritage stories curated by users of PLUGGY using one or more Assets. An exhibition can be of several types: Media Stories, Timelines, Geolocated Tours, Augmented Reality Exhibitions and Games.
- **Exhibition Point (https://pluggy.eu/api/doc/#/Exhibition_Point_CRUD)**: Exhibition Points link Exhibitions and Assets. An Exhibition Point is the usage of an Asset in an Exhibition, and an Exhibition is composed of one or several Exhibition Points. For example Events are the Exhibition Points of Timelines, and Chapters are the Exhibition Points of Media Stories.

The PLUGGY provides basic CRUD methods for manipulation of all aforementioned entities, as shown in the link: https://pluggy.eu/api/doc/#/

The entities of PLUGGY that are entitled Exhibition, Exhibition Point and Asset are being used by the Hyperion Engagement Tool but in a different manner as described below:

- **Exhibition in Hyperion Engagement Tool:** Each citizen story and each business owners story in Hyperion corresponds to an exhibition. In Hyperion, the exhibition has the meaning of a story (either citizen or business story). This exhibition may contain many exhibition points. Apart from the linking to exhibition points, the exhibition may contain **among others**: link to other exhibitions, the date when the exhibition was updated or created, the owner and the creator of the exhibition, and finally the title and description of the exhibition. For instance the title for an exhibition concerning a citizen story could be the following: "Notable landmark in Wiltshire has deteriorated" and a title for an exhibition concerning a business owners story could be the following: "Restaurant in Athens has damage due to hurricane". The description of an exhibition relevant to business owners story could be: "Restaurant Kitro in Athens has suffered damages. A hurricane has caused damage in Restaurant 'Kitro' and is now out of service".

- **Exhibition Point in Hyperion Engagement Tool:** The exhibition point is linked to a specific exhibition (as shown in figure 13) and contains more detailed info regarding the story. Especially for the case of Hyperion business stories, a separate json file is linked to the exhibition point that when downloaded and parsed by the client, it contains the following: (for keeping things simple for the user, this type of json file is not included in citizen stories)
    - **Business damage description** (e.g. Extensive damage to the building. The building has completely flooded over the weekend. As such, the restaurant has been closed.)
    - **Business risk description** (e.g. Since the restaurant is closed for the past week, there is the risk to loose high percentage of customers).
    - **Local authorities info** (e.g. There is absolute devastation in the surrounding area. A fundraiser needs to be set up to help families who lost everything. Police has informed us that all businesses are under water.)
    - **Customers lost percentage** (e.g. -80%)
    - **Priority of the issue** (possible values: Urgent, High, Medium, Low, Very Low)
    - **Criticality of the issue** (possible values: Very High, High, Medium, Low, Very Low)
    - **Relevant Tags** (e.g: restaurant, flooded, damage)

- **Asset in Hyperion Engagement Tool:** Similarly, to PLUGGY assets, an asset in Hyperion stories is a media file with an identified owner, a title, a description, a set of tags and a license, which specifies how this file can be reused. Currently, the kind of assets that the Communities' Engagement Tool can support are images. It has to be noted that the asset contains file metadata (tags, file length, name of file, chunk size, date of creation/update, file title, file description, etc.) and also the ID of the actual file which is required in order to later call the API so as to retrieve the binary data.

This chapter addressed the main data entities of the Communities' Engagement Tool (i.e. "exhibition", "exhibition point", "asset") and their relation to PLUGGY's data model. The Hyperion Engagement Tool takes advantage of the flexibility concerning PLUGGY data model and stores all info in PLUGGY's database by making use of its APIs. To this end, as stated previously the Hyperion citizen stories and Hyperion business owners stories are stored in PLUGGY's back-end in the form of exhibitions.

3.2 Integration with the PLUGGY Authorization Server

PLUGGY is a pluggable platform and multiple applications can access users' content through OAuth2 security protocol. OAuth2 is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. Generally, OAuth2 provides clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without providing credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth2 essentially allows access tokens to be issued to third-party clients by an authorization server, withthe approval of the resource owner. The third-party then uses the access token to access the protected resources hosted by the resource server.

In the case of Hyperion, we consider the third-party clients to be both the Communities' Engagement Tool and HRAP. The resource server is considered to be the back-end of the PLUGGY platform that holds the information concerning citizen and business owners' stories. The resource owners are the users that created the Hyperion stories and who make use of the HRAP and the Communities' Engagement too. Figure 14 depicts the interaction between the components so asfor Hyperion clients (HRAP and Communities' Engagement Tool) to be able to access PLUGGY protected resources (Hyperion Stories). According to figure 14, the client (e.g.Hyperion Communities' Engagement Tool) has to register itself to PLUGGY Authorization services by providing a client ID and a redirect URI. Then, the client is able to participate to PLUGGY OAuth2 workflow which is the following:

1. The user wants to login to the client (either HRAP of Communities' Engagement Tool) and therefore is redirected to PLUGGY login page.
2. The user inserts to the form their PLUGGY credentials.
3. After successful login, the access token is provided to the client from PLUGGY Authorization services.
4. The user through the client (e.g. Engagement Tool) asks a resource (e.g. a Hyperion story) and therefore the access token is passed to PLUGGY Resource server (i.e. PLUGGY back-end).
5. The PLUGGY Resource server validates the token and then it passes to the client the resource.
6. The user is able through the client to view or edit the Hyperion citizen or business owners story.

