www.nautilos-h2020.eu

Organisation CNR
Department ISTI

Fully developed Graphic User Interface

Accompanying report

Date: 05/05/2022
Doc. Version: v.0.6

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101000825 (NAUTILOS). This output reflects only the author's view and the European Union cannot be held responsible for any use that may be made of the information contained therein.

Document Control Information

Settings	Value
Deliverable Title	Fully developed Graphic User Interface
Work Package Title	Data Management
Deliverable number	D8.7
Description	This deliverable consist of the deployment of the project's web portal through which the storage and management of the integrated data regarding NAUTILOS, and the services defined and implemented during Task 8.4 have been organised. The implemented tools and services for data management and visualisation, designed in close connection with end-users requirements, are described.
Lead Beneficiary	CNR-ISTI
Lead Authors	Marco Tampucci, Gabriele Pieri, Federico Volpini
Contributors	ETT; NIVA
Submitted by	Gabriele Pieri
Doc. Version (Revision number)	0.6
Sensitivity (Security):	Public
Date:	05/05/2022

Document Approver(s) and Reviewer(s):

NOTE: All Approvers are required. Records of each approver must be maintained. All Reviewers in the list are considered required unless explicitly listed as Optional.

Name	Role	Action	Date
Flávio Martins	Reviewer	*Approve*	04/05/2022

Document history:

The Document Author is authorised to make the following types of changes to the document without requiring that the document be re-approved:

- Editorial, formatting, and spelling
- Clarification

To request a change to this document, contact the Document Author or Owner.

Changes to this document are summarised in the following table in reverse chronological order (latest version first).

Revision	Date	Created by	Short Description of Changes
V0.1	21/01/22	Gabriele Pieri, Marco Tampucci	First structure of the deliverable ready, shared with other contributors.

V0.3	11/02/22	Gabriele Pieri	Finalised structure.
V0.4	28/03/22	M. Tampucci, F. Volpini	Preliminary version finalised.
V0.5	6/04/22	M. Tampucci, G. Pieri	Further update.
V0.6	5/05/22	G. Pieri	Final version after revision

Configuration Management: Document Location

The latest version of this controlled document is stored in <location>.

Nature of the deliverable		
R	Report	
DEC	Websites, patents, filing, etc.	
DEM	Demonstrator	
O	Other	√

Dissemination level		
PU	Public	√
CO	Confidential, only for members of the consortium (including the Commission Services)	

ACKNOWLEDGEMENT

This report forms part of the deliverables from the NAUTILOS project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101000825. The Community is not responsible for any use that might be made of the content of this publication.

NAUTILOS - New Approach to Underwater Technologies for Innovative, Low-cost Ocean observation is an H2020 project funded under the Future of Seas and Oceans Flagship Initiative, coordinated by the National Research Council of Italy (Consiglio Nazionale delle Ricerche, CNR). It brings together a group of 21 entities from 11 European countries with multidisciplinary expertise ranging from ocean instrumentation development and integration, ocean sensing and sampling instrumentation, data processing, modelling and control, operational oceanography and biology and ecosystems and biogeochemistry such, water and climate change science, technological marine applications and research infrastructures.

NAUTILOS will fill-in marine observation and modelling gaps for chemical, biological and deep ocean physics variables through the development of a new generation of cost-effective sensors and samplers, the integration of the aforementioned technologies within observing platforms and their deployment in large-scale demonstrations in European seas. The fundamental aim of the project will be to complement and expand current European observation tools and services, to obtain a collection of data at a much higher spatial resolution, temporal regularity and length than currently available at the European scale, and to further enable and democratise the monitoring of the marine environment to both traditional and non-traditional data users.

NAUTILOS is one of two projects included in the EU's efforts to support of the European Strategy for Plastics in a Circular Economy by supporting the demonstration of new and innovative technologies to measure the Essential Ocean Variables (EOV).

More information on the project can be found at: https://www.nautilos-h2020.eu/

COPYRIGHT

TABLE OF CONTENTS

ACKNOWLEDGEMENT ... 4

COPYRIGHT ... 4

TABLE OF CONTENTS ... 5

EXECUTIVE SUMMARY ... 6

LIST OF FIGURES ... 7

LIST OF ACRONYMS AND ABBREVIATIONS ... 7

I. INTRODUCTION: TASK OBJECTIVES AND DELIVERABLE OUTLINE ... 8

II. DATA STORE AND MANAGEMENT ... 8

1. @lgawarning ... 10

III. STRUCTURE OF THE WEB-INTERFACE/PORTAL ... 10

1. Framework description ... 11
2. Implementation of the Data Visualisation Layer ... 12
3. Access/security ... 13

IV. GUI TOOLS AND SERVICES (SEARCH AND DISCOVERY) ... 14

1. Data Filtering ... 15
2. Data Rendering ... 16

V. DEVELOPED DEVICES ... 18

1. Smartphone microplastic NIR scanner device ... 18
2. @lgawarning mobile app and device ... 20

VI. CONCLUSIONS ... 23

APPENDIX 1: REFERENCES AND RELATED DOCUMENTS ... 24

EXECUTIVE SUMMARY

Following the store and management of data organised in Task 8.1, this document represents an accompanying report for the tools and services developed in Task 8.4 to visualise and use data collected within the NAUTILOS project. The web portal Graphical User Interface, integrated into the project website, is an end-to-end data and information management, search, discovery and access system providing the following: (i) a summary of the data and products available through a catalogue and searching features to facilitate end-users access; (ii) a digital map based on layers displaying data and data products compatible with OGC standards and following INSPIRE recommendations; (iii) dynamic tools with panning and zooming facilities allowing users to view data, data products and metadata, and to combine and use the map layers; and, (iv) services for accessing, data, metadata, and products.

NAUTILOS

List of Figures

Figure 1 NAUTILOS Map viewer. 9
Figure 2 Scheme of NAUTILOS GUI data flow. 10
Figure 3 Example of Layers object. 12
Figure 4 Example of a Layer definition. 13
Figure 5 Example of the Data Visualisation Layer. 14
Figure 6 Zoom Controller. 14
Figure 7 Example of Layer Controller. 15
Figure 8 Filters button. 15
Figure 9 Time Series controller. 15
Figure 10 Filter form example. 15
Figure 11 Retrieved data are displayed as clusters. 16
Figure 12 (a) Markers clusters and the shape of one of them; (b) by increasing the zoom level, the clusters will be divided into smaller clusters; (c) if the markers are distant enough from each other, they are displayed. 17
Figure 13 Data rendered as time series. In detail, green and red markers belong to different layers. 17
Figure 14 Popup example. 18
Figure 15 Smartphone app and Near Infra-Red scanner for plastics identification. 18
Figure 16 Raw spectra for the 9 different polymers acquired by the smartphone NIR Scanner. 19
Figure 17 Spectra from the polymers after data treatment (log, average, 2nd derivative) from the selected wavelength range of 850-950 nm. 19
Figure 18. Multivariate model showing the separation of the different polymer, except for HDPE and LDPE which overlap slightly (blue and purple symbols). 20
Figure 19 @lgawarning Setup 21
Figure 20: Report visualisation on @lgawarning app. 21
Figure 21: @lgawarning map viewer and list reports viewer. 22
Figure 22: @lgawarning report creation page 22

List of Acronyms and Abbreviations

Abbreviation	Definition
GUI	Graphical User Interface
NIR	Near Infra-red
CS	Citizen Science
UX	User experience
WMS	Web Map Services
WFS	Web Feature Services
AI	Artificial Intelligence
OGC	Open Geospatial Consortium
CSS	Cascading Style Sheets

I. Introduction: task objectives and deliverable outline

This document represents an accompanying report to the developed Graphical User Interface (GUI), its deployment onto the NAUTILOS web portal, and the tools and services defined and implemented during Task 8.4.
The tools and services for data management and visualisation were planned and implemented in close connection with the end user's requirements defined in previous deliverables of WP8. Services and tools have been designed and implemented, maintaining the predefined roles to grant access and fruition of the interface to different users.

