



## EPN 2024 RI

### EUROPLANET 2024 Research Infrastructure

H2020-INFRAIA-2019-1

Europlanet 2024 RI has received funding from the European Union's Horizon 2020 Research and Innovation Programme under

Grant agreement no: 871149

### Deliverable D12.1 Observation Campaign Website and Observational Alert System

Due date of deliverable: 31/01/2021

Actual submission date: 29/01/2021

Nature: O

Dissemination level: PU

Work package: WP12

Lead beneficiary: AMU

Contributing beneficiaries: IWF-OEAW, OU, OBSPARIS, UOE,  
UPV/EHU, VU

Document status: Final

Start date of project: 01 February 2020. Duration: 48 months

Project Co-ordinator: Prof Nigel Mason, University of Kent

1. **Nature:** R = Report, P = Prototype, D = Demonstrator, O = Other

2. **Dissemination level:**

PU	PP	RE	CO
Public	Restricted to other programme participants (including the Commission Service)	Restricted to a group specified by the consortium (including the Commission Services)	Confidential, only for members of the consortium (excluding the Commission Services)

## Task 12.2: Coordination of Observations. Development of Alert System

### 1. Executive Summary / Abstract

Our task is to develop a generalized Alert System for observations, which will notify and allow participating observatories to select appropriate targets across the diverse range of planetary science topics. This will provide both regular monitoring of targets and alerts for events requiring time-critical and/or spatially distributed observations (e.g. stellar occultations by asteroids). The service will be based on the existing alert system software created to coordinate amateur observations of asteroids in support of the ESA Gaia mission (Gaia-GOSA, [www.gaiagosa.eu](http://www.gaiagosa.eu)). This service currently provides targets for asteroid light curve observations based on the observer's location and the available targets at the time. Targets of interest in the new service will involve: atmospheres of the giant planets (like convective storms or planetary disturbances), Mars and Venus observations, and ephemeris for their observation, asteroids, comets, exoplanets, and other targets for which observations are needed. This year the prototype version of the system was created. It can be found at: [www.astro.amu.edu.pl/parsec](http://www.astro.amu.edu.pl/parsec). However, further tests and development is needed before public release of the system.

After a rough concept plan for fulfilling the task was developed, the needs for an alert system have been established during an internal meeting. We have chosen a toolset for the service development: PARSEC (PIAnetaRy SciencE Collaboration tool) which is a webapp created in Python, Flask and bootstrap frameworks with additional C++ code and postgresql database. After the early development stage, the main features were programmed, thereby enabling request views on the main page, request filtering by observation date, target category, ownership, campaign view, and a notification system. The user can also establish preferences for notifications and follow requests/campaigns. In the „Requests” view there is the possibility of adding, editing, deleting, commenting the observational needs. Similarly, in „Campaigns” the user can start and coordinate observing campaigns of objects or events.

The first early prototype version was finished by the end of 2020 and internal tests of the service have started. For further information on the service check the presentation at <https://bit.ly/3luMUuN>.

### 2. Performed stages of work:

#### 1. Preparation

1. Assessment of the team capabilities
2. Concept plan of actions needed to fulfill the task
3. Review of existing web technologies
4. Selection of the technologies needed:
  - Frontend: HTML, CSS, JavaScript and Bootstrap framework,
  - Backend: Python, Flask and C++,
  - Data storage: SQLite, Postgresql database.

#### 2. First mockup of the service

1. Getting Python libraries
2. Running Flask server
3. Creating file structure and rough project architecture
4. Creating database
5. Creating first templates for web pages

#### 3. Implementing web service features and code refactoring (iterative process)

#### 4. Development of target visibility calculation software

1. Development of necessary tools for orbit and visibility calculations (C++)
2. Implementing backend for running background computing tasks
3. Incorporating redis-server into the project
4. Creating task queues with rq
5. Linking target visibility calculation software with the web service via background tasks

5. Deployment on the server
  1. System installation on the dedicated server computer
  2. Preparation of the environment, software installation
  3. Database creation
  4. Web service software installation
  5. Web server configuration (nginx, gunicorn)
  6. Setting up system services

## 3. Web service features

### 6. Guest Account

When the main page is viewed, and a user is not logged in, Guest Account is used. The default location (longitude and latitude) is set (Greenwich) that can be changed. Visible targets from this location are shown (calculation of visible targets not supported for Guest User yet). Guest User can view all the data available in the service, but cannot add / remove any Requests or Campaigns or comments.

