



Deliverable

D5.2 Initial Digital Twin Prototype

| | | |
|-------------------------------|---|---|
| Project Acronym: | DUET | |
| Project title: | Digital Urban European Twins | |
| Grant Agreement No. | 870697 | |
| Website: | www.digitalurbantwins.eu | |
| Version: | 1.0 | |
| Date: | 10/06/2021 | |
| Responsible Partner: | ATC | |
| Contributing Partners: | VCS | |
| Reviewers: | Internal Philippe Michiels (IMEC) Walter Lohman (TNO) Gert De Tant (OASC) Jiri Bouchal (ISP) External Pieter Morlion Andrew Stott Yannis Charalabidis Michiel Van Peteghem | |
| Dissemination Level: | Public | X |
| | Confidential – only consortium members and European Commission | |

Revision History

| Revision | Date | Author | Organization | Description |
|----------|------------|---|--------------|--|
| 0.1 | 15.04.2021 | Athanasios Dalianis Marina Klitsi | ATC | Initial structure / TOC |
| 0.2 | 20.05.2021 | Athanasios Dalianis Marina Klitsi | ATC | First draft version |
| 0.3 | 31.05.2021 | Athanasios Dalianis Marina Klitsi, Thomas Adolphi | ATC, VCS | Description reg the map viewer was included |
| 0.4 | 02.06.2021 | Athanasios Dalianis Marina Klitsi | ATC | Updated version sent for Internal Revision |
| 1.0 | 10.06.2021 | Athanasios Dalianis Marina Klitsi | ATC | Final version taking into account reviewers comments |

Table of Contents

| | |
|--|-----------|
| Executive Summary | 6 |
| Introduction | 7 |
| 1.1 Purpose and Scope | 7 |
| 1.2 Relation to other Work Packages | 7 |
| 1.3 Structure of the Document | 7 |
| Development of the DUET platform | 8 |
| 2.1 Components integrated for the Closed Beta release | 9 |
| 2.2 User Interfaces | 11 |
| DUET platform user guide | 13 |
| 3.1 Home | 13 |
| 3.2 Contact Us | 16 |
| 3.3 Join Us | 17 |
| 3.4 Sing your own DUET | 18 |
| 3.4.1 Login | 18 |
| 3.4.2 Users | 20 |
| 3.4.3 Cases | 24 |
| 3.4.4 Data catalog | 29 |
| 3.5 Map Client | 33 |
| 3.5.1 Case support | 33 |
| Load a case | 35 |
| Create a new case and save it via cases-api | 35 |
| Extend and save a case via cases-api | 36 |
| 3.5.2 Add data to viewer and visualize | 37 |
| 3.5.3 Edit legend entry and create a user style | 40 |
| 3.5.4 Inspecting a feature in viewing client | 41 |
| 3.5.5 Blocking a road segment / changing traffic model | 43 |
| 4. Conclusion | 48 |
| 5. References | 49 |

List of Tables

[Table 1: list of epics for the closed beta](#)

[Table 2: Integrated components](#)

List of Figures

[Figure 1: list of epics for the closed beta](#)

[Figure 2: Landing and platform pages](#)

[Figure 3: Landing pages - Home page 1](#)

[Figure 4: Landing pages - Home page 2](#)

[Figure 5: Landing pages - Case details](#)

[Figure 6: Search results](#)

[Figure 7: Contact us page](#)

[Figure 8: Join Us page](#)

[Figure 9: Login](#)

[Figure 10: Wrong credentials](#)

[Figure 11: Forgot password](#)

[Figure 12: Registration requests](#)

[Figure 13: Request rejection](#)

[Figure 14: Users table](#)

[Figure 15: Update user](#)

[Figure 16: Add user](#)

[Figure 17: Delete user](#)

[Figure 18: list of Cases](#)

[Figure 19: Cases table](#)

[Figure 20: Case details - Information](#)

[Figure 21: Case details -Details](#)

[Figure 22: Case details - Location](#)

[Figure 23: Case details - Media](#)

[Figure 24: Case details - Tabs](#)

[Figure 25: Case copying](#)

[Figure 26: Remove](#)

[Figure 27: Data catalog](#)

[Figure 28: Filtering by region](#)

[Figure 29: Add data source](#)

[Figure 30: Generic information 2](#)

[Figure 31: Connection specs](#)

[Figure 32: Connection parameters](#)

[Figure 33: Relation to models](#)

[Figure 34: Duet data catalog view - cases tab](#)

[Figure 35: DUET data catalog view - cases tab - case entry](#)

[Figure 36: data catalog view - cases tab - create or update a case](#)

[Figure 37: data catalog view - cases tab - create a new case dialog](#)

[Figure 38: data catalog view - cases tab - loaded case entry](#)

[Figure 39: data catalog view - cases tab - update case dialog](#)

[Figure 40: Map Client: Add data to viewer and visualize](#)

[Figure 41: DUET catalog view](#)

[Figure 42: DUET catalog view in VCS client with selected dataset searched by parameter "traffic"](#)

[Figure 43: content panel view on the left hand side of the viewer with loaded dataset for Plzen traffic and the default styled data in the viewing client on the right hand side](#)

[Figure 44: editing a layer in viewing client](#)

[Figure 45: Feature inspection](#)

[Figure 46: Feature inspection with additional graph for Capacity vs. Flow comparison](#)

[Figure 47: Feature inspection of traffic model and changing the properties](#)

[Figure 48: blocking a road segment by quick option](#)

[Figure 49: changing the properties of a road segment by editing](#)

[Figure 50: Notification - successful sent of the change to DUET message system](#)

[Figure 51: url with the new traffic data](#)

[Figure 52: DUET visualization client > data settings](#)

[Figure 53: Traffic Results](#)

Executive Summary

This document describes the closed beta version of the DUET platform. This is a first working version of the prototype that offers an initial set of the expected functionalities, and will act as the test-bed for the DUET stakeholders to experience with this initial set of DUET provisions and assess the concepts and knowledge conveyed by the project.

The deliverable is examining the platform from two different aspects; from the internal point of view, in which the functional and architectural structure is presented briefly, and from the end-user point of view, in which the external appearance, look & feel and the overall high level functionalities and interaction potentials are explained. Moreover, the document provides an overview of the current functionalities in the form of a user guide, exposing every available feature so far.

The DUET Platform, far from being a simple container for the individual modules, is a coherent application, where several different components reside and collaborate in harmony. The first prototype is a proof of concept to the DUET end users. It encapsulates most of the underlying technologies and gives a clear and easy to use graphical interface. The benefit of the current architecture is that any additional functionality can be wrapped into a separate component and be added to the platform, provided that it abides by the basic communication standards exposed by the DUET platform architecture. The scope of this practice is to enable future extensions of the platform to arising functionalities, which may maximise the potentials for further exploitation and adoption of the platform beyond the project end.

1. Introduction

This report accompanies and documents the Deliverable D5.2, which is a **prototype** (Closed beta release of the DUET platform).

1.1 Purpose and Scope

This deliverable aims to provide an overview of the closed beta release of the DUET prototype. To this end, the report describes shortly the implementation aspects as well as the current functionality which is explained in the form of a user guide.

1.2 Relation to other Work Packages

The WP5:DUET System Integration aims at providing the implementation aspects for the delivery of the DUET components integration in a unified platform. The design of the DUET platform is driven from the epics and the user interfaces as delivered from the WP2: User-centric design for advanced decision-making practices and WP4:DUET Front-end environment set-up: Analytics, 2D/3D visualizations & simulation respectively.

Following the implementation of the individual modules, this WP delivers an integrated view of the DUET platform to act as the test-bed for setting the DUET Pilots in WP6: Pilot scenarios, deployment, and impact validation and validating the results in real life scenarios. The content presented in this deliverable is subject to refinements and modifications, based on the progress of the technical work packages, as well as during the validation and evaluation phases of the project.

1.3 Structure of the Document

This document reports on the activities and effort placed in the integration of the various technologies and tools provided by the WP3 towards delivering the first release of a functional DUET Integrated prototype. The integration effort is guided by the Agile Software Development methodology, aiming to progress the development work in parallel teams and regularly integrating their output, based on a well-defined design.

The scope of this document is to act as appendix to the current version of the DUET integrated prototype and as such, it is structured as follows:

- Section 2 provides an overview of the development aspects of the DUET platform;
- Section 3 presents a user guide with the main functionality of the DUET closed beta;
- Section 4 concludes this report.

2. Development of the DUET platform

The DUET platform (closed beta version) has been implemented via Sprint iterations following the agile process, as described below, in concrete steps:

- The Pilot Users along with the assigned POs agreed on specific epics (as those defined in WP2)
- The selected Epics were broken down in Stories / Tasks by the POs and the technical team
- The Prioritised Stories became available in DUET Jira from the POs
- The technical DUET team organised bi-weekly Sprints (sprint planning meetings) where team members came together in order to lay out the components for the upcoming round of work.
- The POs Reviewed the finished user stories and provided early and constructive feedback.
- Monthly Sprint Reviews were organised with the rest of the Consortium partners with the aim to present the working software to the Pilots.
- Pilot partners tested the provided functionalities and provided their feedback
- POs translated the feedback provided into Jira stories / tasks

Taking into account the list of epics as defined by the Pilot partners in cooperation with the POs, the list of epics for the closed beta version as refined and given in the technical team is presented in the following table:

Table 1: list of epics for the closed beta

| Closed Beta Epic | Pilot epic | Rationale |
|---|------------|---|
| As a visitor, I can understand the possibilities of the digital twin when I land on the main page, so I can select the right case, pilot. | G8 | The epic makes the whole digital twin more easily to understand and as such guides visitors |
| As a visitor, I can view the digital twin on a laptop, mobile, tablet. | G8 | Making DUET available to more users |
| As a visitor, I want to experience a user-friendly and exciting UI, so I want to keep interacting with DUET. | G8 | - |
| As a city official, I can inspect segments of the road network in the digital twin, so I can find the correct road I want to generate an experiment for | G3 | Selecting roads to be able to do a what if analysis related to traffic |
| As a city official, I want to test-change an obstruction of the traffic on the road, so I can test different traffic scenarios | G3 | Part of the what-if analysis related to traffic |
| As a traffic model, I can see the events that are sent to me, update my state for the experiment and send out the results as an event so other systems can react to this update | G3 | Part of the what-if analysis related to air pollution |
| As a city official, I can assess what the impact is of the traffic on the noise pollution in my city (Pilsen and Flanders) | G2 | Part of the what-if analysis related to noise pollution |
| As a city official, I can assess what the impact is of | G3 | Part of the what-if analysis related to |

| | | |
|---|--|---------------|
| the traffic on the air pollution in my city (Pilsen and Flanders) | | air pollution |
|---|--|---------------|

Also the figure below presents the list of epics as extracted from the DUET Jira.

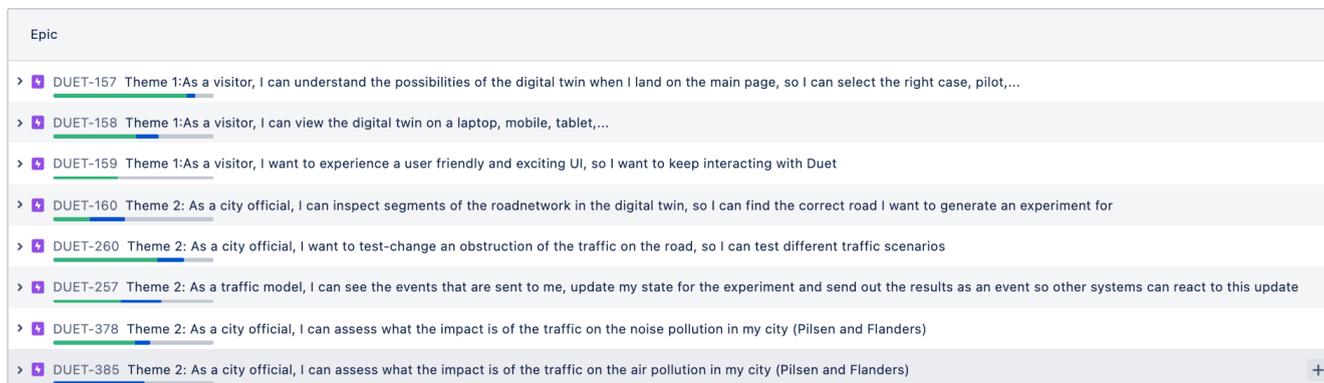


Figure 1: list of epics for the closed beta

2.1 Components integrated for the Closed Beta release

The following table provides an overview of the components integrated for the current release of the platform and the technologies used for their implementation.

Table 2: Integrated components

| Component | Main technologies used | Notes |
|----------------------|--|--|
| API Gateway | Kong Gateway ¹ , Nginx, Cert manager ² | The API Gateway is the component that protects the services and UIs of the platform providing a single point of access. The Kong Ingress Controller ³ is used for accessing the platform services and Nginx Ingress Controller ⁴ for accessing the platform UIs. The Cert Manager is used to automatically issue and renew LetsEncrypt certificates. |
| App Gateway | Java Spring Boot ⁵ , MySQL ^{6 7} | The App gateway was updated to support the communication between the models and the Map Visualiser, providing additionally to the Web sockets mechanism, a polling mechanism that the models can use to retrieve their messages. |
| Data Sources Gateway | Node JS ⁸ | - |

¹ <https://konghq.com/kong/>

² <https://cert-manager.io/docs/installation/kubernetes/>

³ <https://konghq.com/solutions/kubernetes-ingress/>

⁴ <https://kubernetes.github.io/ingress-nginx/how-it-works/>

⁵ <https://spring.io/projects/spring-boot>

⁶ <https://www.mysql.com/>

⁷ <https://www.presslabs.com/docs/mysql-operator/>

⁸ <https://nodejs.org/en/>

| | | |
|-----------------------------|--|---|
| Data Catalog (Data sources) | Node JS, React JS ⁹ , MongoDB ¹⁰ ¹¹ | The first version of the Data Catalog component is implemented from scratch to replace CKAN used in the previous period. |
| Audit Manager | Java Spring Boot, React JS, MySQL | The first version of the Audit Manager is implemented to monitor basic user activity for statistical purposes. |
| Users Manager | Java Spring Boot, React JS, MySQL, Keycloak ¹² , PostgreSQL ¹³ ¹⁴ | The first version of the Users Manager is implemented. This component uses keycloak's services in order to realise functionalities related to the user management through a custom UI, acting as a "Keycloak connector". The implementation is generic enough so that other relevant tools can be connected in the future without changing much the component's services and UI |
| Cases Manager | Java Spring Boot, React JS, MySQL | A new component that will be part of the Management module and is responsible to provide web services and UIs for the management of the project's cases. |
| Map Visualiser | Javascript, Node JS | Updated version mainly to support the integration with the new Data Catalog, the Cases Manager and the communication with the models in the context of Theme 2 related epics |
| KUL Traffic model | Python ¹⁵ , Azure storage ¹⁶ | Updated version mainly to support the Theme 2 related epics |
| P4A Traffic model | Python, Java, MySQL | Updated version mainly to support the Theme 2 related epics |
| Noise model | Python, Java, MySQL | Updated version mainly to support the Theme 2 related epics |
| Air pollution model | Node JS, MySQL | Updated version mainly to support the Theme 2 related epics |
| Message Streaming Platform | Kafka ¹⁷ ¹⁸ | - |

⁹ <https://reactjs.org/>

¹⁰ <https://www.mongodb.com/>

¹¹ <https://github.com/mongodb/mongodb-kubernetes-operator>

¹² <https://www.keycloak.org/about>

¹³ <https://www.postgresql.org/>

¹⁴ <https://github.com/zalando/postgres-operator>

¹⁵ <https://www.python.org/>

¹⁶ <https://docs.microsoft.com/en-us/azure/storage/>

¹⁷ <https://kafka.apache.org/>

¹⁸ <https://strimzi.io/>

| | | |
|--------------------|---|--|
| DUET Landing pages | Wordpress ¹⁹ , MySQL, React JS, DireFlow ²⁰ | A new component that implements part of DUET's offered UIs mainly for dissemination purposes. It is a separate installation hosted outside of the DUET kubernetes cluster. |
|--------------------|---|--|

2.2 User Interfaces

In order to address the user needs, a user-driven approach was followed to produce the necessary dashboard screens via a process of paper sketches, visual mock-ups and iterative versions of the new screens until the desired result was obtained in cooperation with the product owners and the pilot users. The intermediate releases of the platform were presented to the pilot users at the end of the monthly sprints in order to provide their feedback to the Product Owners. The Product Owners reviewed the User Stories upon their completion, gathered the users' feedback and planned the necessary changes with the technical team of the project.

For the purposes of the project, we have set up two environments, one for staging and one for production. All the tests from the technical team and the Product Owners took place in the staging environment, while the production environment was being updated after the monthly sprint pilot reviews in order to allow the pilot users to test the new platform functionalities themselves.

The DUET User Interfaces consist of two main parts, one mainly for dissemination purposes (DUET landing pages) and one that contains the tools that mainly the pilot users interact with (DUET Platform).

In the DUET landing pages, in summary, a user can view basic information about the DUET project, ask a question or provide feedback in the contact us page, request to join the project by creating an account for a specific pilot and more importantly to view information about the most important cases from all the pilots of the project.

In the DUET Platform, in summary, a user depending on his / her role can manage cases and / or users, search and register data sources and interact with the map performing actions like view static data, close a road and view how the traffic, noise and air pollution are affected in the area, etc.

¹⁹ <https://wordpress.com/>

²⁰ <https://direflow.io/>

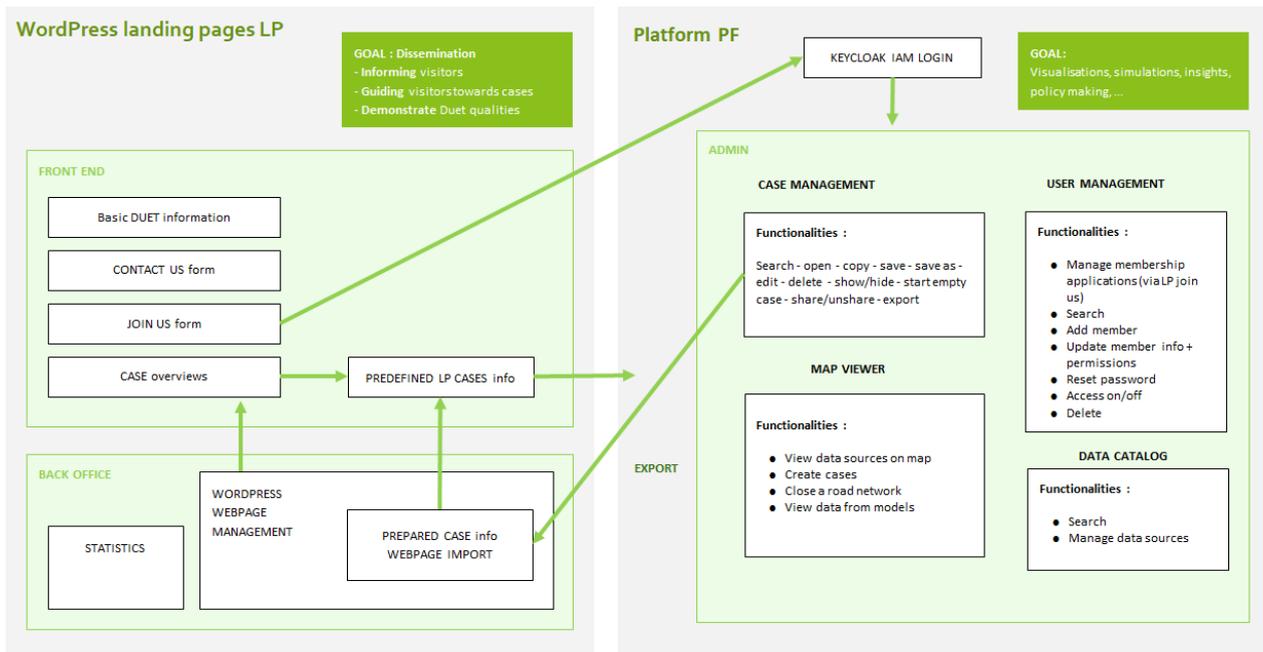


Figure 2: Landing and platform pages

The following section presents a User Guide for the closed beta of the DUET platform. The DUET Consortium has acquired the following URLs for accessing the landing pages: <https://citytwin.eu> for the production environment and <https://staging.citytwin.eu> for the staging environment and two for accessing the platform pages <https://platform.citytwin.eu/app> and <https://platform.citytwin.eu/dev> for the production and development environments respectively.