Two different hardware devices have been implemented among the various tools. Firstly, a smartphone microplastic Near-InfraRed (NIR) scanner device for identifying plastics has been developed and will be described. Furthermore, the integration of another device for collecting and identifying algae species, which will be used fruitfully in Citizen Science (CS) campaigns, has been developed and described. These two devices have been integrated into the NAUTILOS data infrastructure, and their data are flowing to it, and they are part of the project's products.
All the collected NAUTILOS data and products are available and usable through the services deployed through the GUI described in this report.

The report is arranged as follows: section II reports a description of the integration with the data store and management, emphasising the data management infrastructure and connection between it and the web interface; section III describes the web interface focusing on its organisation and structure, the adopted technologies and framework and how the access is monitored; section IV describes the functionalities and tools provided by the interface and how to use them; section V describes the hardware devices as mentioned earlier, used and developed inside the project, through specific tools but supporting and providing data through the NAUTILOS data infrastructure. Finally, section VI concerns the deliverable conclusions.

II. Data store and management

The development of the web portal is adopting and following the recommendations defined in the deliverable D.8.2. In line with the project objectives, and in order to have an end-to-end data and information management, search, discovery and access system, the web portal and, in particular, the web data portal is an essential component of the system.

The portal implements a WebGIS viewer with a dynamic map that provides the user with zooming and panning features, as well as with features to include and select data and data-products. Each dataset is enriched with metadata, and the viewer offers the user tools to preview these metadata contents (Figure 1)

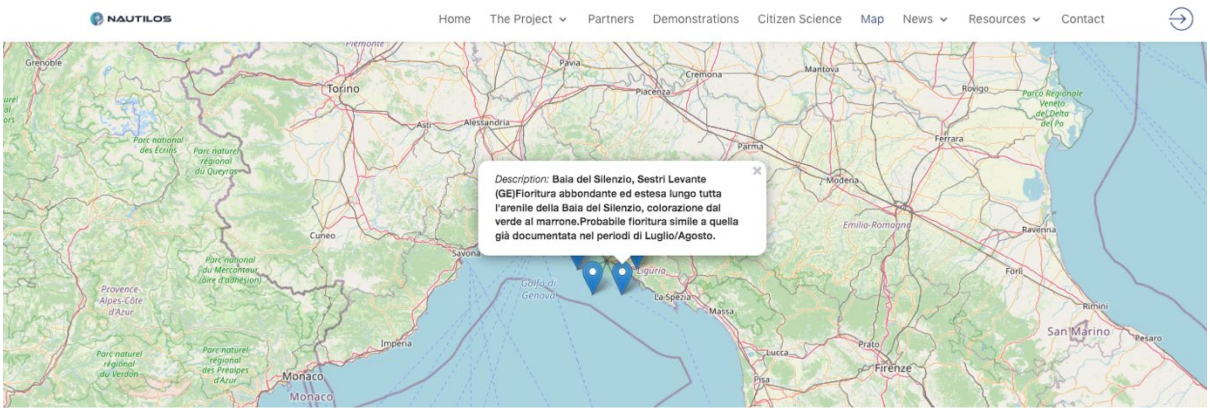

Figure 1 NAUTILOS Map viewer.

As soon as new data start flowing into the NAUTILOS Data Infrastructure (after M18), the current version of the system will be enriched with a catalogue tooltip that will guide the user into a user-friendly data visualisation experience.

As described in the Data Management Plan (DMP, deliverable D1.3) and the deliverables D8.1 and D8.2, the NAUTILOS infrastructure organises the data within two primary data management and publishing systems: ERDDAP[1] and GeoServer[2]. While ERDDAP enables easy data access for time-series, in situ sparse data, remote sensing or gridded data, GeoServer offers the framework to host and publish vectorial data and map layers.
Both the systems implement common OGC standards, and data can be recalled by, e.g. WMS/WFS and REST queries.
Although ERDDAP and GeoServer are very powerful and easy tools, they are designed for professional users and technicians; hence the role of the NAUTILOS Map viewer is to implement a visual user experience (UX) that enables any possible NAUTILOS stakeholder to consume and use NAUTILOS data and contribute to maximising the NAUTILOS outcomes.

The scheme shown in Figure 2 summarises the NAUTILOS data flow focusing on components used for data visualisation on the GUI and sharing.

The NAUTILOS Map viewer is also a tool that the engaged Citizen Scientists can use to see and discuss their data while participating in the NAUTILOS CS activities and campaigns (see also WP10 and WP12). CS data are going to flow into the NAUTILOS system by a series of CS tools and applications, which dataflow, GUI and UX are described in brief in the following.

---

[1] Simons, R.A. 2020. ERDDAP. https://coastwatch.pfeg.noaa.gov/erddap . Monterey, CA: NOAA/NMFS/SWFSC/ERD.
[2] Open Source Geospatial Foundation. http://geoserver.org/

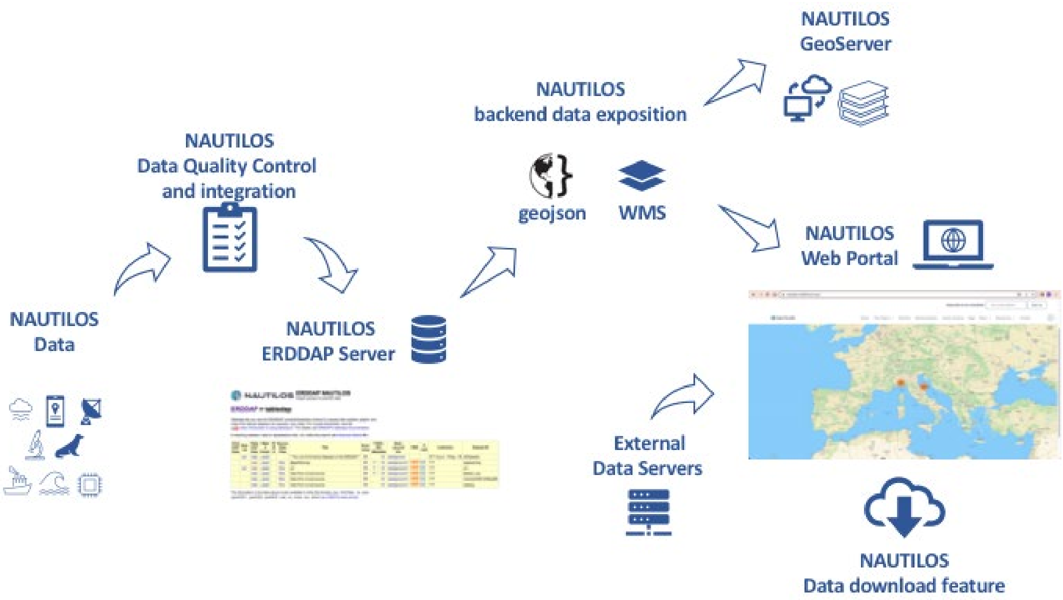

Figure 2 Scheme of NAUTILOS GUI data flow.