### 7. User Account

#### 1. Registration

Anyone can create an account in the service. Upon clicking the „Register“ button a registration form is displayed. User has to specify the username and email (used for logging in), default location, and password.

#### 2. Logging in

Registered users can log in with specified email address and password. „Forgot password?“ button is available. By clicking it the user will receive an email with the link to set new password. This feature is tested but not working at the moment (requires a mail server, not created yet).

#### 3. Account settings

Registered users can set a profile picture (displayed in comments), change user name and email, and change password. Also, Registered users can specify any number of locations (and define the default one) used in Request filtering. Preferences regarding Notifications can also be set.

Locations

📍 Royal Observatory, Greenwich	lon: 0.0, lat: 51.5	Set as default	<span style="border: 1px solid red; border-radius: 50%; padding: 2px;">×</span>
📍 Poznań	lon: 16.0, lat: 52.0	Default	<span style="border: 1px solid red; border-radius: 50%; padding: 2px;">×</span>

Add Location

Preferences

- Notify me when someone comments on my Request
- Notify me when someone comments on a Request I follow
- Notify me when Request I follow is deleted
- Notify me when new Request is added to a Campaign I am following
- Notify me when Request is removed from a Campaign I am following

By following a Campaign you automatically follow all requests in it. You will be notified according to the preferences regarding Requests.

- Notify me via e-mail

Save

## 8. Main view (Requests)

The main page of the web service displays the list of Requests posted by all the users. It is sorted by the requested observation date. The user can see the targets names, campaign (if they belong to one), observational date and following status. The user can follow/unfollow the requests from here. The user can see the description and go to the single Request view upon clicking the arrow button.

Target	Campaign	Date	Follow
⌵ (4) Vesta		2020-12-21	<span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px;">×</span>
⌵ (1) Ceres	Dwarf planets	2020-12-23	<span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px;">×</span>
⌵ (4) Vesta		2020-12-28	

Location: Poznań ▾

Start Date: 📅 2020-12-21

End Date: 📅 2021-01-21

- Minor bodies
- Planets
- Comets
- Only mine
- Followed

Filter

### 1. Filtering

The list is limited to the time period of one month starting from the current date, and ending 30 days later. The date can be narrowed down to any dates from this period. Displayed Requests are also filtered by the visibility from a location. Location can be set to any location from specified locations list. Also, the user can narrow down the type of the target (Minor bodies, Planets, Comets), choose only his/her Requests, or display only the followed ones.

## 9. Campaign view

The user is presented with the list of defined campaigns, with some basic info (ownership, number of requests, following status). Upon clicking the name, the user is redirected to the single Campaign view.

# Campaigns:

Irene

following

0 requests

Dwarf planets

yours

1 request

## 10. Requests

### 1. My requests view

The user can view only his/her requests in a special view. They are not filtered in any way except by the ownership. More information is shown (number of followers, number of comments, creation date). By clicking the name, single Request view is shown.

# My Requests:

Target	Campaign	Time span	Added	Followers	Comments
<a href="#">(4) Vesta</a>		2020-12-21 — 2020-12-31	2020-12-21	0	0
<a href="#">(1) Ceres</a>	<a href="#">Dwarf planets</a>	2020-12-23 — 2020-12-23	2020-12-21	1	0

### 2. Single request view

In this view more information is presented. The user can follow/unfollow a request from here as well. If the user is the owner of the request, he/she can edit or delete the request from here. At the bottom there is a comment section.

[Edit](#) [Delete](#)

Request for  
**(4) Vesta**  
📅 2020-12-21 — 2020-12-31  
No. people following: 0