3. DUET platform user guide

Screenshots of the main pages are presented in the following subsections and describe the platform's main functionalities. By opening a web browser and entering the platform's Uniform Resource Locator (URL) <https://citytwin.eu> the user gets into the Home page of the Integrated Prototype.

3.1 Home

Via the landing page the user is able to 1) access to the project’s website in order to learn more about the project 2) access to the “Join us” form and to the DUET video (Figure 3). In addition, the user can view a tag cloud (“Search a DUET Case”) related to the terms associated with the pilot cases published. The size of each tag indicates the appearance of the tag in the provided cases, that is, the larger the tag the higher the number of cases that are associated with it. Clicking a tag searches for cases with this tag and the user is navigated to the results page presenting all the relevant cases (Figure 6).

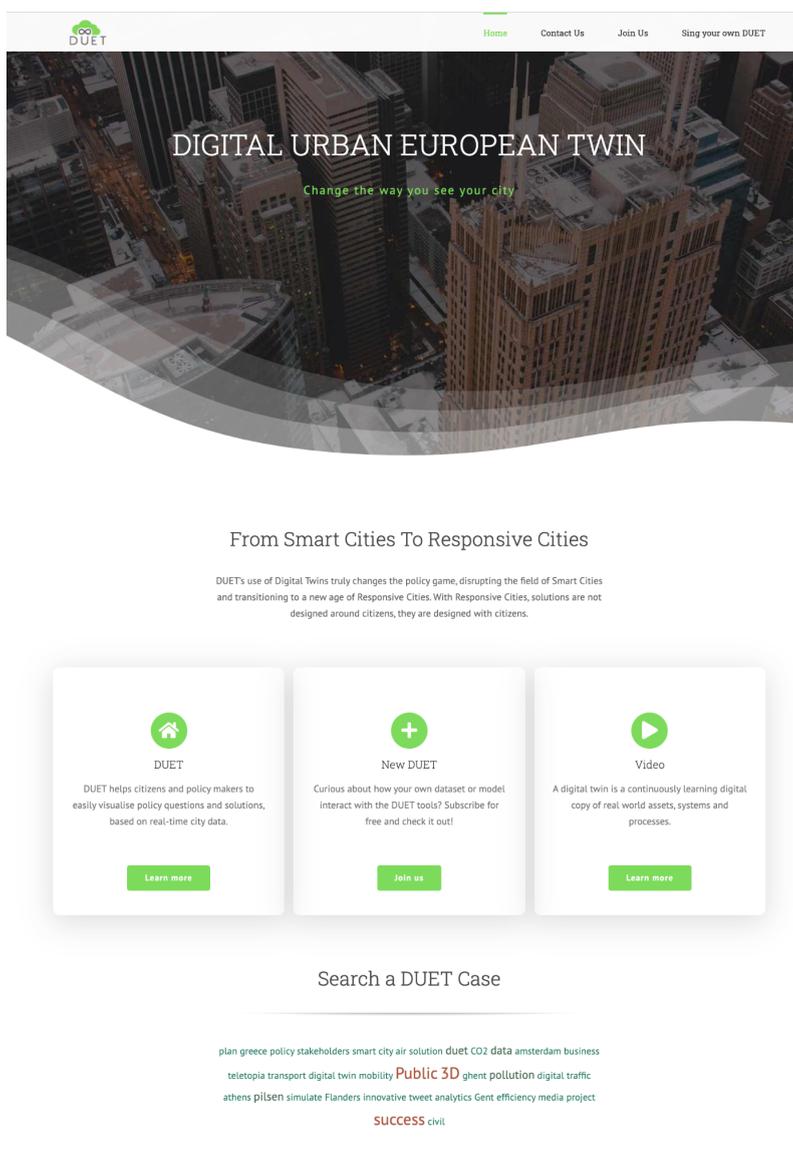


Figure 3: Landing pages - Home page 1

The user can search the DUET cases via the map that is available in the Homepage - the map is another way to provide easy access to the cases- or via the Section “Join DUET cases” Figure 4).

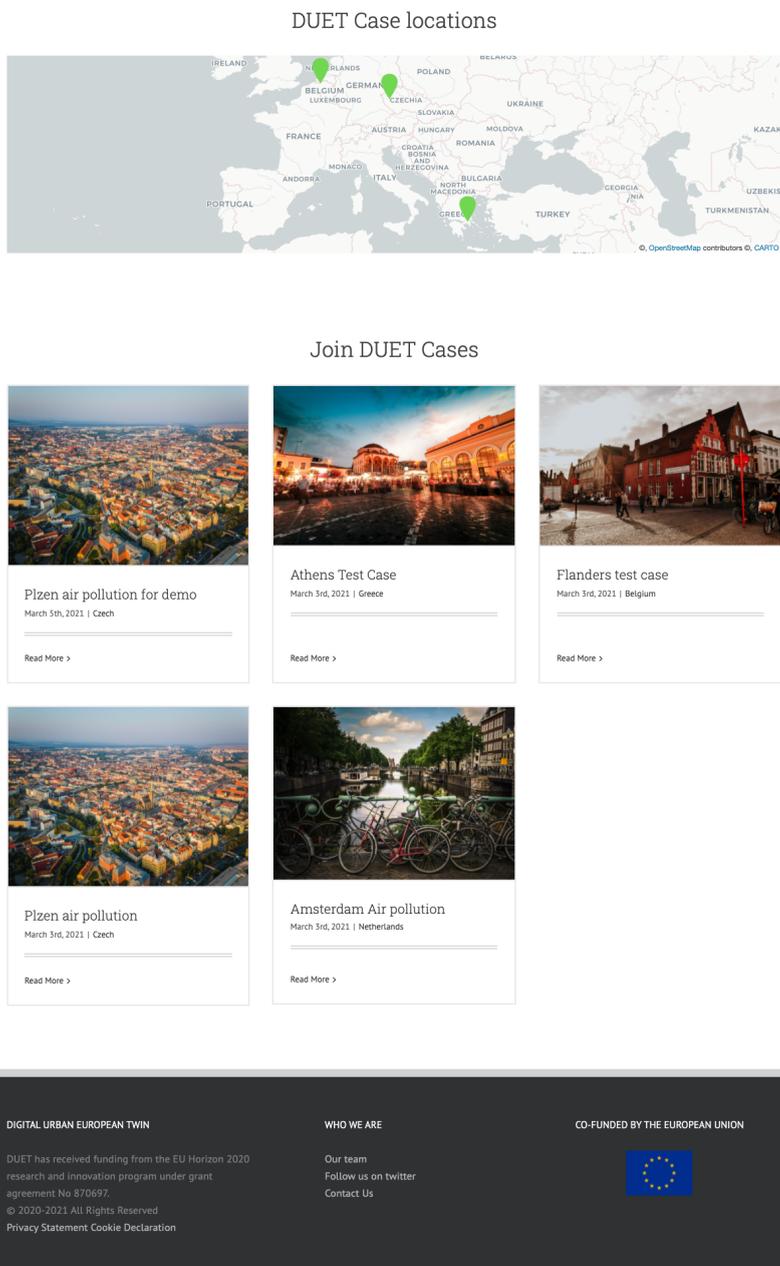


Figure 4: Landing pages - Home page 2

Clicking on the title of the case or the “Read more” button, in the search results page of the home page, leads to the case details page (Figure 5), where the user can view more information about the selected case, like the challenge and approach of the case, the data it uses, as well as available images or videos. Clicking on the “See this case live on DUET” redirects the user to the map viewer. Details about the functionalities of the map viewer are included in Section 3.5.



< Previous



STATUS : ongoing

LOCATION : Plzen, Czech Republic

TARGET AUDIENCE : citizens of Plzen

CHALLENGE

Visualize air pollution model in Plzen. Air pollution is caused by solid and liquid particles and certain gases that are suspended in the air. These particles and gases can come from car and truck exhaust, factories, dust, pollen, mold spores, volcanoes and wildfires.

APPROACH

Civil servants use the DUET solution to combine models with air pollution information to simulate the effects of the pollution in the air.

EXPECTED OUTCOME

3D visualization of the air pollution of Plzen. As a result, the air pollution plans were slightly adapted and after a new series of simulations, a congestion free solution was selected.

Categories

> Belgium

> Czech

> Greece

> Netherlands

Recent cases

> Plzen air pollution for demo

> Athens Test Case

> Flanders test case

> Plzen air pollution

> Amsterdam Air pollution

Monthly

> March 2021

How to use Used data Used models

User friendly information explaining how this case can be used for other purposes.
Description of what information is relevant and what you should look for on the DUET platform.
And most important, direct link to the Platform part of duet:

See this case live on DUET

Video



Images



March 5th, 2021 | Czech

Related Posts



Figure 5: Landing pages - Case details

Search Results



Flanders test case

CHALLENGE Visualize a case from the map. Air pollution is caused by solid and liquid particles and certain gases that are suspended in the air. These particles and gases can come from car and truck exhaust, factories, dust, pollen, mold spores, volcanoes and wildfires. STATUS : ongoing LOCATION :

March 3rd, 2021 | Belgium

[Read More >](#)



Plzen air pollution

CHALLENGE Visualize air pollution model in Plzen. Air pollution is caused by solid and liquid particles and certain gases that are suspended in the air. These particles and gases can come from car and truck exhaust, factories, dust, pollen, mold spores, volcanoes and wildfires. STATUS : ongoing LOCATION : Plzen, Czech Republic TARGET

March 3rd, 2021 | Czech

[Read More >](#)

Figure 6: Search results

3.2 Contact Us

The user can ask a question or provide feedback in the “Contact Us” page (Figure 7). The user selects the subject of his email, provides his/her name and email address, types his/her message for the DUET communication partners and presses “Send”. The message will be sent and a confirmation message will appear.

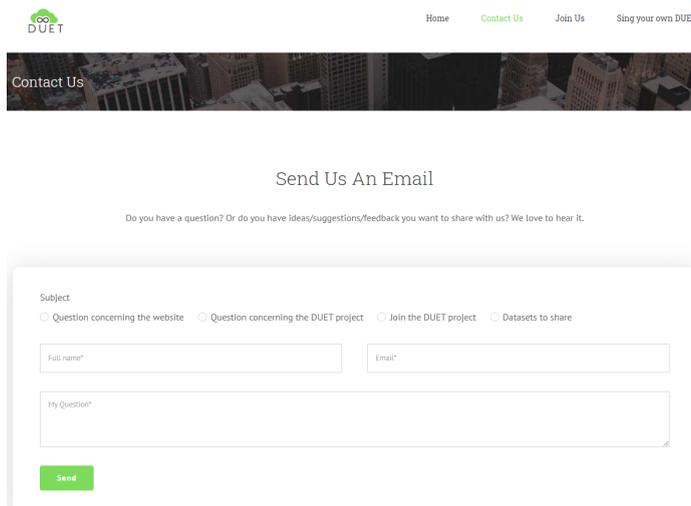


Figure 7: Contact us page

3.3 Join Us

If a user wishes to register to the DUET platform, he can do so in the “Join Us” page (Figure 8). The user fills in his / her name, email address and password and selects why he /she wishes to register to the DUET platform and the region he lives in so that he / she is assigned to the appropriate pilot. When all fields are completed, the “Submit” button will be activated. In case there is a wrong entry an appropriate message appears upon submission indicating the mistake.

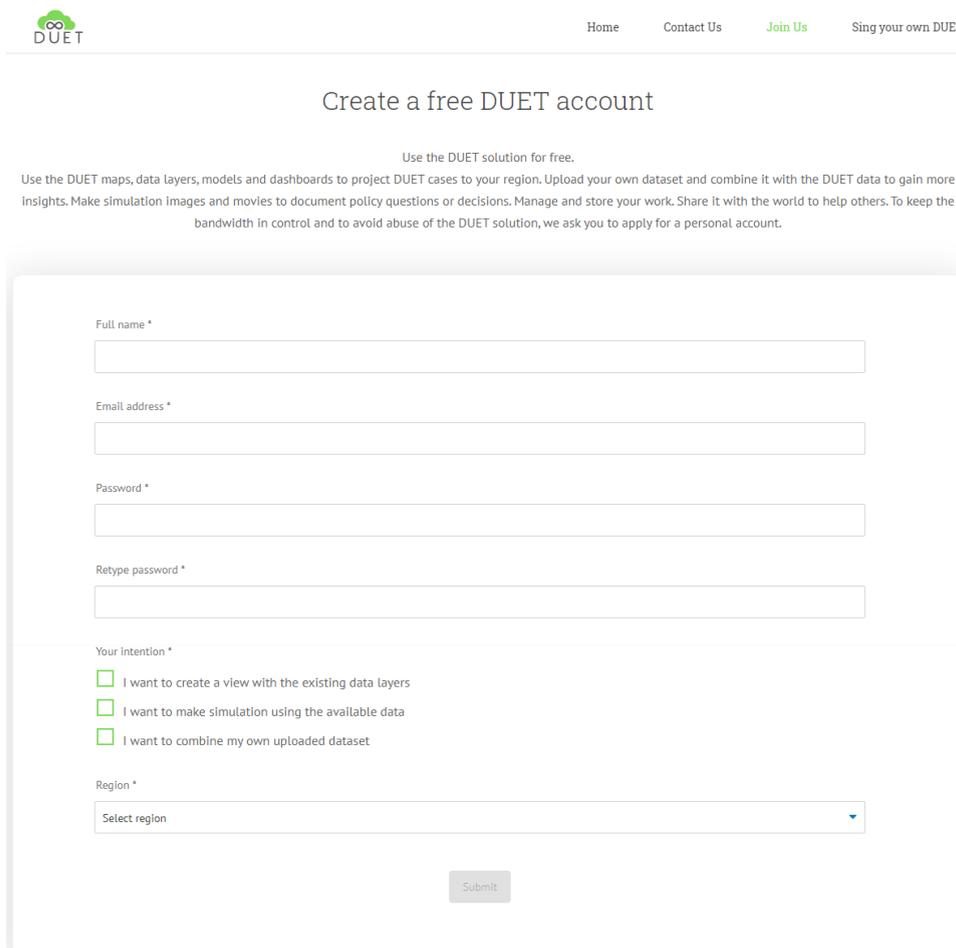


Figure 8: Join Us page

3.4 Sing your own DUET

By clicking on the “Sing your own DUET” link in the menu, the user is redirected to the main functionalities of the platform. The platform consists of a public area where visitors are able to acquire information regarding the DUET and have a basic interaction with some of the tools and private ones that the user needs to log in and have the proper rights in order to have access to.

In order to present all the functionalities of the platform, we consider the super admin user that has permissions to perform all actions.

3.4.1 Login

In order for the user to access all the functionalities of the platform, first has to login. The prerequisites are that the user has registered and the pilot administrator has accepted his registration request. The user needs to fill the form with his email and password in order to login (Figure 9). In case the credentials do not match with the ones registered in the system, an error message is presented (Figure 10). If the user has forgotten his password, he /she can request for a new one by clicking “Forgot password”. A layout appears (Figure 11), where the user can type his / her email and send the request. The system will send a newly generated password in the email provided.

