nuMIDAS

Deliverable 5.3
Dashboard

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101007153.

Project acronym	nuMIDAS
Project title	New Mobility Data and Solutions Toolkit
Project number	Horizon 2020 MG-4-8 – GA No 101007153
Work package	WP5 – Consolidated toolkit for stakeholders
Editor(s) / main author(s)	Steven Boerma (MATM) Luc van der Lecq (MAPTM) Dimitris Tzanis (CERTH)
Reviewer(s)	Sven Maerivoet (TML)
Dissemination level	Confidential
Contractual delivery date	31/10/2022 (M22)
Actual delivery date	31/12/2022 (M24)
Version	v1.0

Legal disclaimer

Document revision history			
Version	**Date**	**Description**	**Editor(s) (Affiliation Short Name)**
v0.1	12/04/2022	Initial version	Steven Boerma (MAPTM)
v0.2	18/05/2022	Adding architecture Three tier model First iteration	Steven Boerma (MAPTM) Luc van der Lecq (MAPTM)
v0.3	03/08/2022	Extending Chapters 5, 6, and 7	Luc van der Lecq (MAPTM)
v0.4	26/09/2022	Adding second iteration activities	Steven Boerma (MAPTM) Luc van der Lecq (MAPTM)
v0.5	05/12/2022	Extending all chapters	Steven Boerma (MAPTM) Luc van der Lecq (MAPTM)
v0.5	22/12/2022	Finishing all technical descriptions	Steven Boerma (MAPTM) Luc van der Lecq (MAPTM) Dimitris Tzanis (CERTH)
v1.0	31/12/2022	Final version	Sven Maerivoet (TML)

Table of contents

List of figures

List of tables

1 Executive summary

Over the past few years, the transport sector is dealing with major challenges attributed to megatrends, such as climate change, shared mobility, and user-eccentricity. Taking into consideration requirements stemming from sustainability and quality of life principles, the need for developing new methods and tools supporting the planning, management, and monitoring of mobility solutions ensuring a well-structured and well-operating mobility system seems more than needed. The scope of this document is to describe the outcomes of creating a tangible toolkit for policymakers, planners, and researchers for new mobility solutions; the nuMIDAS framework and toolkit. On one hand, the toolkit is the implementation of the use cases developed and prototyped in WP3 and described in Deliverables 3.2 and 3.3, and on the other hand, the toolkit is a framework created in such a way that adaptation of use cases and adopting new use cases can be done rather easily due to a generic architecture of the toolkit.

Within this deliverable, an overview of the architecture, the implementation of use cases in two iterations and a description of the set-up of the toolkit based on the three-tiered model are provided.

The architecture is cloud hosted by Amazon Web Services, based on the use of standard toolings like PostgreSQL, S3 data storage, and programming in Python. The toolkit technical design is split up into a three-tiered model, e.g. a data layer, application layer and presentation layer, see Figure 1.

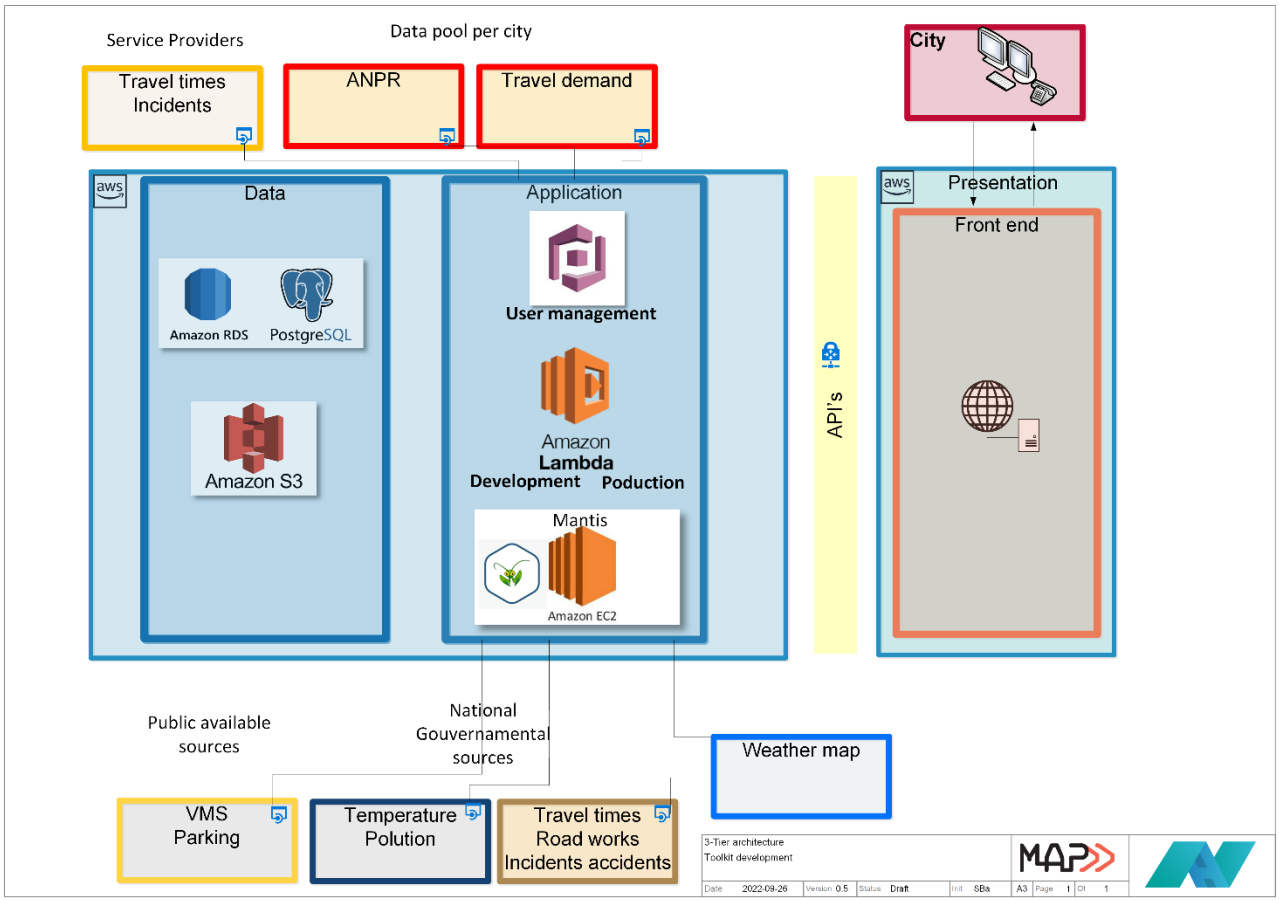


Figure 1: Technical architecture of the nuMIDAS toolkit.

A generic approach is developed for the toolkit using standard naming conventions for a use case, data tables etc. Also, a standard visualisation setup is adopted together with a JSON configuration file to describe the input and output configuration of a use case.

The further development is done in two iterations. The first iteration consists of the setup of the user and city management environment, dividing users into roles with different rights. Next, the data, application and presentation tiers were set up with a development and production environment which enables local testing and cloud deployments. Use Case 1 was developed, first in a hard-coded way to speed up the collaboration and learning process of all the partners. To enable close cooperation of all involved work package meetings, stand-ups and scrum sessions were regularly planned. Code sharing is done thru GitHub and tracking bugs and requirements are organised in Mantis. During the second iteration, with even more extensive cooperation from the partners, the other 5 use case was developed and the generic use case configuration was implemented.

Within a year the toolkit was designed, set up and populated with use cases. During the development of the toolkit and the implementation of the six use cases, the building process speeded up when more code cam available. This gives the nuMIDAS toolkit and its partners a solid starting point for further exploitation. The aim of creating a toolkit with TRL levels 6-7 has thus become reality.

2 Introduction

2.1 About nuMIDAS

The mobility ecosystem is rapidly evolving, whereby we see the rise of new stakeholders and services. Examples of these are the presence of connected and automated vehicles, a large group of organisations that rally to establish various forms of shared mobility, with the pinnacle being all of these incorporated into a large MaaS ecosystem. As these new forms of mobility offerings start to appear within cities, so do new ways in which data are being generated, collected, and stored. Analysing this (Big) data with suitable (artificial intelligence) techniques becomes more paramount, as it leads to insights in the performance of certain mobility solutions, and is able to highlight (mobility) needs of citizens in a broader context, in addition to a rise in new risks and various socio-economic impacts.

Successfully integrating all these disruptive technologies and solutions with the designs of policymakers remains a challenge at current. let alone being able to analyse, monitor, and assess mobility solutions and their potential socio-economic impacts.

nuMIDAS, the New Mobility Data & Solutions Toolkit, bridges this (knowledge) gap, by providing insights into what methodological tools, databases, and models are required, and how existing ones need to be adapted or augmented with new data. To this end, it starts from insights obtained through (market) research and stakeholders, as well as quantitative modelling. A wider applicability of the project's results across the whole EU is guaranteed as all the research is validated within a selection of case studies in pilot cities, with varying characteristics, thereby giving more credibility to these results. Finally, through an iterative approach, nuMIDAS creates a tangible and readily available toolkit that can be deployed elsewhere, including a set of transferability guidelines, thus thereby contributing to the further adoption and exploitation of the project's results.

nuMIDAS, the New Mobility Data and Solutions Toolkit, started at the beginning of 2021 under the Horizon 2020 programme and its is being developed by a European Consortium, composed of 9 partners from 6 countries: Belgium, Czech Republic, Greece, Italy, The Netherlands, and Spain.

2.2 Pilot cities and use cases

The project builds on a distributed selection of case studies in pilot cities to provide a geographic coverage of the EU. The four pilot cities are Barcelona (Spain), Milano (Italy), Leuven (Belgium), and Thessaloniki (Greece).

Table 1: List of nuMIDAS case studies.

	Problem	Solution	City
UC1	Fleet size optimisation of shared mobility operators [PPMS]	Optimisation model	Milan
UC2	Spatial optimisation of shared mobility (operators) [OASM]	Spatial allocation model	Milan
UC3	Traffic insights related to emissions [AQVE]	Short- and mid-term emissions and traffic intensity forecasting	Barcelona
UC4	Impact analysis of parking space reduction policy [PLFP]	Planning support system (GIS, simulation model, visualisation)	Leuven
UC5	Transport planning improvement (P&R and public transport) [IOMA]	OD matrix using ANPR	Barcelona
UC6	Traffic management scenarios assessment [ATMS]	Dashboard with visualisation of traffic management scenarios and related KPIs	Thessaloniki

2.3 Consolidated toolkit for stakeholders

As part of WP5 of nuMIDAS project, a dashboard has been developed that provides researchers, planners, and policymakers with a visualisation of the results derived from methods and tools from the six case studies related to connected and automated vehicles and shared mobility forms that are incorporated into a large MaaS ecosystem. The goal of the dashboard is to make policy decisions measurable and quantifiable, making impact assessment more tangible and directly usable.

The development process of the dashboard is characterised by:

- rapid prototyping to evaluate acceptance by prospective users regarding:
    - preferred visualisation
    - data combinations
    - type of visual element (graphs, tables, maps)
- a micro-services approach to rapidly develop and adapt reliable building blocks

2.4 Purpose of this document

This document describes the architectural design, development, and technical setup of the nuMIDAS toolkit and the incorporation of the methods and tools into algorithms calculation of the outcome of the six use cases based on input data and parameters. The development of the use case is described shortly accompanied by the agile development approach which was followed in a natural approach. Next to that, the flexible setup of the toolkit is explained in what way the three-tiered model is implemented.

For the usage of the tool and explanation of the data requirements, usage procedure, interpretation of the results and the next steps please consult Deliverable 5.4 Transferability guidelines. For user interaction and usability and feasibility of the use case implementations please consult Deliverable 5.1 Usability and feasibility report.

2.5 Structure of this document

Chapter 3 describes the architectural design of the toolkit. Chapter 4 sets out the development of the six use cases within the toolkit in two iteration steps. Chapters 5, 6, and 7 describe the three tiers of the nuMIDAS toolkit. Chapter 8 concludes the development of the nuMIDAS project period.

2.6 Acronyms

AWS	Amazon Web Services
CSS	Cascading style sheets
EC	European Commission
GA	Grant agreement
HTML	Hypertext markup language
ISO	Internation Standards Organisation
JSON	JavaScript object notation
KPI	Key performance indicator
MaaS	Mobility-as-a-service
MFD	Macroscopic fundamental diagram
nuMIDAS	New Mobility Data and Solutions Toolkit
OD	Origin-destination
S3	(Amazon) Simple Storage Service
Three-tiered model	Modular client-server architecture that consists of a presentation, application, and data tier
UC	Use case
WP	Work package

3 Architecture

The First steps were to create a common picture of the architectural approach on the toolkit and the look and feel of the dashboard. Ideas were developed on the architecture and a representation was made to elaborate on the set-up of the toolkit as a whole, see Figure 2.

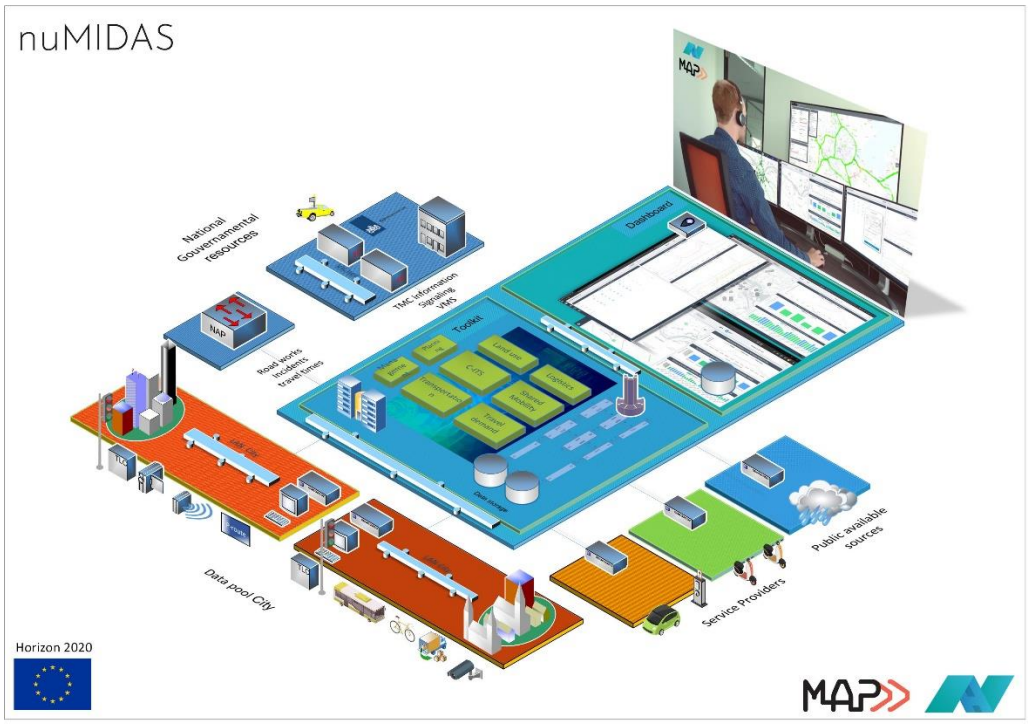


Figure 2: Conceptual architecture of the nuMIDAS toolkit.

Together with the partners' ideas were developed how to deploy the dashboard side and cooperate with the so-called backend where the Python code of Use Case calculation is made available. As MAPtm uses Amazon Web Services (AWS) to host their dashboards this approach was adopted by the nuMIDAS partners. A first draft of the technical architecture was created to get all the partners up to speed on the layout and further development of the nuMIDAS toolkit. The technical architecture is shown in Figure 3

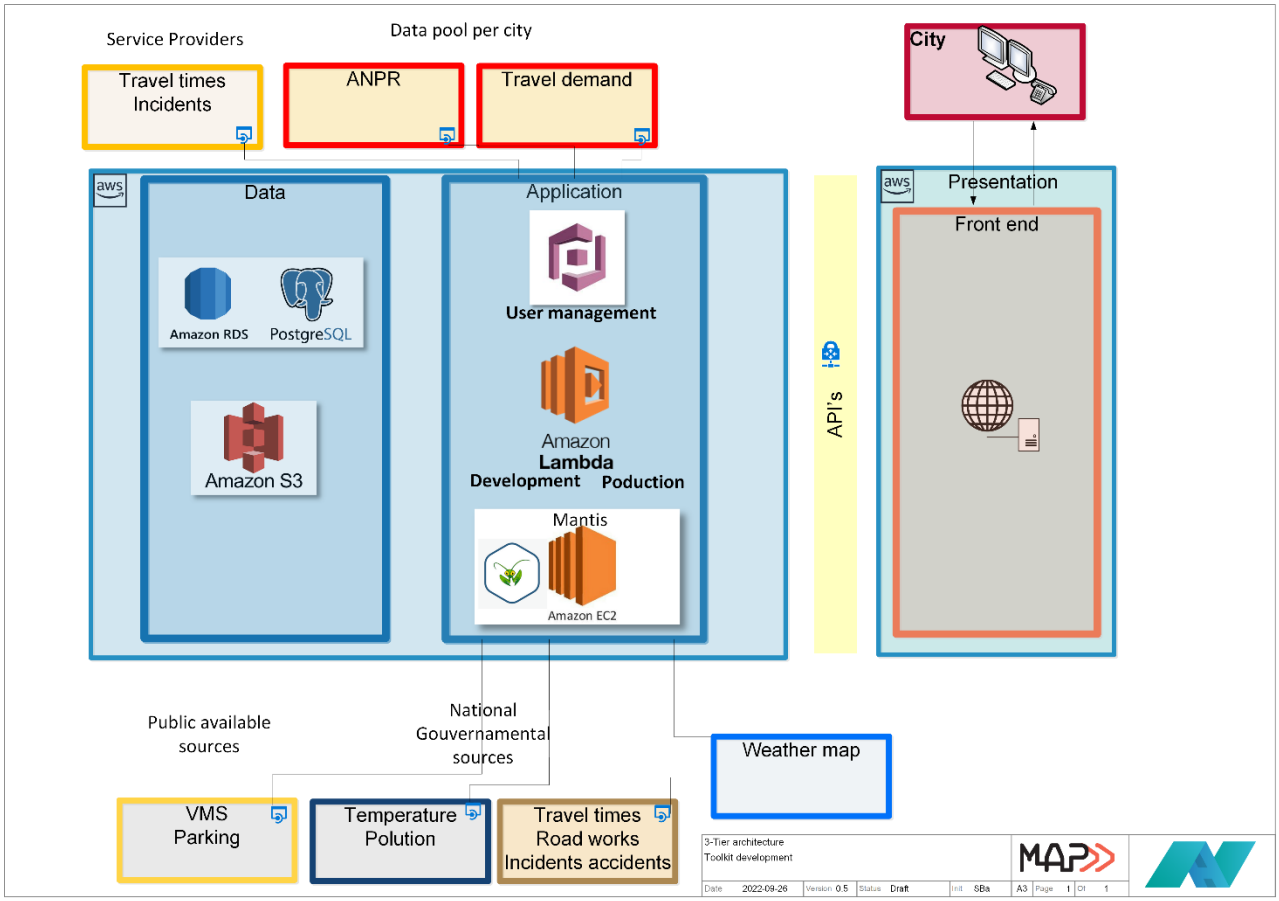


Figure 3: Technical architecture of the nuMIDAS toolkit.