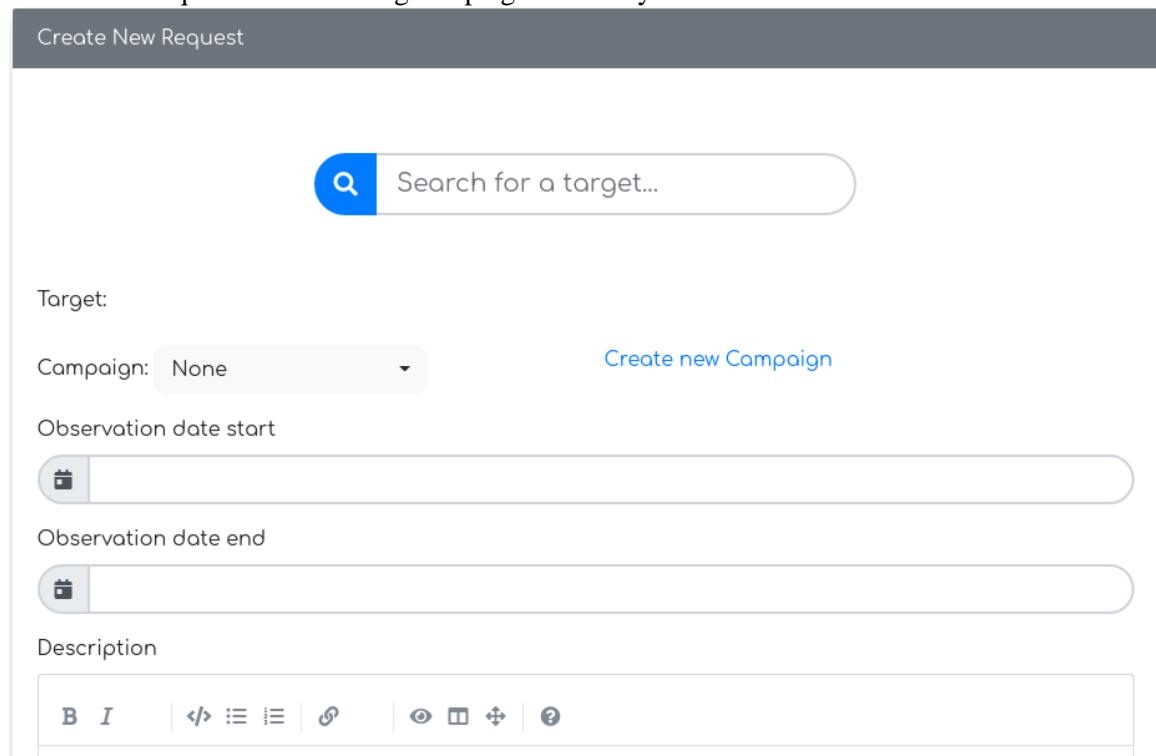
category: minor body      author: user2      posted: 2020-12-21

Add to Campaign:  [Add](#)

Example Description

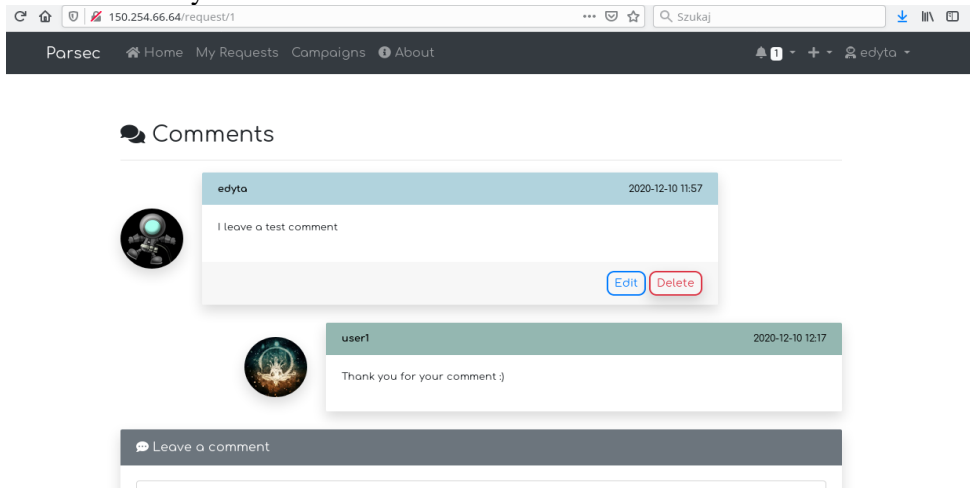
### 3. Adding new Request

By clicking the „+“ button the user can choose to create a new Request. In this view a user is asked to specify the target (selected from the database), observation start/end dates and description. The user can add the request to the existing campaign owned by him/her.



#### 4. Commenting

Users can comment the Request in a form of continuous discussion. Comments can be deleted and edited by Comment owners.