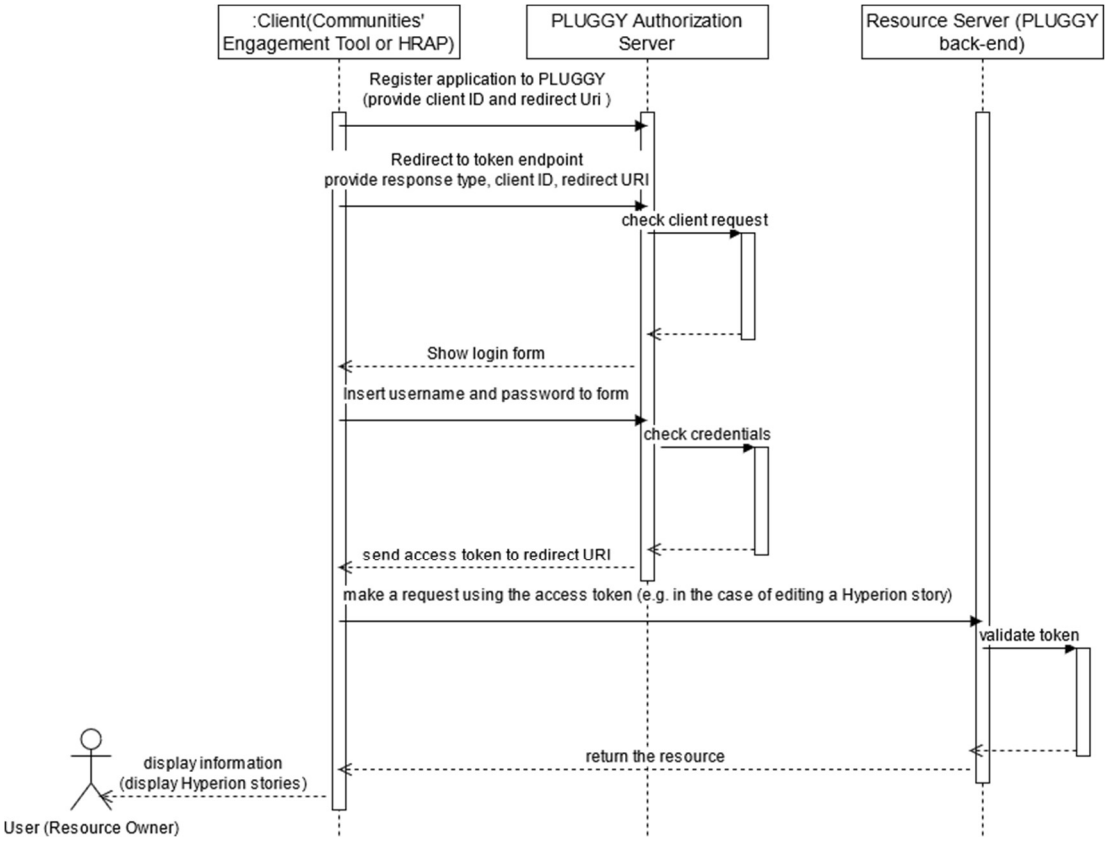

Figure 14: Sequential diagram which depicts the interaction between Hyperion clients (Engagement Tool and HRAP), PLUGGY Authorization Server and PLUGGY Resource Server.

Figure 15 depicts the PLUGGY login page (link). Users are redirected to this page in order to enter their credentials and be authorized by PLUGGY Authorization services.

After successful authentication, the access token is granted to the client and the useris able through the client app to view or edit the Hyperion stories. It has to be noted that through the PLUGGY login page, users are also able to reset password or create new accounts.

Finally, as stated previously, the registration of the application client to PLUGGY Authorisation services is a prerequisite for integrating into PLUGGY security mechanism (i.e. OAuth2 protocol). However, this step is automatic for any developerand it is provided through PLUGGY platform (client registration link.). As shown in figure 16, the developer of the app has to provide a client ID and a redirection URI to PLUGGY platform so as to initialize the integration of the app with PLUGGY OAuth2 mechanism.

Every application can be integrated to PLUGGY platform and the security workflow and the procedure is straightforward. The Hyperion Engagement ICT Tool has already completed this procedure. However it has to be noted that in order to just view the PLUGGY content (without editing permissions), no security integration is needed since most 'retrieval' APIs are not locked (PLUGGY APIs documentation link). This could be the case with HRAP.

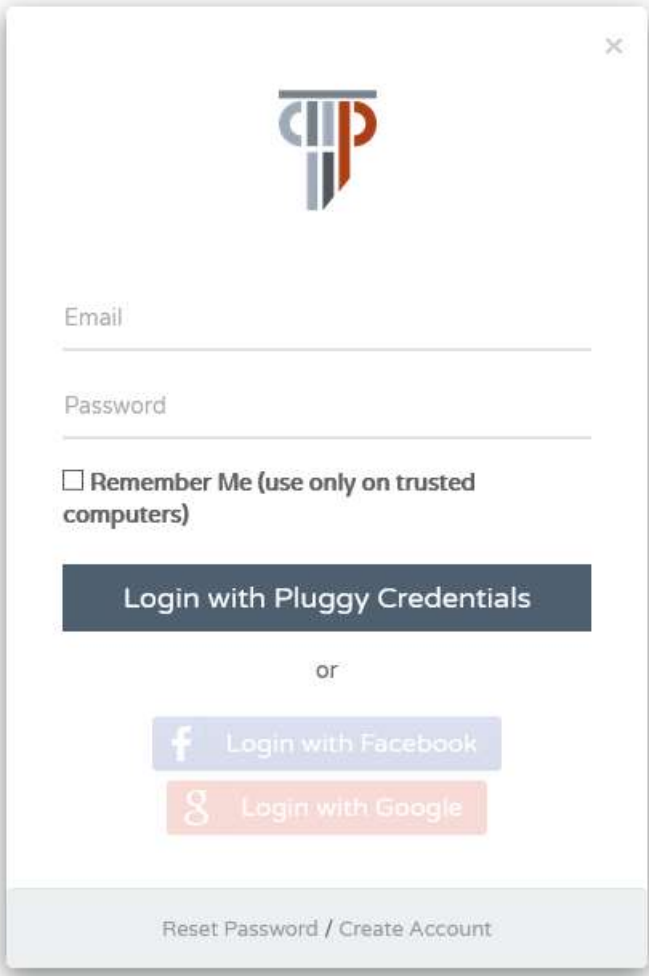

Figure 15: Users are redirected to PLUGGY login page. They enter their PLUGGY credentials in order for the client (HRAP or Communities' Engagement Tool) to get the needed access token and access the resources.

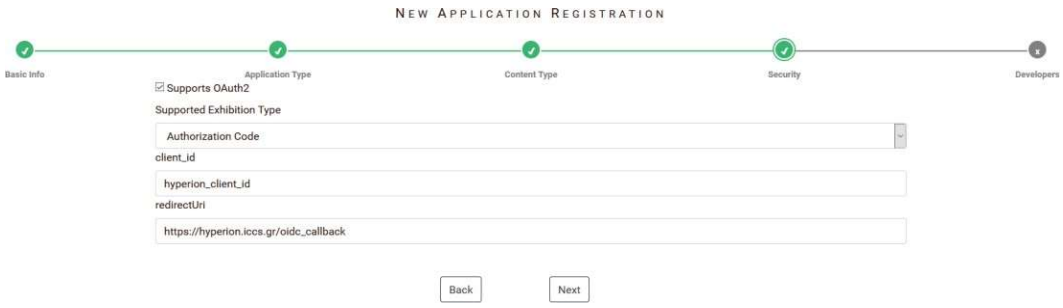

Figure 16: Part of the wizard for registering an external app to PLUGGY OAuth2 mechanism. The user enters a desired client ID and a redirection URI.