The architectural approach consists of the 3-Tier setup with a presentation tier, an application tier, and a data tier.

Presentation tier

The presentation tier is the interface and communication layer with the user. The main purpose is to display and obtain information. The presentation layer is developed in HTML, CSS en JavaScript.

Application tier

The application or logic tier is the main body of the nuMIDAS toolkit. All the business logic is in this tier like user management, functions for the nuMIDAS use case and also the bug tracking tool Mantis. Function/applications in the application tier can be triggered by the presentation tier or run autonomously based on a clock function or monitoring new data input in de data tier. The application tier can add, delete, and modify data in the data tier.

Data tier

In the data tier information processed by the application tier is stored. From the presentation tier also configuration information is read and input is stored in the data tier for further use by the application tier.

For the first setup, AWS is configured to contain a PostgreSQL database to store geographical and relational data. Flat data can be stored in an S3 bucket. If needed one or more virtual machines (EC2) can be configured. Calculation and data handling is a Lambda function that is called from the dashboard side. In the dashboard the user must be enabled to:

- Create a scenario within a use case
- Set parameters for a scenario
- Choose affected objects if needed
- Run the calculation of a scenario
- Edit, rerun, or delete a scenario

4 Development

4.1 First iteration

Within the consortium, the next steps of development have been discussed and the first steps are made to develop requirements and coordinate the work to get the toolkit up and running for the first use case.

First requirements were drawn up for:

- Draft layout
- English – (multi-language in the future)
- Users for all cities, use cases
- admin, manager, specialist
- adding users (behind the screen)
- creating flexible code (over cities and use cases)
- Transferability (T5.3)
- Setup storyboard for creating workflow descriptions

Visualisations

- built up of the frontend
- graphs, geo representation
- comparison of calculations

Based on a created mockup the first version of the dashboard of the toolkit was developed. Alongside a storyboard describing the step-by-step actions, a user can do like a walkthrough of the functionality of the toolkit, see Figure 4.

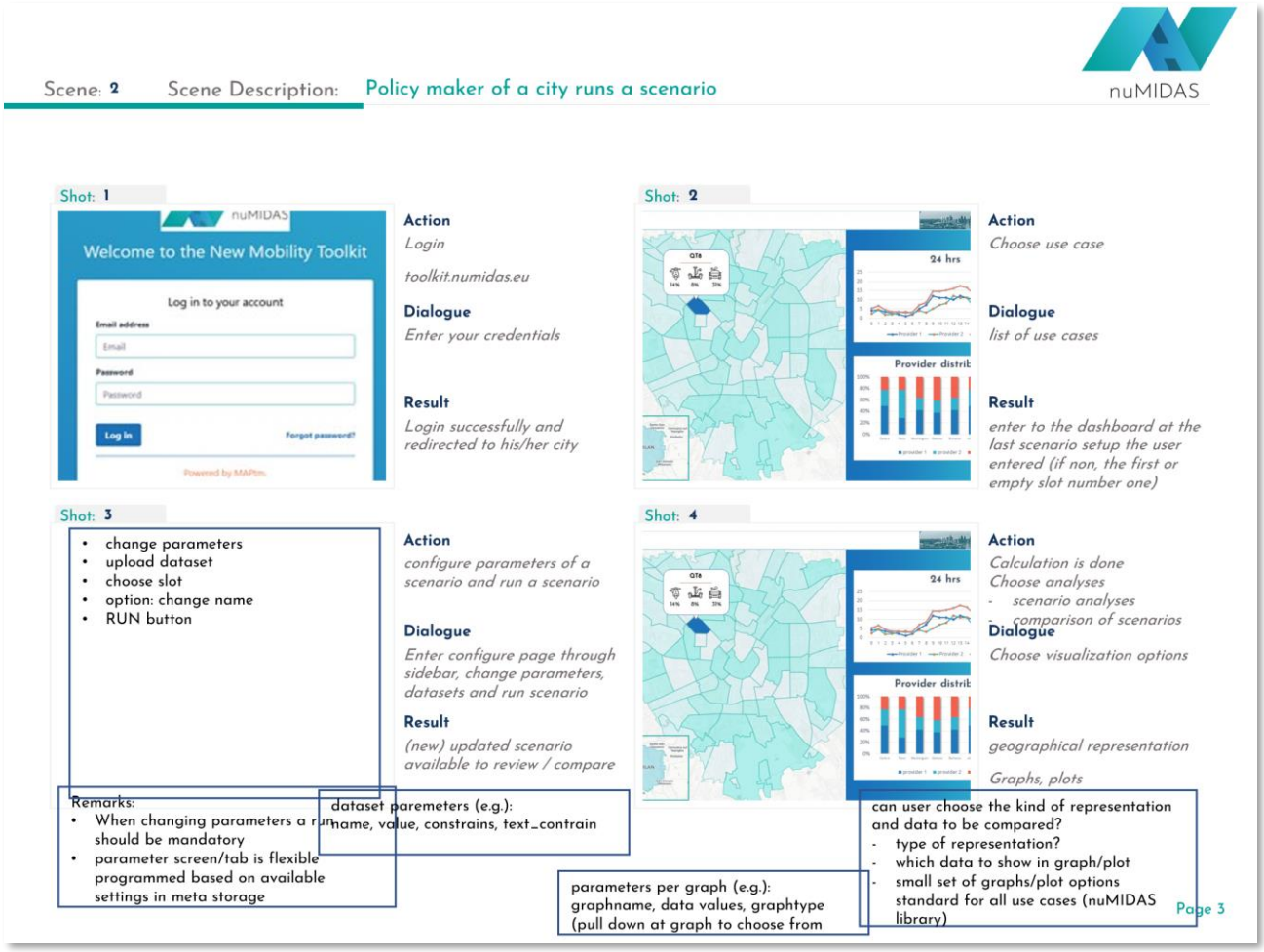


Figure 4: Storyboard general use case actions.

Based on the analysis of the storyboard translation is made into functions:

- Login & user management
- Definition of pages
- Definition of menus/tabs
- Definition of use case functions

The dashboard design is elaborated using a high-level flow diagram which describes the coherence of the dashboard, see Figure 5. Within a use case scenarios are defined to enable users to run different calculations within the use case and save and present the results.

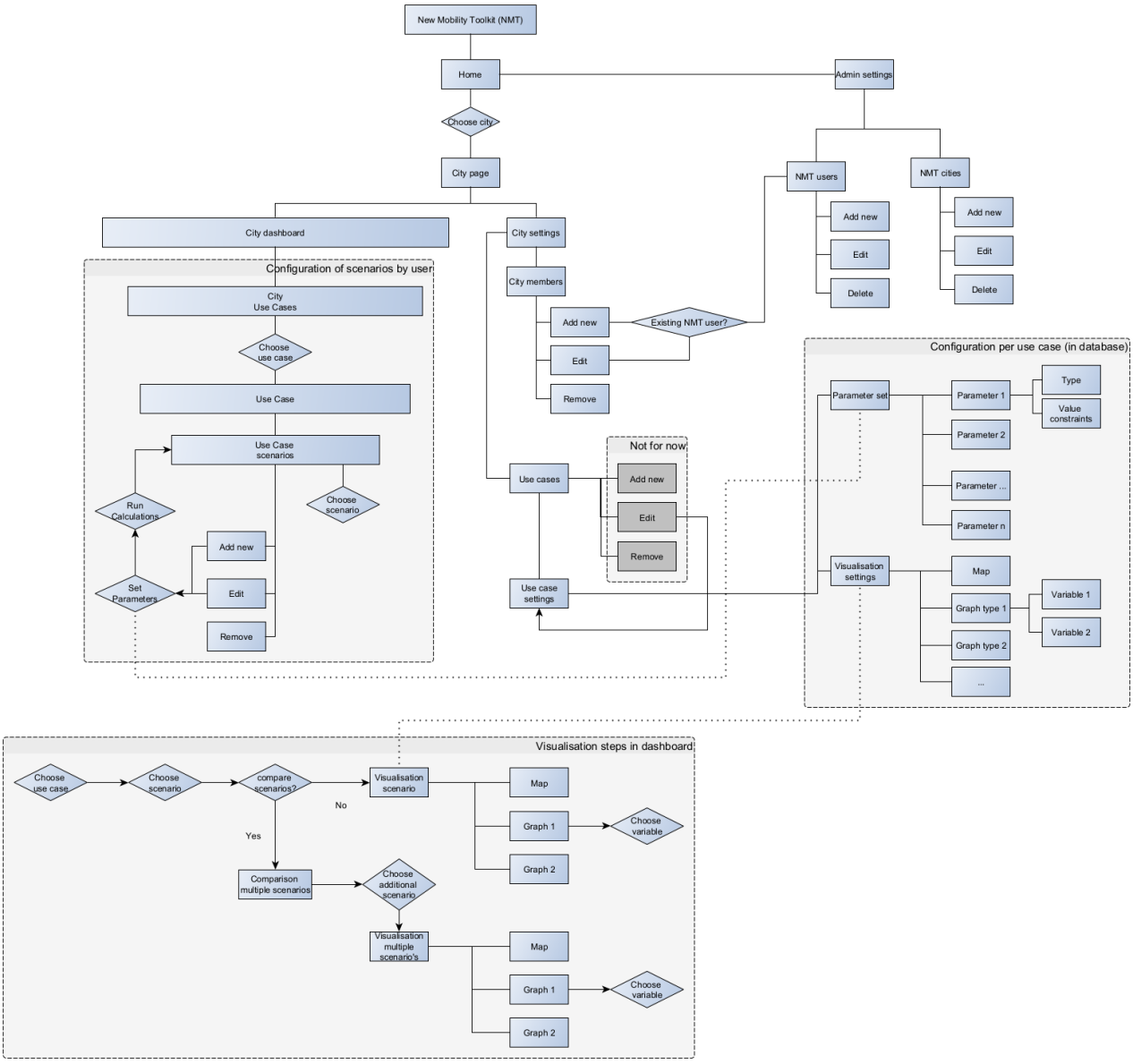


Figure 5: Dashboard flow diagram.

During this first iteration, the dashboard functions are built and extended. In this iteration, the functionality is partially hardcoded in the presentation tier (frontend) to get a first (user)experience based on which the more generic, flexible, and transferable second iteration could be built upon. The end-user experience is collected within Task 5.1. In parallel, the implementation of a generic solution is created. Figure 6 shows the nuMIDAS dashboard.

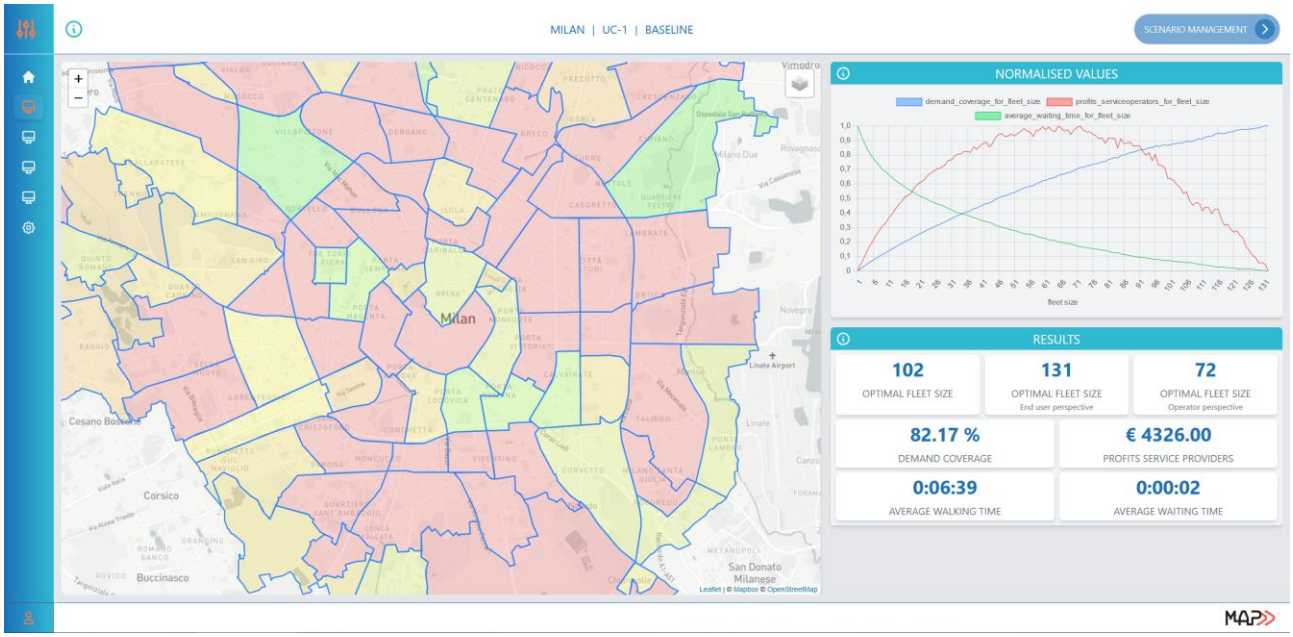


Figure 6: nuMIDAS dashboard.

4.1.1 Installing Amazon Web Services

Configuration

For nuMIDAS a new account is created. This means all the services deployed within the nuMIDAS account are transparently billed and only partners with access rights can add remove or change AWS services for the nuMIDAS toolkit. The user management of the toolkit access, which is a different user pool of AWS organised in Cognito. Cognito enables the 'add user' 'sign-up', 'sign-in', and access control to the web and mobile apps quickly and easily. Cognito is HIPAA eligible and PCI DSS, SOC, ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, and ISO 9001 compliant. Within the Cognito environment, the user access to the toolkit and the access on the city level is arranged, see Figure 8, left side. Within the toolkit, three roles are defined, see Figure 7.

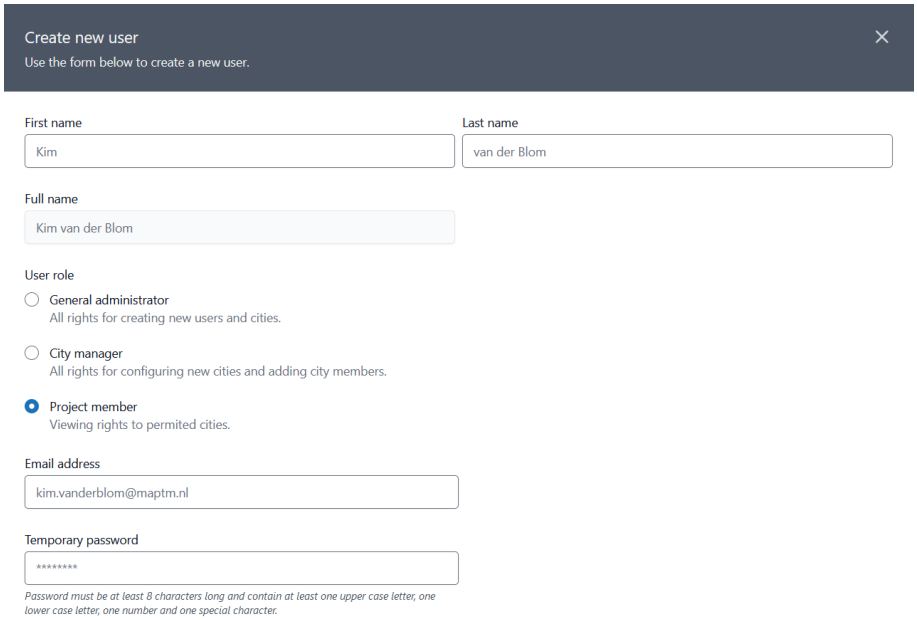


Figure 7: Toolkit user roles.

The general administrator can perform all configuration tasks within the toolkit. A city manager has the same rights but applies only to the city in which the role is appointed. A project member can create, edit delete and run scenarios within the use case he/she has access to.