#### 11. Campaigns

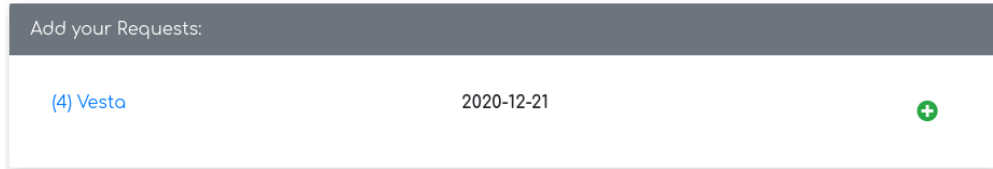
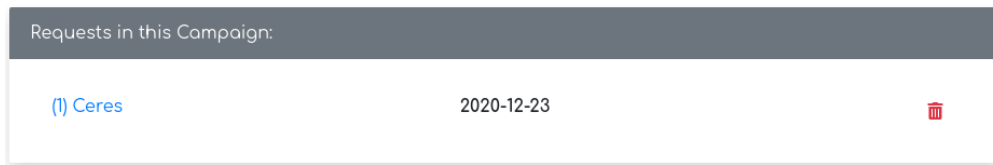
Campaigns serve the role of Request aggregates under one name.

##### 1. Adding / Removing

Similar to adding Requests, a new Campaign can be created by clicking the „+“ button. Only the name and description are required.

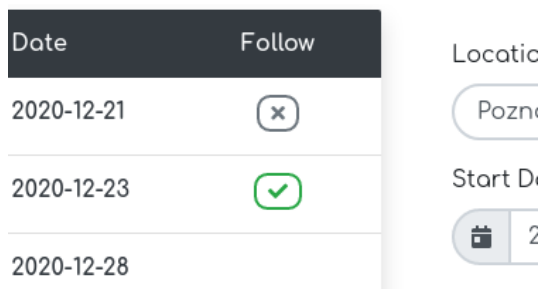
##### 2. Campaign view

The single campaign view displays the name, owner and the description, and the list of the targets added to the Campaign. If the user owns the Campaign, he/she can edit and remove it. Also, he/she can add owned requests from here.



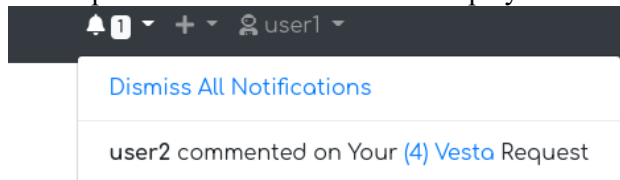
## 12. Following

Users can follow Requests and Campaigns. This provides filtering capabilities in the Main view and enables notifications.



## 13. Notifications

The web service has a notification system. The users are informed when several happen in the service, i.e. someone comments owned or followed Request, when followed Request is deleted, when Requests are added to or removed from a followed Campaign. The notification's settings are available in the User's profile. The notifications are displayed in the web service and can be also sent via email.



## 14. Target list

While creating a Request, the user searches the database for the target. There is no option to specify your own target.

3) Brunhild (minor body)

---

23) Shapleya (minor body)

---

30) Riceia (minor body)

### 1. MPC file

Currently the targets are loaded from MPCORB.DAT file downloaded from Minor Planet Center website. The file provides named and unnamed asteroids with their current orbital elements used for the visibility calculation. In the future the list will be extended by adding planets, comets and other targets of interests.

#### 15. Visibility calculations

Targets in requests are only displayed in the Main view if the User can observe them. The visibility of the target is calculated based on the target's orbital elements and the chosen location by the User. This calculation is done every time the location is changed in the Main view. Calculations are performed in the background in parallel (not blocking the web service) per user. A number of worker programs are enabled on the server to facilitate this. The code is written in C++.