4. PLUGGY APIS for retrieving and accessing Hyperion Stories

Since all info of Communities' Engagement Tool is stored in PLUGGY's back-end, the content can be retrieved by making use of its REST APIs. These APIs have been documented online in Swagger UI in this link. As shown in the aforementioned link, the PLUGGY API provides basic CRUD methods to manipulate Exhibitions, Exhibition Points and Assets.

The most important API for HRAP is the one for retrieving Exhibitions/Stories by making use of GET HTTP method.

4.1 Customizable Method for retrieving Exhibition/Stories

The most important API for HRAP is the one for retrieving Exhibitions/Stories by making use of a GET HTTP method.

Table 1: Customizable method for retrieving exhibitions

Customisable method for retrieving exhibitions		
URL:	https://pluggy.eu/api/v1/exhibitions	
Accept:	application/json	
HTTP Method	GET	
URL Query parameters	kind=	For the Engagement Tool it should always have the value "ContentExhibition"
	limit=	Max records per page to return
	type=	Type of story. In the context of Hyperion the values are either "hyperion_citizen_story" or "hyperion_business_owners_story"
	sort=	Results will be sort in specific order. Possible values: "trending", "recent", "title". (e.g. if sort=title then results will be sorted based on Hyperion stories title.)
	tags=	search by tag (stories that contain this tag will be returned)
	page=	number of page to be returned (given that apis make use of pagination)
	q=	full-text query (e.g. q="damage" can be used for searching stories that contain the word "damage")
	lng=	longitude of geo location

	lat=	latitude of geo location
Example of a call for getting Hyperion citizen stories	https://pluggy.eu/api/v1/exhibitions?kind=ContentExhibition&limit=10&type=hyperion_citizen_story	

The response of the aforementioned example is shown below: (It has to be noted that many attributes of the JSON are mostly related to PLUGGY and not to Hyperion. To this end, the fields that are important and related to Hyperion are colored in red. However in the context of the Communities' Engagement Tool development, more fields from this JSON could be exploited in the future if needed.). The most important attributes from the JSON below to be used by Hyperion client (HRAP) are the following:

- Title of the exhibition/story
- Description of the exhibition/story
- ID of the exhibition/story
- Location of the exhibition/story
- ID of the exhibition point of this exhibition/story. This is particularly useful only in the case of business owners stories where this ID can be used by the client through a later API call in order to retrieve more detailed info about the hyperion business story (e.g. local authorities info, customers lost percentage, priority of the issue, etc.)
- The asset IDs and metadata within the exhibition point. (An asset holds the metadata of an image)
- The media IDs of assets (These IDs are important in order to download images with another API calls)
- Tags of the exhibition point (these are tags that describe the exhibition/story)


```
{
  "success": true,
  "data": {
    "result": [
      {
        "_id": "5fe31e14333a33a7f10e4dff",
        "exhibitionPoints": [
          {
            "content": {
              "contentType": "json",
              "content": "6022aad1c5e1d5e4501e938c"
            },
            "legal": {
              "isOwnWork": false
            },
            "assets": [
              {
                "legal": {
                  "isOwnWork": false
                },
                "public": false,
```

```
                "tags": [
                  "wear away",
                  "damage",
                  "winter",
                  "monument"
                ],
                "mediaContent": [
                  {
                    "_id": "6025279d10ed9eee408f7fc4",  //Media ID of asset

                    "filename": "monument.jpg",
                    "contentType": "image/jpeg",
                    "length": 101914,
                    "chunkSize": 261120,
                    "uploadDate": "2021-02-11T12:48:29.024Z",
                    "aliases": null,
                    "metadata": null,
                    "md5": "ca53ff536b0cc015a761ac3e71bdf94c"
                  }
                ],                              //End of mediaContent of asset
                "kind": "ContentAsset",
                "_id": "5fe32179333a33a7f10e4e03",
                "social": {
                  "views": 283,
                  "lastMonthViews": [],
                  "allViews": [],
                  "likes": [],
                  "comments": [],
                  "report": []
                },
                "location": {
                  "geo": {
                    "type": "Point",
                    "coordinates": [
                      -1.9443841343187198,
                      51.42361745514588
                    ],
                    "zoom": 14
                  }
                },
                "updatedAt": "2021-02-11T12:48:28.074Z",
                "createdAt": "2020-12-23T10:52:41.607Z",
                "title": "Lansdowne Monument has suffered weather damage",
                "description": "Monument has been neglected and it has
started to wear away.",
                "owner": "5c96c0a28235f62760f6ef31",
                "creator": "5c96c0a28235f62760f6ef31",
                "type": "image",
                "__v": 19,
                "coverAbsoluteUrl":
"/api/v1/assets/5fe32179333a33a7f10e4e03/media/6025279d10ed9eee408f7fc4/face
book",
                "id": "5fe32179333a33a7f10e4e03"   //Asset ID              }
            ],                                          //End of assets list
            "public": false,
            "tags": [                           //Tags of exhibition point
              "monument",
              "winter",
              "frost action",
              "damage"
            ],
            "mediaContent": [],
            "kind": "ContentExhibitionPoint",
            "_id": "5fe320ec333a33a7f10e4e00",
            "social": {
```

```
        "views": 0,
        "lastMonthViews": [],
        "allViews": [],
        "likes": [],
        "comments": [],
        "report": []
      },
      "updatedAt": "2021-02-09T15:31:29.750Z",
      "createdAt": "2020-12-23T10:50:20.028Z",
      "title": "CH site deterioration",
      "owner": "5c96c0a28235f62760f6ef31",
      "creator": "5c96c0a28235f62760f6ef31",
      "__v": 7,
      "location": {
        "geo": {
          "type": "Point",
          "coordinates": [
            0,
            0
          ],
          "zoom": 12
        }
      },
      "previewMedia": {
        "_id": "6025279d10ed9eee408f7fc4",
        "filename": "monument.jpg",
        "contentType": "image/jpeg",
        "length": 101914,
        "chunkSize": 261120,
        "uploadDate": "2021-02-11T12:48:29.024Z",
        "aliases": null,
        "metadata": null,
        "md5": "ca53ff536b0cc015a761ac3e71bdf94c"
      },
      "coverAbsoluteUrl": "/public/images/imagenotavailable_200.png",
      "id": "5fe320ec333a33a7f10e4e00" //ID of the exhibition point
    }
  ],                                //End of exhibition point list
  "exhibitions": [],        //relevant exhibitions/stories
  "mediaContent": [],
  "kind": "ContentExhibition",
  "arrangements": [],
  "social": {
    "views": 548,
    "lastMonthViews": [],
    "allViews": [],
    "likes": [],
    "comments": [],
    "report": []
  },
  "updatedAt": "2021-02-09T15:31:29.717Z", //Date of exhibition update
  "createdAt": "2020-12-23T10:38:12.178Z",
  "owner": {
    "profile": {
      "firstName": "Nikos",
      "lastName": "Tousert"
    },
    "mediaContent": [],
    "kind": "UserPerson",
    "_id": "5c96c0a28235f62760f6ef31",
    "fullName": "Nikos Tousert",
    "id": "5c96c0a28235f62760f6ef31"
  },
  "creator": {
    "profile": {
```

```
            "firstName": "Nikos",
            "lastName": "Tousert"
          },
          "mediaContent": [],
          "kind": "UserPerson",
          "_id": "5c96c0a28235f62760f6ef31",
          "fullName": "Nikos Tousert",
          "id": "5c96c0a28235f62760f6ef31" //ID of story/exhibition creator
        },
        "__v": 1,
        "description": "The Lansdowne Monument above Cherhill, near Calne,
has suffered weather damage.",     //Description of exhibition/story
        "legal": {
          "isOwnWork": false
        },
        "location": {  //location of exhibition/story
          "geo": {
            "type": "Point",
            "coordinates": [
              -1.9460804108354293,
              51.42286816395111
            ],
            "zoom": 14
          }
        },
        "public": false,
        "tags": [                           //Tags of exhibition/story
              "monument",
              "winter",
              "frost action",
              "damage"
            ],
,
        "title": "Notable landmark in Wiltshire has deteriorated. ",//story
title
        "type": "hyperion_citizen_story",     //type of exhibition
        "weight": 1,
        "previewMedia": {
          "_id": "6025279d10ed9eee408f7fc4",
          "filename": "monument.jpg",
          "contentType": "image/jpeg",
          "length": 101914,
          "chunkSize": 261120,
          "uploadDate": "2021-02-11T12:48:29.024Z",
          "aliases": null,
          "metadata": null,
          "md5": "ca53ff536b0cc015a761ac3e71bdf94c"
        }
      }
    ],
    "total": 1
  }
}
```