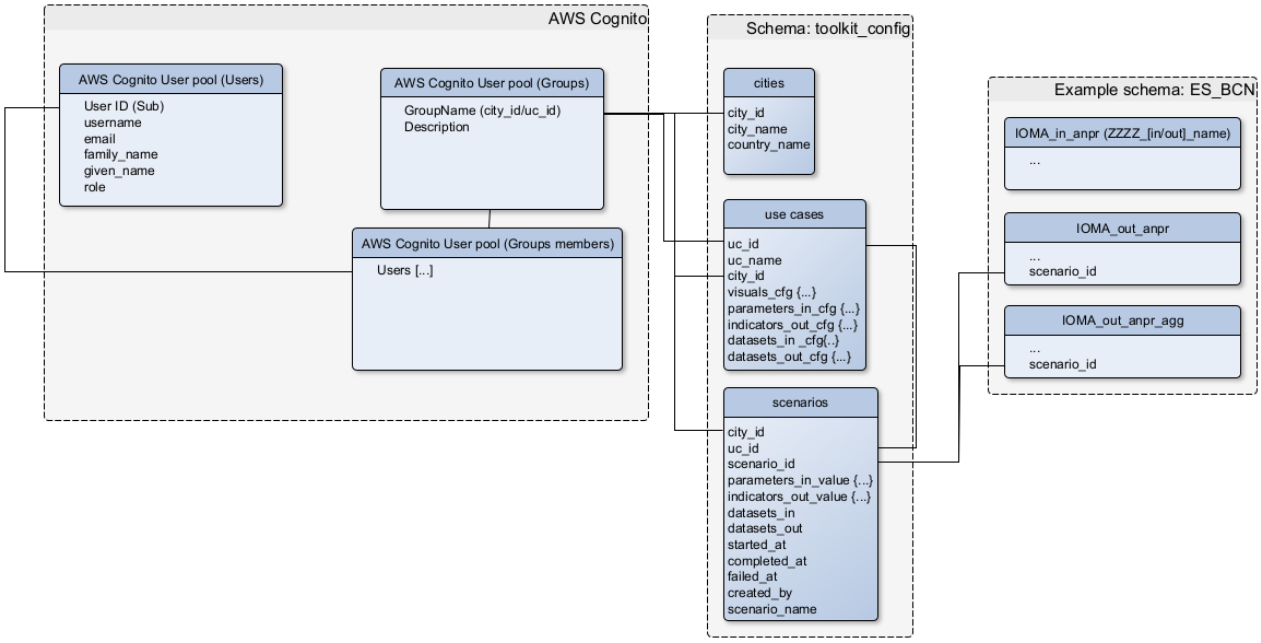


Figure 8: Organisation user pool, configuration, and city use cases.

The created use cases and scenarios are organised in database tables (Postgres) within the AWS environment, see also Figure 8

3 tiers deployment

The user administration and use case configuration coincide with the use case dashboard and backend calculations. When setting up a use case first of all the environment of the use case must be created on the user management level and the input dataset levels. This process is shown in Figure 9.

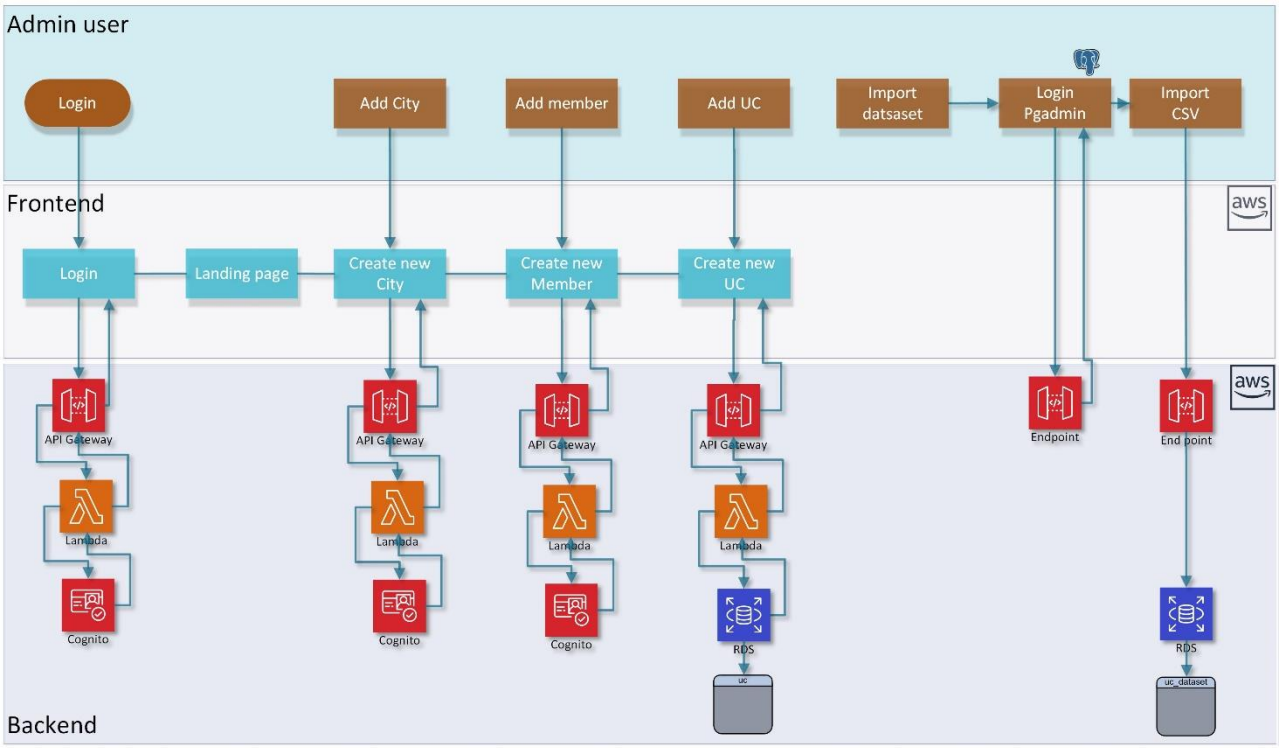


Figure 9: Setup use case from a city managers perspective.

Of course, the backbone of a use case is the application tier where all functions are programmed in a Python script which accesses the AWS environment to obtain input data, and input parameters and calculate output results and calculation datasets. The Python calculation scripts are deployed as a Lambda function that is called from the presentation tier (frontend) to start the function. The general process is shown in Figure 10

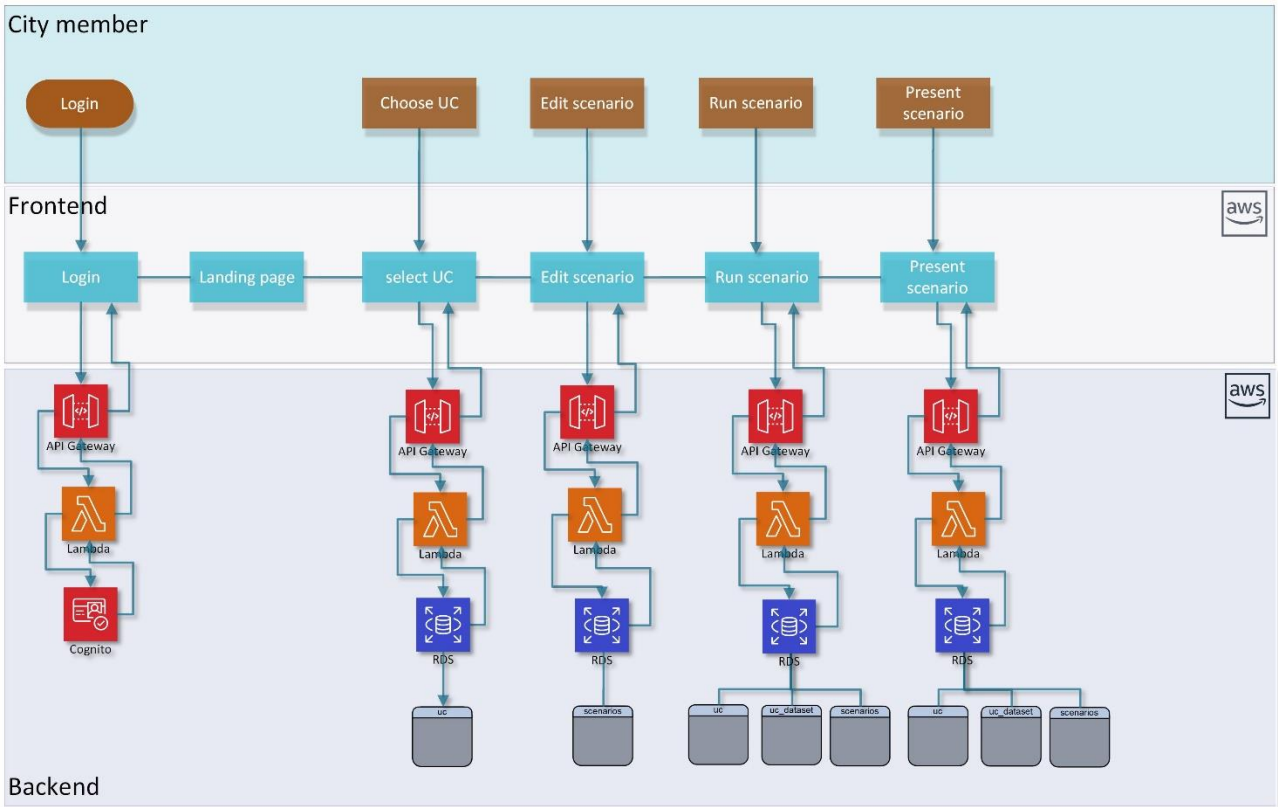


Figure 10: Setup use case calculations from an end-user perspective.

For the first use case, a simple setup of the data storage is developed which is converted in the second iteration to the flexible setup of the data tier as explained in Chapter 7.

4.1.2 Enabling collaboration and agile development thru scrum sessions

From the start of the project, the cooperation within the nuMIDAS consortium was very good due to combining efforts on the work packages and practically weekly online meetings. For the toolkit development, even more, intensive collaboration was needed to enable quick drafting of the first results and enable (end) users to start working with the toolkit. To support this the next actions were executed:

- Setting up a collaboration environment (Github)
- Setting up a tracking and reporting environment (Mantis)
- Support of agile software development of frontend and backend thru dedicated scrum sessions
- Started end of Q5 up until Q24, execution of the usability analysis.

GitHub enables the developers to cooperate in software development and keeps track of revised versions of the code.

The purpose of the Mantis environment is twofold. Bug tracking by the developers and requirements collection of the users, combined with the tracking and reporting of task 3.3.

To enable quick development of the first use case, on-demand, short online meetings were held to get acquainted with the AWS environment and working order and start the deployment of the code developed in WP3 within the toolkit.

Started but of great value for further development the usability analysis coincides with all the work done in WP3, WP4 and WP5. Figure 11 shows the mantis interface accessible through mantis.numidas.eu.

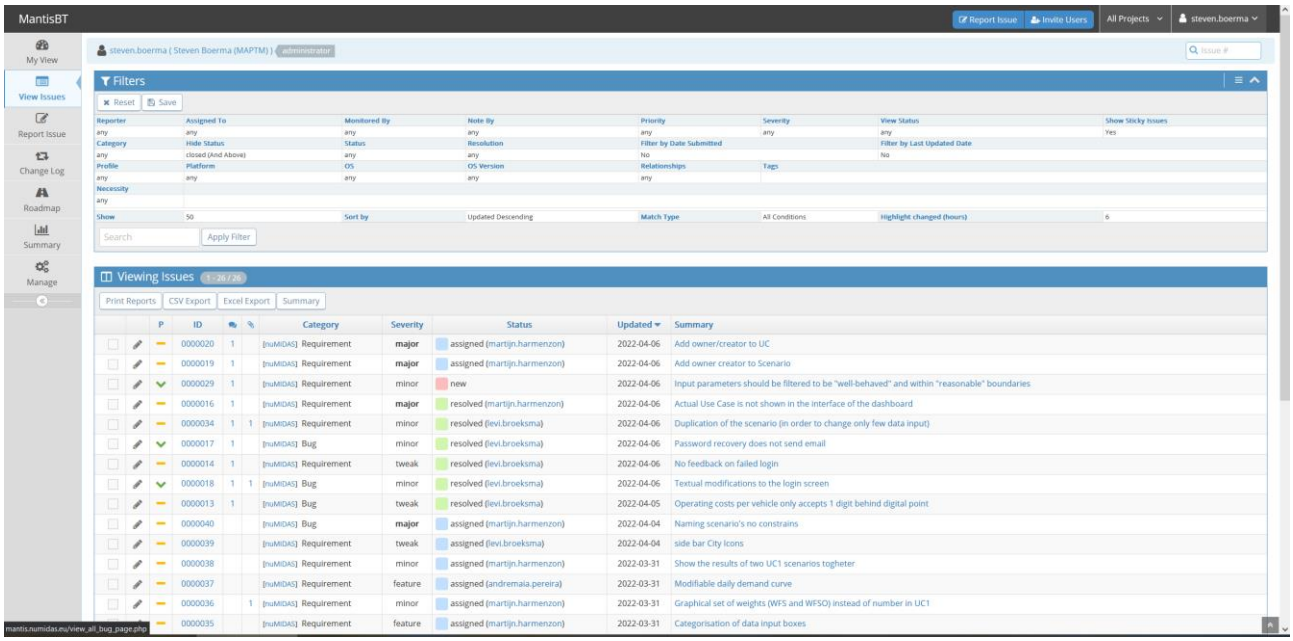


Figure 11: mantis.numidas.eu interface.

4.1.3 Design of a flexible and transferable environment

While developing the first use case in the toolkit parallel, based on the first experiences, a small design team created several drafts of a more flexible and transferable environment for the second iteration. The main goal was the use of clear naming conventions and configuration to enable the developers to understand and recognise all the components of a use case and city within the toolkit environment.

In the naming convention a country code, city codes and use case codes are used, see Section 7.1. Within the Postgres environment and the S3 storage, all datasets are grouped by city.

Started after developing the user management, the next step was extending the functionality of the toolkit. This includes the ability to add and edit cities, users, use cases and scenarios, as well as configuring the use case datasets in an organised and flexible manner using the structure shown in Figure 12.

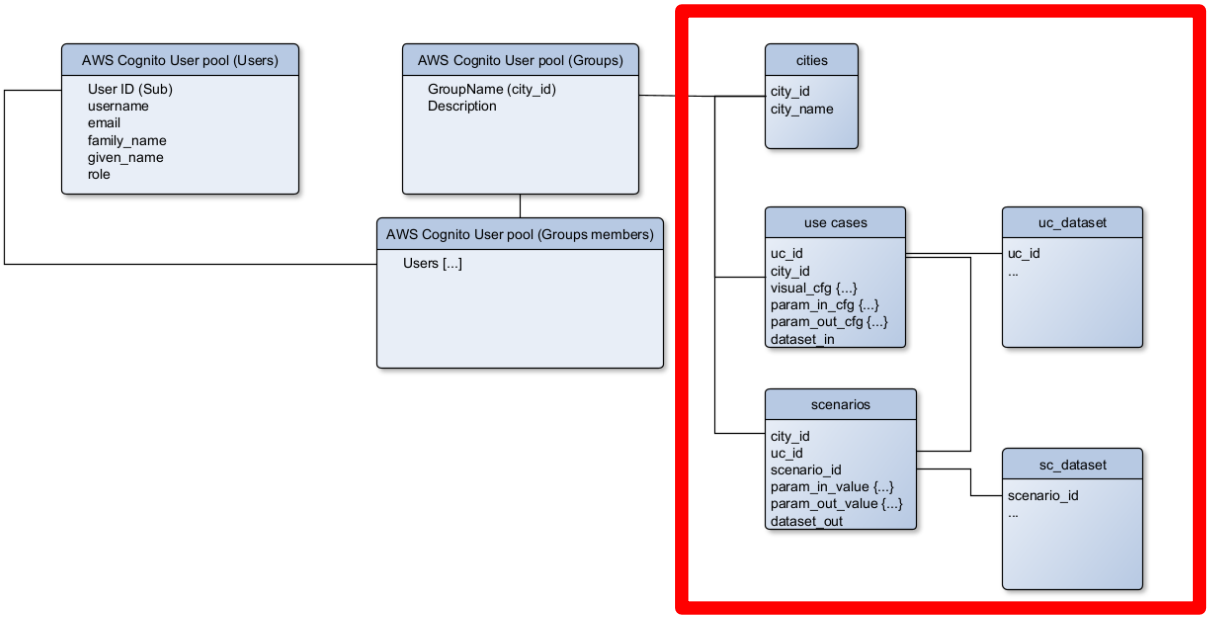


Figure 12: Set up of city management including datasets.

On the scenario level, a standard configuration approach was implemented and stored in the scenario table that is located in the general database schema, separate from the city schemas in the PostgreSQL database. Next to city_id, uc_id and scenario_id the whole scenario information for the application tier and the presentation tier is stored in the description of parameters in, and indicators out. The information is coded in a JSON string where all information on parameters with their constraints, datasets in, data sets out and indicators like KPIs can be found, see Figure 13. Also for the presentation tier, a standard set of screen configurations were defined and coded into the JSON configuration.

visual_cfg JSON

```
{
  vis_1 : {
            vis_name : "Normalised Values",
            vis_type : "line_chart",
            vis_param : {
                         x_axis : "fleet_size",
                         y_axis : [
                                   "demand_coverage_for_fleet_size",
                                   "profits_serviceoperators_for_fleet_size",
                                   "average_waiting_time_for_fleet_size"
                                  ]
                        }
          },
  vis_2 : {
            vis_name : "milan_map",
            vis_type : 'map"
          }
}
```

param_in_cfg JSON

```
{
  {
    param_id : "1",
    param_name : "EDD",
    param_objecttype : "integer",
    param_description : "xxx"
  },
  {
    param_id : "2",
    param_name : "ST",
    param_objecttype : "string",
    param_description : "xxx"
  },
  {
    param_id : "3",
    param_name : "WFS",
    param_objecttype : "double",
    param_constraint : " < 1"
    param_description : "xxx"
  },
  ...
}
```

param_out_cfg JSON

```
{
  {
    param_id : "1",
    param_name : "OFS",
    param_objecttype : "integer",
    param_description : "xxx"
  },
  {
    param_id : "2",
    param_name : "OFS_S",
    param_objecttype : "integer",
    param_description : "xxx"
  },
  {
    param_id : "3",
    param_name : "OFS_SO",
    param_objecttype : "integer",
    param_description : "xxx"
  },
  ...
}
```

param_in_value

```
{
  {
    param_id : "1",
    param_val : "5000"
  },
  {
    param_id : "2",
    param_val : "station-based"
  },
  {
    param_id : "3",
    param_name : "0.5"
  },
  ...
}
```

param_out_value

```
{
  {
    param_id : "1",
    param_val : "102"
  },
  {
    param_id : "2",
    param_val : "131"
  },
  {
    param_id : "3",
    param_val : "72"
  },
  ...
}
```

Figure 13: Setup JSON configuration.