1. @LGAWARNING

@lgawarning is a citizen science app to collect, classify and measure the quantity of algae in a water sample directly at sea. The application is a mobile application taking a photo of the sample (see section V-2 for details on the methodology), gets the datum (time, latitude and longitude) from the mobile phone, collects the notes from the user and stores all these data into a remote database that is generating a data product in NAUTILOS.
A dedicated ERDDAP dataset hosts the collected metadata (e.g. sampling site position, sample volume, notations) and the pictures.

The designed workflow enables the possibility to apply AI algorithms to classify the species (and make warnings if a harmful species is identified) in delayed mode and attach this information (and annotations) to the original data (picture source).

III. STRUCTURE OF THE WEB-INTERFACE/PORTAL

The Web Interface oversees displaying the data resulting from the ERDDAP datasets available at the data server. These datasets can contain both data provided by the project partners and data gathered through the Citizen Science applications and devices as described in the Deliverable D8.8 "Citizen Science tools and interface". The number of datasets will increase during the NAUTILOS project.
Thus, the Web interface will provide a map layer where all the data will be displayed. In order to correctly display the data, the GUI will also be compatible with the OGC standards[3] (e.g., WMS) and Georeferenced data format (e.g., GeoJSON[4]). Finally, the Web interface will also provide functionalities such as panning and zoom to explore the map and the possibility to view, hide or filter the data gathered from the data server.

[3] Open Geospatial Consortium, https://www.ogc.org/standards
[4] GeoJSON Working Group, https://geojson.org/

In detail, this section describes the technologies, frameworks and tools exploited to develop the web portal as described before. The first subsection discusses the framework adopted to develop the Web portal. Subsection 2 describes the implementation of the functionalities mentioned above, and Subsection 3 illustrates the access modalities.

1. Framework description

The GUI, as discussed before, will be composed of an explorable digital map where the data retrieved from the data server will be geographically displaced. This is obtained by integrating into the Webpage a WebGIS client. Among all the available WebGIS clients, Leaflet[5] has been chosen. Leaflet is an open-source WebGIS client based on JavaScript and supports most mobile and desktop browsers. It can easily interface with WMSs, and Vector Layers or load the features from GeoJSON files to show data, and relative information, on a Tile Layer (generally a map). It manages several object types like:

- Raster Types (TileLayer and ImageOverlay)
- Vector Types (for instance, Path and Polygon)
- Grouped Types (LayerGroup, FeatureGroup and GeoJSON)
- Controls (for example, zoom and layers).

Leaflet is also supported by a broad community that contributes to enriching the WebGIS functionalities by developing and freely releasing a set of dedicated plugins.
Another key value is that Leaflet is fully compatible and easy integrable in Angular[6]. Angular is a powerful web development platform based on TypeScript, designed to build web applications that can be executed on any web browser (a compiled Angular application is "translated" into JavaScript language), even smartphone ones. In the current release, the GUI is straightforward and light; in this scenario, the usage of Angular could be an avoidable overload to the GUI itself; despite this, Angular has been taken into account because, in the forthcoming releases, it will be a valuable tool since a more complex GUI could require a powerful tool to implement and manage all the possible and future functionalities.
Thus, the Data Visualisation Layer has been developed in pure JavaScript, integrating the Leaflet Library and jQuery[7], a common JavaScript library used to simplify the development. The GUI has also been optimised and extended with the usage of two Leaflet plugins:

- *Markercluster*[8]: this plugin groups markers of a layer that are close to each other, showing clusters of markers based on the map's zoom level; this plugin is used to avoid performance loss due to the high number of available markers to be displayed.
- *TimeDimension*[9]: this plugin adds functionalities to manage time-based data; thanks to this plugin, a slider bar is shown and allows to view temporal evolution of the data.

In addition, to guarantee usability in every context, such as in case of access through a smartphone, the Bootstrap toolkit[10] has been used to build the layout making it mobile-friendly. Indeed, Bootstrap offers a set of toolkits and design templates based on HTML, CSS,

[5] https://leafletjs.com
[6] https://angular.io
[7] https://jquery.com/
[8] https://github.com/Leaflet/Leaflet.markercluster
[9] https://github.com/socib/Leaflet.TimeDimension
[10] https://getbootstrap.com/

and JavaScript dedicated to developing a responsive Web GUI. A responsive interface adapts the webpage layout to the viewing environment and is rendered well on various devices and different resolutions.

2. Implementation of the Data Visualisation Layer

Besides the integrated libraries, the Data Visualisation Layer consists of 4 interconnected files:

- "*index.html*": the entry point of the Graphical User Interface; it defines the layout of the map layer and includes all the necessary components;
- "*script.js*": the main script file; it is the core of the map layer;
- "*configuration.js*": a file containing the data URL and some constants;
- "*layers.js*": the Layers to be added to the map; it defines an object called "*Layers*" that contains all the layers that will be rendered into the map.

The "*Layers*" object is defined as shown in Figure 3. *Layers* is an object of objects; indeed, it contains an object for each available layer that, in general, refers to a separate dataset. In the GUI, each one of these layers can be activated or deactivated separately from the other ones.

```
var Layers = {
   dataset1: { … }
   dataset2 : { … }
   …
   datasetN : { … }
}
```

Figure 3 Example of Layers object.

Figure 4 contains an example of a layer generation, showing a dataset named for simplicity *Nautilos*. Each layer is composed by a set of properties and function: (i) "*infoUrl*", (ii) "*geoJsonUrl*", (iii) "*popupContent*", (iv) "*fillColor*" and (v) *"filters".*
In detail:

- *infoUrl* is a function that returns the URL of the JSON that describes data columns and their ranges; it is used during the initialising phases to download the list of the features contained in the dataset and relative ranges to be considered during filtering operation.
- *geoJsonUrl* is a function returning the URL to download the GeoJSON which contains the data to be shown in the layer; this obtained URL also contains the filter values specified, through the interface, by the user that will be applied during the data retrieval; filters are both relative to the layer and globally applied to all the available layers.
- *popupContent* is a function that specifies the HTML code used to show the pop-up content that will be displayed when a marker is selected. The code is partially static (name of the shown fields and representation styles) and dynamic (values extracted from the GeoJSON).
- *fillColor* is a variable containing the Hex colour code of the markers in the layer in case of time series rendering.

- *filters* is a set of functions returning the filter type; there is a function for each value that can be filtered, and each one can assume values "*RANGE*" or "*VALUE*" depending on the type of filter.

```
Nautilos: {
   …
   infoUrl: () => `${Configuration.URL.infoUrl}cnr/index.json`,
   geoJsonUrl() {
  return
`${Configuration.URL.tabledapUrl}cnr.geoJson?SN,longitude,latitude,depth,temperature,salinity,oxygen,fluorescence,
time&${this.cond}${MyMap.globalFilters}`
   },
   popupContent: (feature) => `
       <em>Temperature:</em> <strong>${feature.properties.temperature}</strong><br/>
       <em>Depth:</em> <strong>${feature.properties.depth}</strong><br/>
       <em>Acquired on:</em> <strong>${feature.properties.time}</strong>
   `,
   fillColor: "#ff0000",
   filters: {
       depth: () => Configuration.constants.RANGE,
       temperature: () => Configuration.constants.RANGE,
       salinity: () => Configuration.constants.RANGE,
       oxygen: () => Configuration.constants.RANGE,
       fluorescence: () => Configuration.constants.RANGE
   },
   …
},
```

Figure 4 Example of a Layer definition.