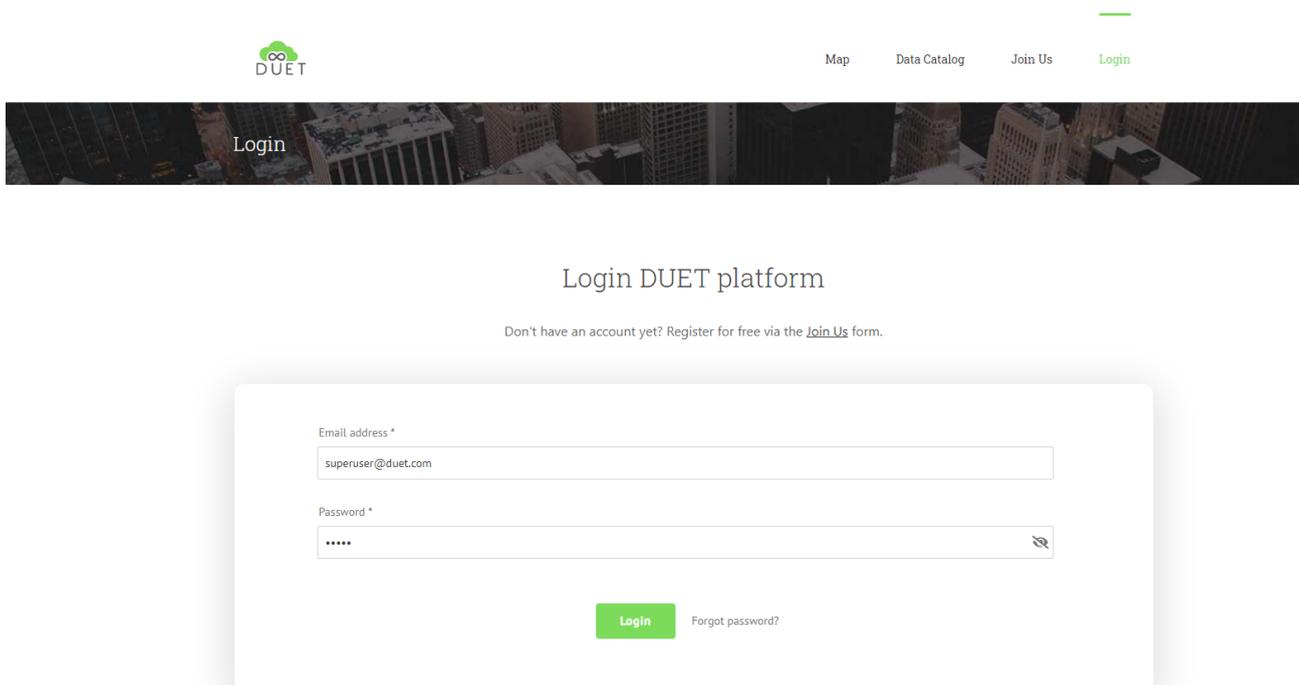


Figure 9: Login

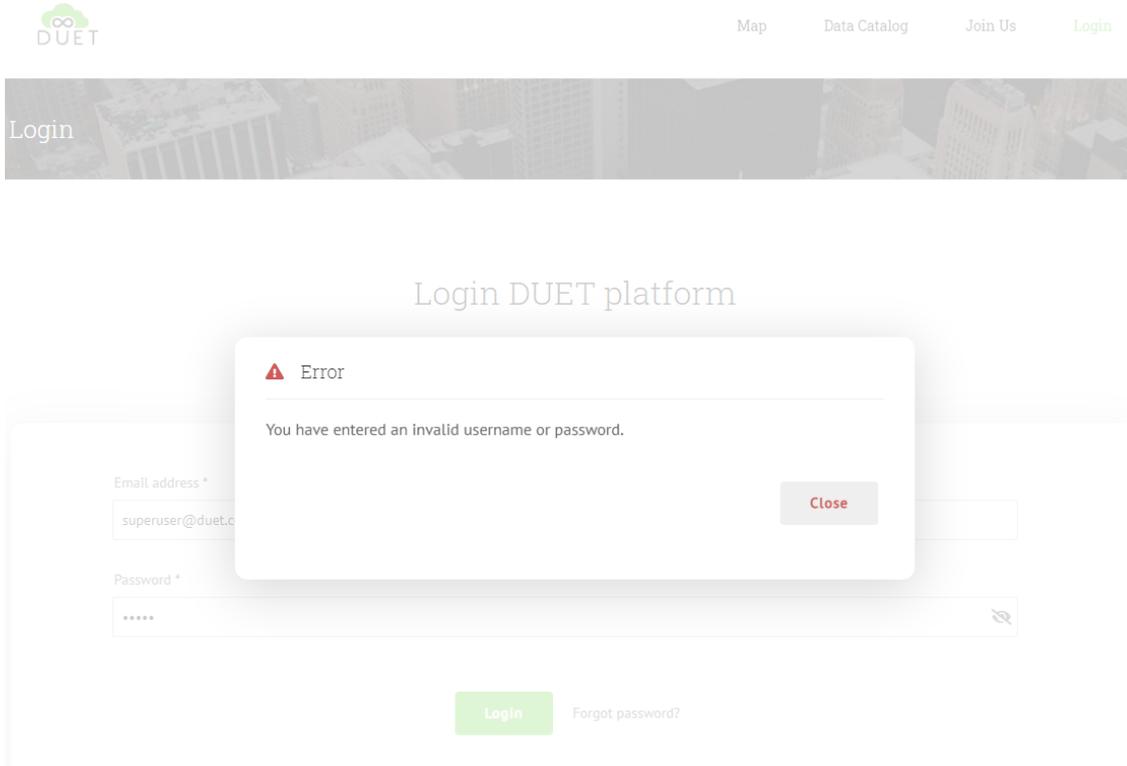


Figure 10: Wrong credentials

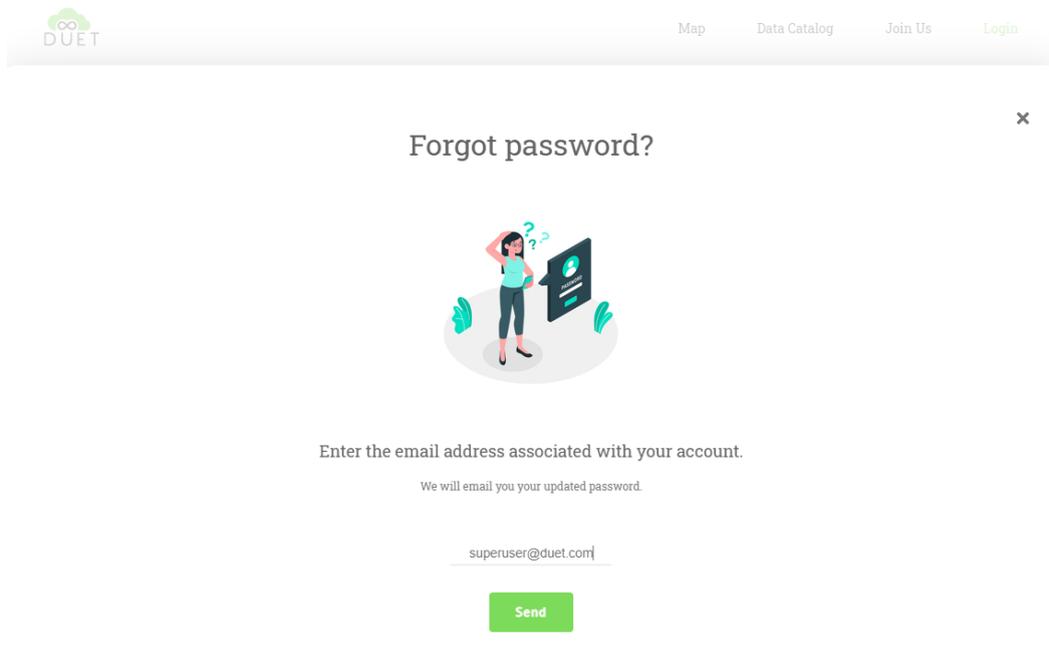


Figure 11: Forgot password

3.4.2 Users

If the logged in user has the proper rights e.g. pilot admin, he / she can manage the users and registration requests of his / her pilot (super admins can manage users and requests from all pilots).

When a visitor applies for registration in the Join Us page, the request is stored in the system and is presented to the authorized user of the pilot. This user can approve or reject a request by clicking “Yes” or “No” respectively in the “Approved” column for a specific request. In the case of a rejection a confirmation message appears first (Figure 12). Upon approval or rejection the list is updated to reflect the action (tick for approval and x for rejection) and a relevant message is sent to the email provided in the request.

| USERS | | REQUESTS | | |
|-------------------|-------------------------------|----------|---------------------------|--|
| Name | Email | Country | Intentions | Approved |
| Philippe Michiels | philippe.michiels.ext@imec.be | Flanders | combine, view, simulation | <input type="button" value="No"/> <input type="button" value="Yes"/> |
| ATC foundation | test@hotmail.com | Athens | view | <input type="button" value="No"/> <input type="button" value="Yes"/> |
| Garifalia Sebou | g.sebou@atc.gr | Athens | view | <input type="button" value="No"/> <input type="button" value="Yes"/> |
| Marina Klitsi | m.klitsi@atc.gr | Athens | combine, view, simulation | ✓ |
| ATC foundation | tenant1@uni.com | Athens | view | <input type="button" value="No"/> <input type="button" value="Yes"/> |
| chef gert | gert@cruisincuisine.be | Flanders | combine, view, simulation | ✓ |
| Jurgen Silence | jurgen.silence@sibetec.be | Flanders | combine, view, simulation | ✓ |
| ATC foundation | bug2@gmail.com | Athens | simulation, combine | ✗ |
| ATC foundation | bug1@gmail.com | Pilsen | view | ✗ |
| MAPINA ΚΑΙΤΣΗ | mklitsi@gmail.com | Athens | view | ✗ |

Figure 12: Registration requests

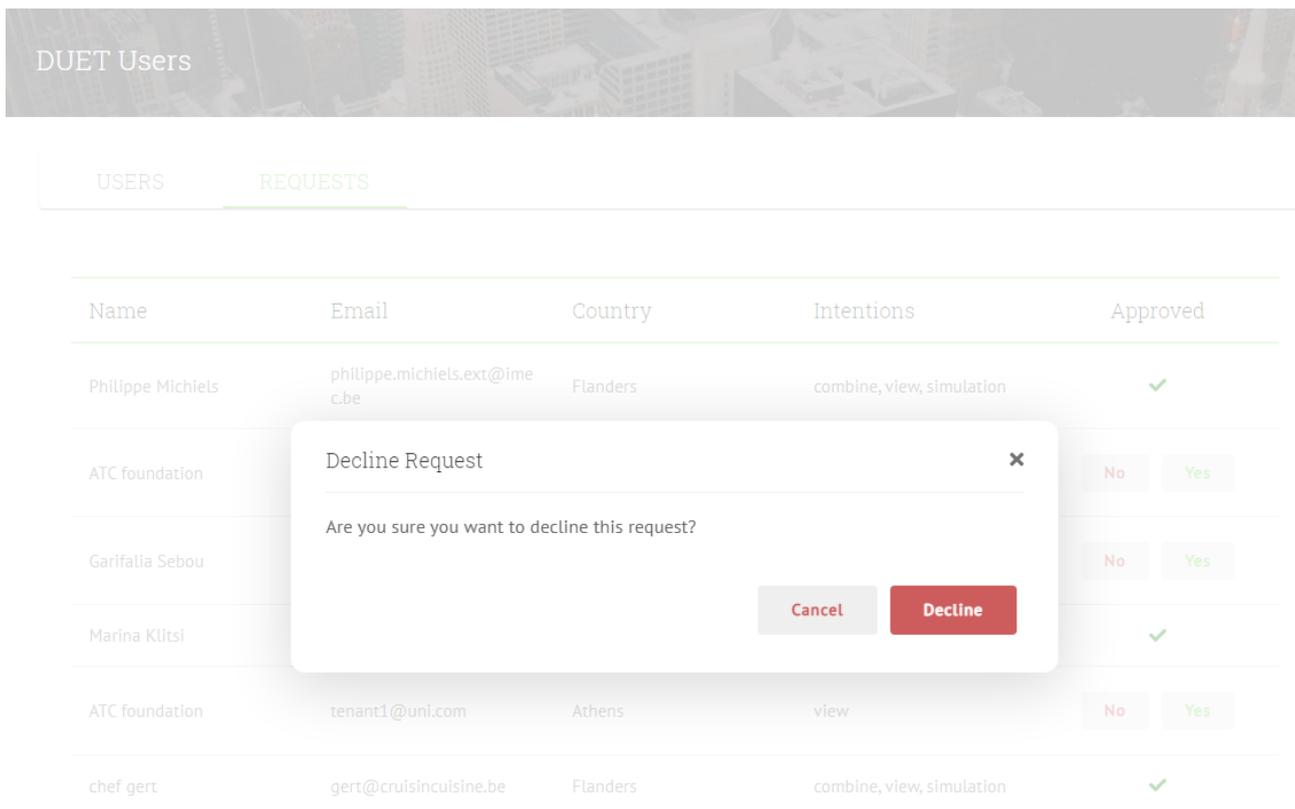


Figure 13: Request rejection

The authorized user can also view information about the approved registered users of his / her pilot (Figure 14). The authorised user can add a new user, update the details of an existing one or delete a user. For updating a registered user’s details, the user clicks on the “pencil” button and a layout appears presenting the relevant information (Figure 14) . In this layout, there details that can be updated are the name, email, role, the intentions and the pilot the user is registered to. Also the admin can deactivate or activate the user’s account and reset his / her password. When the password is reset, the new password is sent by email to the registered user.



USERS REQUESTS

[Add User](#)

| Name | Email | Updated | Role |
|----------------------|-------------------------------|------------|-------|
| ✓ Philippe Michiels | philippe.michiels.ext@imec.be | 2021-05-30 | user |
| ✓ Ilia Christantoni | i.christantoni@daem.gr | 2021-05-28 | user |
| ✓ Jurgen Silence III | jurgen.silence@gmail.com | 2021-05-28 | admin |
| ✓ Marina Klitsi | m.klitsi@atc.gr | 2021-05-25 | user |
| ✓ John D | i.dematis@atc.gr | 2021-05-13 | user |

Figure 14: Users table

User Information ✕

✓ Online / Press to deactivate [Reset password](#)

Name *

Email *

Pilot *

Role *

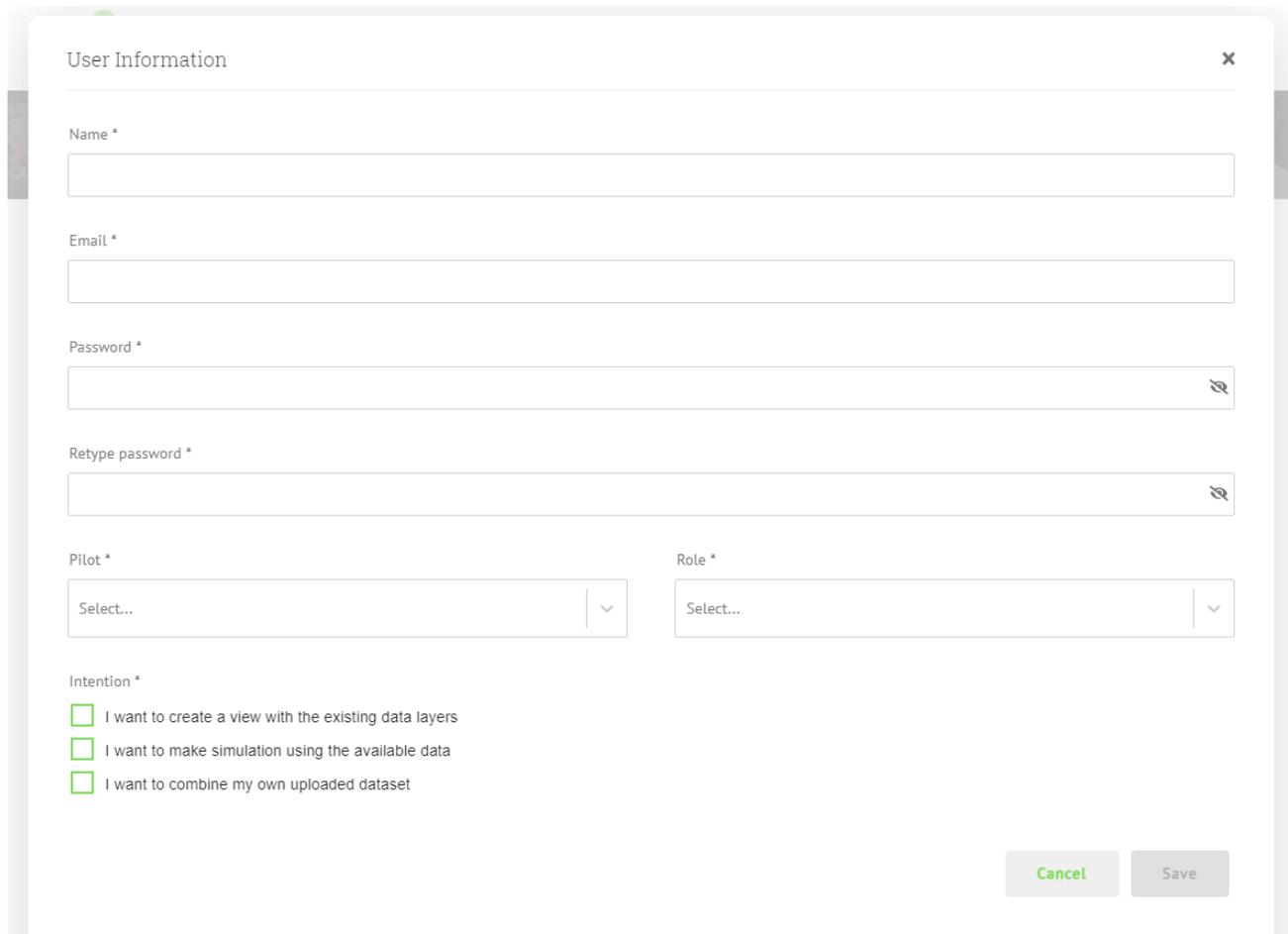
Intention *

- I want to create a view with the existing data layers
- I want to make simulation using the available data
- I want to combine my own uploaded dataset

[Cancel](#) [Save](#)

Figure 15: Update user

Similarly a new user can be added by clicking “Add user”. A new layout appears where the new user’s data can be filled in, the name, the email, the password, the pilot it belongs to, the intentions and the role in the system. In the case of the password this has to be inserted twice for confirmation to avoid any mistakes.



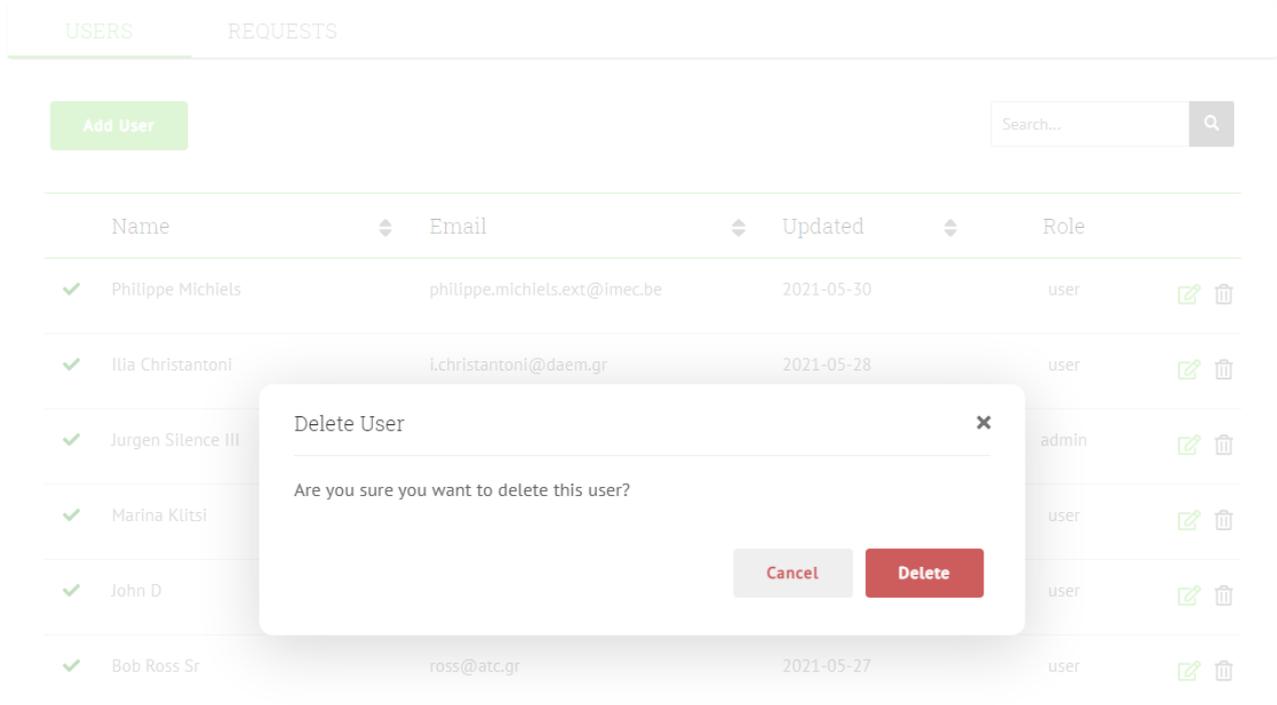
The image shows a 'User Information' form with the following fields and options:

- Name *
- Email *
- Password * (with a toggle icon)
- Retype password * (with a toggle icon)
- Pilot * (dropdown menu with 'Select...' and a downward arrow)
- Role * (dropdown menu with 'Select...' and a downward arrow)
- Intention * (checkboxes for:
 - I want to create a view with the existing data layers
 - I want to make simulation using the available data
 - I want to combine my own uploaded dataset)

At the bottom right, there are 'Cancel' and 'Save' buttons.

Figure 16: Add user

A user can also be deleted by clicking on the “recycle bin” button. A confirmation message appears to avoid any deletions by accident (Figure 17).



The screenshot displays the 'USERS' tab in the DUET interface. At the top left, there is a green 'Add User' button. To the right is a search bar with the placeholder text 'Search...'. Below these elements is a table with columns for Name, Email, Updated, and Role. The table contains several user entries, each with a green checkmark on the left and a trash icon on the right. A modal dialog titled 'Delete User' is open in the foreground, centered over the table. The dialog contains the text 'Are you sure you want to delete this user?' and two buttons: 'Cancel' (grey) and 'Delete' (red). The background table is partially obscured by the modal.

| Name | Email | Updated | Role |
|--------------------|-------------------------------|------------|-------|
| Philippe Michiels | philippe.michiels.ext@imec.be | 2021-05-30 | user |
| Ilia Christantoni | i.christantoni@daem.gr | 2021-05-28 | user |
| Jurgen Silence III | | | admin |
| Marina Klitsi | | | user |
| John D | | | user |
| Bob Ross Sr | ross@atc.gr | 2021-05-27 | user |

Figure 17: Delete user

3.4.3 Cases

A list of the cases can be viewed in the “Cases” page (Figure 18). The role of the user affects how many cases he / she can view, a “simple” user can view only the cases he /she has created, a pilot admin can view all the cases of the pilot, while a super admin can view all the cases from all the pilots.