In this development phase, the aim is to create a tangible, user-friendly dashboard for end users, such as policymakers and traffic engineers. A user with the role of administrator or city manager can create the basic configuration of a city and specific use cases in that city. The main end users of the toolkit are city members, which can add, edit, delete scenarios, and analyse scenario results (actions within a use case). A design choice was made to start simple and solid regarding setup and configuration in the toolkit:

1. On the settings page in the front end, cities, users, city members, and use cases can be created by the dashboard user depending on his/her rights. This creates the basic skeleton in the toolkit for the city and its use case(s).
2. To get a specific use case fully up and running with scenarios, the following actions are required in the backend (configuration and data environment preparation by a nuMIDAS engineer):
    - Insert visual config and parameter config into the database (JSON setup) in the toolkit config part of the database
    - Create a schema for the city if a new one is created
    - Create the necessary Input and output tables/buckets for the use case within a city's schema
    - Populate the input tables with the data needed for the backend calculations
3. After a specific use case has been configured, a dashboard user can then start to create scenarios by customising parameters and input values and running calculations based on changed inputs.

How this is reflected in the toolkit will be elaborated on in the next paragraphs.

4.1.4 UC 1: Pre-planning shared mobility services

Based on the storyboard created for nuMIDAS to clarify the workflow, see Figure 14, UC was incorporated hard coded with a first draft of the database set up to have a first learning experience. Figure 15, Figure 17, and Figure 18 show the developed dashboard.

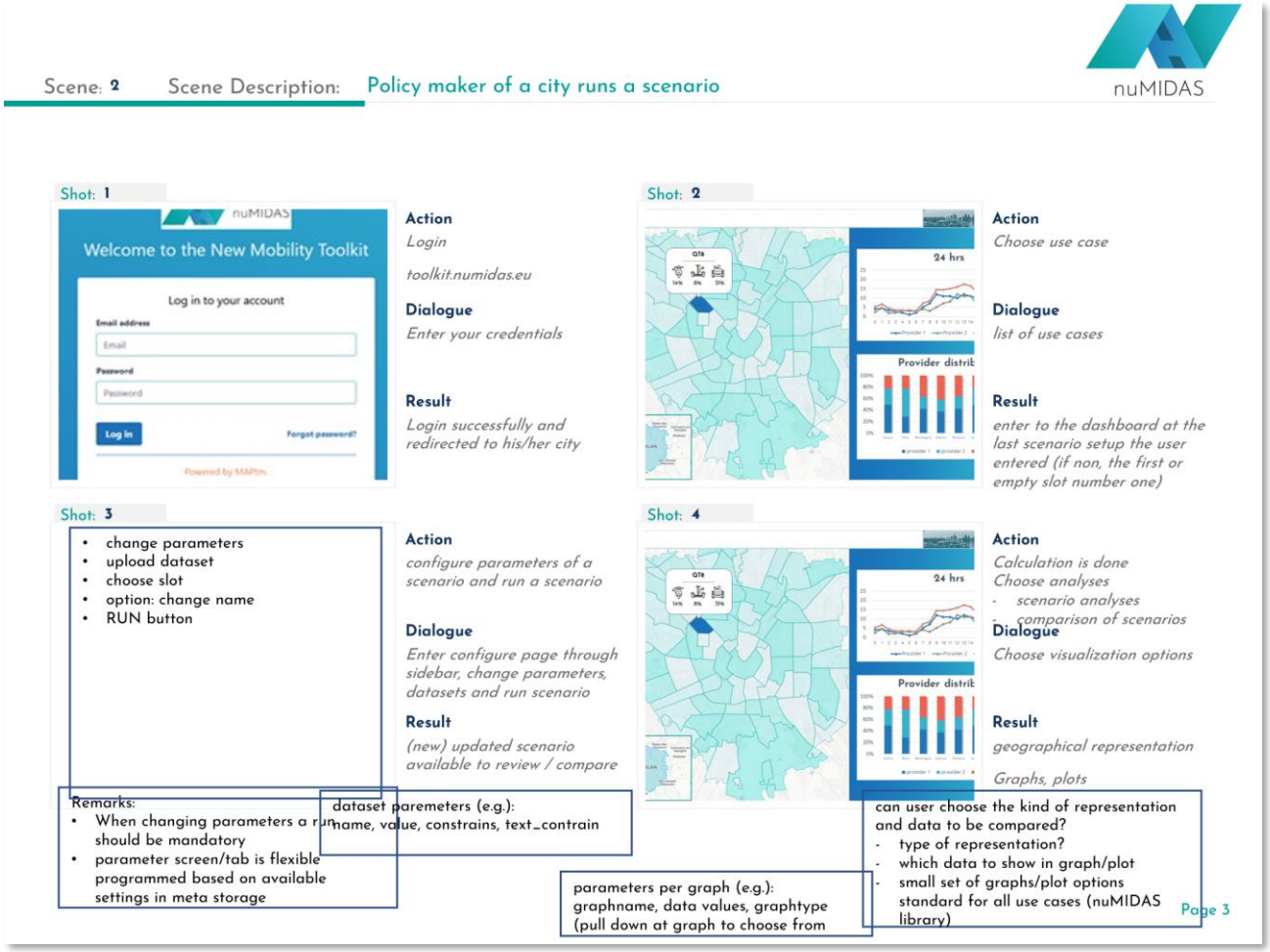


Figure 14: Storyboard UC1.

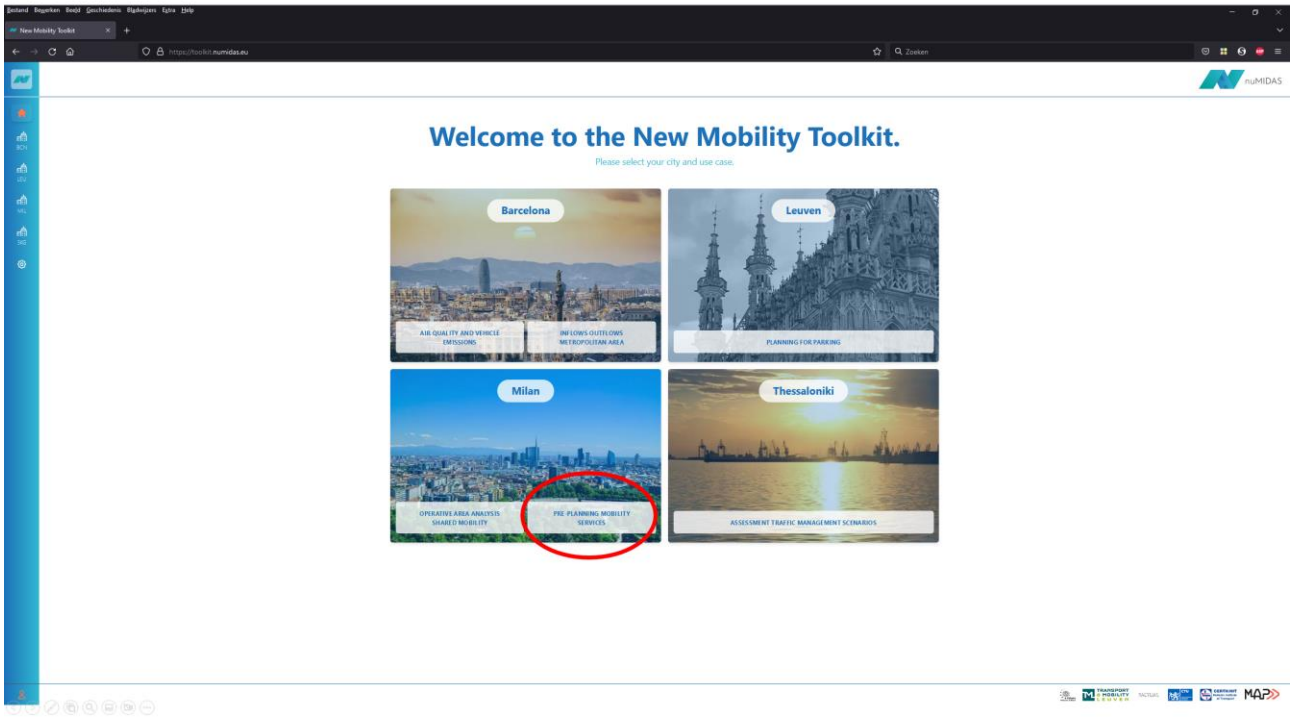


Figure 15: Use case selection.

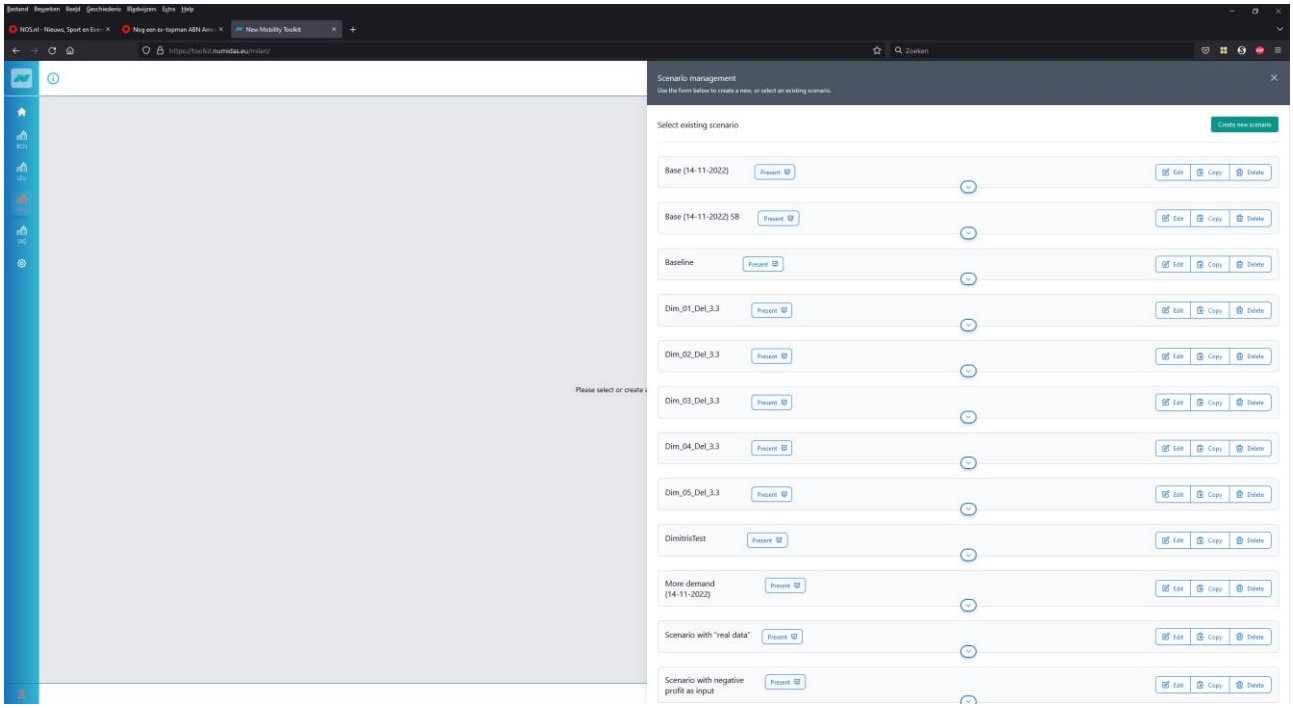


Figure 16: UC1 scenario selection.

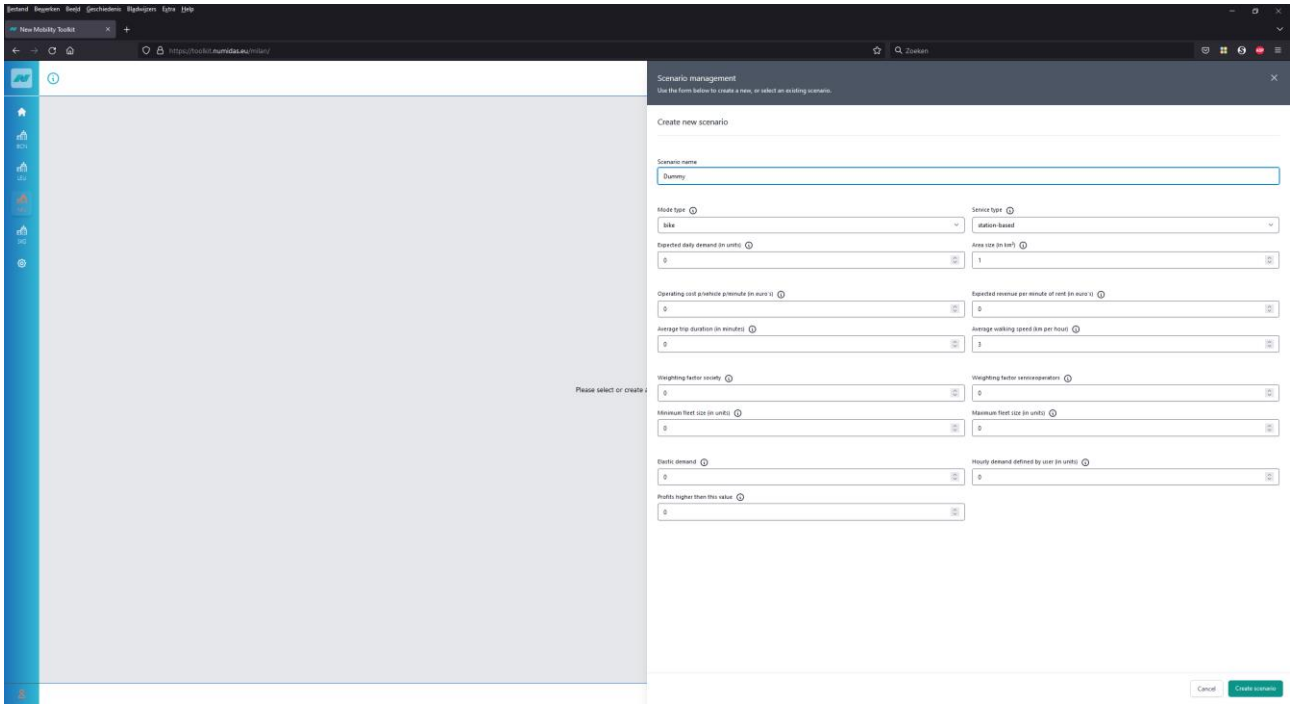


Figure 17: UC1 input.

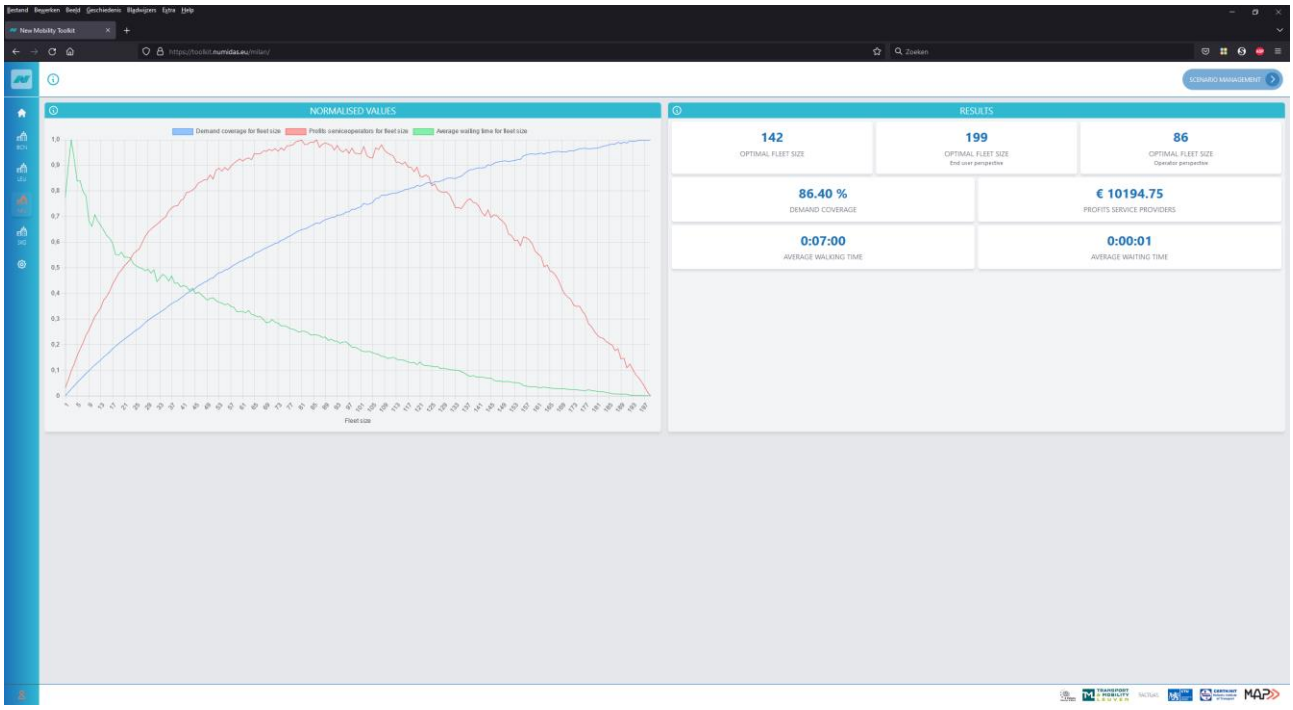


Figure 18: UC1 presentation.

For the usage of the tool and explanation of the data requirements, usage procedure, interpretation of the results and the next steps please consult Deliverable 5.4 Transferability guidelines. To that end, this paragraph and the next paragraph will only give an overview of the look and feel of the dashboard design.

4.2 Second iteration

Based on the first results of building UC1, a workflow document per use case was established for the further development. In this regard, see also the work done in Task 5.1 Usability and feasibility, reported in Deliverable 5.1.

1. Requirement gathering and discussing inputs and outputs of the application layer (Input WP3)
2. Creating a use case flowchart
3. Customer journey mapping (end-user)
4. Use case requirement analysis
5. First release UC
6. User testing (technical and functional)
7. Requirement and debug evaluation and implementation

The development order of the use cases was UC4, UC2, UC5, UC6, and UC3, based on available necessary information and level of planning detail about the use case approach. In the next paragraphs, the created workflow, the input, and the presentation are shown to give a brief overview of the look and feel of each use case.