4.2 GET the content of the Exhibition Point for Business owners stories

As shown in the following table, the ID of the Exhibition Point that is contained in the JSON described in chapter 4.1 can be used to get the content (json file) of the relevant

Exhibition Point. The response is a json file (attachment). It has to be noted that this is applicable only to business stories (in the case of citizen stories the client will get an empty json)

Table 2: Method for retrieving Exhibition Point content

Method for retrieving Exhibition Point content		
URL:	https://pluggy.eu/api/v1/exhibitions/{exhibition_id}/exhibition-points/{exhibition_point_id}/content	
content-disposition:	attachment; filename= "content.json"	
HTTP Method	GET	
URL path parameters	exhibition_id	The ID of the relevant exhibition/story
	exhibition_point_id=	The ID of the exhibition point
Example of a call for getting Hyperion citizen stories	https://pluggy.eu/api/v1/exhibitions/5fe321d5333a33a7f10e4e06/exhibition-points/5fe3233c333a33a7f10e4e08/content	

The example of the call in the previous table will force the download of a json file as an attachment. This JSON file is the content of the exhibition point and contains further info concerning the Hyperion story. In the case of a citizen story the response would always be an empty json whereas in the case of a business story it would look like figure 18. The priority and criticality values along with their meanings are listed in table 3.

{"business_damage_description":"Extensive damage to the building. The building has completely flooded over the weekend. As such, the restaurant has been closed",

"business_risk_description":"Since the restaurant is closed for the past week, there is the risk to loose high percentage of customers.",

"local_authorities_info":"There is absolute devastation in the surrounding area. A fundraiser needs to be set up to help families who lost everything. police has informed us that all businesses are under water.",

"customers_lost_percentage":"-80",

"criticality":4,

"priority":4}

Figure 18: Example of Exhibition Point Content for the business owners story. This JSON contains the damage of the business, the risk of the business, information useful to local authorities and also the priority and criticality of the issue

Table 3: Priority and Criticality values and their meaning

Priority	Criticality	Text
5	5	Urgent
4	4	High
3	3	Medium
2	2	Low
1	1	Very low

4.3 GET method for retrieving asset/image metadata

As shown in the following table, the ID of the asset that is contained within the exhibition point in the JSON described in chapter 4.1 can be used to get the metadata of the relevant image. Based on chapter 4.1, the exhibition point contains an asset with ID: 5fe32179333a33a7f110e4e03. Thereafter, table 4 contains the example of the request for getting the aforementioned asset metadata.

Table 4: GET method for retrieving Asset metadata

Method for retrieving Asset/image metadata		
URL:	https://pluggy.eu/api/v1/assets/{asset_id}	
accept:	application/json	
HTTP Method	GET	
URL path parameter	asset_id	The ID of the relevant asset
Example of a call for getting Hyperion citizen stories	https:// pluggy.eu/api/v1/assets/5fe32179333a33a7f10e4e03	

The result of the aforementioned example call is a JSON with the metadata of the image as shown below:


```
{
  "success": true,
  "data": {
    "legal": {
      "isOwnWork": false
    },
    "public": false,
    "tags": [
```

```
      "wear away",
      "damage",
      "winter",
      "monument"
    ],
    "mediaContent": [
      {
        "_id": "6025279d10ed9eee408f7fc4", //ID of the actual image
        "filename": "monument.jpg",
        "contentType": "image/jpeg",
        "length": 101914,
        "chunkSize": 261120,
        "uploadDate": "2021-02-11T12:48:29.024Z",
        "aliases": null,
        "metadata": null,
        "md5": "ca53ff536b0cc015a761ac3e71bdf94c"
      }
    ],
    "kind": "ContentAsset",
    "_id": "5fe32179333a33a7f10e4e03",
    "social": {
      "views": 285,
      "lastMonthViews": [],
      "allViews": [],
      "likes": [],
      "comments": [],
      "report": []
    },
    "location": {
      "geo": {
         "type": "Point",
        "coordinates": [
          51.42361745514588,
          -1.9443841343187198
        ],
        "zoom": 14
      }
    },
    "updatedAt": "2021-02-11T12:48:28.074Z",
    "createdAt": "2020-12-23T10:52:41.607Z",
    "title": "Lansdowne Monument has suffered weather damage",
    "description": "Monument has been neglected and it has started to wear
away.",
    "owner": {
      "profile": {
        "firstName": "Nikos",
        "lastName": "Tousert"
      },
      "mediaContent": [],
      "kind": "UserPerson",
      "_id": "5c96c0a28235f62760f6ef31",
      "fullName":  "Nikos  Tousert",
      "id": "5c96c0a28235f62760f6ef31"
    },
    "creator": {
      "profile": {
        "firstName": "Nikos",
        "lastName": "Tousert"
      },
      "mediaContent": [],
      "kind": "UserPerson",
      "_id": "5c96c0a28235f62760f6ef31",
      "fullName":  "Nikos  Tousert",
      "id": "5c96c0a28235f62760f6ef31"
    },
```

```
    "type":
    "image","__v":
    19,
    "coverAbsoluteUrl":
"/api/v1/assets/5fe32179333a33a7f10e4e03/media/6025279d10ed9eee408f7fc4/face
book",
    "id": "5fe32179333a33a7f10e4e03"
  }
}
```


As shown in figure 19, the asset metadata contains info concerning the image, title, description, location,tags, update and creation date, etc.