Add Case

Q

| | Title | User | Pilot | Updated | | LP Export |
|---|----------------------|---------------------------|----------|------------|--|--|
| ? | demo closed beta | gert@cruisincuisine.be | Pilsen | 30.05.2021 | | ↪ ✎ 📄 🗑 |
| ✖ | My case - Copy | m.klitsi@atc.gr | Athens | 30.05.2021 | | ↪ ✎ 📄 🗑 |
| ✖ | My case | m.klitsi@atc.gr | Athens | 25.05.2021 | | ↪ ✎ 📄 🗑 |
| ✖ | Sofia's case | s.alysandratou@atc.gr | Athens | 21.05.2021 | | ↪ ✎ 📄 🗑 |
| ✖ | John's case | john@atc.gr | Athens | 21.05.2021 | | ↪ ✎ 📄 🗑 |
| ? | test - Copy | gert@cruisincuisine.be | Pilsen | 17.05.2021 | | ↪ ✎ 📄 🗑 |
| ? | test | jurgen.silence@sibetec.be | Flanders | 11.05.2021 | | ↪ ✎ 📄 🗑 |
| ? | test case from VCS | jurgen.silence@sibetec.be | Flanders | 11.05.2021 | | ↪ ✎ 📄 🗑 |
| ? | test_11_05 | mklitsi@gmail.com | Athens | 11.05.2021 | | ↪ ✎ 📄 🗑 |
| ✔ | base case for Athens | mklitsi@gmail.com | Athens | 07.05.2021 | | ↪ ✎ 📄 🗑 |

<
1
2
3
>

Figure 18: list of Cases

A user can create a new case, edit, copy or delete a case, as well as export it to the landing pages for review where a DUET administrator will decide if it should be published to the landing pages or not (Figure 19).



Figure 19: Cases table

Clicking on the “Edit” button for a specific case, the layout presented in Figure 20 is presented to the user holding all the information of the selected case. This information is organized in five sections, basic information, details, location, media and tabs.

In the first section (Figure 20) the user can update the title of the case, the target audience (a small description of to whom this case is addressed too), the status of the case (whether it is started, ongoing or completed) , the tags this case is associated with and a case overview.

In the details section (Figure 21), the user can update the text regarding the challenges of the case, the approach and the expected outcome of the case.

The next section is about the location of the case (Figure 22) where the user can update the city, country and geolocation coordinates of the case. Usually this information is filled automatically in the creation of the case in the map viewer.

In the media section the user can add, or remove images related to the case, either by dragging the images on the “drag n drop some images..” area or by clicking on the area and selecting one or more images from the computer (Figure 23).

The last section is about the tabs about the data sets and models that are used in the case and instructions on how to use the case in general (Figure 24).

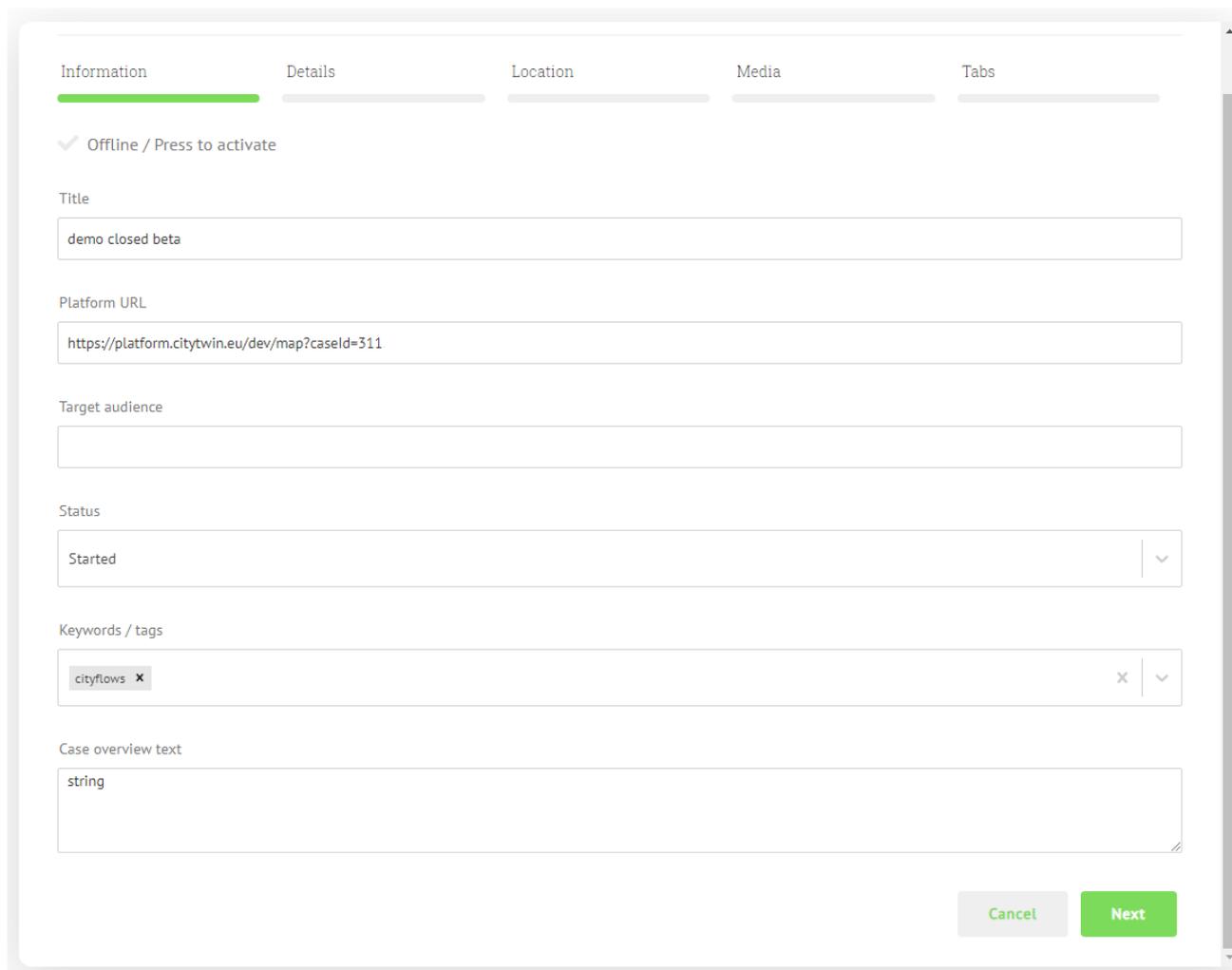


Figure 20: Case details - Information

Add Case ✕

Information Details Location Media Tabs

Challenges

Approach

Expected outcome

[Previous](#) [Next](#)

Figure 21: Case details -Details

Add Case ✕

Information Details Location Media Tabs

City

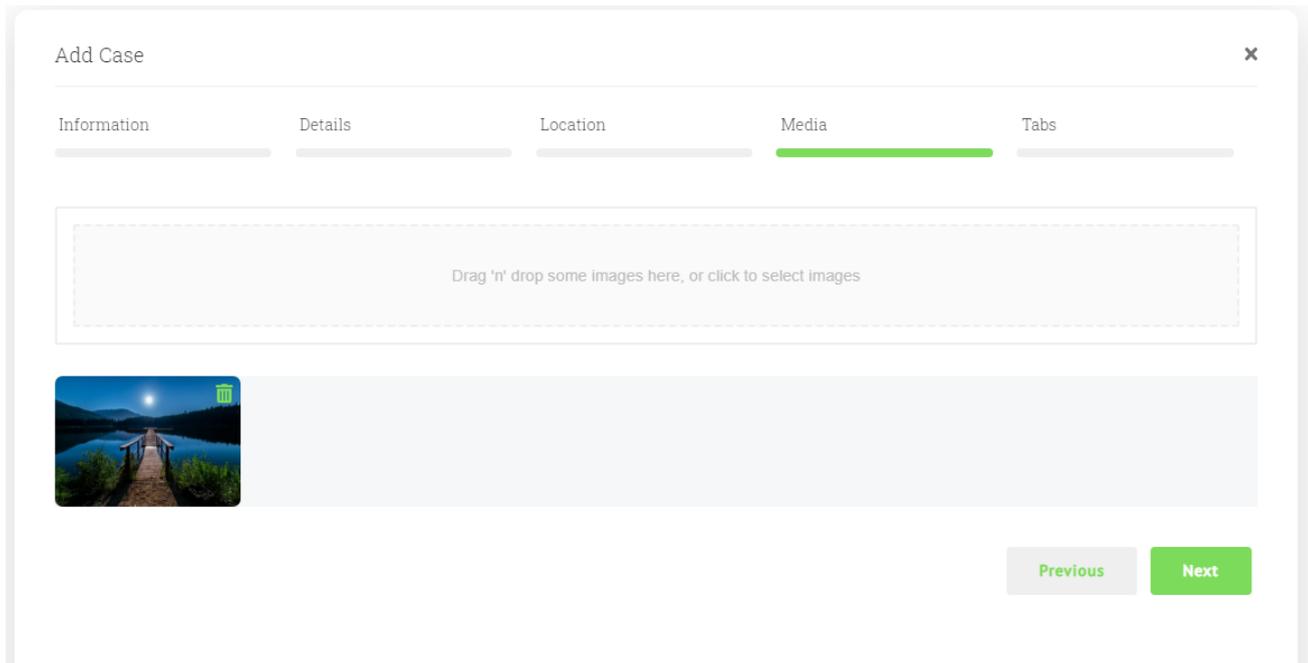
Country

X - Coordinate

Y - Coordinate

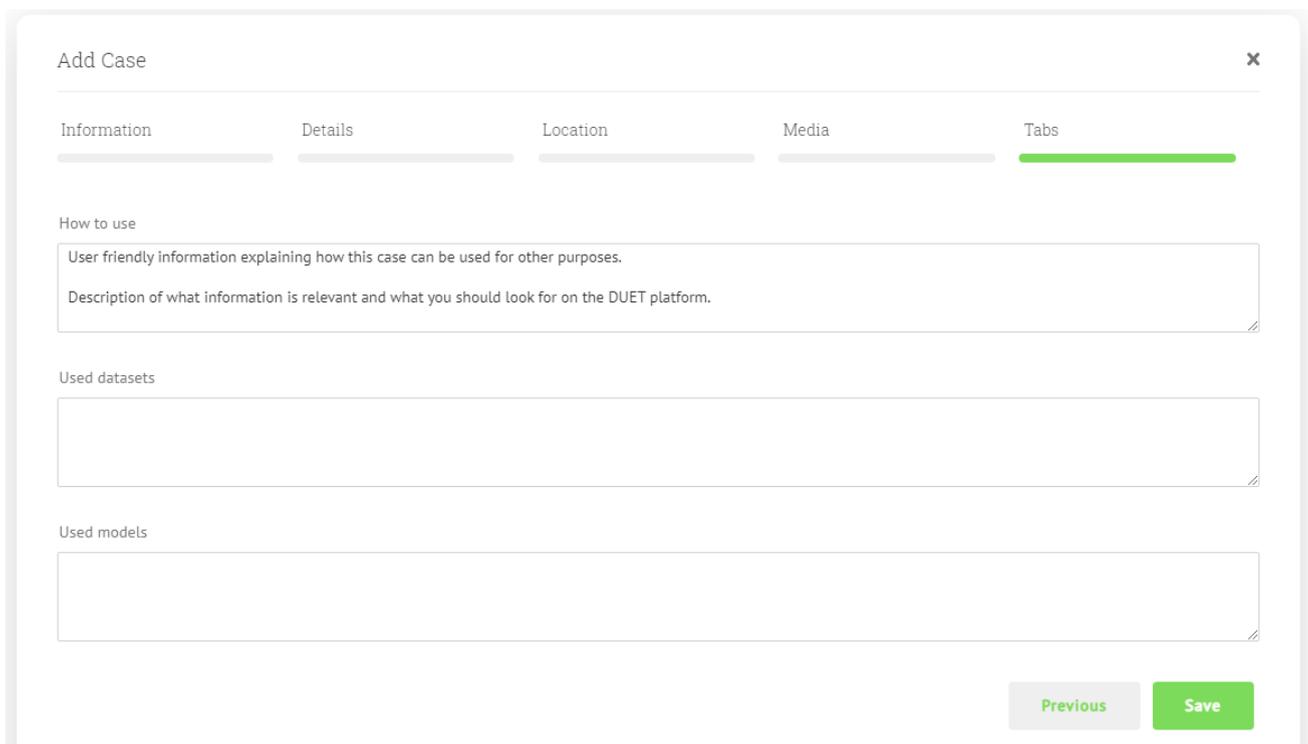
[Previous](#) [Next](#)

Figure 22: Case details - Location



The screenshot shows the 'Add Case' dialog box with the 'Media' tab selected. The dialog has a title bar 'Add Case' with a close button. Below the title bar are five tabs: 'Information', 'Details', 'Location', 'Media', and 'Tabs'. The 'Media' tab is highlighted with a green underline. The main content area contains a large dashed box with the text 'Drag 'n' drop some images here, or click to select images'. Below this is a horizontal strip where one image of a wooden bridge over a lake at night is visible, with a trash icon in the top right corner. At the bottom right, there are two buttons: 'Previous' (disabled) and 'Next' (active).

Figure 23: Case details - Media



The screenshot shows the 'Add Case' dialog box with the 'Tabs' tab selected. The dialog has a title bar 'Add Case' with a close button. Below the title bar are five tabs: 'Information', 'Details', 'Location', 'Media', and 'Tabs'. The 'Tabs' tab is highlighted with a green underline. The main content area contains three text input fields. The first is labeled 'How to use' and contains the text: 'User friendly information explaining how this case can be used for other purposes. Description of what information is relevant and what you should look for on the DUET platform.' The second is labeled 'Used datasets' and is empty. The third is labeled 'Used models' and is empty. At the bottom right, there are two buttons: 'Previous' (disabled) and 'Save' (active).

Figure 24: Case details - Tabs

Adding a case follows the same process like updating one with the difference that all the fields are empty.

The user can also copy a case by clicking on the “copy” button, in order to prepare a new case based on a case already defined. The new case has all the fields pre filled with the values of the original case.

| Title | User | Pilot | Updated | LP Export | |
|------------------|-----------------|--------|------------|------------|--|
| ✖ My case - Copy | m.klitsi@atc.gr | Athens | 30.05.2021 | | |
| ✖ My case | m.klitsi@atc.gr | Athens | 25.05.2021 | 25.05.2021 | |

Figure 25: Case copying

For deleting a case the user needs to click on the “recycle bin” button and confirm in the deletion message presented (Figure 26).

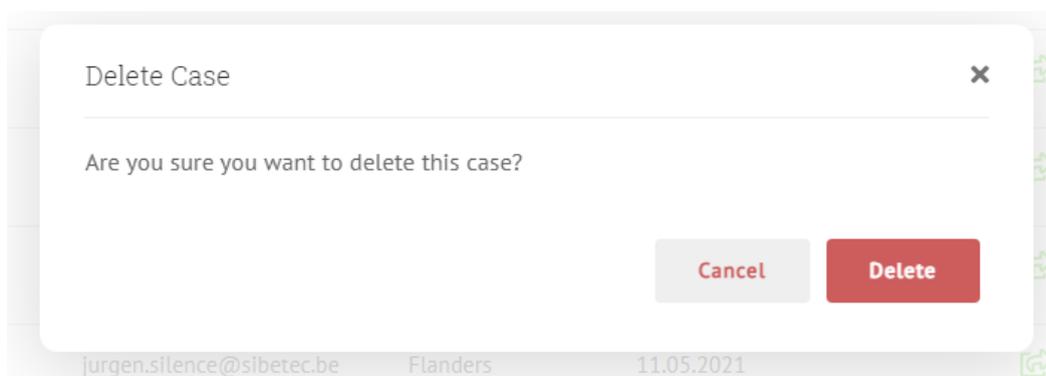


Figure 26: Remove

3.4.4 Data catalog

In the data catalog page, the user can search for registered datasources in the system or add a new one. Initially the data sources list is populated with all the available data sources in the system (Figure 27).

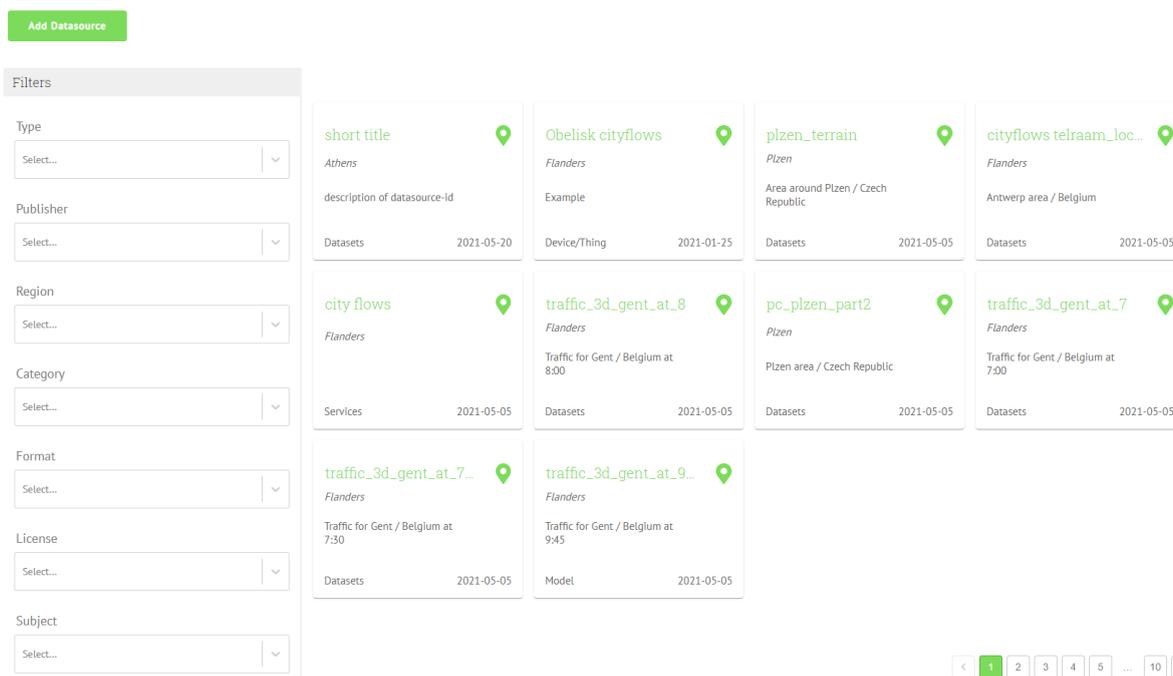


Figure 27: Data catalog

For searching for data sources of specific characteristics, the user can select from the filters on the left side of the page, the type of the data sources (e.g. whether it is about rest apis or web hooks), the publisher of the datasource, the region (one of the three pilots), the format of the of the datasource etc. The values of these filters are populated based on the data sources registered and the number next to the values refers to the number of the data sources associated with the specific value, eg. 36 registered data sources in the Flanders region.