When the GUI is accessed, the application connects to the URLs returned by the *infoUrl* functions downloading the available JSON, then from it, retrieves the ranges for each filter of each layer, properly sets them and renders the base map. After these steps, the map is fully interactive, it can be zoomed in and out, and data can be rendered and filtered.
Due to the huge amount of data that can be contained in the data server, the GUI starts without retrieving any data, and they will be downloaded only once a layer is selected. This solution avoids the initial burden, and at the same time, it eases the entire process of retrieving only the data selected by the user.

3. Access/security

To access the Data Visualization Layer, users have to authenticate into the system; the GUI also offers the possibility to register freely. The registration and authentication operations are obtained through a dedicated set of REST APIs. The GUI performs this operation by accessing the REST APIs through dedicated HTTP requests.

As previously mentioned, the ERDDAP database, which contains datasets acquired and collected from the project, is an open database accessible to everyone without any restriction. Thus, the users' database has been developed as an external database to guarantee the protection of the stored data. On the other hand, to promote open access to the project datasets, the registration is free, and the authentication is used to collect information for purely statistical purposes. Thus, the simple information requested from the users is the following: a username, a password, and nationality.

IV. GUI TOOLS AND SERVICES (SEARCH AND DISCOVERY)

The Data Visualisation layer is integrated into the NAUTILOS official website and can be reached at this address: https://www.nautilos-h2020.eu/map/ or from within the NAUTILOS website menus.

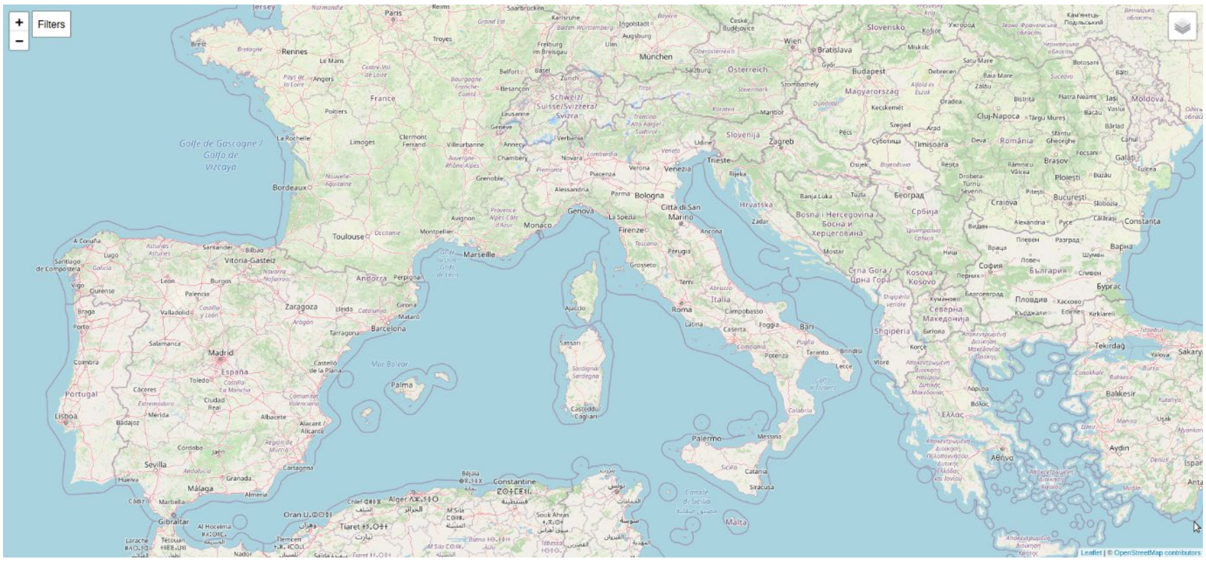

Figure 5 Example of the Data Visualisation Layer.

As shown in Figure 5, the GUI is very simple and shows an explorable map where retrieved data will be displayed. Like other GUIs showing maps (e.g., Google Maps), the map is explorable through *drag&drop* interaction and can be zoomed in or out through a mouse wheel, natural gestures (in case of access from a smartphone) or through the Zoom Controller (Figure 6) that can be found in the top left corner of the map.

Users can request the data to the data server activating the available layers through the Layers Controller (Figure 7) placed in the top right corner of the map (when hovering on this, it shows the available layers that can be selected and displayed in the map). Data can be filtered through the "*Filters*" button near the Zoom Controller (Figure 8) that opens the modal window containing the filters applied to the layers. It is also possible to activate the time series from this modal window through the "*Render data as Time Series*" button that will display data following their chronological order. A dedicated controller called *TimeDimension* is then available (Figure 9); the controller has the following buttons and slider:

- Backward-Play/Stop-Foward Buttons: buttons to play/jump to the desired DateTime;
- DateTime indicator: it shows the current DateTime;
- DateTime Slider: it slides towards the desired DateTime;
- Fps Slider: it adjusts the playback speed.

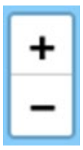

Figure 6 Zoom Controller.

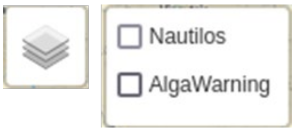

Figure 7 Example of Layer Controller.

Figure 8 Filters button.

Figure 9 Time Series controller.

1. Data Filtering

Available datasets may contain a considerable amount of data, but users could require only a subset of data that fulfils some conditions. This result can be achieved using data filtering. Clicking on the Filters button (the button is enabled if one or more layers are selected), the GUI will display a modal window, as shown in Figure 10, where the user can fill the forms specifying desired conditions. Then, clicking on "*Apply Filters*", the desired state will be applied, filtered data will be requested to the data server, and the already active layers will be reloaded to show only the selected data.