4.2.1 UC 2: Operative area analysis shared mobility

Figure 19 shows the flowchart of use case 2 on which the interface is developed. Figure 20 and Figure 21 show the developed dashboard. For a scenario, several parameters can be set for a specific to choose shared mobility mode. The user can choose if the already served area will be the starting point. On the map, the user can select zones to be included mandatory in the calculation and zones to be excluded from the operative served area. Many alternative solutions are calculated based on the user's input. In the presentation mode, the alternatives and their KPI values are shown in a list. Each alternative can be chosen to show the served areas on the map.

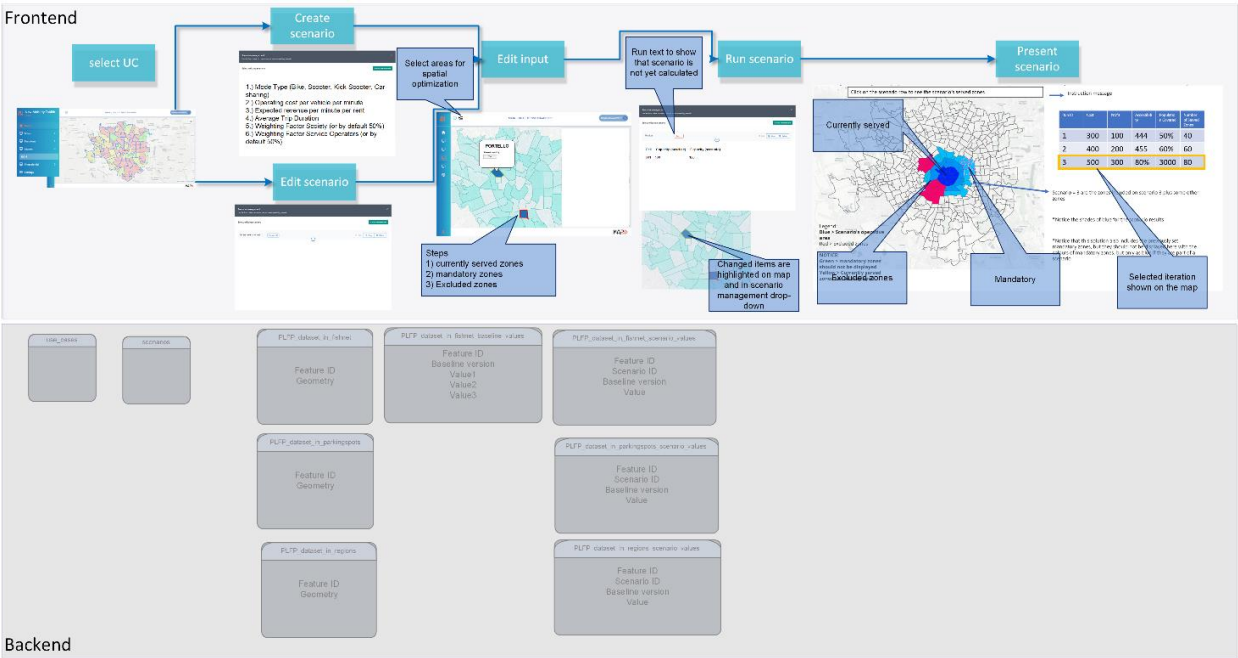


Figure 19: Workflow design UC2.

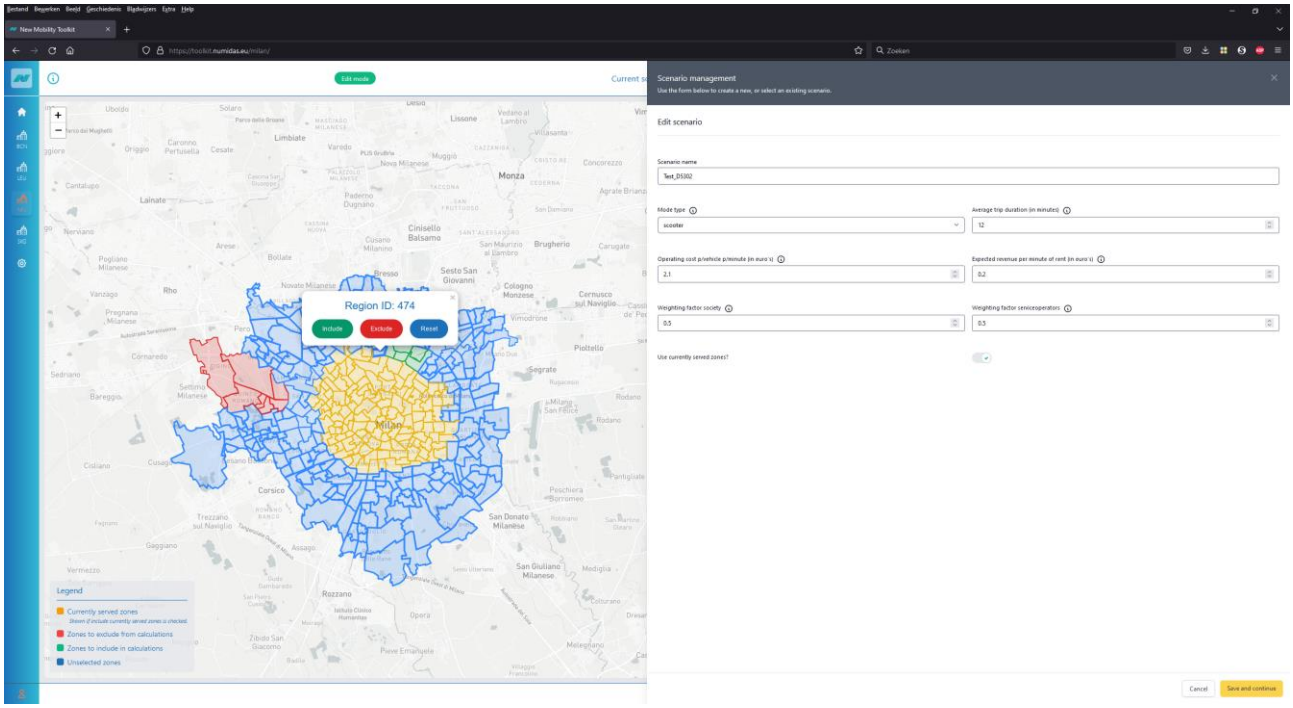


Figure 20: UC2 input.

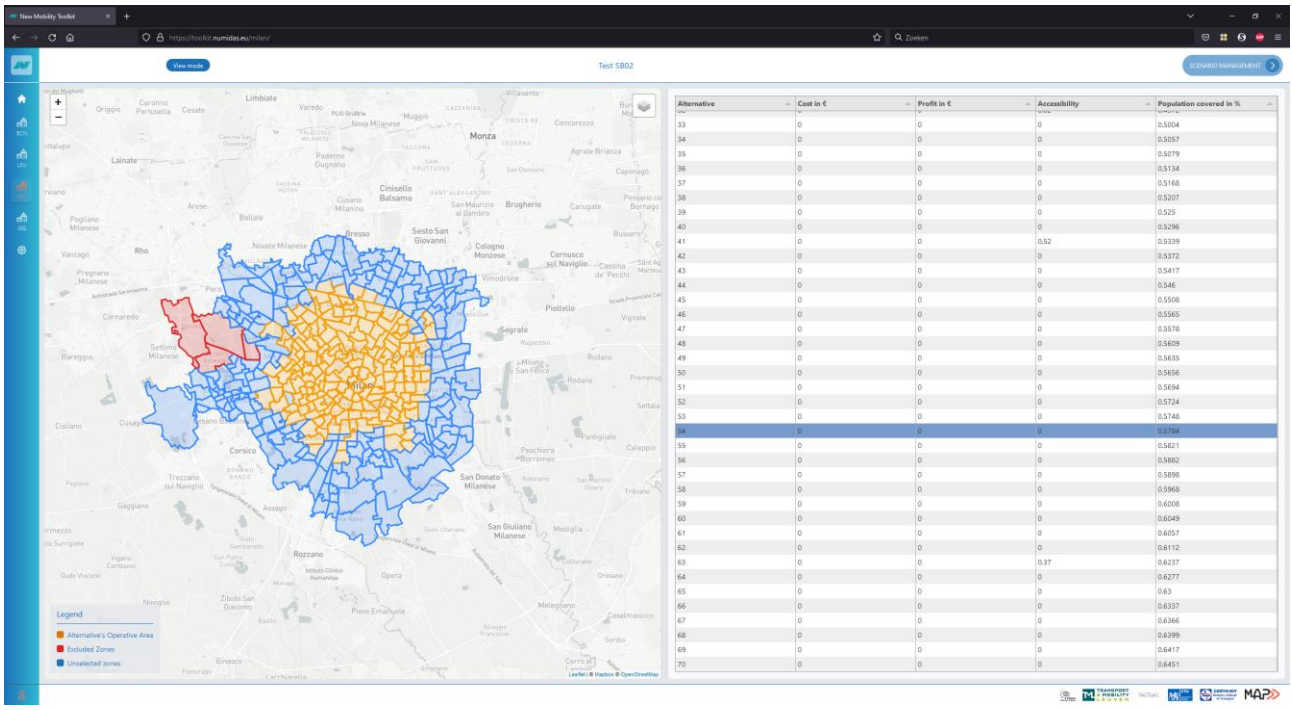


Figure 21: UC2 presentation.

4.2.2 UC 3: Air quality analysis and forcasting

Figure 22 shows the flowchart of use case 3 on which the interface is developed. The setup of the backend was straightforward and not elaborated in the chart. Figure 23 and Figure 24 show the developed dashboard. In the input phase, a time period can be selected and the ANPR cameras for which the data should be shown. In the presentation mode, ANPR cameras can be selected to show the relevant environment graphs for the selected camera only.

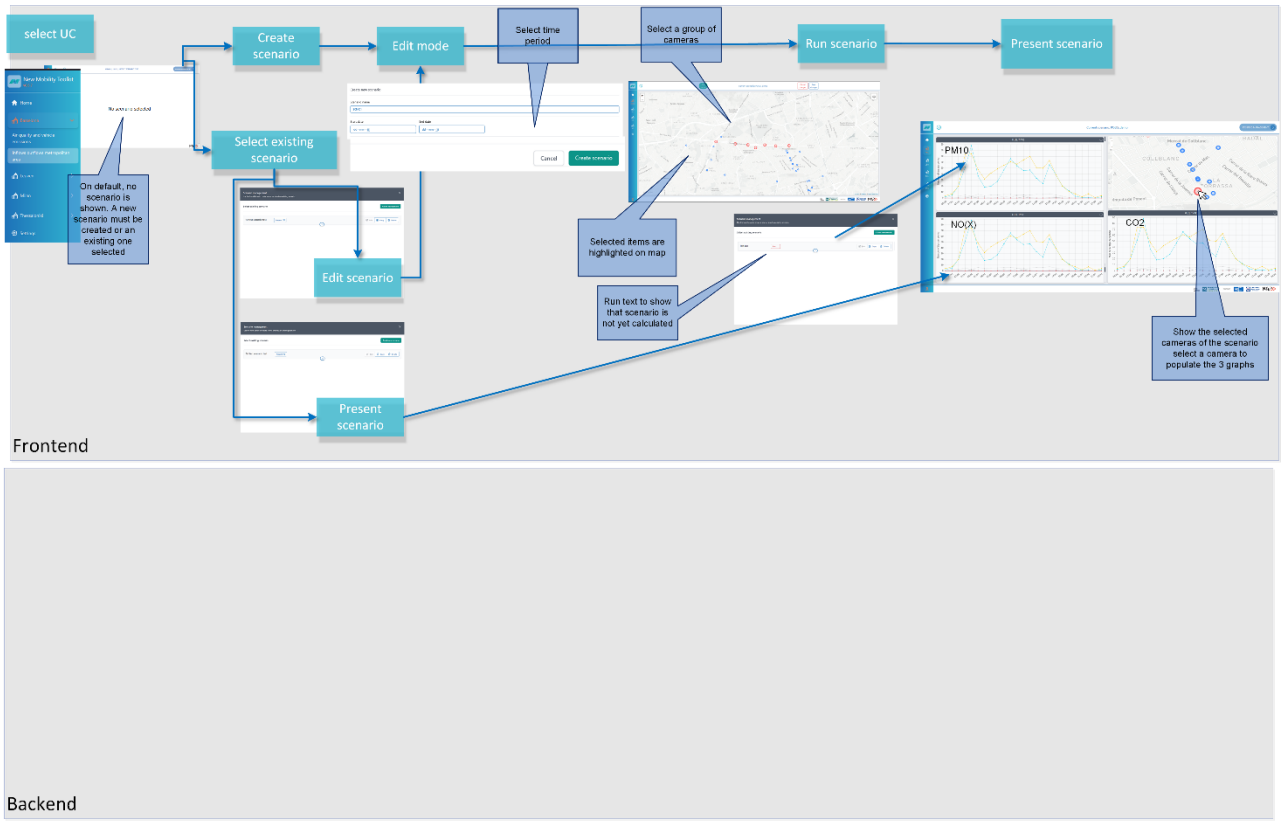


Figure 22: Workflow design UC3.

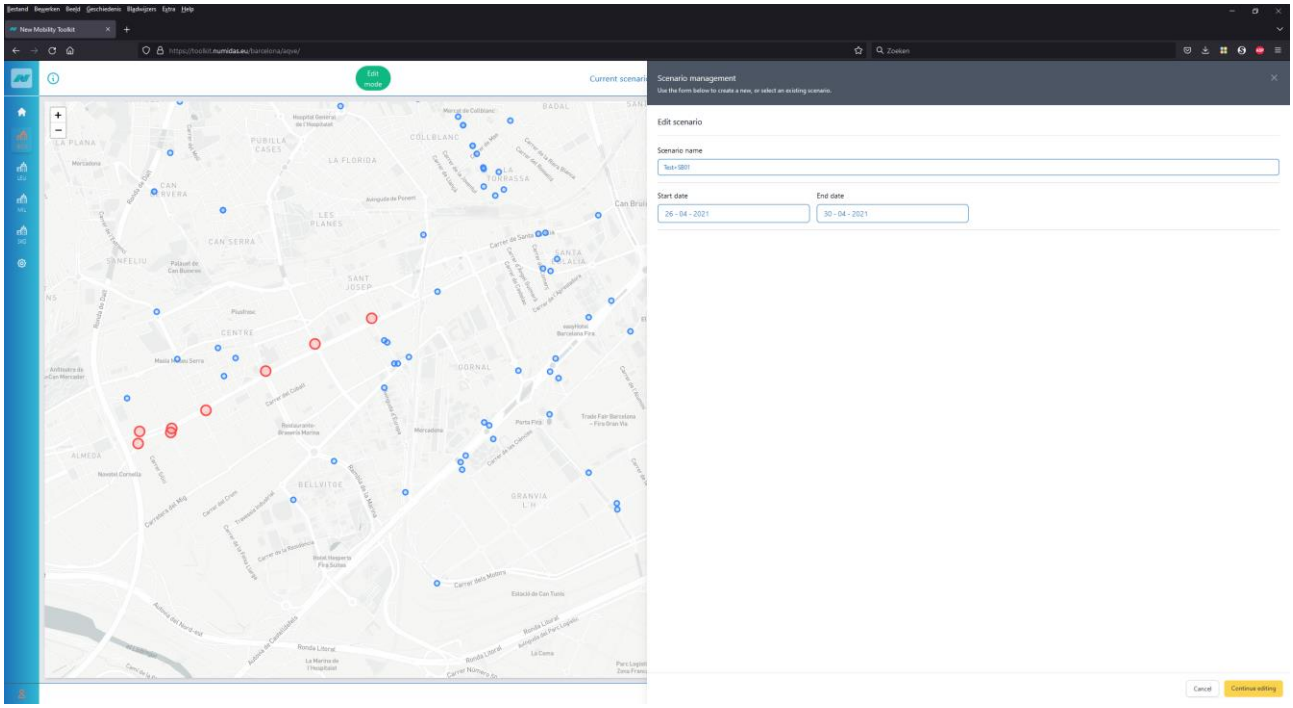


Figure 23: UC3 input.

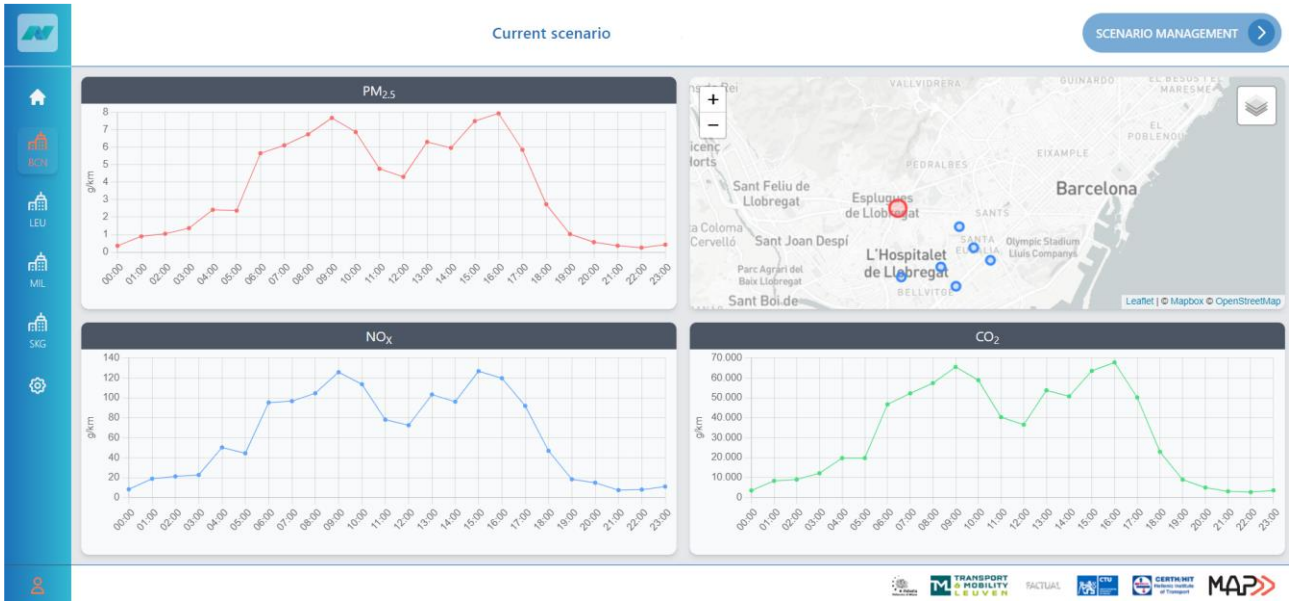


Figure 24: UC3 presentation.