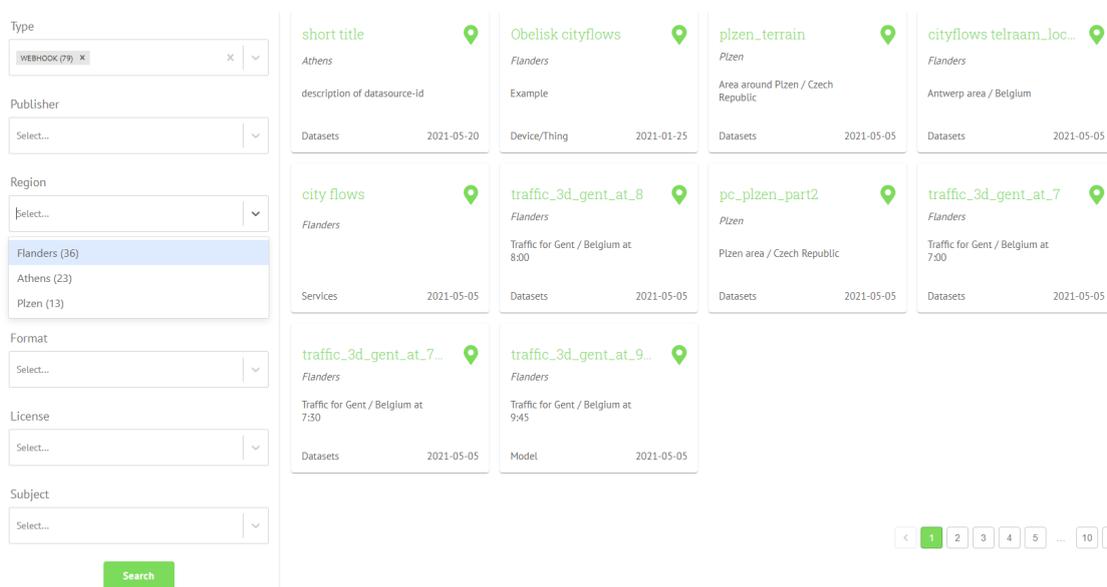


Figure 28: Filtering by region

Clicking on the “Add datasource”, presents the overlay depicted in Figure 29. The user needs to complete several fields divided in different sections. First the user must complete generic information about the data source like region, title, category and subject. All the fields with a * next to the fields name are mandatory.

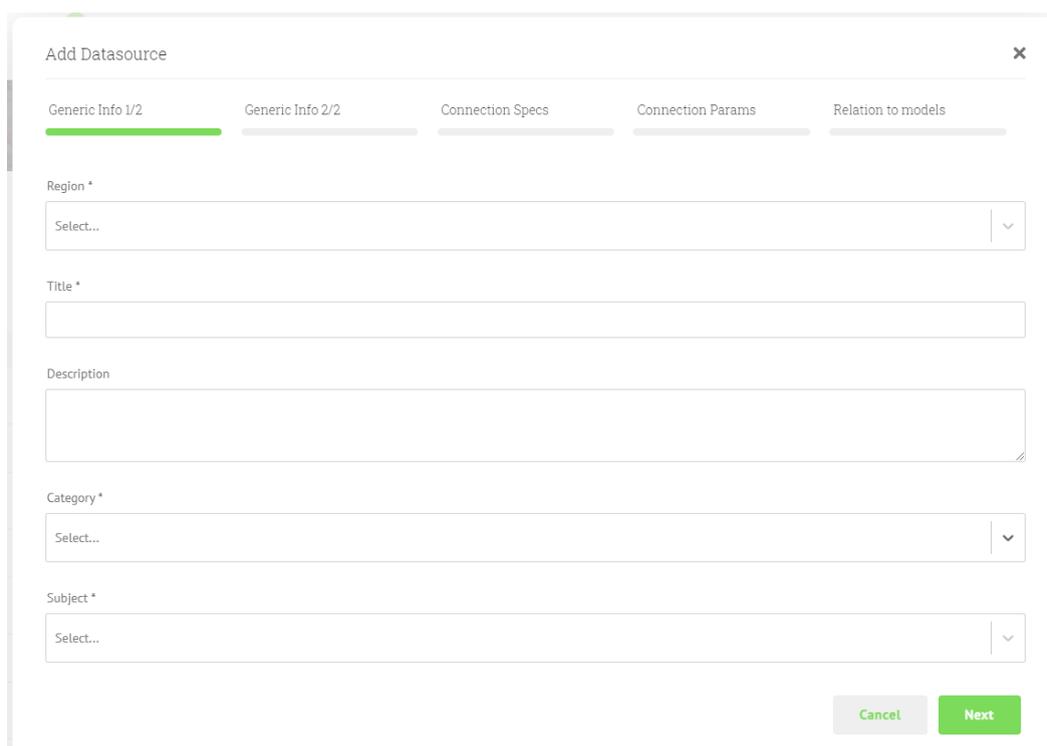


Figure 29: Add data source

In the next step the user needs to select from the relevant lists, the publisher of the datasource, the format of the data and the license scheme of the datasource (Figure 30).

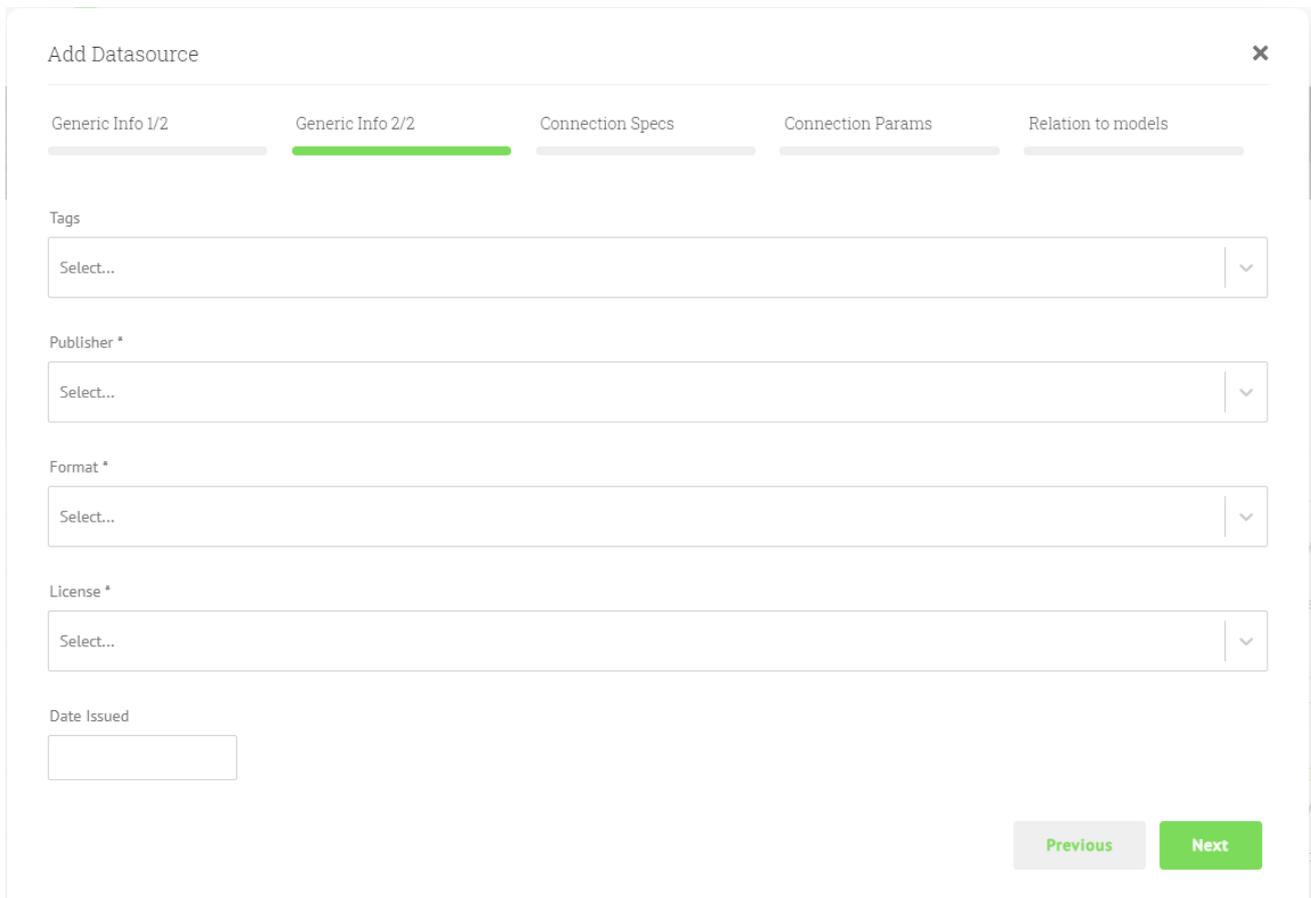
The screenshot shows a web form titled 'Add Datasource' with a close button (X) in the top right corner. The form is divided into five steps: 'Generic Info 1/2', 'Generic Info 2/2' (the current step, highlighted with a green bar), 'Connection Specs', 'Connection Params', and 'Relation to models'. Below the progress bar, there are five input fields: 'Tags' (a dropdown menu with 'Select...' and a downward arrow), 'Publisher *' (a dropdown menu with 'Select...' and a downward arrow), 'Format *' (a dropdown menu with 'Select...' and a downward arrow), 'License *' (a dropdown menu with 'Select...' and a downward arrow), and 'Date Issued' (a text input field). At the bottom right, there are two buttons: 'Previous' (a light gray button) and 'Next' (a green button).

Figure 30: Generic information 2

Next is the connection specification of the datasource, that is, how a client can consume this datasource, so the user needs to select the type of the datasource and provide any connection details like the url that the datasource is accessible and any credentials needed (Figure 31).

The screenshot shows the 'Add Datasource' dialog with the 'Connection Specs' tab selected. The tabs are: Generic Info 1/2, Generic Info 2/2, Connection Specs (active), Connection Params, and Relation to models. The form contains the following fields:

- Type * (dropdown menu with 'Select...' and a downward arrow)
- Datasource URL (text input field)
- Client ID (text input field)
- Client Secret (text input field)
- Schedule (text input field)

At the bottom right, there are two buttons: 'Previous' (disabled) and 'Next' (active).

Figure 31: Connection specs

In the connection parameters, the user defines any query or path parameters needed for the connection with the data source. The user first selects the type of the parameters and then clicks on the plus button to add a parameter. In the key text field, type the name of the parameter e.g. “name” and in the type text field the type of the parameter e.g. string, integer etc.

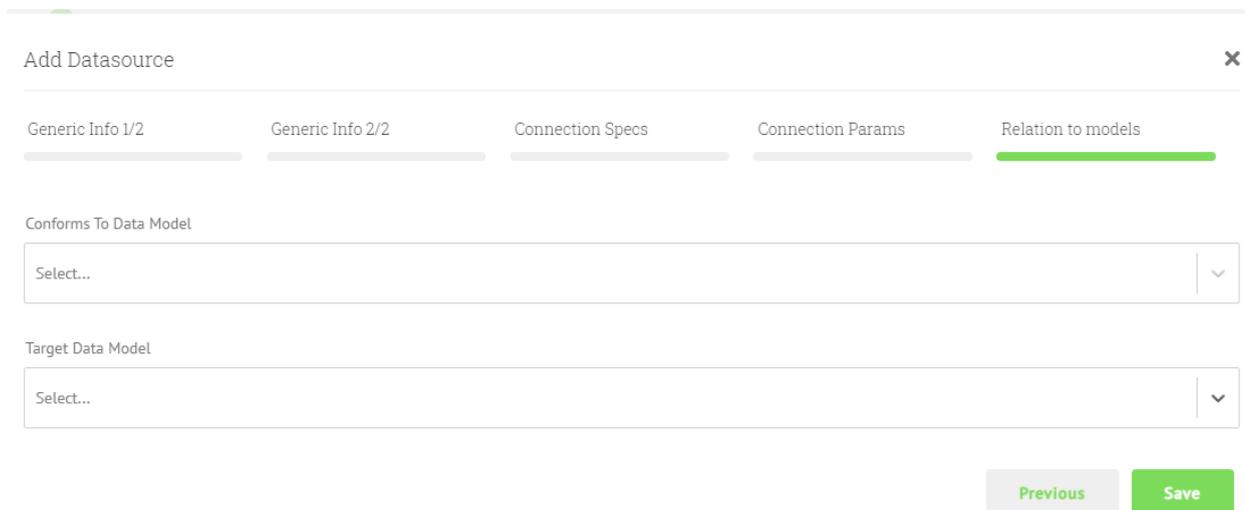
The screenshot shows the 'Add Datasource' dialog with the 'Connection Params' tab selected. The tabs are: Generic Info 1/2, Generic Info 2/2, Connection Specs, Connection Params (active), and Relation to models. The form contains the following elements:

- Parameter Type (dropdown menu with 'QueryParams' and a downward arrow)
- Table with two columns: 'name' (text input) and 'string' (text input). A green minus sign is on the right.
- + Add parameter (button)
- JSON snippet: [{"key": "name", "type": "string"}]

At the bottom right, there are two buttons: 'Previous' (disabled) and 'Next' (active).

Figure 32: Connection parameters

The last section the user defines if there is any compatibility that the datasource has with any of the registered data models in the system.



The screenshot shows a web interface for adding a data source. At the top, there is a title bar 'Add Datasource' with a close button 'x'. Below it is a horizontal progress bar with five tabs: 'Generic Info 1/2', 'Generic Info 2/2', 'Connection Specs', 'Connection Params', and 'Relation to models'. The 'Relation to models' tab is currently active and highlighted in green. Under this tab, there are two dropdown menus. The first is labeled 'Conforms To Data Model' and the second is labeled 'Target Data Model'. Both dropdowns have 'Select...' as their placeholder text. At the bottom right of the form, there are two buttons: 'Previous' (disabled) and 'Save' (active).

Figure 33: Relation to models

3.5 Map Client

The following sections present the main functionalities that are available to the user via the map client.

3.5.1 Case support

Cases can be described as entries into the system, with a view specific for a certain analysis, discussion, policy action/decision, ... Depending on the selected case, datasets are selected, settings are filled in, relevant models will be presented and dashboards will be created in the platform component of the DUET portal. Cases are registered via cases-api and provided to users via the DUET platform or the visualization client directly. Each case holds a lot of information as intended use, expected outcome and used datasets, thus each case can be loaded and used inside the DUET environment.

As for the DUET portal, we defined 3 case variants:

- **Prepared cases** where the initial settings are predefined/preset to get an optimal visualisation of a specific case and its accompanying analysis as a starting point. This category includes the showcases, which are exceptional cases fully worked out, being key examples for similar (future) studies.
- **Semi generic cases** offer more exploration freedom to the visitor/user. For these pilot related and epic-based (see Deliverable 2.3) cases, the initial settings and selected datasets can be easily adapted according to the personal interest or angle of approach of the visitor/user.
- **Generic cases** are defined by the user himself/herself. It is the user who decides which of the available data layers, models or dashboard will be utilized. Supported by the information available on the password protected pages (the user needs to apply for a password first), datasets, models and dashboards can be freely used and additional datasets can be **uploaded** to create an own - tailor made - DUET with access to all available datasets and tools.

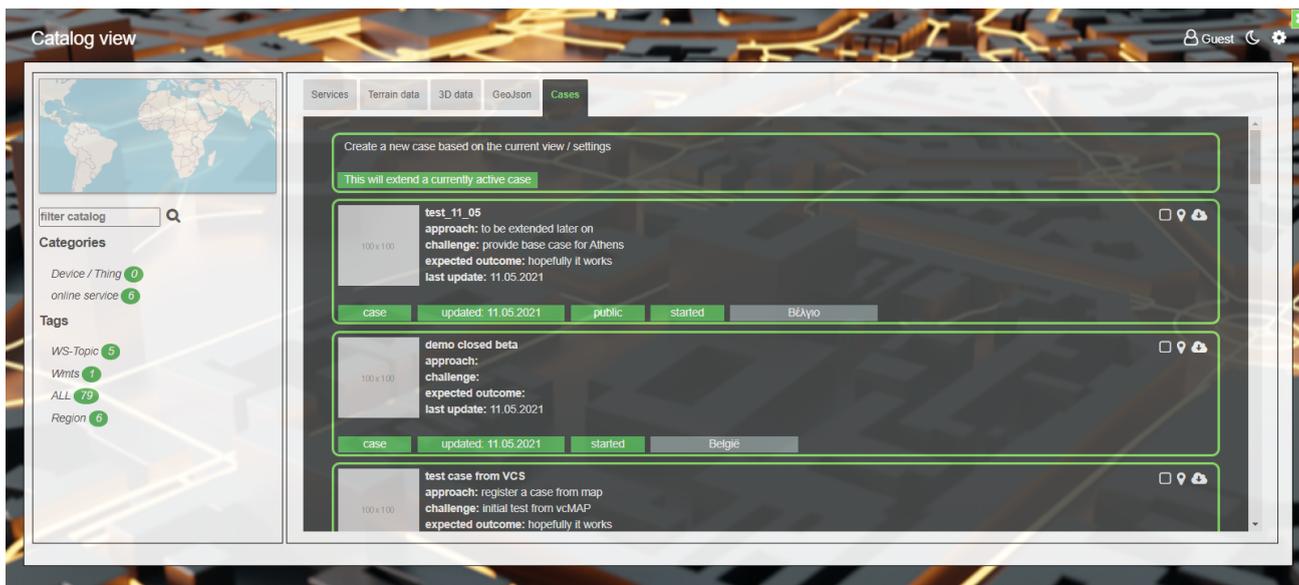


Figure 34: Duet data catalog view - cases tab

Cases are presented to the user inside the visualization component in the catalog view’s cases tab. Here all listed cases of the cases-api will be shown to the user. Inside the tab several functions are provided to the user.

1. overview about existing cases in DUET environment, see “Load a case”
2. creation of a new case, see “Create a new case and save it via cases-api”
3. update / extension of an existing case, see “Extend and save a case via cases-api”

As figure 34 & figure 35 show, the whole content of the DUET cases catalog is presented to the user as a scrollable list of entries. The list is sorted by issue / update date and uses the sortBy and sortOrder functionality of the DUET cases-api. Each entry in the list of cases owns some descriptive and some functional items. The descriptive items are listed on the left hand side of each entry.

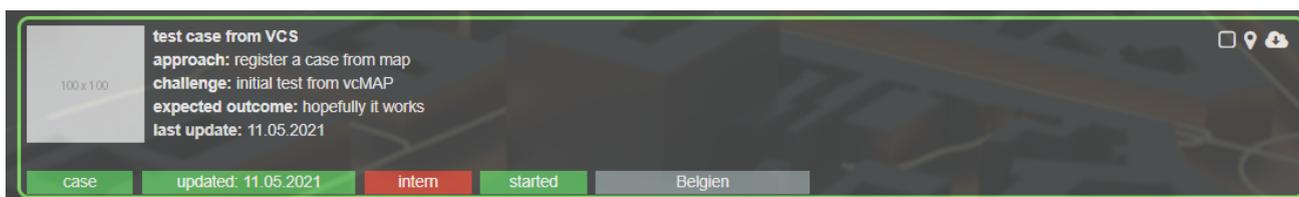


Figure 35: DUET data catalog view - cases tab - case entry

The descriptive elements are:

1. a thumbnail of the case, if available. if not a placeholder will be shown followed by:
2. title of the case
3. approach description of the case
4. challenge description of the case
5. expected outcome description of the case
6. last update date of that case

Thus the most relevant information except for the descriptive elements such as title etc. are highlighted. The highlighted elements are:

1. case
2. update date
3. target Audience - depending on audience the tag is green = public, red = intern, yellow = all others,
4. status of the case - depending on status the tag is green = started, red = stopped, yellow = all others
5. region or city where the case is located

On the upper right corner the functional elements of each case entry are located. Here, similar to a catalog entry the functional items are:

1. show the extent of the case - square element. Clicking on that item takes the extent (BBOX) of the case and shows it inside the overview map.
2. jump to position in map - place marker. Clicking on that item takes the BBOX of that case and calculates a center position and jumps to that position in the main view.
3. load case - download sign. Clicking on that item fetches a case and adds the datasets associated to that case to the list of map datasets, see chapter “Load a case”.