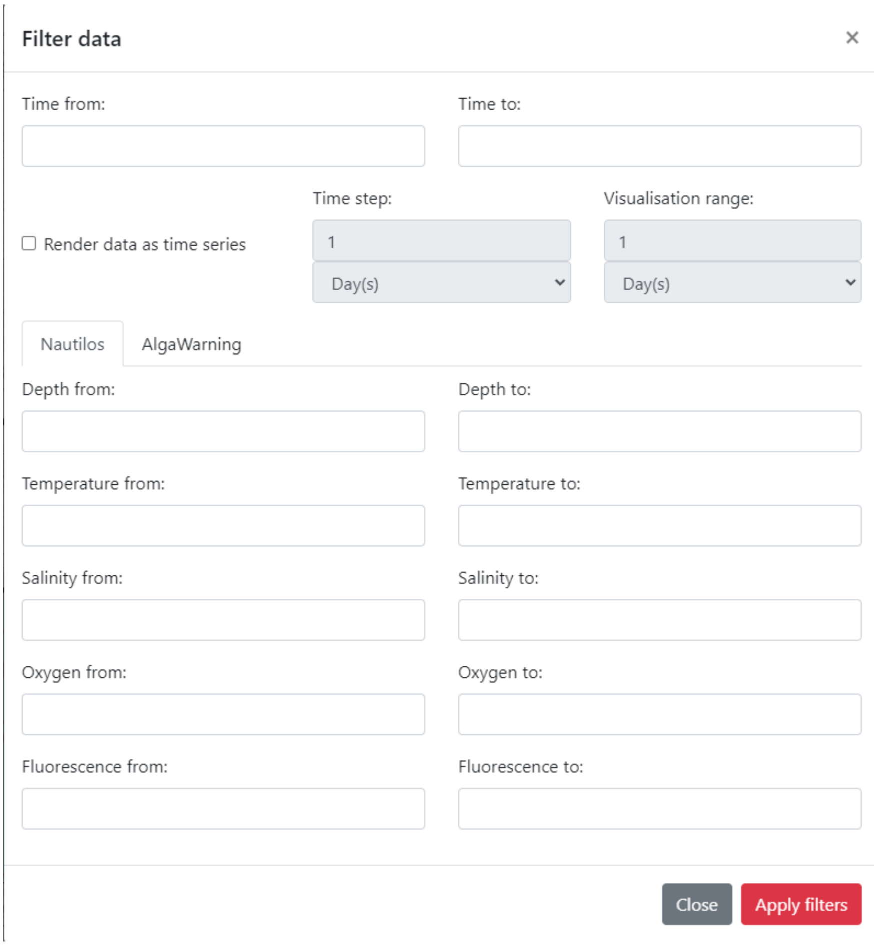

Figure 10 Filter form example.

As shown in Figure 10, the filters in the form are divided into two categories:

- Global filters, such as data time intervals and visualisation options, are applied to each layer.
- Layer filters: each layer has its properties that can be filtered, usually by range or value.

Global filters are applied to all the selected layers. If some layer has no data that fulfils the conditions, a proper warning is given to the user.
From the modal window, it is also possible to activate the Time Series rendering by checking the "*Render data as time series*" checkbox; the user can also specify (i) the time-step interval between each data through "*Time step*" and the time-range of the data that will be shown on the map at the same time through "*Visualisation range*". By default, the period intervals are set to 1 day (i.e. meaning that data from 1-day will be shown at every step).

2. Data Rendering

Once a layer is selected, the GUI requests the data contained into the relative dataset to the data server and shows them on the map. By default, data are rendered in clusters, avoiding an excessive burden on the rendering due to a possibly huge amount of data that could compromise the GUI performance and usability. The data are reloaded every time a set of filters is applied, re-rendering all the previously selected layers.

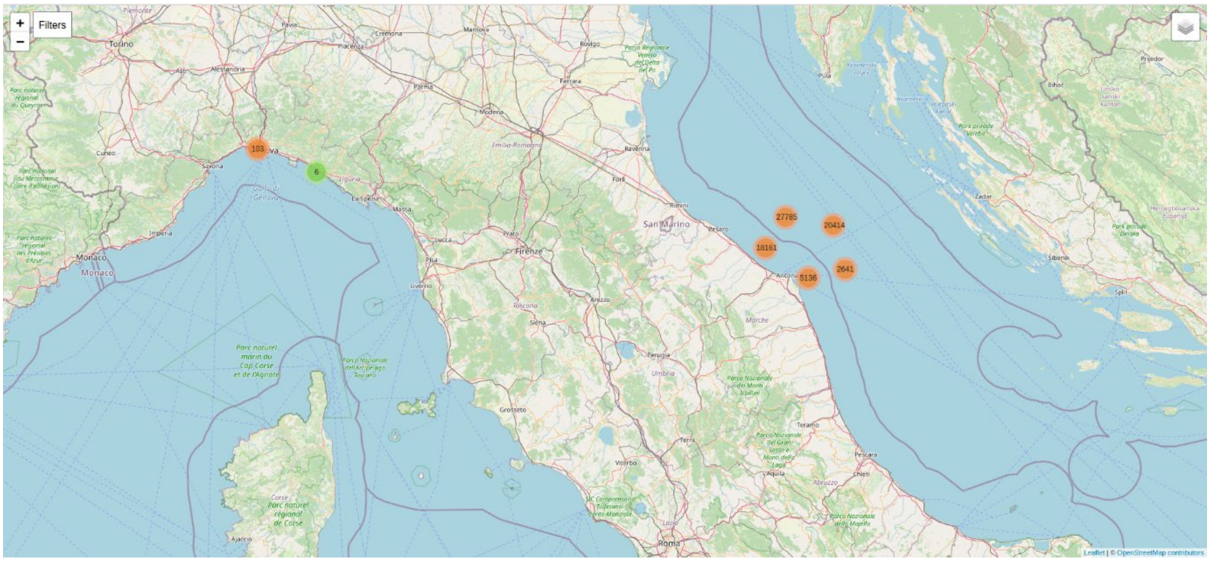

Figure 11 Retrieved data are displayed as clusters.

Figure 11 shows the GUI with the data displayed as clusters. Clusters are represented by filled circles and the number of contained markers in their centre; clusters are characterised by different colours that change depending on how many markers are included in the specific cluster. When a cluster is hovered by the mouse, the cluster's shape (defined by the geographical position of each contained marker) is shown (Figure 12-a). By clicking on a cluster, the map zooms in until the cluster shape fits the screen, then the markers are grouped in new and smaller clusters, and the new clusters are rendered (Figure 12-b). If the zoom level is close enough and the markers are distant enough, the markers are no longer grouped in clusters and are directly displayed on the map (Figure 12-c).

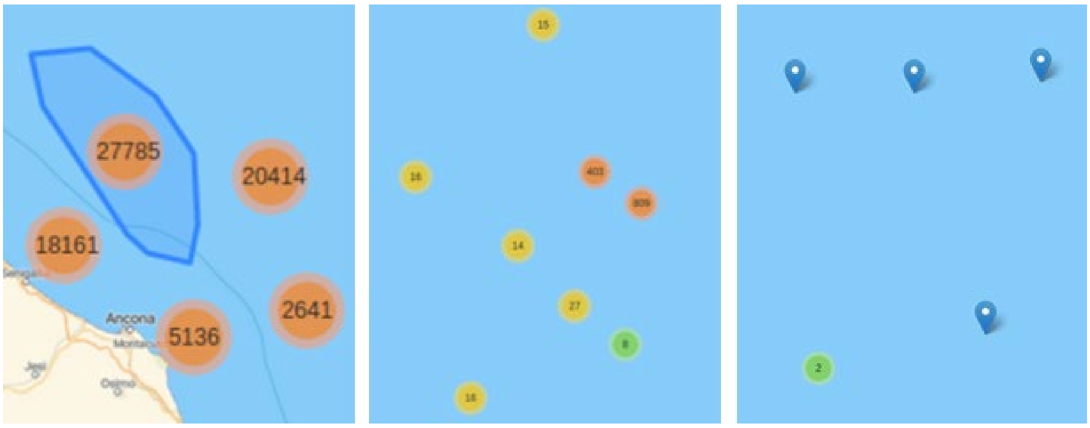

Figure 12 (a) Markers clusters and the shape of one of them; (b) by increasing the zoom level, the clusters will be divided into smaller clusters; (c) if the markers are distant enough from each other, they are displayed.