4.2.3 UC 4: Planning for parking

Figure 25 shows the flowchart of use case 4 on which the interface is developed. Figure 26 and Figure 27 show the developed dashboard. Inputs of a scenario are divided into the parameter settings slide-over and the selection of areas on the map where the user (edit mode) can reduce the parking capacity. The present mode shows available Parking spots/Average demand or searching time for each area.

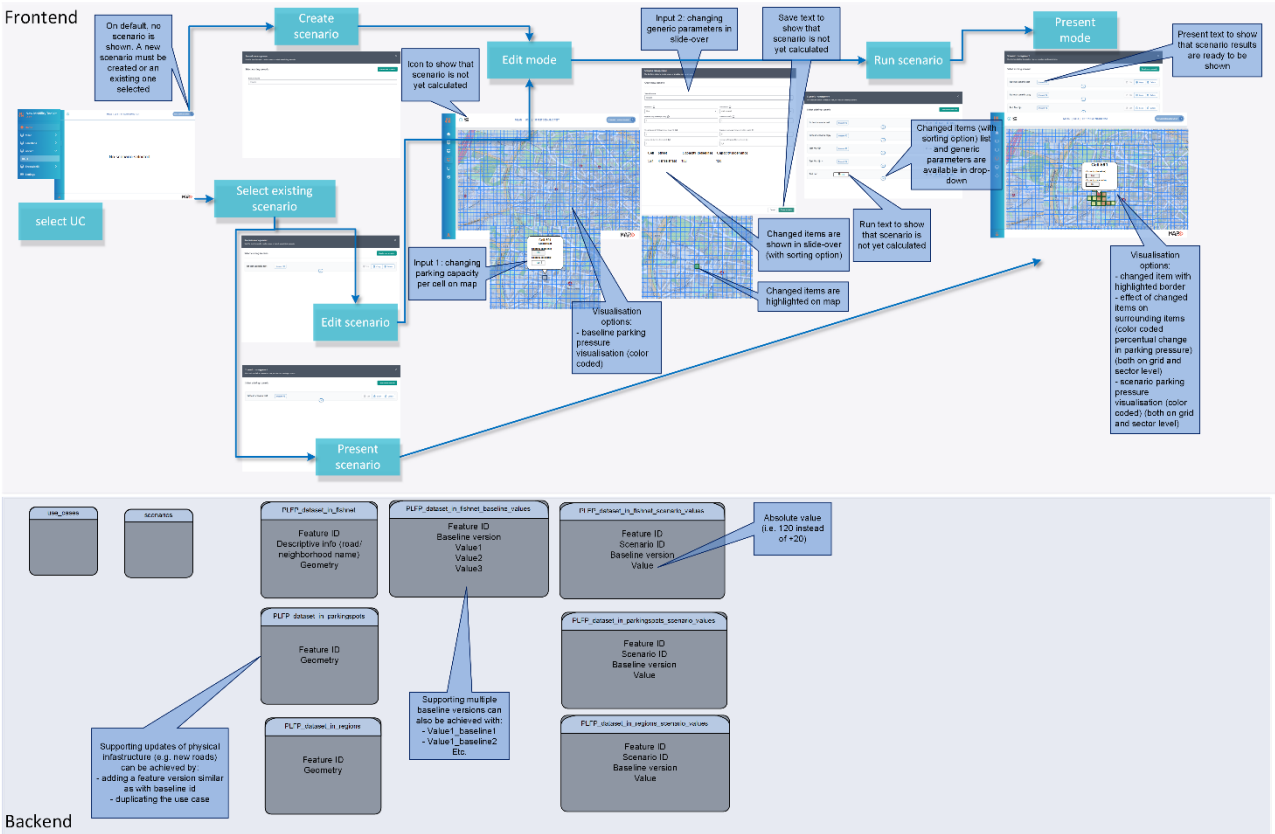


Figure 25: Workflow design UC4.

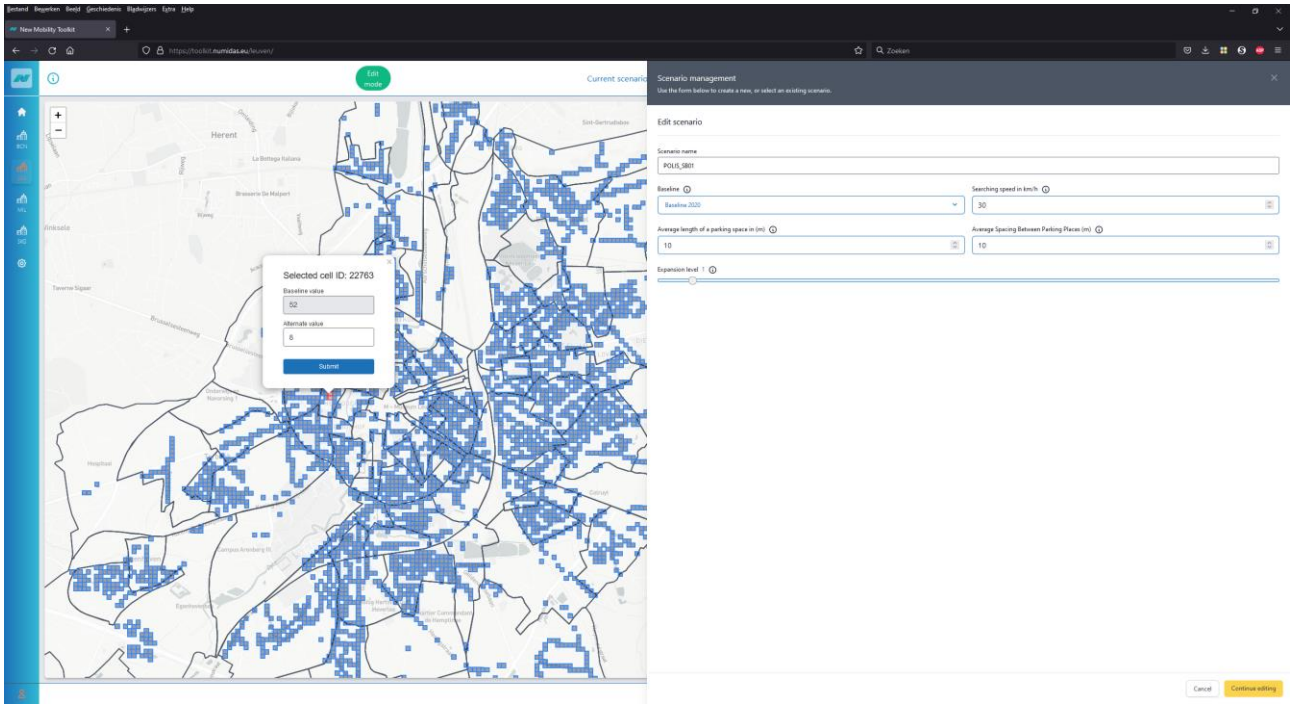


Figure 26: UC4 input.

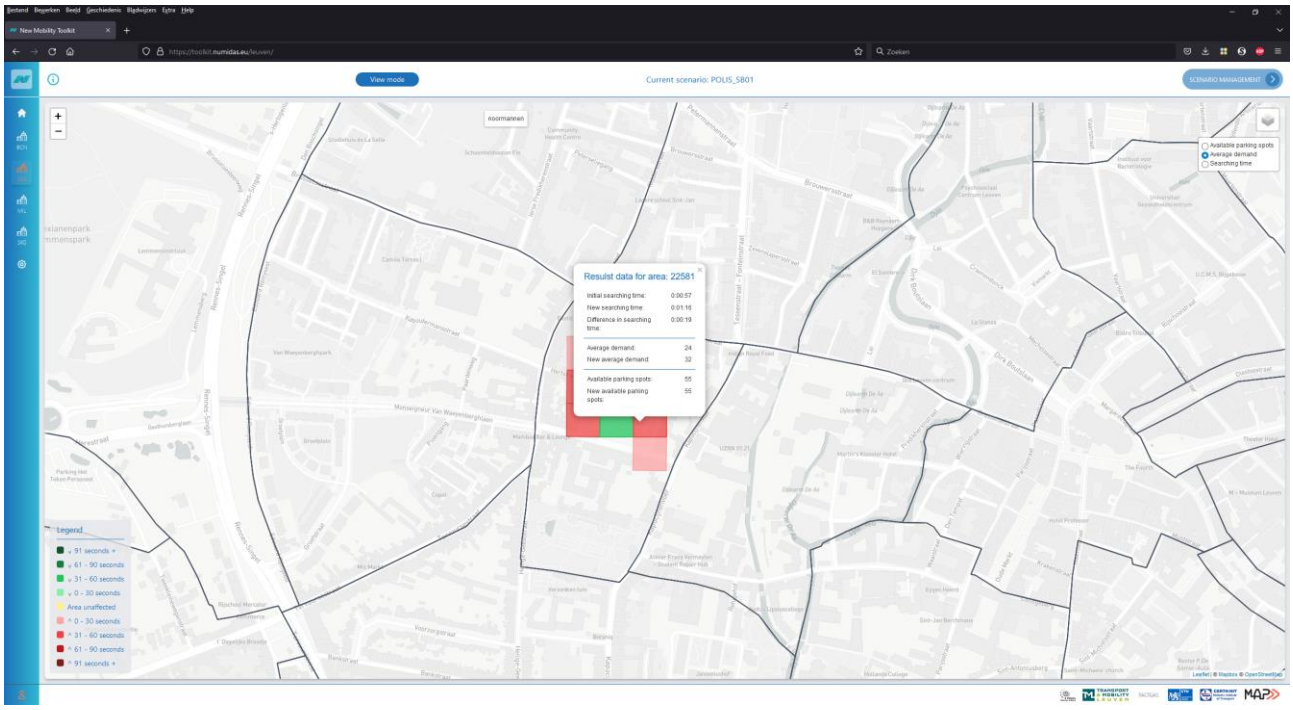


Figure 27: UC4 presentation.

4.2.4 UC 5: Assessment of inflows and outflows in a metropolitan area

Figure 28 shows the flowchart of use case 4 on which the interface is developed. Figure 29 and Figure 30 show the developed dashboard. On the input side, a temporal period including a time resolution can be selected (input parameter slide-over). After that, a number of ANPR cameras can be selected in the map presentation (edit mode). After running the scenario, an overall O/D table is calculated for the selected period and cameras and shown in the present mode. By clicking a value in de matrix, the average daily graph is shown and the related O/D relation is shown on the map. So for now, only the O/D table representation is created for the use case, as opposed to a full O/D map view for all O/D pairs.

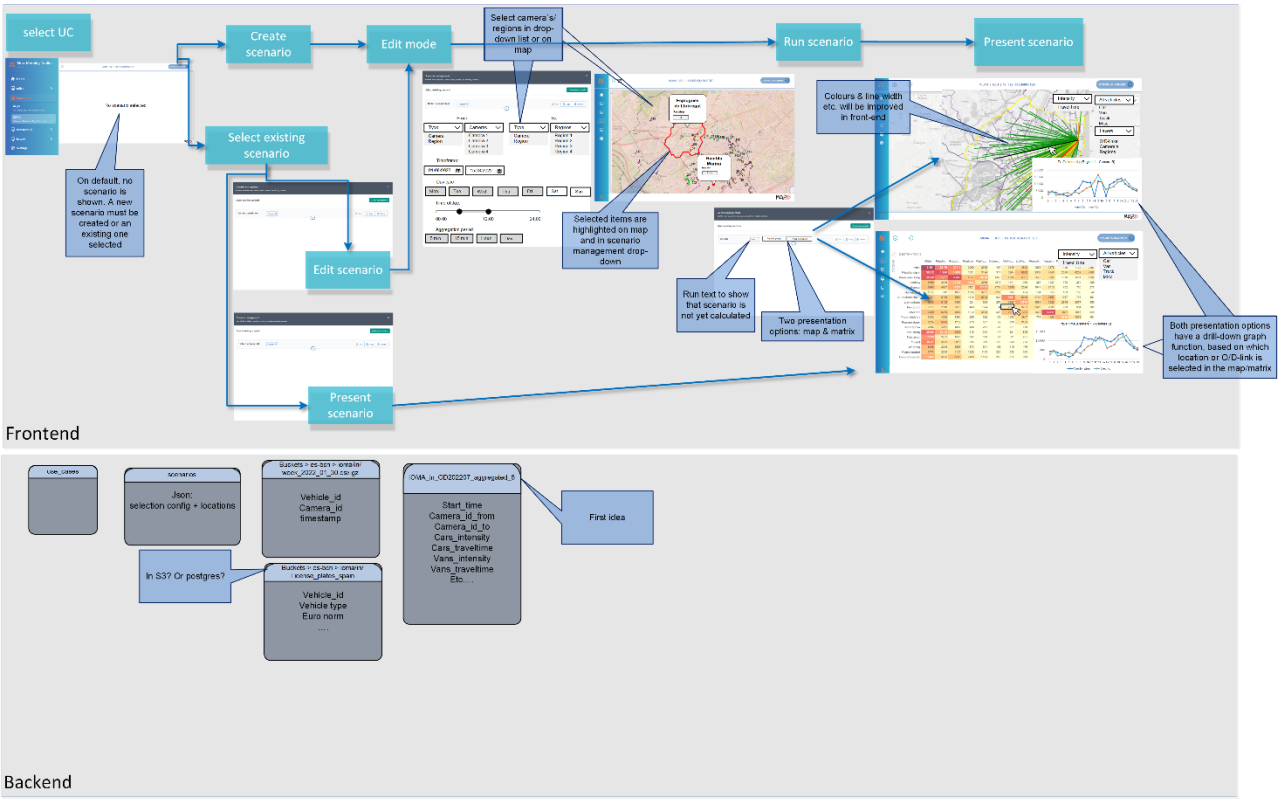


Figure 28: Workflow design UC5.

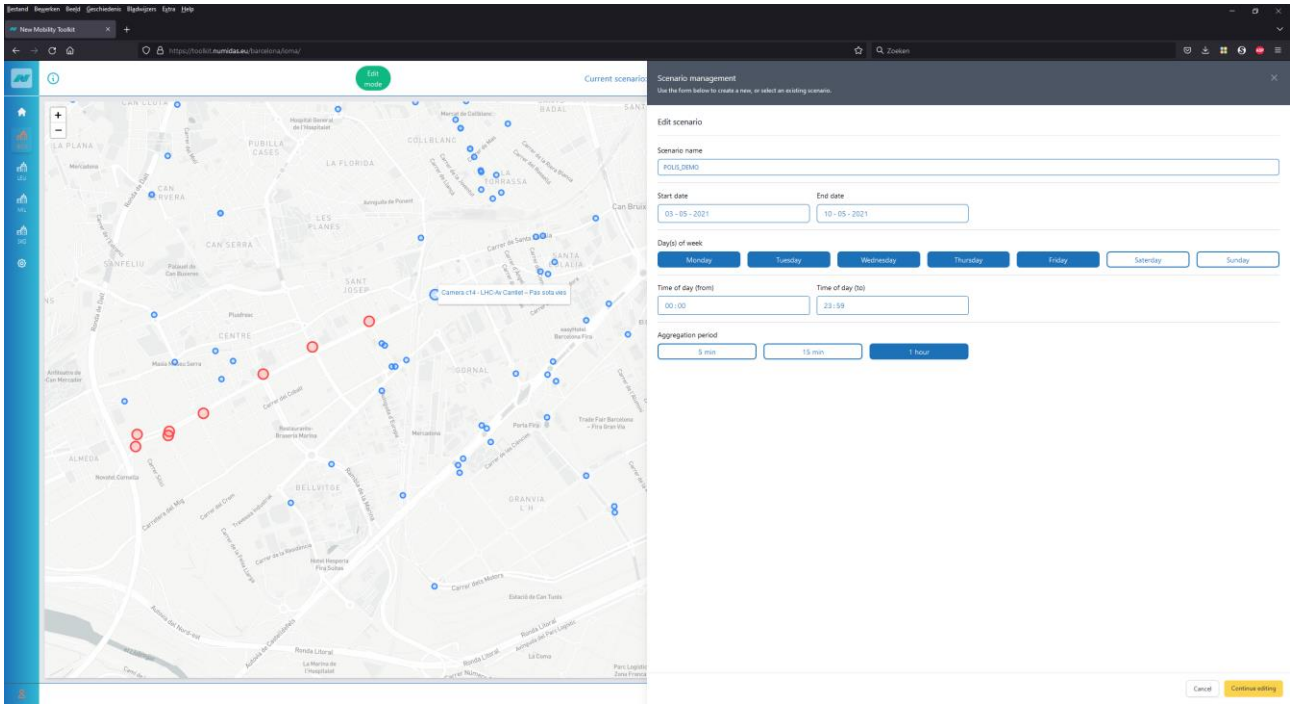


Figure 29: UC5 input.