4.4 GET method for retrieving the image as an attachment

The JSON response described in the previous chapter holds the attribute "_id" which is located within mediaContent list. As shown in the following table, we can make useof this ID(media_id) in order to get the actual image as an attachment.

Table 5: GET method for retrieving image as an attachment

Method for retrieving the image as an attachment		
URL:	https://pluggy.eu/api/v1/assets/{asset_id}/media/{media_id}	
accept:	Image/webp,*/*	
HTTP Method	GET	
URL path parameters	asset_id	The ID of the relevant asset which can be found in the exhibition point assets list
	media_id=	The ID of the media which can be found in asset metadata
Example of a call for getting Hyperion citizen stories	https://pluggy.eu/api/v1/assets/5fe32179333a33a7f10e4e03/media/6025279d10ed9eee408f7fc4	

After making this call, the client (HRAP) will get the actual image as an attachment. It has to be noted that an exhibition point may have many assets and each asset exactly one media file, such as an image.

5. Mobile phone application

5.1 General description

Hyperion application aims to create a user-friendly tool with the latest user interface and usability issues/trends which is focused on museum enthusiasts and active citizens' or travelers' needs. The general idea of the app is the creation of an Asset, an elementary unit of content in PLUGGY, a media file with an identified owner, a title, a set of tags, and a license, which specifies how this file can be reused. Moreover, emphasis is placed on the retrieval of the assets created in and their innovative presentation to users, to experience and better understand the changes imposed by climate change and extreme events.

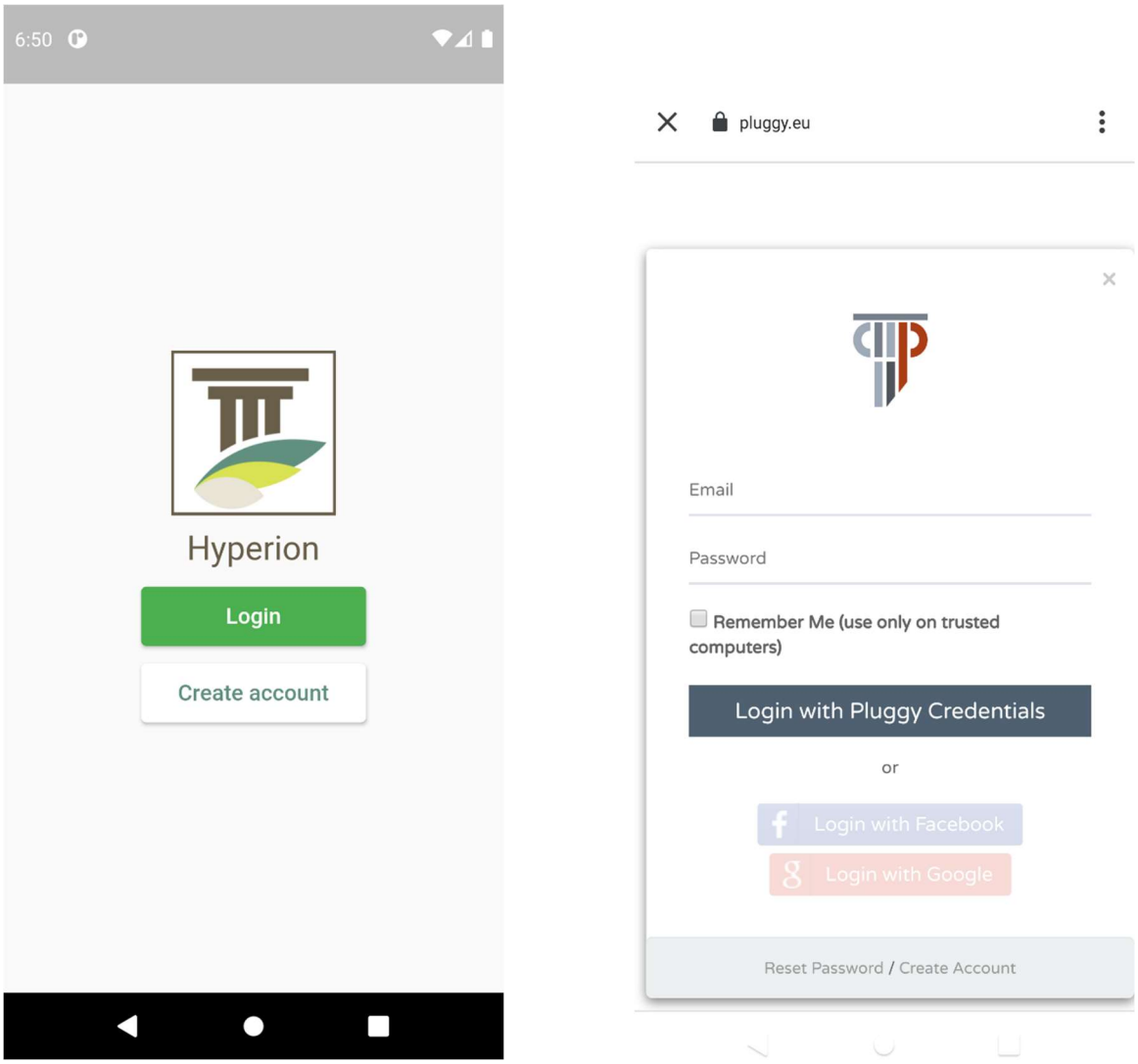

Figure 17: Login Screen (left) and PLUGGY authorization (right).

The app also utilizing PLUGGY's API. The Hyperion app makes use of PLUGGY social platform, for the same reasons Communities' Engagement ICT Tool implements it. The user logs in or creates an account in the application through the PLUGGY platform. The basic functionality, which is the creation and posting of an asset using PLUGGY's REST API, includes several steps such as specifying the location of the event that happened, adding a title and some tags for the asset to describe in a better way the situation of the event and finally selecting an image from photo gallery that illustrates the event.