Once the option "*Render data as time series*" is checked (and the filter is applied), data representation will change. A circle will represent each value coloured depending on each layer's "*fillColor*" value proper and contained in the file *js/Layers.js* (Figure 13). Indeed, the time series modality will be activated, and data in each active layer will change according to the values specified in the *Time step* and *Visualisation range* fields of the filters.

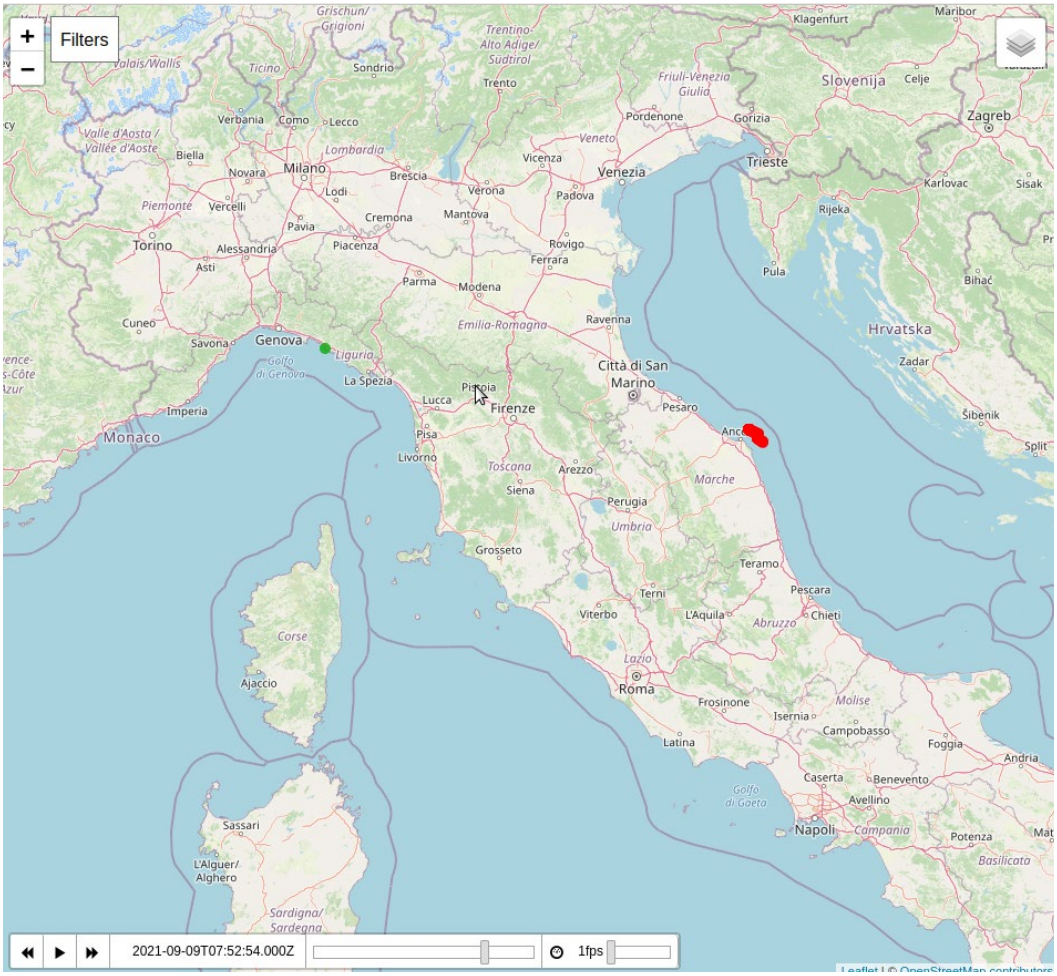

Figure 13 Data rendered as time series. In detail, green and red markers belong to different layers.

As shown in Figure 14, the user can interact with the displayed markers both in the classic and time-series rendering. Clicking on a marker (or on a circle) will show a pop-up reporting further information proper to the specific data represented by the marker. The additional information to be displayed is defined in the property "*popupContent*" of each Layer (contained in the file *js/Layers.js*).

Figure 14 Popup example.

V. DEVELOPED DEVICES

This section contains the description of the specifically developed hardware devices which can be used more specifically for the CS activities but whose data will be collected and shared within the NAUTILOS data management.

1. SMARTPHONE MICROPLASTIC NIR SCANNER DEVICE

The development of the smartphone app and Near Infra-Red (NIR) scanner (SCiO consumer physics) (Fig. 15) has been performed on a developers license while awaiting the finalisation of the app. For the development of the plastic litter database 7 virgin polymers and 400 household articles were scanned focussing on the most used plastic polymers (PE, PA,PC, PET, PP, PS and PVC), an example of the raw spectra acquired from 740 – 1070 nm (13514-9346 cm^{-1}) in shown in Figure 16. This NIR region (~780-1070 nm) contains information on the polymer composition of plastic litter, while spectral data in the visible light and UV (<~780 nm) does not have data that can be used to separate the different polymer types.

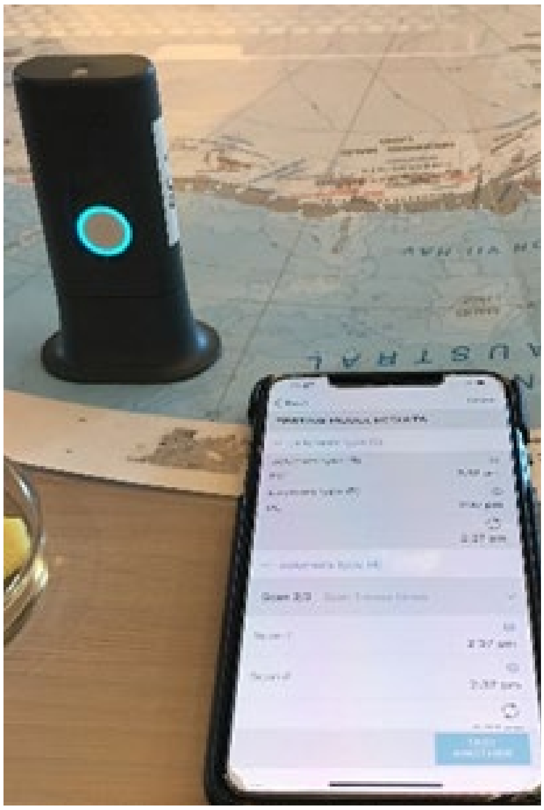

Figure 15 Smartphone app and Near Infra-Red scanner for plastics identification.