## 4. Summary and future prospects

The web service is still in early stages of development. Existing features need to be tested and new features need to be implemented. List of planned features and improvements:

- Adding more targets to the database (planets, exoplanets, comets, stars)
- Enabling the email server to support email notifications and password reset
- Visibility calculation for Guest Users
- Visibility calculations for targets other than minor planets
- Creating Admin panel
- Enable database migrations and backups
- Upgrading the interface feel and look

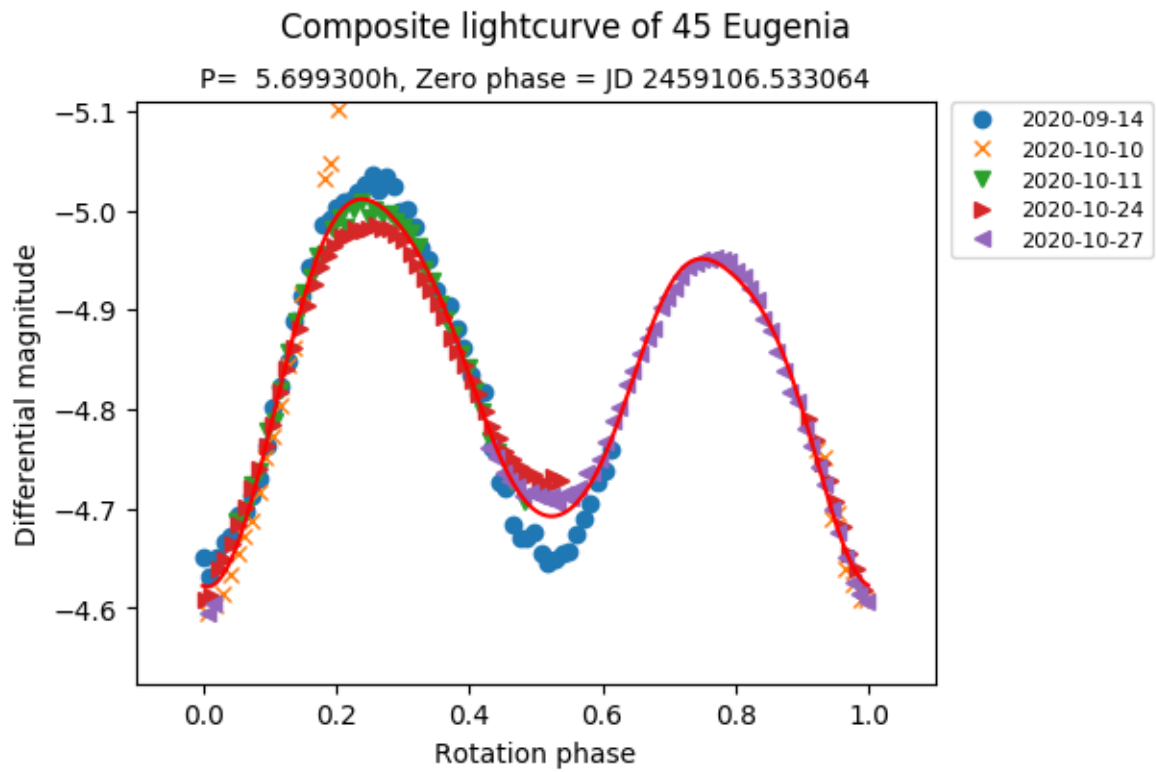
In the upcoming year, tests on the system will proceed and new functionalities will be added. The frontpage will be created, thereby joining other existing services for various planetary science topics. The User Guide will be created to make new observers familiar with the system and show all capabilities. After this, the service will be made publicly available for users. The further development will then depend on the user's needs and requests.

## 5 . Activities with impact on the science:

During the whole year we constantly developed and upgraded the GaiaGOSA system and coordinated several observing campaigns. Few new users have joined to the Service. There was an observing campaign of the occultation event of (551) Ortrud asteroid which took place on the 6-th October and was visible across Europe. There is also always a list of targets for which we need coordinated observations from different places depending on the visibility. After gathering and processing the data, within this year, with help of amateur observers, we were able to complete the rotation period for several asteroids, and the results will be used in further studies, mainly for spin and shape modeling. In the last Figure we present a high quality lightcurve of 45 Eugenia completed thanks to observations gather by GaiaGOSA user.

**There was also one scientific paper published:** Dudziński, G, Podlewska-Gaca, E., Bartczak, P. et al.,, Volume uncertainty of (7) Iris shape models from disc-resolved images,, Monthly Notices of the Royal Astronomical Society, Volume 499, Issue 3,





**The influence of Covid-19 pandemic:** the building of Astronomical Observatory Institute at AMU, the operational hub of the programme, was closed in October and part of November. Thus there was no physical access to the computer server where the Service will be run which caused a small delay in our work.