Load a case

Loading a case inside the visualization component, works on clicking on the download case sign. By doing so, the whole map configuration will be loaded and the associated data sources will be added to the legend (content) tab of the visualization component. In addition to that, user defined styles will be fetched from case definition and will be added to the map as well, to be used in the style section of each layer, see “Edit legend entry and create a user style”.

Create a new case and save it via cases-api

If a user wants to create a new case, the following steps have to be undertaken:

1. choose datasets from the data catalog and add them to the map, see “Add data to viewer and visualize”
2. name / rename the content entries, see “Edit legend entry and create a user style”
3. create an appropriate style, see “Edit legend entry and create a user style”
4. afterwards open the data catalog again and switch to the cases tab
5. follow the guidelines below

At the top of the cases tab, the uppermost list element (see figure 36) provides users the option to save a new case or update an existing case. By simply clicking on the list element a new dialog opens, providing some UI elements relevant for creating or updating a case.

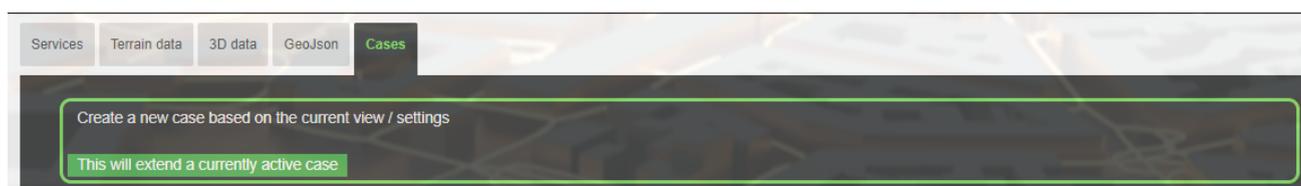


Figure 36: data catalog view - cases tab - create or update a case

As figure 37 shows, the “save case” dialog asks for some user input by text fields. Main fields of user required inputs are the same fields as used for describing a case inside the duet case list view, except for the tags fields that are mainly used in [Duet platform \[https://staging.citytwin.eu/\]](https://staging.citytwin.eu/) word cloud, filter function for cases, etc. By clicking on “Save case” Button some other required fields will be generated by the underlying

API. Thus, country, city, position will be created via geocoding using the Nominatim API [3] and fetching the relevant parameters via posting the BBOX of the current view extent. Thus it is guaranteed that each case has an extent and a location. Afterwards all relevant parameters will be taken and pushed via HTTP protocol POST to the DUET CASES API.

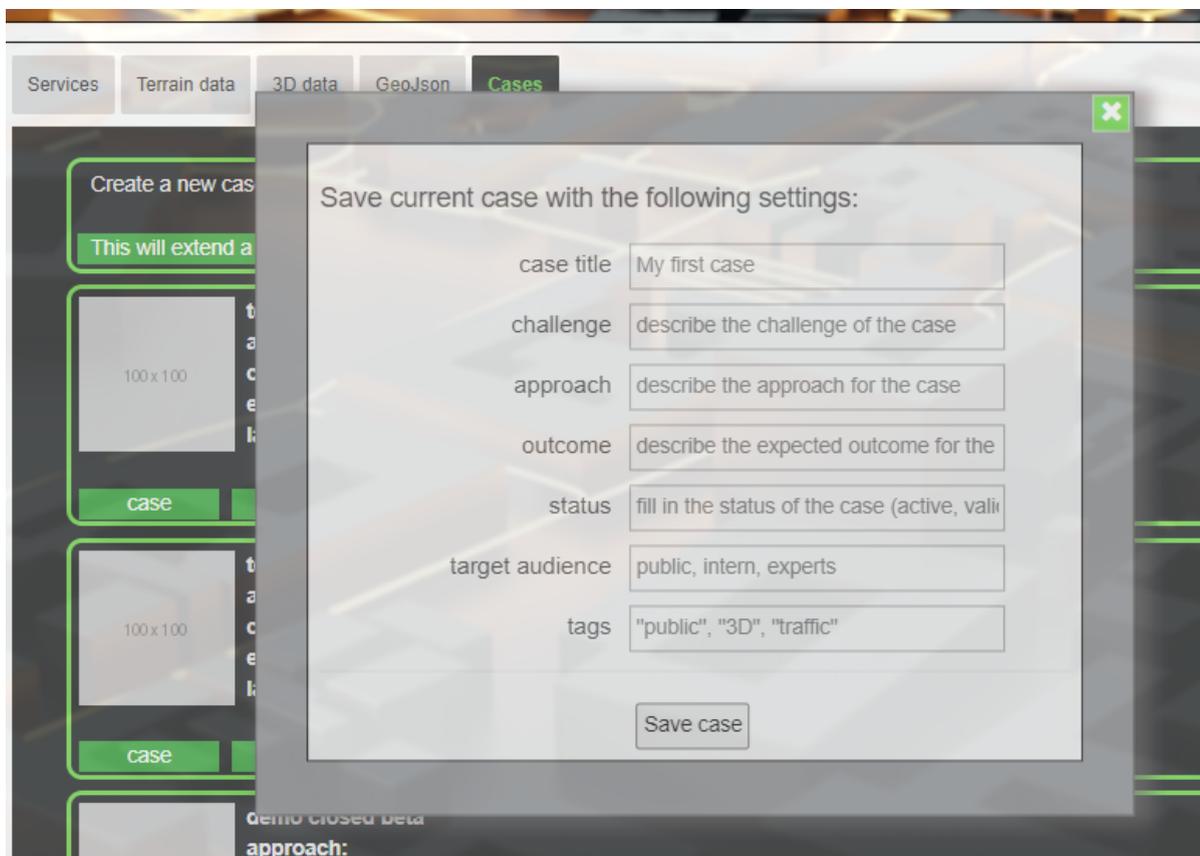


Figure 37: data catalog view - cases tab - create a new case dialog

The user will be informed about success or failure via notifications.

Extend and save a case via cases-api

If a user wants to create a new case, the following steps have to be undertaken:

1. Load a case from cases tab, see Load a case
2. Choose datasets from the data catalog and add them to the map, see Add data to viewer and visualize
3. name / rename the content entries, see Edit legend entry and create a user style
4. create an appropriate style, see Edit legend entry and create a user style
5. afterwards open the data catalog again and switch to the cases tab
6. follow the guidelines below

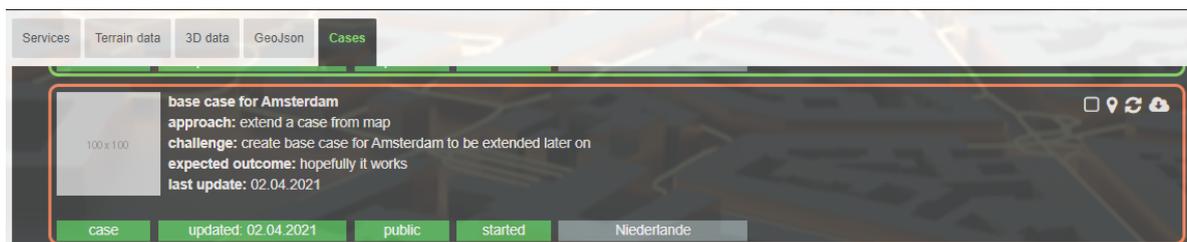


Figure 38: data catalog view - cases tab - loaded case entry

In addition to figure 38, where a standard case entry is shown, figure 39 shows an already loaded case - highlighted by a “coral” colored frame. A new symbol is added to the functional section, showing a reload / update sign. The user can now see which case is loaded and is able to extend that case by adding new styles or datasets to that case. The first either by creating a new style as described in [Styling non-time dependent data based on individual attributes](#) or [Styling time dependent data based on individual attributes](#). The latter by choosing dataset entries from the list of datasets and adding it to the visualization component.

If a user clicks on that symbol or the main entry point for saving a case, the “Save Case” dialog opens (see figure 39), showing now instead of default value filled fields - the information of the current loaded case. The user can now keep the fields or change them and can then decide whether to save that case as a new case or simply update the existing case. By doing the latter, the existing case will be overwritten with the settings and properties of the new one.

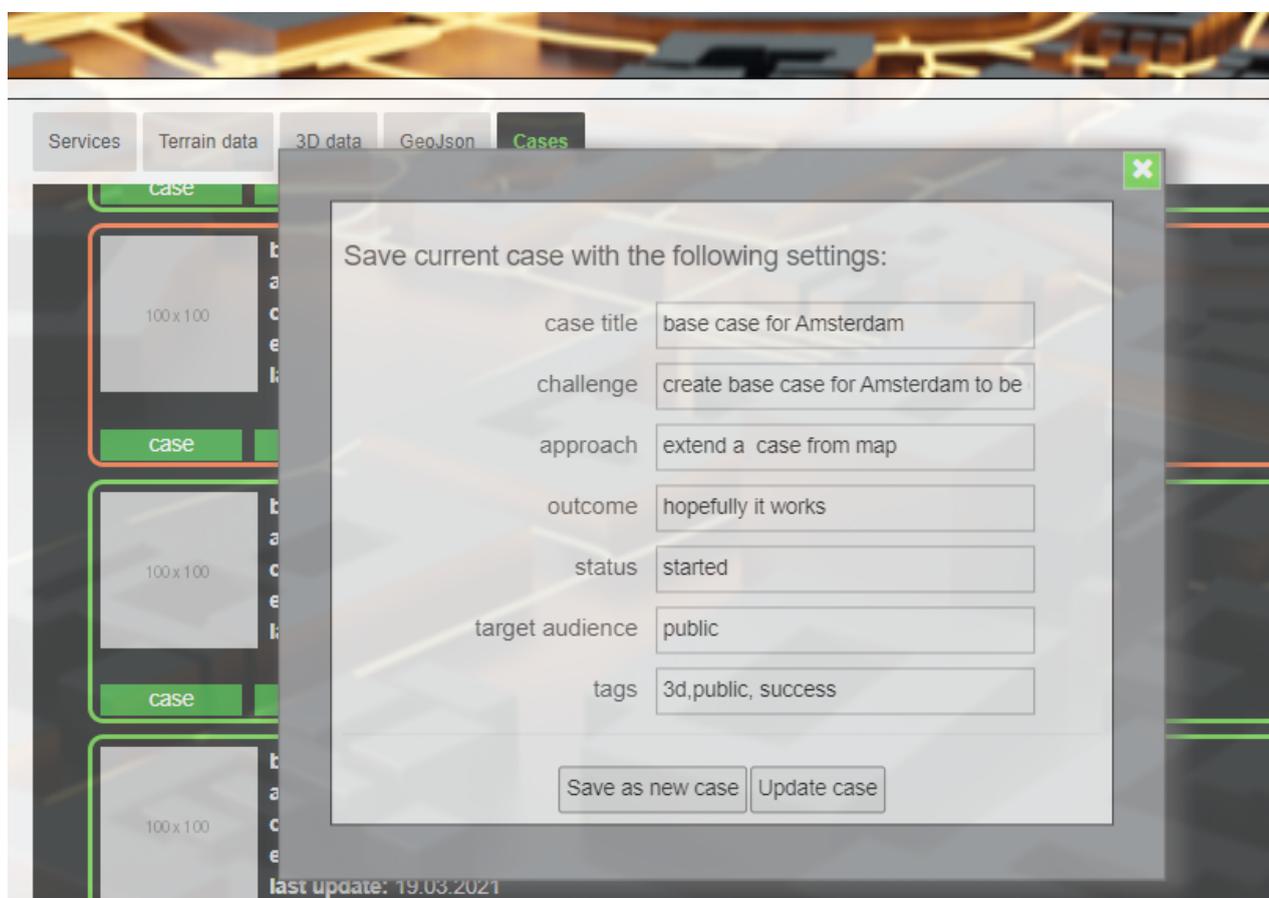


Figure 39: data catalog view - cases tab - update case dialog

3.5.2 Add data to viewer and visualize

By clicking one of the available POI's (step 1/figure 40) the user will be directed to one of the pilot regions. By clicking on the “Data” (step 2 / figure 40) the user can open the Data Catalog.

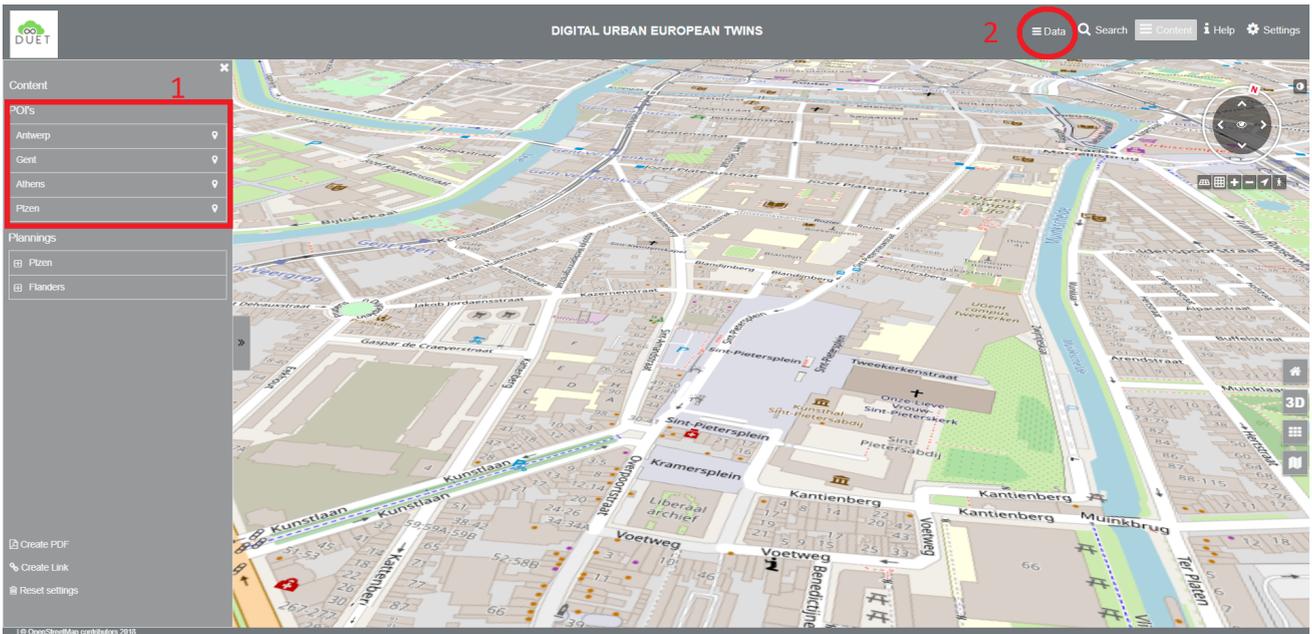


Figure 40: Map Client: Add data to viewer and visualize

Doing 1 at first will enforce the viewer to jump to a certain location. The current view point + 20km circumference will be used to filter the data of the catalog against the user's position. Thus on start-up of the data catalog only datasets with respect to the user's region will be shown.

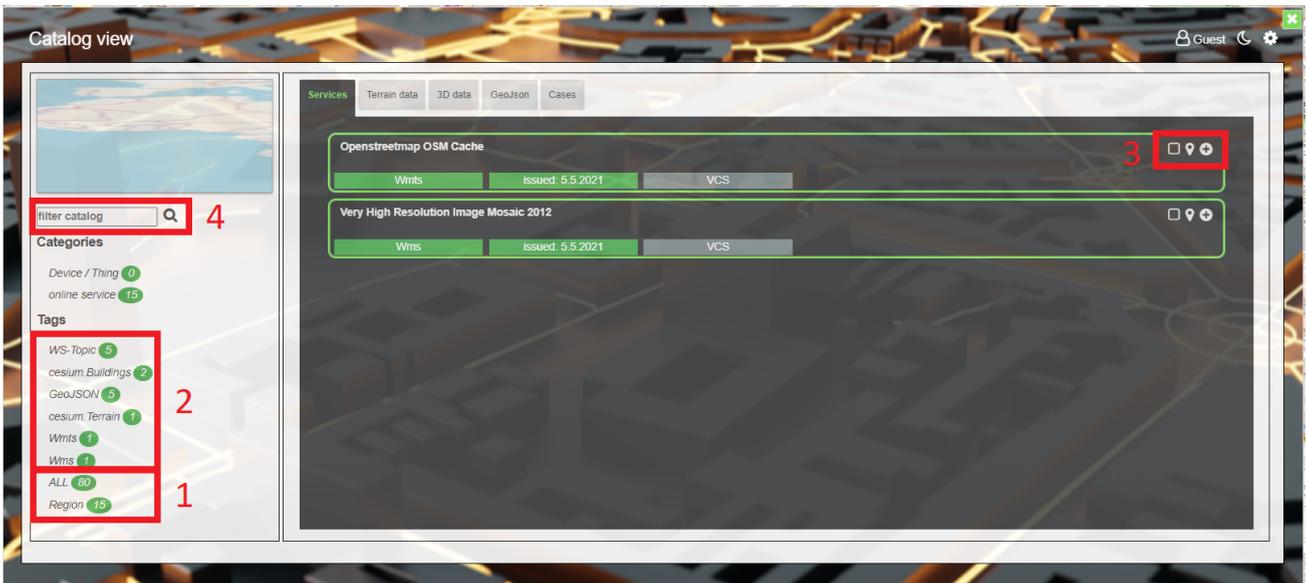


Figure 41: DUET catalog view

1. The tags in 1 (figure 41) shows the number of datasets at all or in the current region. Clicking on one of the tags enables a switch between Regional and Total datasets.
2. The tags in 2 indicate what kind of datasets are available with respect to the chosen option of 1.
3. The symbols in 3 provide functionality to the user, as specified below (from left to right)
 - a. shows the extent of the dataset in the overview map on the left hand side
 - b. jumps to the center point of the dataset in main map view
 - c. adds the dataset to the map content
4. typing a word (place, type or part of dataset) searches the data catalog against that word and will provide the datasets fulfilling the search parameter inside the respective datatype tabs. Thus the

search phrase “air” will bring up 5 datasets for example. However the search function is developed for the whole data catalog and provides a full list of datasets fulfilling the search parameter. The results **are not bound** to the current region.

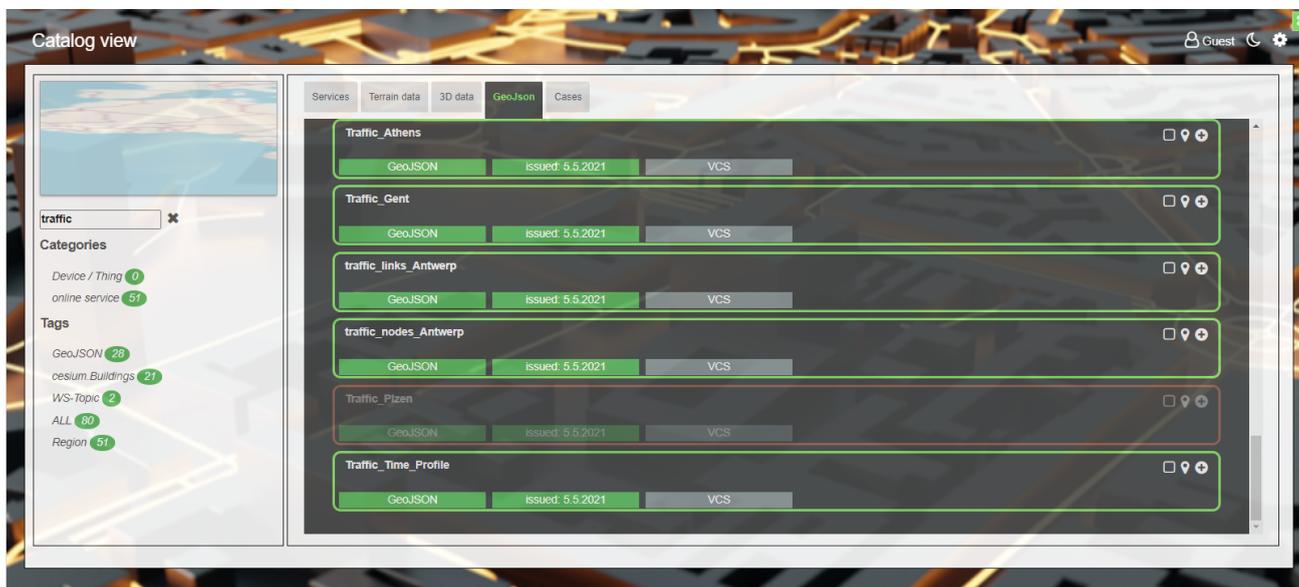


Figure 42: DUET catalog view in VCS client with selected dataset searched by parameter “traffic”

Once the correct dataset is found, the user can add the data to the map by clicking on the plus sign on the right hand side of each dataset. By doing so, the dataset will be greyed and bordered in coral color. This indicates that the dataset is already in use by the client. Even after closing the catalog view and reopening it, the datasets in use will be highlighted like that.