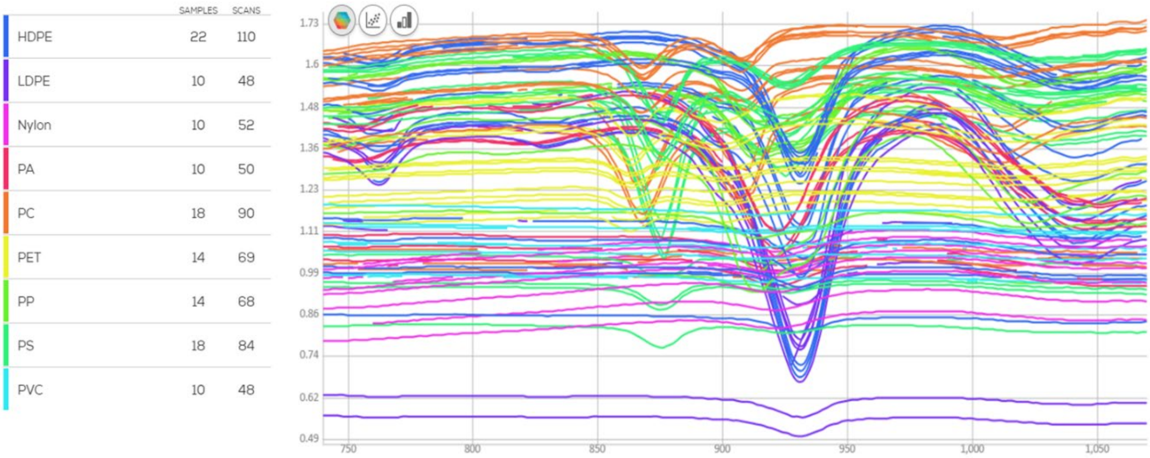

Figure 16 Raw spectra for the 9 different polymers acquired by the smartphone NIR Scanner.

The raw spectra were treated in different ways to obtain as much information as possible for the matching the polymer spectra. The following data treatment of the spectra was found to give the best results using log transformation followed by averaging of the spectra, and taking the 2nd derivative in the selected wavelength range of 850-950 nm (Figure 17). This data set was normalised to unite variance (standard normal variate; SNV) and a multivariate model using principal component analysis was calculated. Several models were tested and the final model containing 407 spectra was uploaded. The final model was not able to distinguish between high density and low density polyethylene, which are chemically too similar to be able to be separated from each other by the model as can be seen in Figure 18.

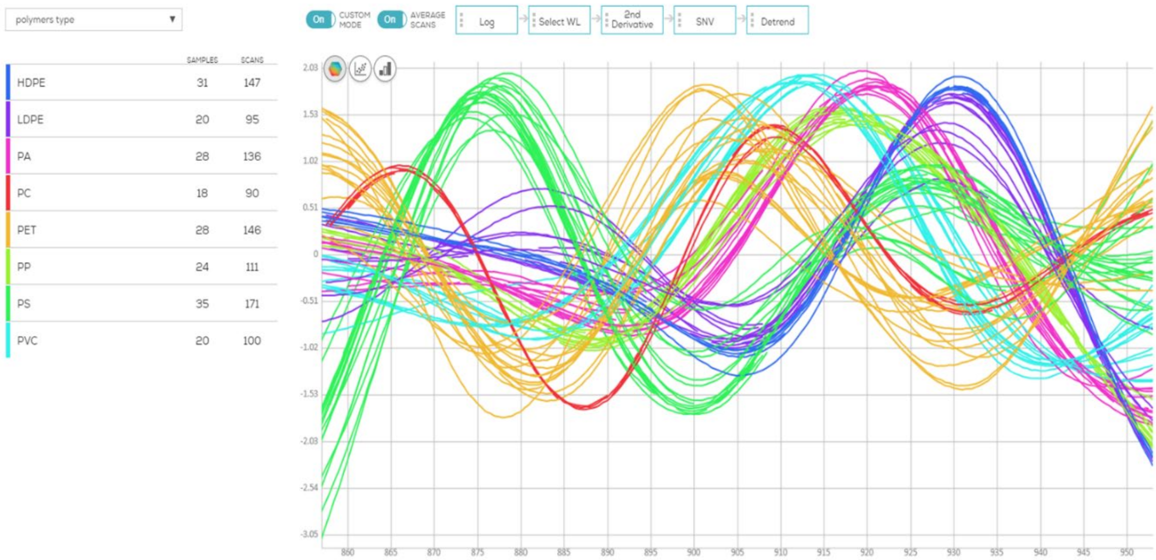

Figure 17 Spectra from the polymers after data treatment (log, average, 2nd derivative) from the selected wavelength range of 850-950 nm.

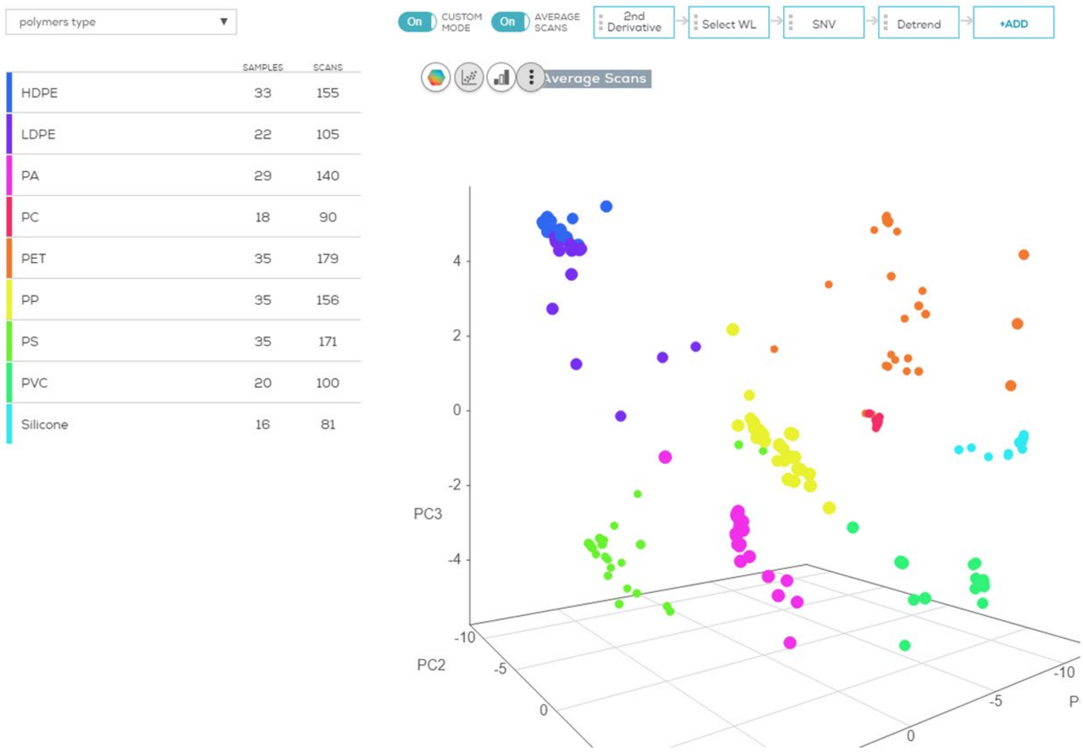

Figure 18. Multivariate model showing the separation of the different polymer, except for HDPE and LDPE which overlap slightly (blue and purple symbols).

The use of the application and smartphone NIR scanner is further described in D8.8, where the smartphone NIR scanner was used with citizen scientists on a cruise expedition ship and university students in Norway. The polymer identification of different items gives additional information on potential sources and origin, especially fragments including meso plastic, when shape and colour do not give any information on potential sources or usage. Upload functionality will be will established in the final app, all development has been performed on a developer license.

2. @LGAWARNING MOBILE APP AND DEVICE

@lgawarning mobile app is the user interface to collect and analyse water samples. It has to be used in association with the DIPLE device, a portable microscope that works with a smartphone (https://diple.smartmicrooptics.com/). Figure 19 shows the basic @lgawarning setup.