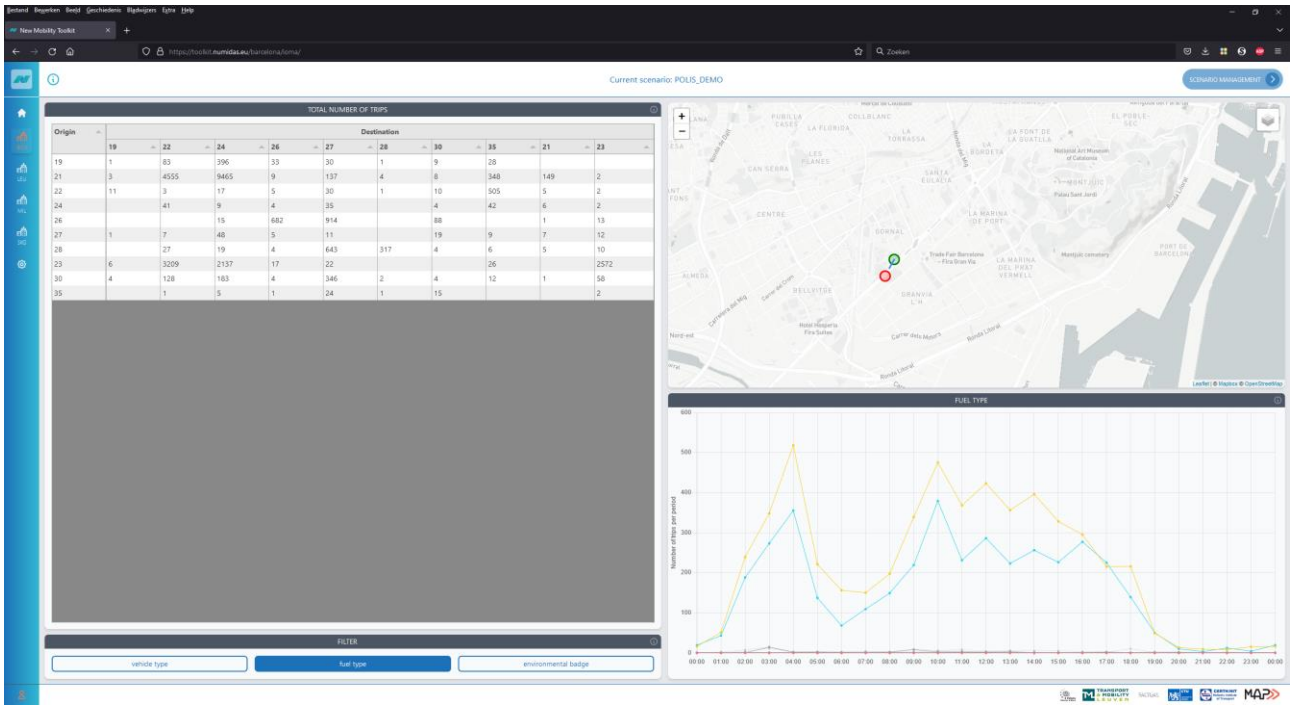


Figure 30: UC5 presentation.

4.2.5 UC 6: Assessment of traffic management scenarios

Figure 31 shows the flowchart of use case 2 on which the interface is developed.

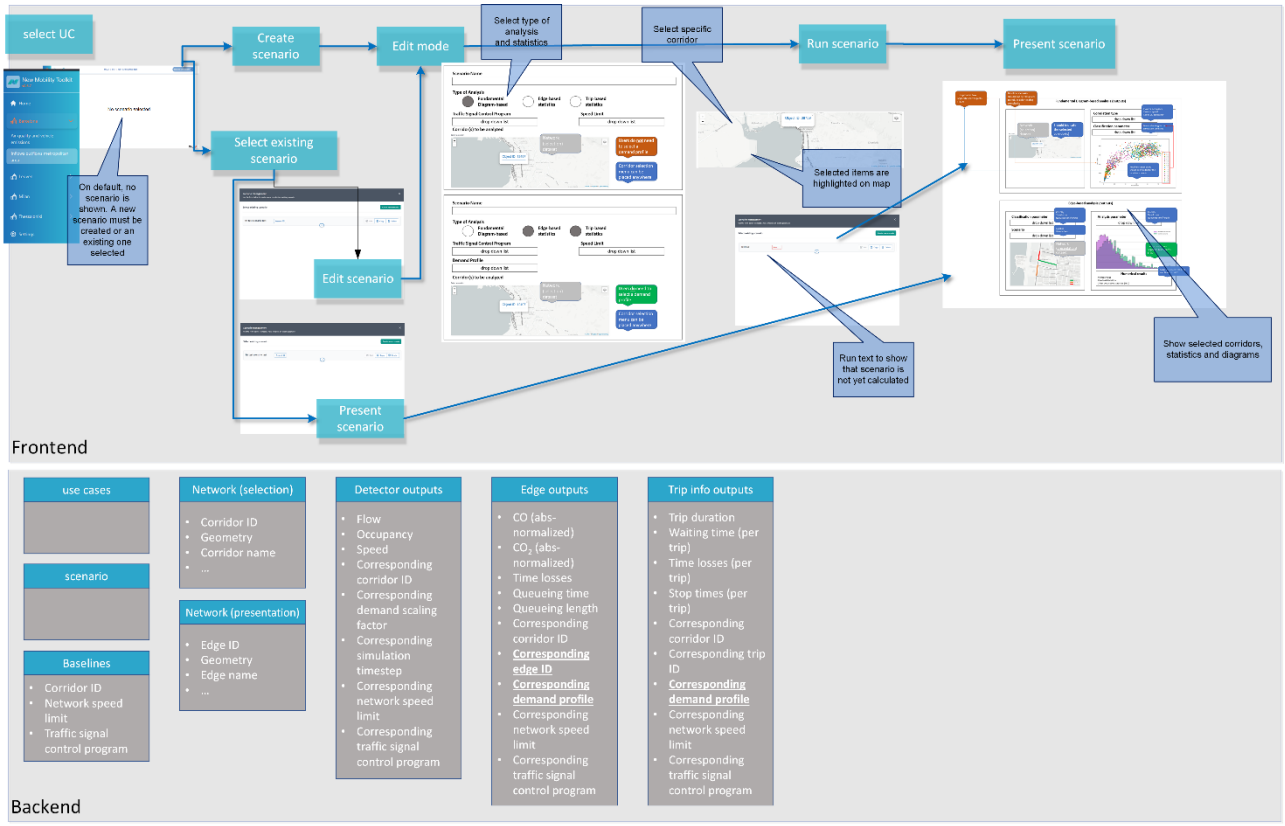
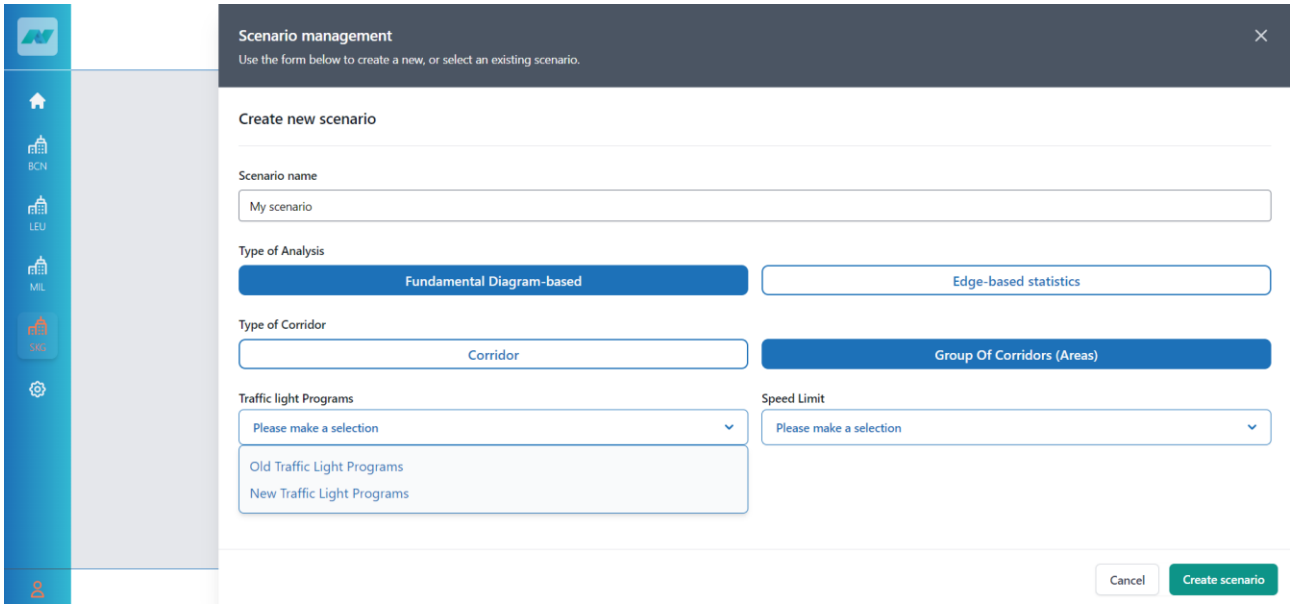


Figure 31: Workflow design UC6.

The screenshots in Figure 32 shows the developed dashboard on the input side. The user can specify a scenario name, and then what type of analysis is needed, i.e. based on a macroscopic fundamental diagram (MFD) or on the edges of the road network. Further input details the types of corridors that are used, i.e. a single corridor or aggregated results on a group of corridors forming an area. Finally, the user can select a combination of traffic management options, with two different traffic light control settings on the one hand, and various different global speed limits on the other hand.



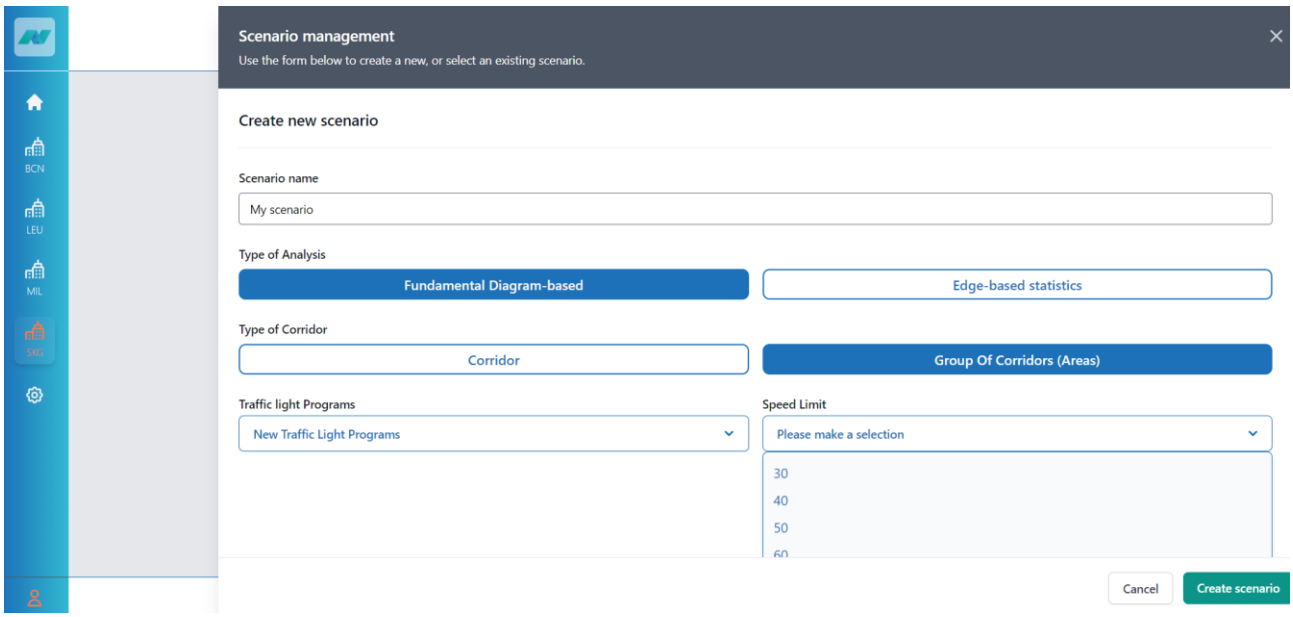


Figure 32: UC6 input.

Next, Figure 33 presents some screenshots of possible outputs (more are provided in Deliverable D4.1). Examples are shown for the MFD of a corridor (top), the same but now compared with a baseline (middle), and an edge-based group analysis (bottom).

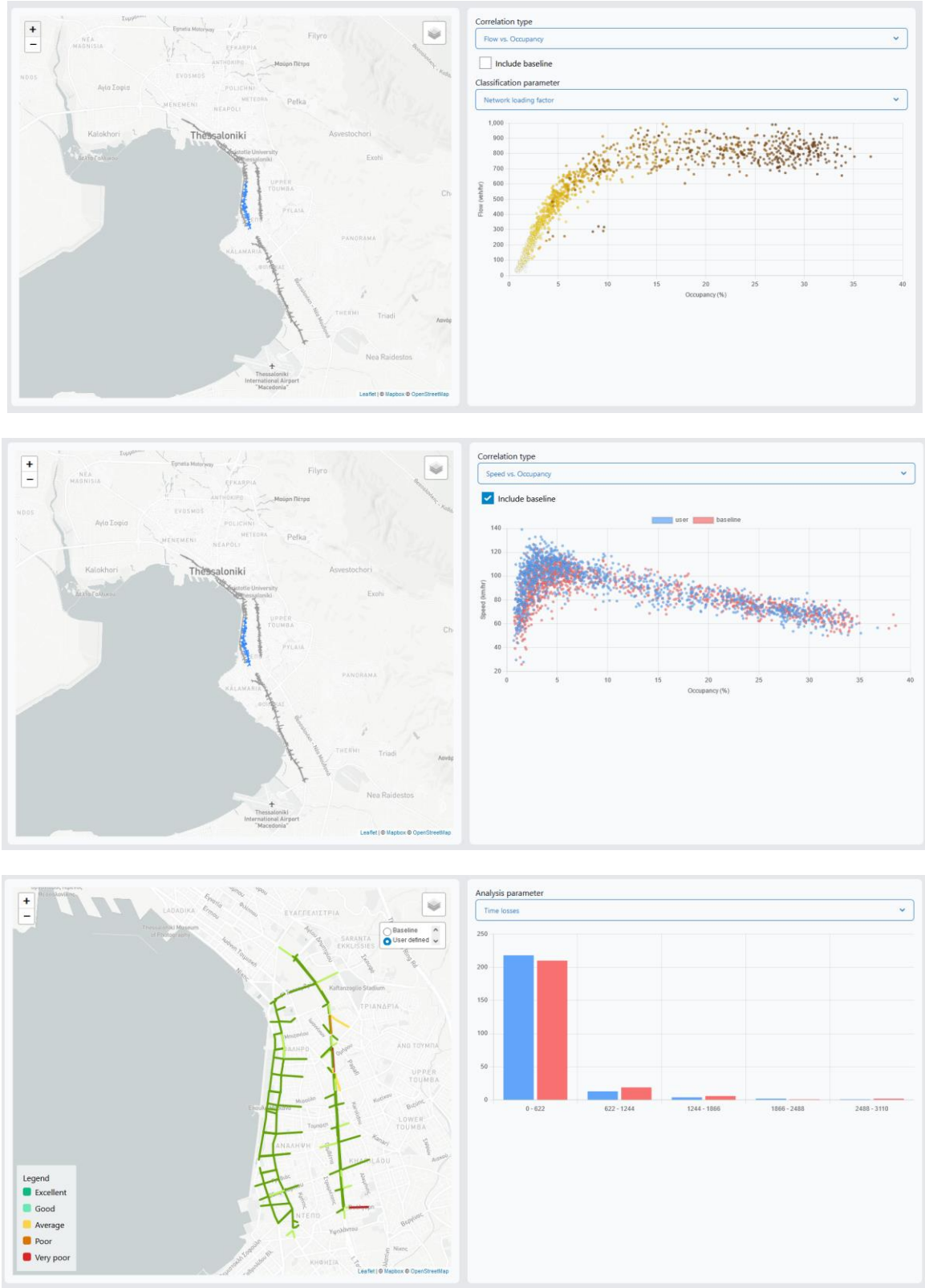


Figure 33: UC6 presentation.

5 Presentation tier

5.1 Look and feel

Based on the nuMIDAS stylesheet/guide book a first mockup is created for the dashboard to discuss layout and functionality, see Figure 34.

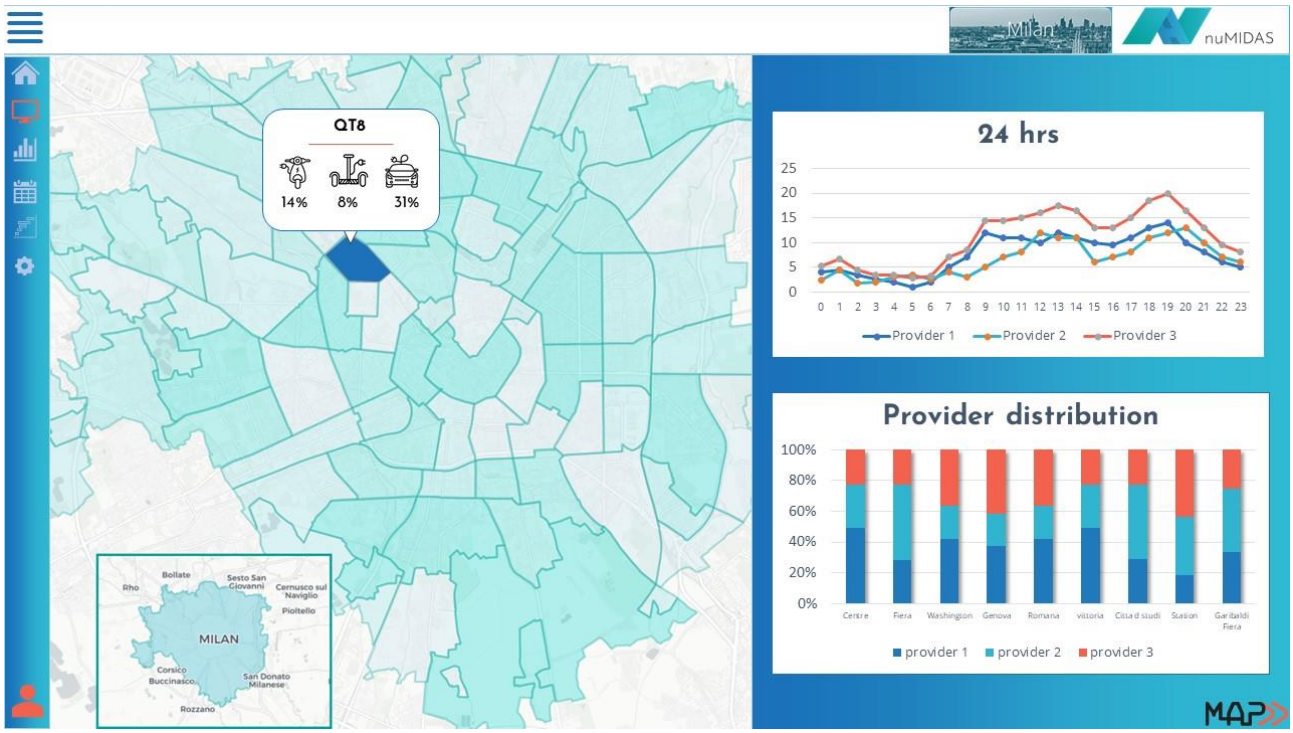


Figure 34: Mockup of the nuMIDAS dashboard.