5.2 Integration with PLUGGY Authorization Sever

Starting the application, the user can log in or create a new account through PLUGGY, as depicted in Figure 2. Hyperion mobile app is regarded as a third-party client that can access users' content through Open Authorization (OAuth 2.0) security protocol. By selecting the Login or Register button the client participates in PLUGGY Auth 2.0 workflow, as it follows:

1. The user wants to login or register to the application and therefore is redirected to PLUGGY login/registration page.
2. The user inserts in the form their PLUGGY credentials, or creates a new account.
3. After the successful login, the access token is provided to the client from PLUGGY authorization services.
4. The user is redirected back to the application, provided with an access token that will enable them to have access to the protected resources hosted by the resource server.
5. Actions like posting an asset or viewing/having access to other entities, will be validated by PLUGGY resource server, using the authentication token.

5.3 User Interface

Having successfully logged in, users asked whether they allow their location to be detected, to be displayed on the map. Users' coordinates are used not only as a part of the creation of the Asset, but at the same time are included in various parts of the interface so that the application is more user friendly. If users consent to the use of GPS, they are redirected to an intro page. Users' location has been updated and is presented in the center of the screen. Two basic buttons, one that leads to an overview with previous reports that have been done, and a second, that allows the creation of a new report, are the main functionalities that provided.

5.4 Creating a Story

Clicking the "Add new report" button, the map with the user's location is displayed in the screen. *(If something went wrong or the user wants to find the address for which he wants to make a new report, he could search for the location or navigate in the map)*. At the bottom right of the screen, there is a button that focus back to mobile's coordinates in the map, and in the bottom center there is another button that leads to the steps that the user must follow to upload the post.

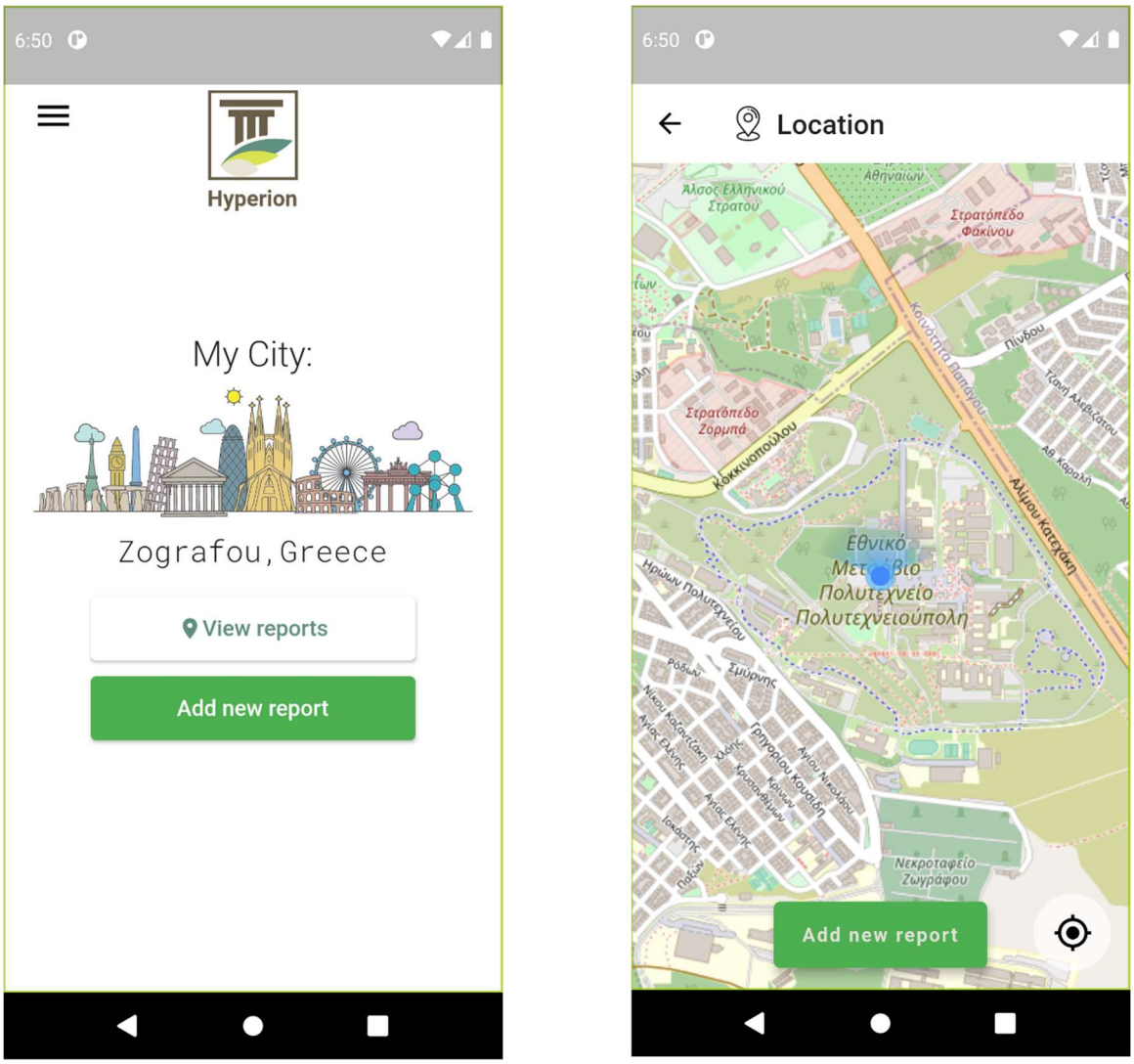

Figure 18: Introduction Screen (left) and location service (right).

The process of uploading a post, is described in detail, so that it is easier for users to avoid unnecessary steps that may confuse them and may lead to a time-consuming procedure. The steps that users must follow are:

(a) Adding a photo. An image is selected from the gallery, which has been previously taken by the user and will be uploaded to show the damage on the monument or the historical site.

(b) Defining a title. A simple and brief description of the event.

(c) Selecting ICOMOS tags.

(d) Defining tags. Information that will give a more complete picture of the event and automatically link the story with other stories/events that have already created in PLUGGY platform and have common tags

Once all this information is gathered, it is combined into an Asset and then posted to PLUGGY. Having completed the process, a notification appears on the screen informing users that the report posted successfully.