The smartphone is mounted on the DIPLE. The @lgawarning App allows an easy and supervised collection of the high-quality sample picture and adds complementary metadata (user position, notes, etc.).

Figure 20 shows an example of complete report visualisation displaying:

- Sample image;
- Coordinates of the image;
- Date and time of the last update;
- Additional information added by the user.

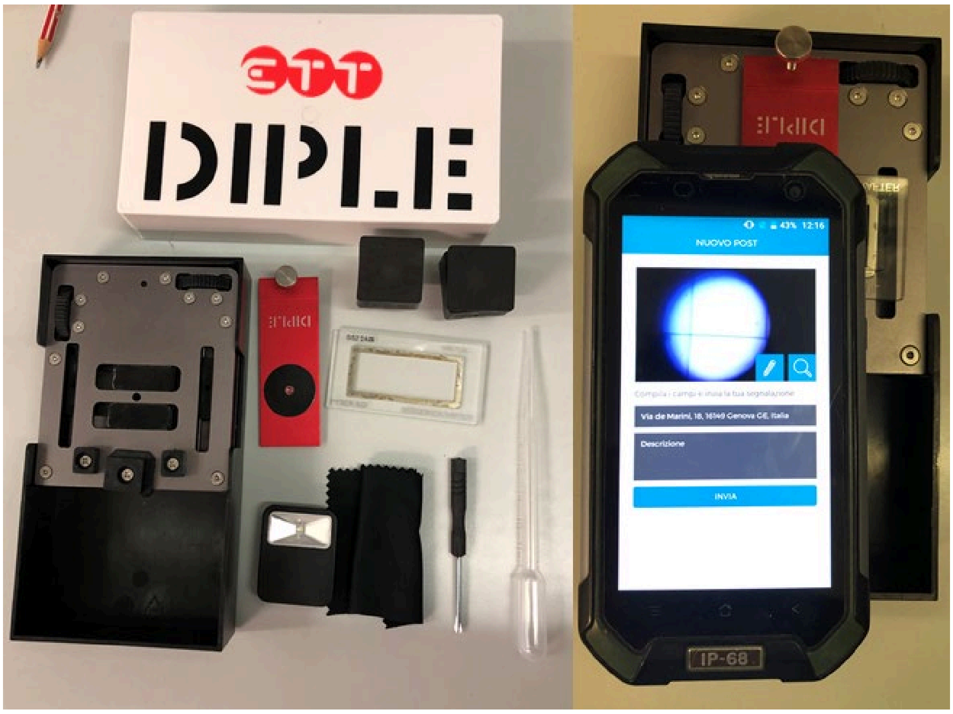

Figure 19 @lgawarning Setup

Figure 20: Report visualisation on @lgawarning app.

The log and the collected samples are always accessible via both the web page and the @lgawarning mobile app (Figure 21)

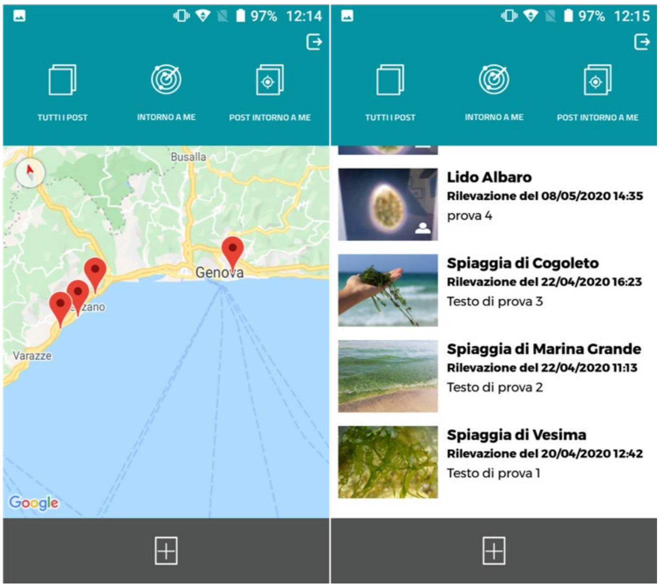

Figure 21: @lgawarning map viewer and list reports viewer.

Sample data and metadata updates are managed as different records (called *posts*) linked together (Figure 22). The field shown in the interface are actually in Italian and represent the information that can be added by the user, as mentioned above (i.e. image, place/coordinates and description).

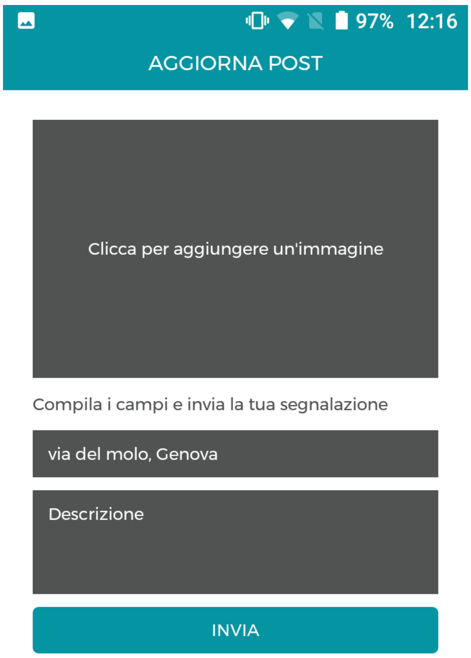

Figure 22: @lgawarning report creation page.

Before sending the report to the web portal, it is possible to crop images and count items by tapping directly on the image.

VI. CONCLUSIONS

This document represents the accompanying report for the developed Graphical User Interface, describing its design and deployment within the NAUTILOS web portal and the tools and services implemented as part of the Task 8.4 activities.
Both the GUI and the services and tools have been designed and implemented in strict connection with Tasks 8.1 and 8.2 activities and the consequent deployed data infrastructure.

Firstly, a recap of the data store and management and how the developed GUI is integrated within the whole data management plan is described.
Then, the description of the structure of the web portal, in terms of frameworks, implementation details and access modalities, is reported. Subsequently, going into more detail, the implemented tools and services are described focusing on the *search & discovery* features designed for the end-users of the web portal.
Finally, two devices that have been implemented specifically to support the Citizen Science activities are described focusing on their integration within the project data management infrastructure. The two devices are a smartphone-based microplastic Near-InfraRed (NIR) scanner for identifying plastics and one for collecting and identifying algae species based on a microscope and a smartphone. More details on these two devices relating to their role in the CS tools are given in the specific deliverable D8.8, "Citizen Science tools and interface".

All the collected NAUTILOS data and products are available and usable through the services deployed through the GUI described in this report. During the NAUTILOS project, and in particular, in the course of the testing and demonstration phase, the current version of the system will be enriched with new data that will flow to the NAUTILOS Data Infrastructure and a catalogue tooltip that will guide the user into an extended and user-friendly data visualisation experience.

APPENDIX 1: REFERENCES AND RELATED DOCUMENTS

ID	Reference or Related Document	Source or Link/Location
1	D8.2 Interoperability requirements definition	NAUTILOS OwnCloud and SyGMa platform
2	D1.3 Data Management Plan	NAUTILOS OwnCloud and SyGMa platform
3	D8.1 Technical documentation and operational field primary data capture systems	NAUTILOS OwnCloud (under submission)
4	D8.8 Citizen Science tools and interface	NAUTILOS OwnCloud (under submission)