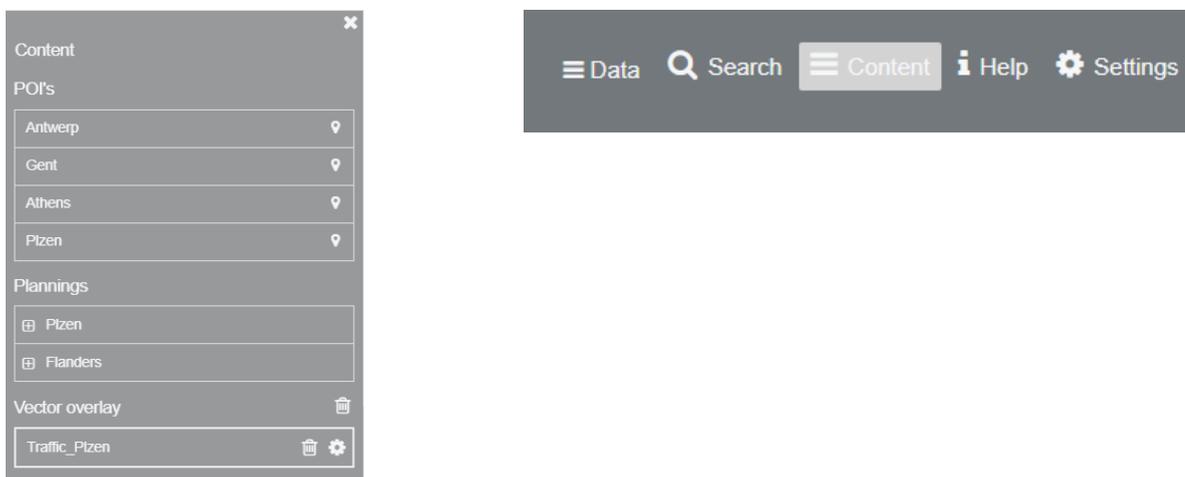


Figure 43: content panel view on the left hand side of the viewer, triggered to be open by clicking on content on the right hand side of the header line

Once a dataset is added to the map an entry in the content legend is created. Depending on the type of the dataset it will appear in different so-called groups. Groups are:

- Vector overlay - for all datasets of type geojson including model results
- WMTS data - for all datasets of type **Web Map Tile Service**
- WMS data - for all datasets of type **Web Map Service**
- TMS data - for all datasets of type **Tiled Map Service**
- Pointcloud - for all datasets of type Cesium Pointcloud

- 3D data - for all datasets of type Cesium 3D Tiles
- Terrain data - for all datasets of type Cesium Terrain

Future versions of the viewing client will of course support more common data types and formats, such as vector tiles. However in the current stage of the underlying API, these are the only supported data types.

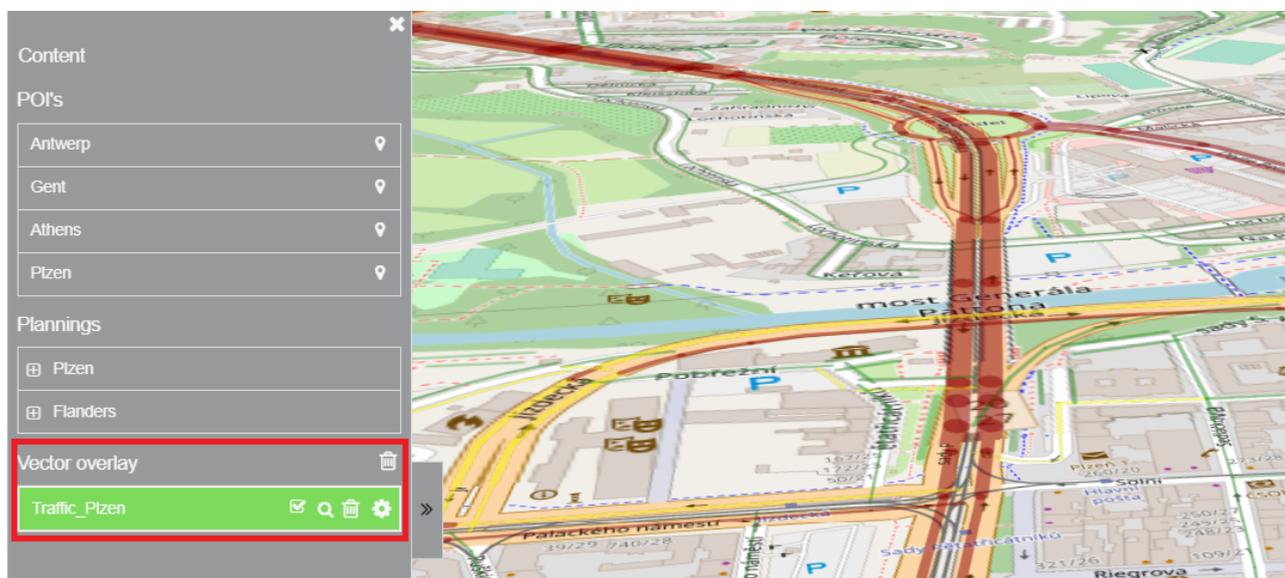


Figure 43: content panel view on the left hand side of the viewer with loaded dataset for Plzen traffic and the default styled data in the viewing client on the right hand side

To view the dataset in the map client, the user has to click on the legend entry to enable the **“layer”** in the viewer. This will enforce the client to load the content of the dataset in the view and visualize it with a default style if needed. An enabled layer in the viewing client is highlighted in green and marked by a check on the right hand side of the layer (layer functionality). Layer functionalities are:

1. enable / disable layer in the view
2. zoom to Layer (not fully developed yet)
3. remove layer from view and legend
4. layer settings for editing a layer entry and styling (see following chapter)

3.5.3 Edit legend entry and create a user style

A user has the possibility to edit the automatically created legend entry and to define its own style for vector overlays and 3D data. By clicking on the settings symbol a new panel right to the content panel opens and provides several options to edit the layer. These options will be further extended by time, for sure. Right now a user has the possibility to edit the legend entry name, applying an already defined style or creating a new style based on the attributes of the dataset. The figure below shows the different options and steps to do.

1. clicking on setting symbol opens the settings panel to the right of the content panel
2. typing in a new title in 2 and clicking on the button **“Set Title”** will change the title of the layer in the content panel, but the original name will be still kept.
3. choosing a style from style dropdown and clicking on **“Apply style”** will assign the chosen style to the layer.
4. here a user can define his own style for a layer based on attributes of the data. Currently only lines and polygons can be styled, but support for further geometries (points) and features (Text, Label)

will be developed in upcoming versions of the client. Right now a user can define the color of a feature by styling them all with the same color (checking the checkbox “Style all features with the same color?”) or based on an attribute chosen from the list of attributes. The same is valid for stroke definition. The stroke of a line or polygon can be defined as to be equal for all features by checking the checkbox or based on attribute by choosing an attribute from a dropdown list of attributes.

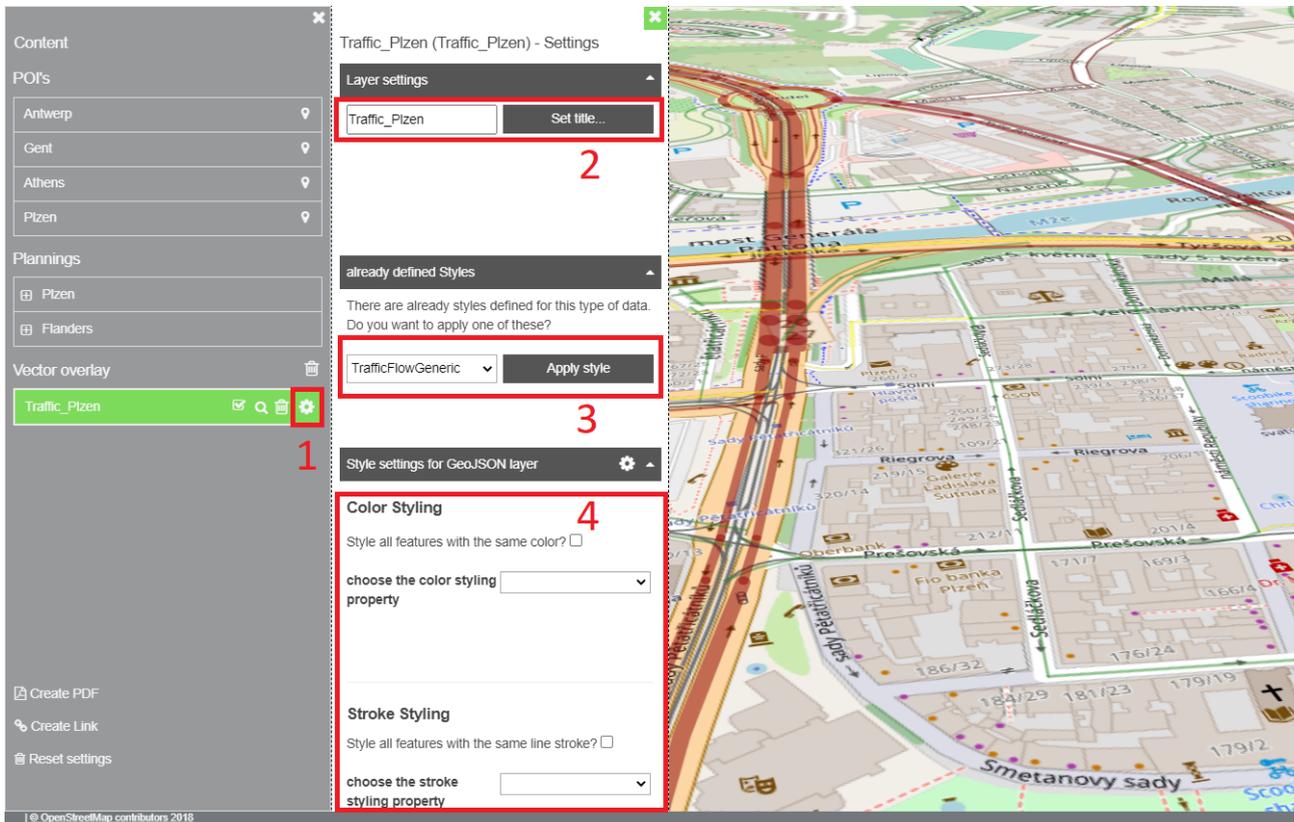


Figure 44: editing a layer in viewing client

3.5.4 Inspecting a feature in viewing client

Once a dataset is loaded (see Section 3.5.2) a user is able to inspect the shown features by clicking on it. The current development supports the inspection of one feature only. A comparison of two or more clicked features using the inspection method is not supported right now.

If a user clicks on a feature inside the viewing client a new panel on the left hand side of the viewer appears, showing the feature properties. Supported datasets for inspection are in general all kind of vector datasets except of Point Clouds, namely:

- Vector overlays
- 3D Data

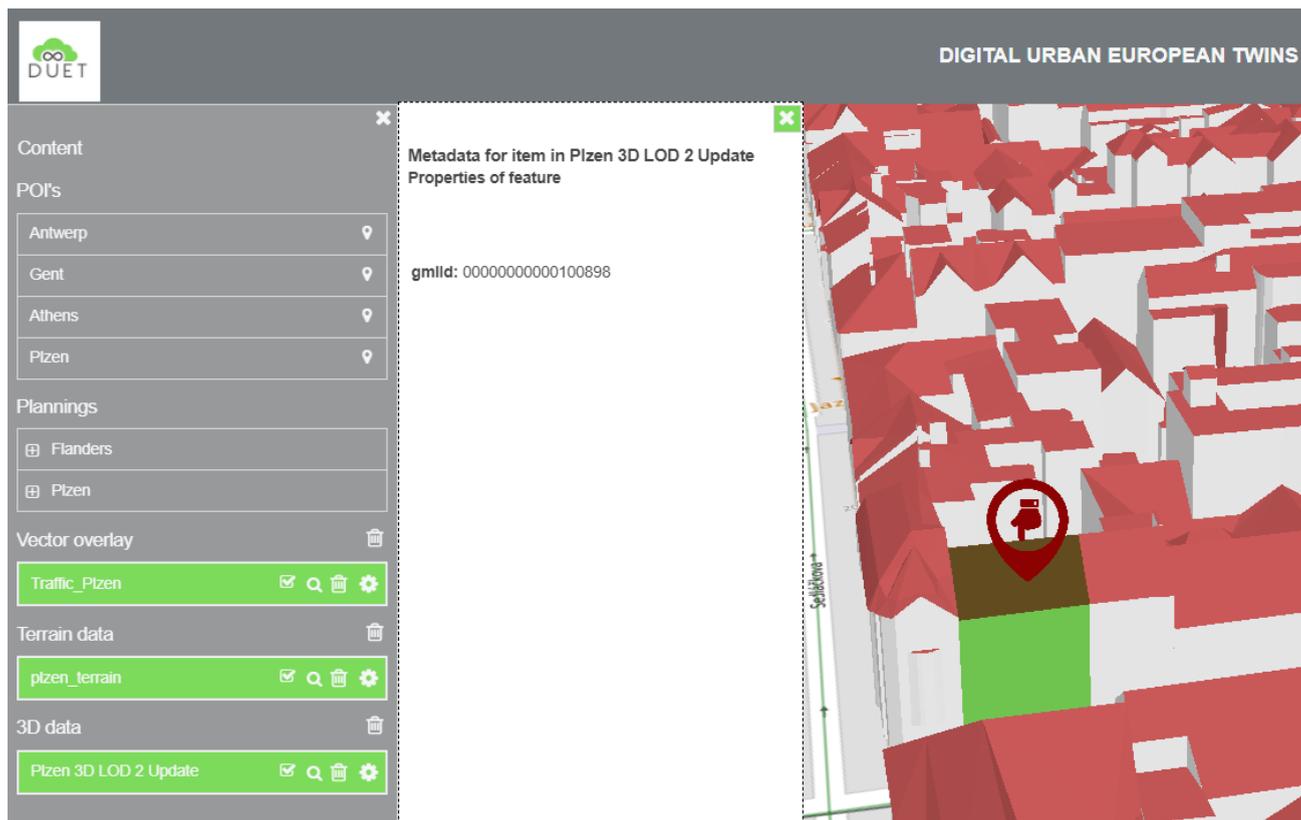


Figure 45: Feature inspection

As shown above, a click on a feature triggers several functions in the viewer.

1. A red marker is shown at the clicked position (in 3D).
2. the clicked object will be highlighted in green
3. an inspection panel opens on the left hand side of the viewing client providing information about the clicked object.
 - a. Information about the layer (dataset) the feature is related to
 - b. depending on the context (properties) of the feature a graph highlighting a specific properties of the feature (speed, capacity vs. flow, etc) is provided (see figure below
 - c. the properties of the feature

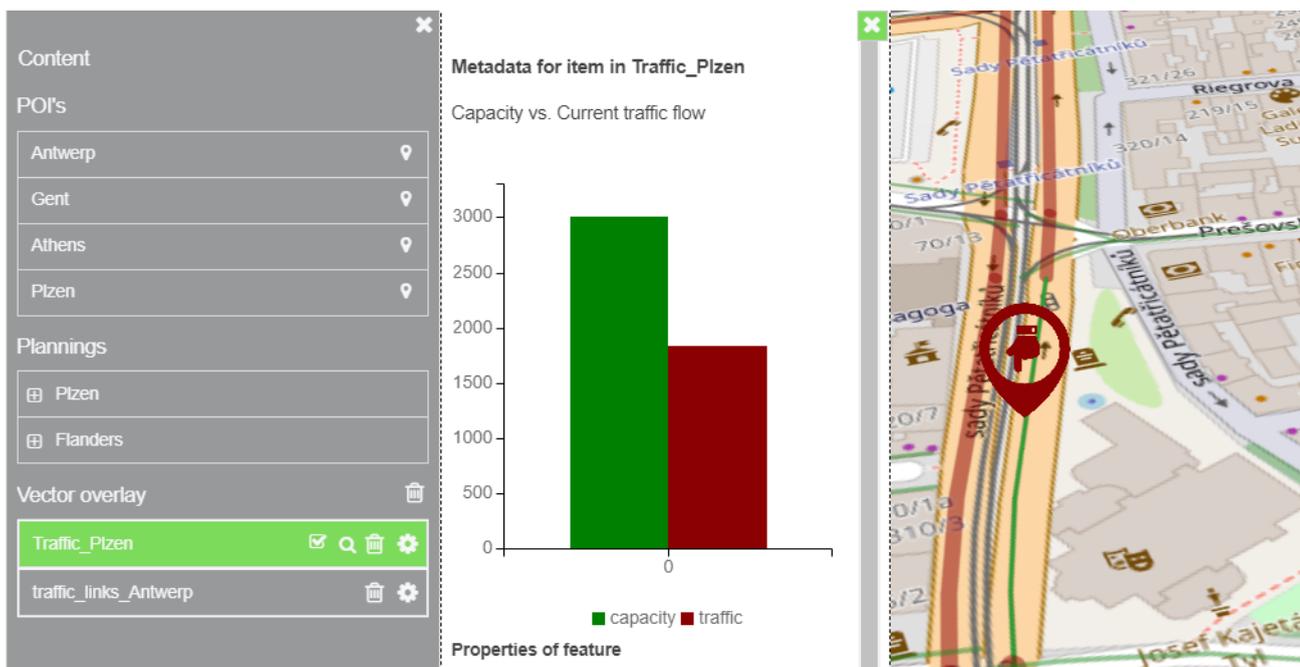


Figure 46: Feature inspection with additional graph for Capacity vs. Flow comparison

3.5.5 Blocking a road segment / changing traffic model

Once a traffic dataset is loaded (see section 3.5.2) a user is able to inspect the shown features by clicking on them and to change the properties of it to enforce a re-calculation of the traffic model.