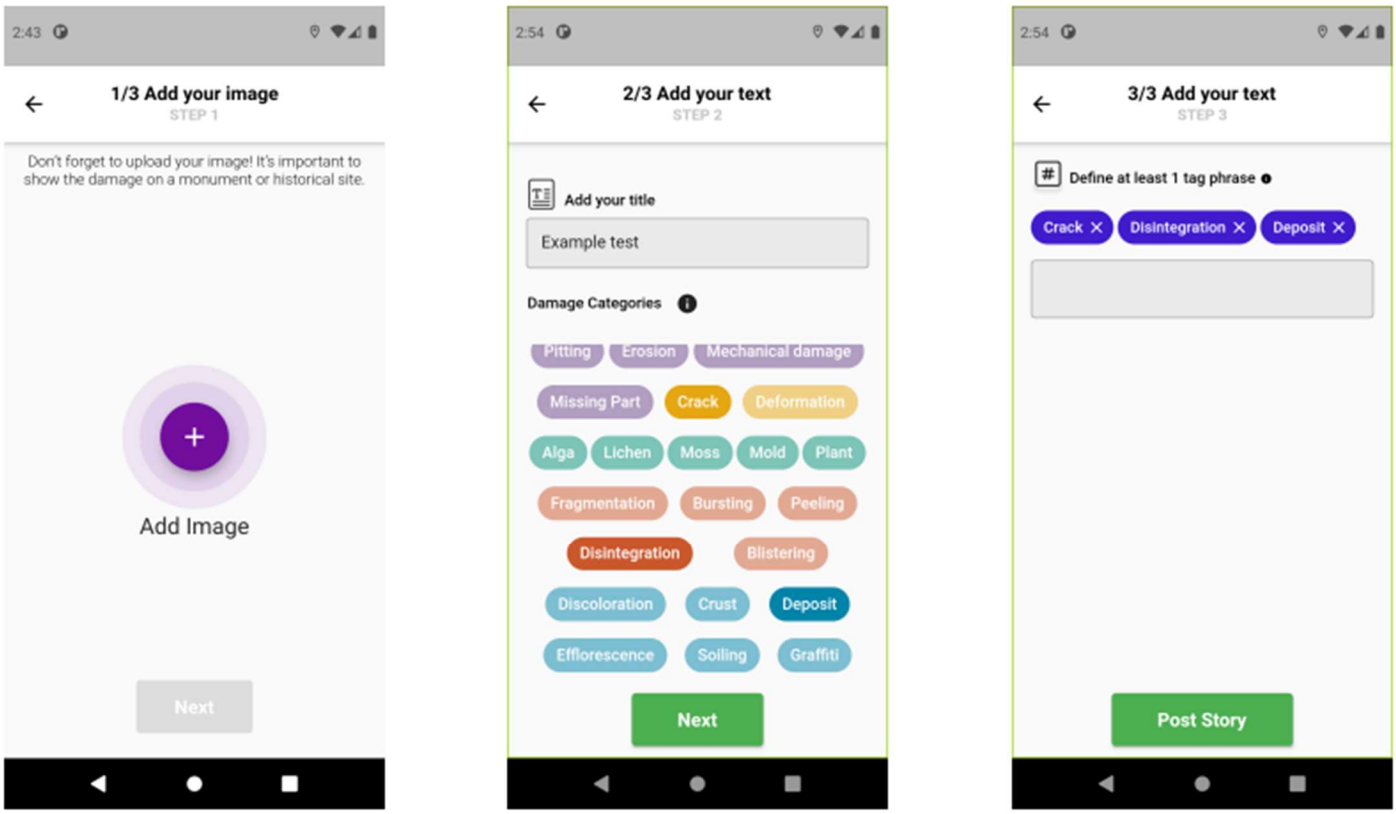

Figure 19: Step 1: Add image, Step 2: Add title and ICOMOS tags, Step 3: Add general tags.

5.5 ICOMOS damage categorization

Hyperion application also provides recommended tags chosen from the main categories that are defined in ICOMOS-ISCS glossary on stone deterioration patterns. ISCS aims at facilitating the publication, dissemination and presentation of state-of-the-art reviews on pre-identified issues. The ISCS glossary constitutes an important tool for scientific discussions on decay phenomena and processes. The 5 categories that are used in the application are: Crack and deformation, Detachment, Features induced my material loss, Discoloration and deposit, Biological colonization. Every category includes definitions that describe different damages. Some of the damage categories definitions are:

- CRACK: Individual fissure, clearly visible by the naked eye, resulting from separation of one part from another.
- FORMATION: Change in shape without losing integrity, leading to bending, buckling or twisting of a stone block.
- BLISTERING: Separated, air-filled, raised hemispherical elevations on the face of

stone resulting from the detachment of an outer stone layer. This detachment is not related to the stone structure.

- BURSTING: Local loss of the stone surface from internal pressure usually manifesting in the form of an irregularly sided crater.
- EROSION: Loss of original surface, leading to smoothed shapes.
- MECHANICAL DAMAGE: Loss of stone material clearly due to a mechanical action.
- DEPOSIT: Accumulation of exogenic material of variable thickness.
- DISCOLOURATION: Change of the stone color in one to three of the color parameters: hue, value and chroma.
- BIOLOGICAL COLONIZATION: Colonization of the stone by plants and micro-organisms such as bacteria, cyanobacteria, algae, fungi and lichen.

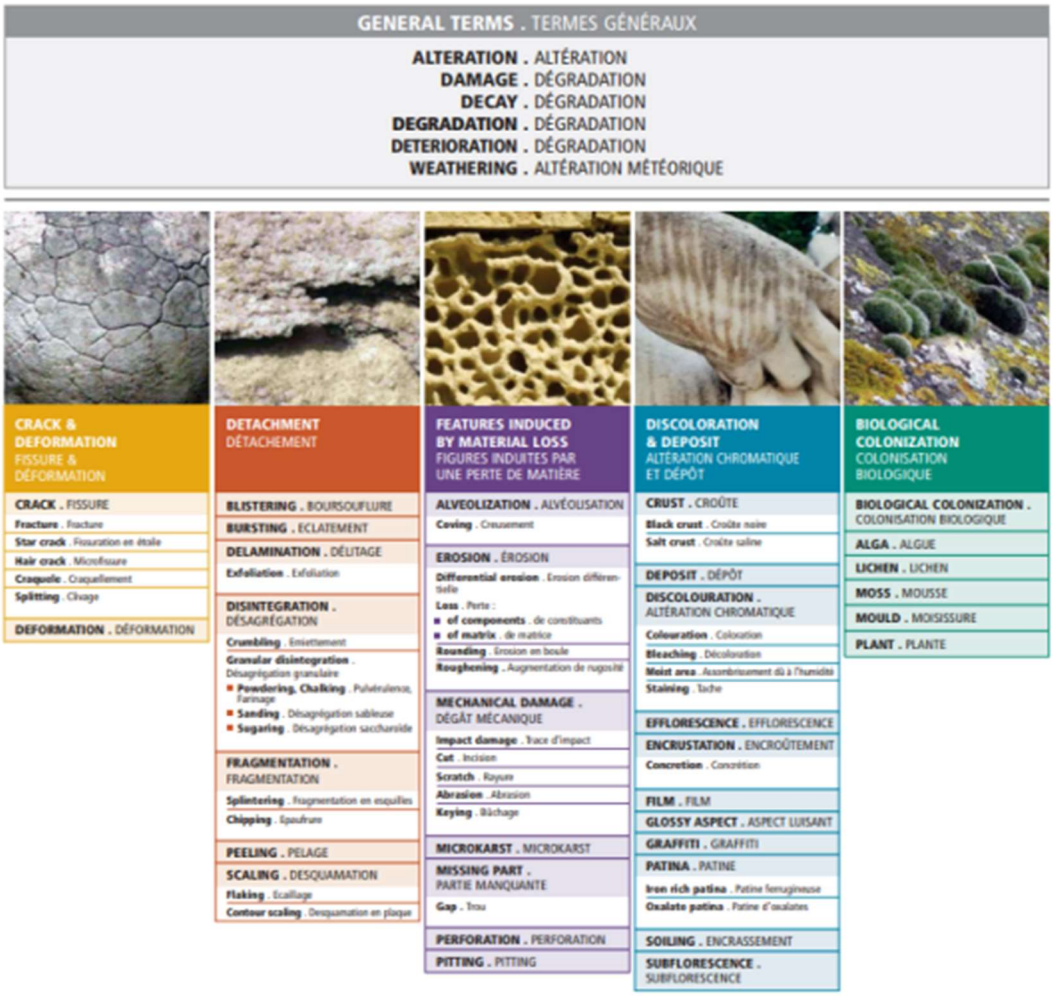

Figure 20: ICOMOS categories.

5.6 Overview

5.6.1 User stories

Users can choose between viewing a map with all the stories near them or a map that displays only the assets that they have added. Proceeding to the map, there are markers in the places where the reports took place. By clicking to a marker, a card in the bottom of the screen appears which updates and shows all the important information that have been retrieved for the specific point. An image of the event, the title, and an indicative tag, describe briefly the asset that has been selected. Also, there is a button that leads the user to the PLUGGY platform, in case they want to explore more about other assets and events.

5.6.2 Around Me Section

In the menu options, users are given the opportunity to inspect a map with assets that have been posted around them. The point in the map that is defined as the center of the circle that includes all those events, is the real time location of the user, as it was calculated by the GPS device. Every asset is represented with a card in the bottom of the screen, the same approach that was followed in the User Stories section. This implementation increases the experience of users, making them part of a greater picture and showing a view of the application use in general.

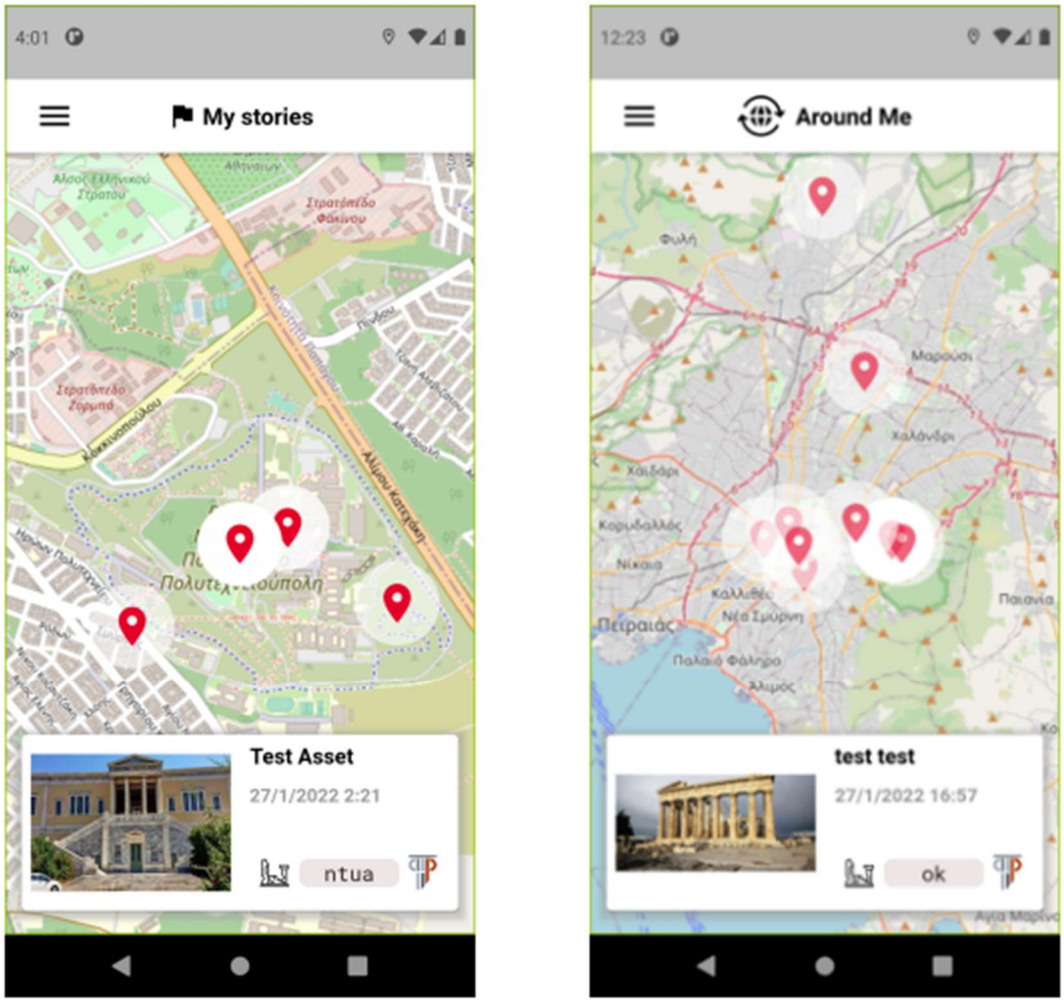

Figure 21:"My stories" screen (left) and "Around me" screen (right).

6. Conclusions

This deliverable reports about the Communities' Engagement Tool and the Hyperion Mobile Application and their integrationinto PLUGGY platform. The functionalities and the design of the tools were based on Deliverables D2.2 and D2.3 which document the overall Hyperion use cases and architecture but they were also driven by project discussions during which all partners' comments were taken into consideration. In addition to the presentation of tool's functionalities through screenshots and documentation, a separate chapter has been devoted for documenting in detail the PLUGGY APIs needed by HRAP for retrieving resources

generated by this tool (i.e. citizen and business stories). Moreover, the integration of the tool with PLUGGY's authorization services as well as the data modelof the tool, are all described in detail. This document aims to serve as a roadmap for both the other Hyperion components that wish to analyze and retrieve Hyperion stories through PLUGGY APIs and also for users of the tool.