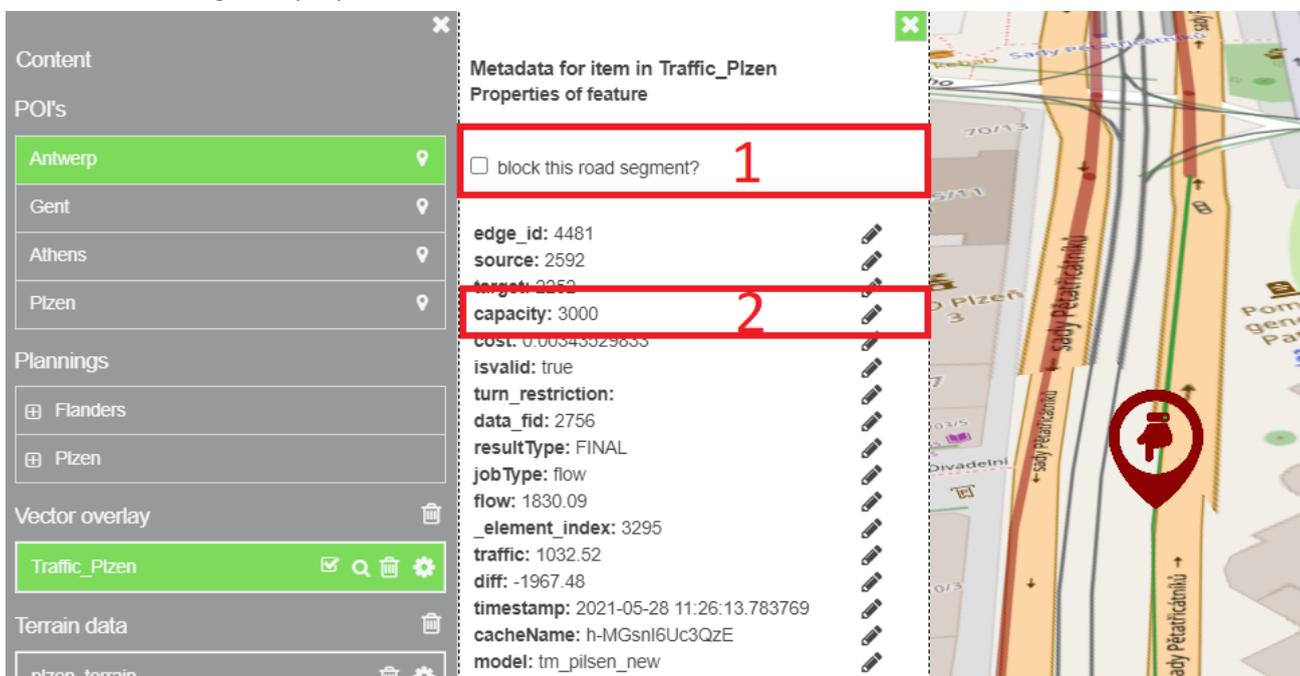


Figure 47: Feature inspection of traffic model and changing the properties

As it is shown in the figure above, inspecting a traffic dataset gives a user now the possibility to change properties of the feature. Two options are provided to do this:

1. Quick blocking a road segment by enabling the checkbox to block this road segment. This option **will set the capacity of this road segment to zero.**

2. changing one of the displayed properties to a user defined value. For example reducing the capacity of this road segment to a value different from zero, by
 - a. either double clicking on the shown value and typing in a new value, or
 - b. by clicking on the pen symbol on the right hand side and then typing in a new value.

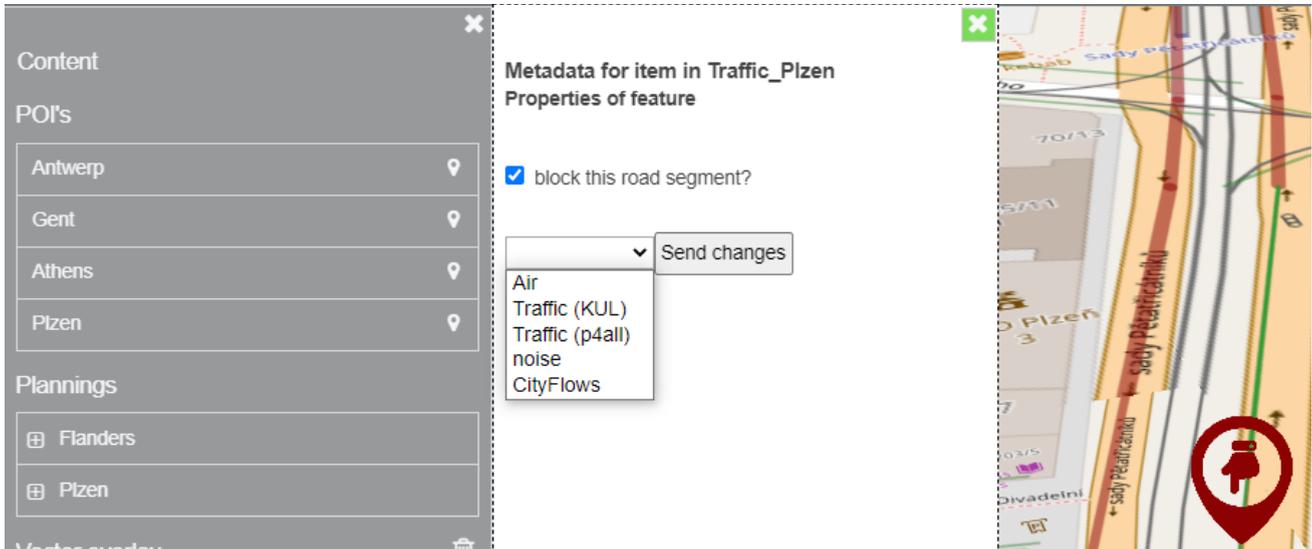


Figure 48: blocking a road segment by quick option

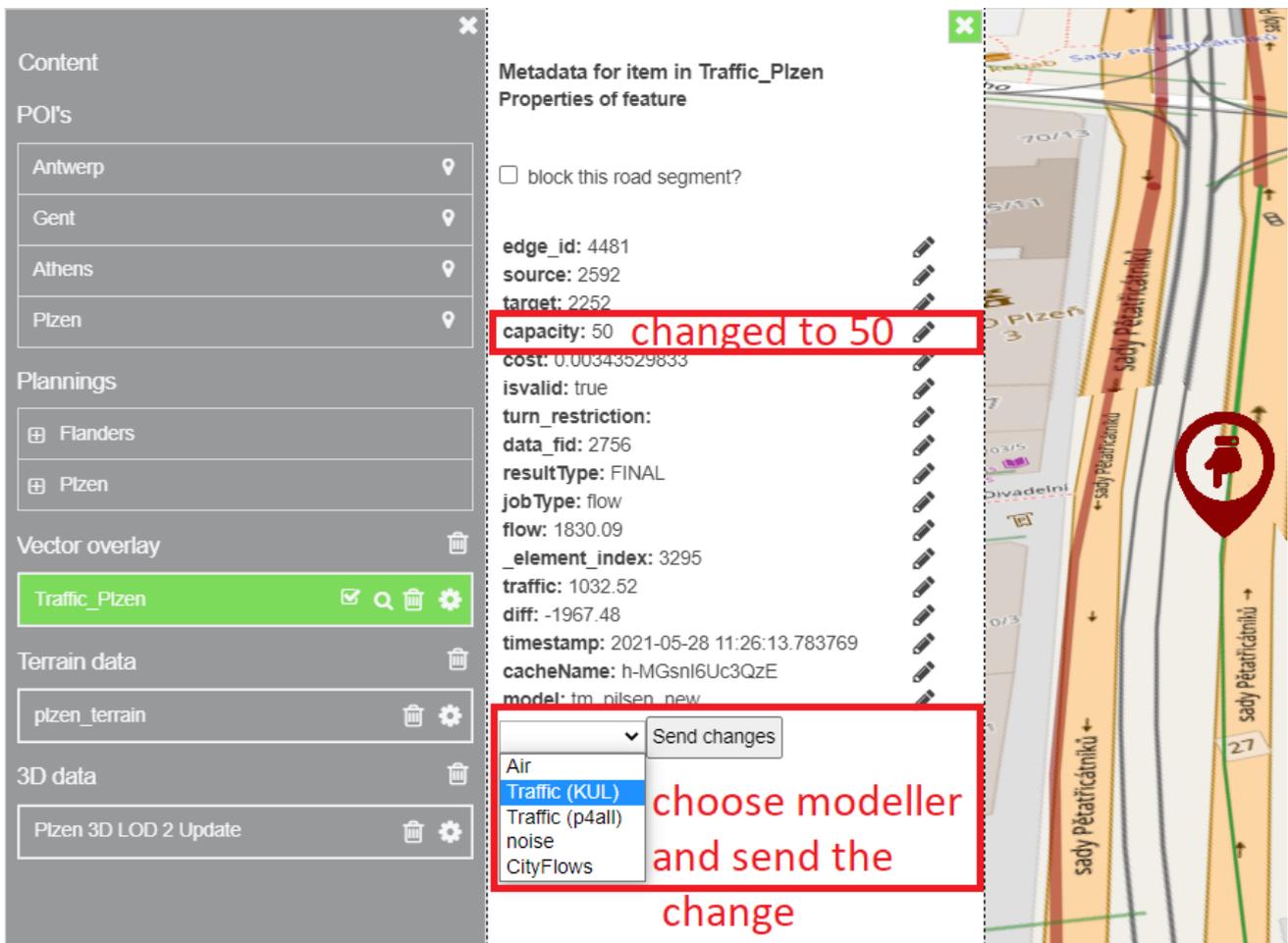


Figure 49: changing the properties of a road segment by editing

In each case, the change has to be sent to the respective modeller. Thus the user has to choose the modeller (Traffic KUL for Flanders region, Traffic p4all for Plzen region) and afterwards send the changes to the modeller.

A sent change will trigger the modellers to perform a new run of the traffic model and to provide the results back.



Figure 50: Notification - successful sent of the change to DUET message system

A notification (see above) will inform the user about the successful sent of the change to DUET message system. The modeller will poll for new incoming messages and will take the change sent by the user and will perform a new model run. Afterwards the results will be made available with the client specific scenario and a message will be sent back to the DUET message system once the resulting data is available for the client. The user's client will again show a notification (see below) when the data is available for integration.

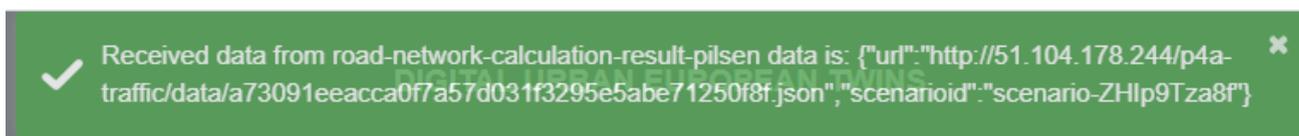


Figure 51: url with the new traffic data

This message shows the url, where the new traffic data can be found and the user's client specific scenario id.

Depending on the settings the user has specified under data => settings (see figure below), the incoming results will be automatically added to the content of the map (automatically add option **enabled**) or added to the available datasets list (automatically add option **disabled**).

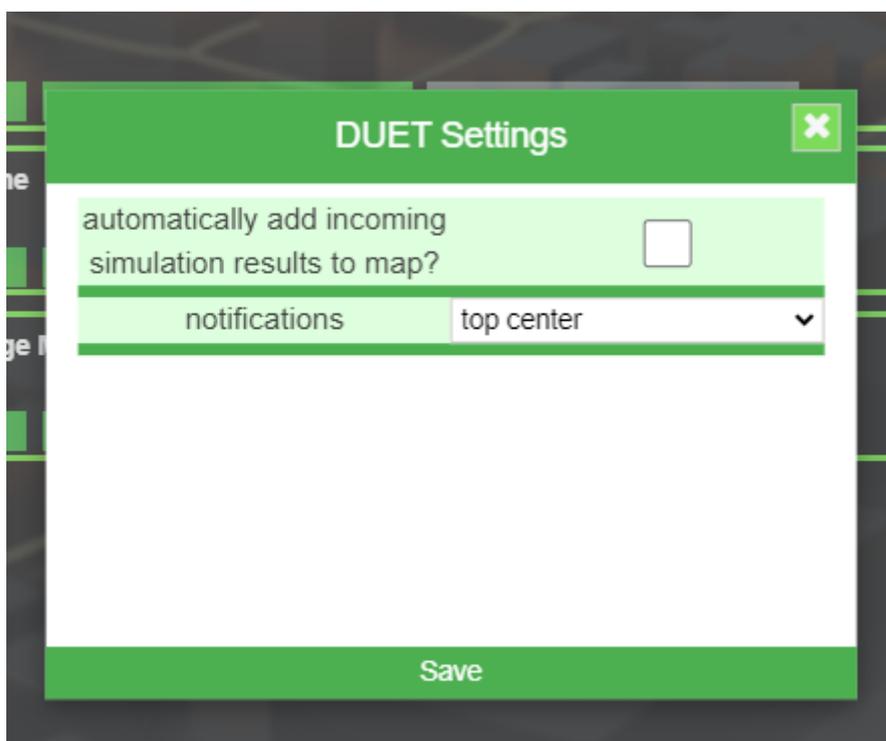


Figure 52: DUET visualization client > data settings

Depending on the defined settings the new datasets will appear either in the content panel or in the data panel.

Option - automatically add incoming data enabled

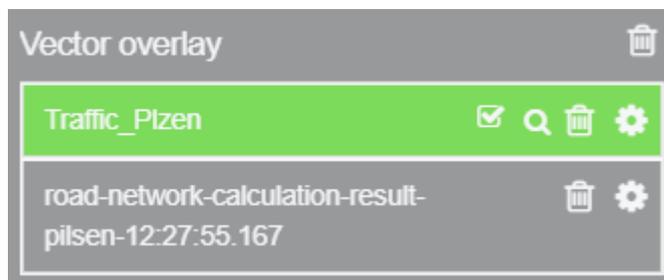


Figure 54: Vector overlay

If this option is enabled in the settings, incoming resulting datasets will be automatically loaded and added to the content panel under “Vector overlay”. Each added dataset is named with the topic / model it comes from together with the incoming time. Thus a traffic result can be identified by “**road-network-calculation-result-pilsen-12:27:5.167**” indicating that it is a result for the road network in the Plzen region provided at 12 o’clock 27 minutes and 55.167 seconds the issuing day. As shown above the original layer is enabled, whereas the resulting layer is disabled after automatic adding to the content panel. For enabling it, the user has to click on the respective entry and the dataset will be loaded, styled and visualized in the main view.

Option - automatically add incoming data disabled

If this option is disabled in the settings, incoming resulting datasets will be added to the catalog view in the respective tab, in this case “Traffic results”.

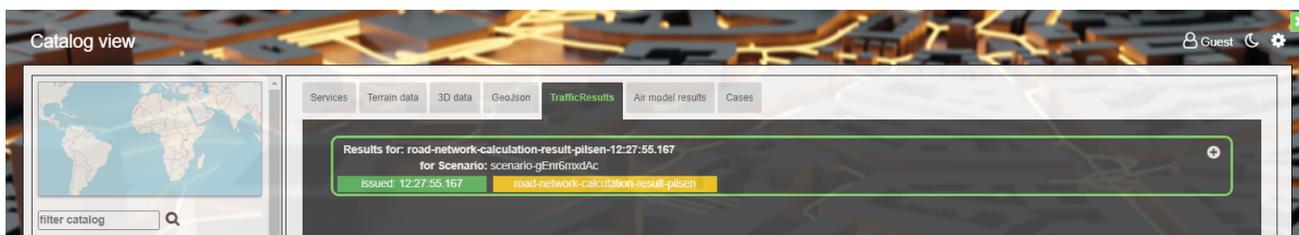


Figure 53: Traffic Results

Each Theme has its own tab, thus results can be found under:

1. Traffic result tab for results of any traffic modeller
2. Noise results tab for results of any noise modeller
3. Air results tab for results of any air pollution modeller
4. Cityflows results for results of Cityflows modeller

Theme specific tabs are only visible if a dataset is available. The datasets per Theme are ordered by issuing time. Thus the upper most dataset is the newest one.

Thus, the resulting dataset for the made traffic change can be found in tab Traffic results. Each added dataset is named with the topic / model it comes from together with the incoming time. Thus a traffic result can be identified by **“road-network-calculation-result-pilsen-12:27:5.167”** indicating that it is a result for the road network in the Plzen region provided at 12 o'clock 27 minutes and 55.167 seconds the issuing day. The dataset can be added to the map content by clicking on the plus symbol on the right hand side of each dataset.

By doing so, the dataset will be added to the content panel and will appear under **“Vector overlay”**

As shown left the original layer is enabled, whereas the resulting layer is disabled after automatic adding to the content panel.

For enabling it, the user has to click on the respective entry and the dataset will be loaded, styled and visualized in the main view.

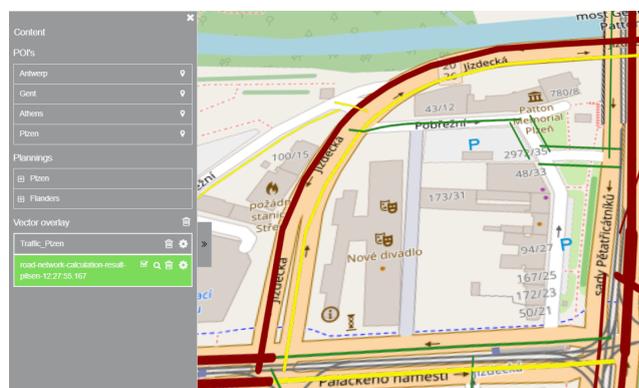
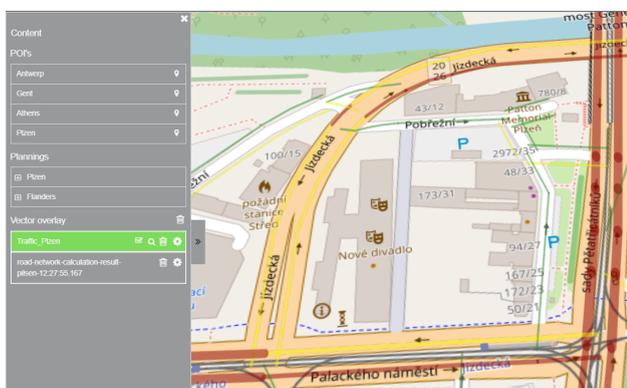
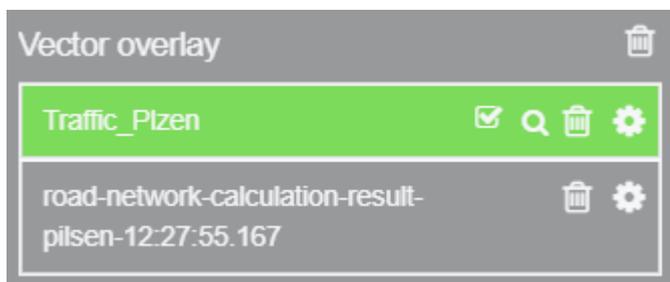


Figure 54: Original traffic data before change (left) - resulting traffic data (right) after change

The models are set up in a way that a traffic model change (at least for Plzen region) will trigger a change in Noise and Air pollution as well. Thus besides a new traffic result also an air pollution model result as well as a noise pollution result can be expected by the user. Both results will be automatically added to the map or listed in the catalog view of the visualization component.

4. Conclusion

This document is the report describing the prototype deliverable D5.2 about the closed beta version of the DUET platform. This is the closed beta version of the platform that offers an initial set of the expected functionalities, and will act as the test-bed for the DUET stakeholders to experience with this initial set of DUET provisions and assess the concepts and knowledge conveyed by the project.

The implementation and integration is not limited to the current level. Most DUET modules, as well as the platform itself are designed to be flexible and easily customisable. The overall architecture of the system facilitates modifications and makes the platform improvement to be easy, at any point. This will be the aim of the following period: to continue the enhancement of this platform capabilities, make it more stable, more sophisticated and more efficient for the target end users, by plugging additional functionalities and calibrating existing ones.

This version of the prototype will be enriched with more functionalities to support the three pilots by M24 (November, 2021), so that the planned first evaluation iteration will be performed in the next period (M19 – M23). The results of this evaluation cycle, along with the already scheduled work in this and the other technical WPs, will lead to another prototype release of the DUET integrated platform by M30 (May 2022).

5. References

- [1] DUET D5.1 System Architecture and Implementation Plan
- [2] DUET D4.4 visualization Components v1