For the first iteration of the toolkit, the front end of the dashboard was custom-made and hard-coded. So, one of the first actions was to do some rebuild towards a (partly) dynamic frontend. This was implemented using configuration tables in the backend and exposing these to the frontend through a series of new endpoints. For example, Figure 35 shows the landing page with an expanded sidebar that is dynamically populated by the configured cities and their corresponding use cases.

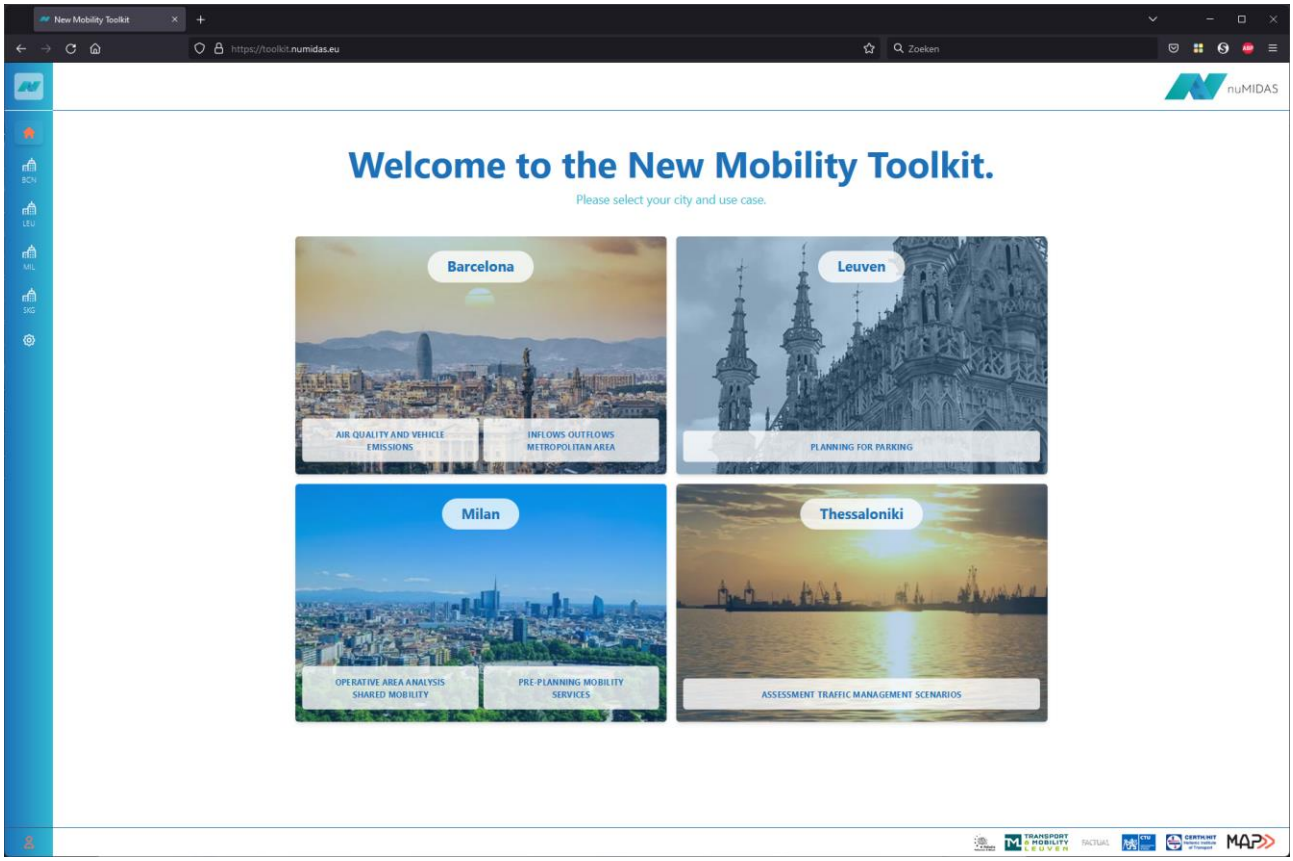


Figure 35: Toolkit landing page with dynamically populated sidebar.

City management was added to the settings page including a dynamic coupling with the specific use cases, as shown in Figure 36 ans Figure 37.

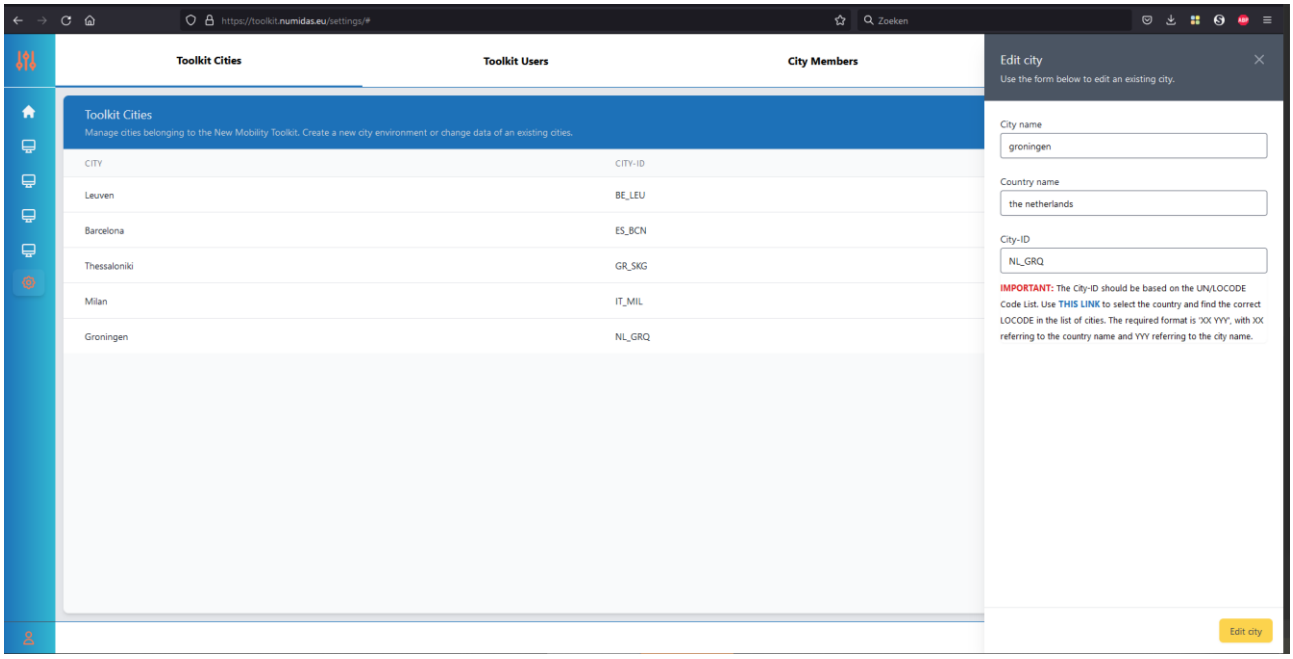


Figure 36: City management page

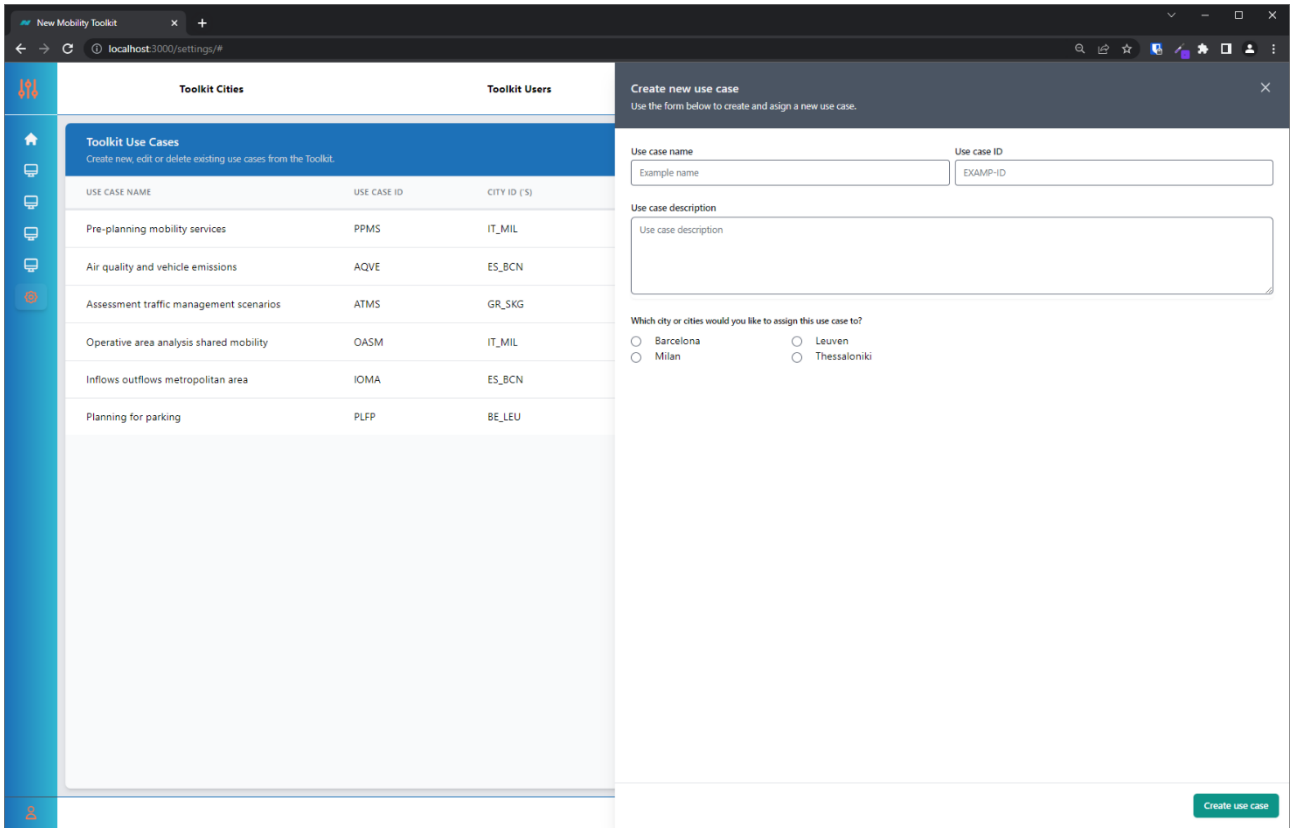


Figure 37: Use case settings page.

Furthermore, a number of requirements originating from T5.1 and bug reporting (which were all traced in Mantis) were dealt with.

5.2 Panel templates

For quick configuration of a new use case four panels were predefined and can easily be chosen from in the presentation tier, see Figure 38.

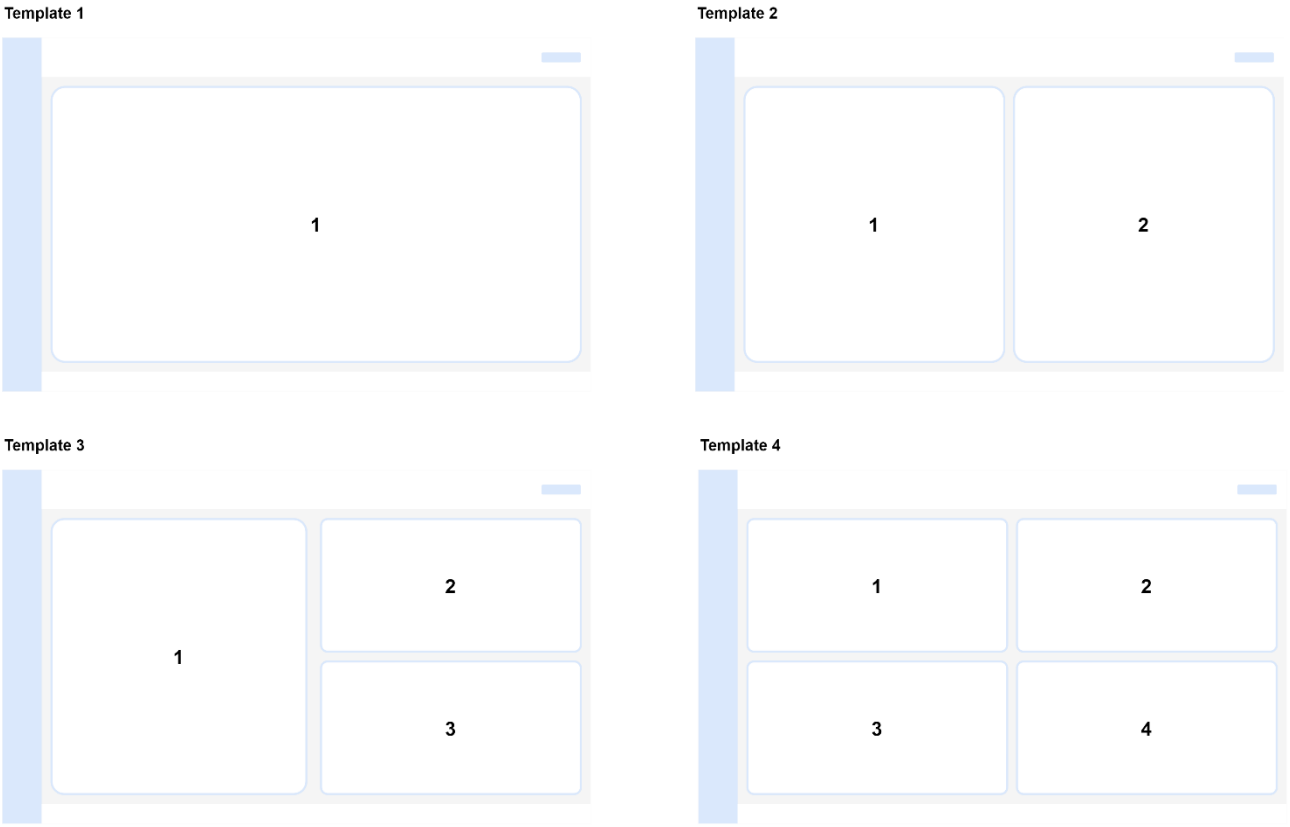


Figure 38: Panel templates.

6 Application tier

For the application layer, an extended communication between front-end and back-end have been developed. There are two levels (ways) of communication, the first one from the front-end to back-end and the second one from the back-end to the front-end.

The firsrt level of communication between the users and the developed tools refers to parsing the input data to the algorithms and fire them up. For this purpose, a well structured database has been created where for all the input parameters the values are stored. After the storage of the input values to the respective database the front-end through an endpoint fires up the algorithms at the back-end. The algortihms receive as input through the endpoint the city id, the use case id, and the scenario id in order to retrieve the correct input parameters.

The second level of communication involves the way the calulcate results of each algorithm are stored and retrieved from the front-end. For this purpose, and based on the use case and the nature of the ouput values two different implementations are adopted. The first implementation, implies that the output values produced by the algorithm are stored at the same database where the input parameters are stored, while the second one, implies that the ouput values are stored to different tables for specific use cases. During the second implementation the scenario id parameter is used in order to separate the ouputs amongh themselves.

6.1 API endpoints

The nuMIDAS toolkit consist of various tiers, as is shown in Chapter 3. These tiers communicate with eachother using REST APIs. An API endpoint specifies the location of certain toolkit resources or functions. The request and response specifications and endpoints were coordinated during the development of the toolkit, so that data can move through the tiers as intented and the required toolkit functionality is realised.

6.1.1 Use case endpoints

The following endpoints have been exposed from the data and application perspective of the toolkit:

- UC1 Pre-planning mobility services
  - UC1 static parameter
  - City id parameter
  - Use case id parameter
  - Scenario id parameter
- UC2 Operative areas anaylsis
  - UC2 static parameter
  - City id parameter
  - Use case id parameter
  - Scenario id parameter
- UC3 Air quality analysis and forecasting
  - UC3 static parameter
  - City id parameter
  - Use case id parameter
  - Scenario id parameter

- UC4 Planning for parking
    - City id parameter
    - Use case id parameter
    - Scenario id parameter
- UC5 Inflows and outflows in a metropolitan area
    - UC5 static parameter
    - City id parameter
    - Use case id parameter
    - Scenario id parameter
- UC6 Assessment of traffic management scenarios
    - UC6 static parameter
    - City id parameter
    - Use case id parameter
    - Scenario id parameter

6.1.2 Presentation tier endpoints

The following endpoints have been exposed from the presentation perspective of the toolkit:

- User management:
    - Authorisation and password functions
    - Creating, updating, and deleting users
    - Creating, updating, and deleting user groups
- City management:
    - List of all configured cities including descriptive information
    - Add, edit, and delete cities
- Use case management:
    - List of all configured use cases including descriptive information
    - Add, edit, and delete use cases
- Scenario management:
    - List of all scenarios per use case including descriptive information, metadata, inputs, and outputs
    - Add, edit, and delete scenarios (including helper functions for separate results tables)
    - Scenario progress functions
- Miscellaneous:
    - Dynamic site navigation functions (landing page, header, and side-bar)
    - Static geodata functions for map views in use cases

7 Data tier

7.1 Naming convention

The following naming convention is in use for referring to countries, cities and use cases.

Table 2: Country codes.

Country code (XX)	Country name
ES	Spain
BE	Belgium
GR	Greece
IT	Italy

Table 3: City codes.

City code (YYY)	City name
MIL	Milan
LEU	Leuven
SKG	Thessaloniki
BCN	Barcelona

Table 4: Use case codes.

Use Case code (ZZZZ)	Use Case name
OASM	Operative area analysis shared mobility
PLFP	Planning for parking
ATMS	Assessment traffic management scenarios
PPMS	Pre-planning mobility services
AQVE	Air quality and vehicle emissions
IOMA	Inflows outflows metropolitan area

The country-ID and City-ID is based on the UN/LOCODE Code List. The required format is 'XX YYY', with XX referring to the country name and YYY referring to the city name.

7.2 PostgreSQL

Data tables within the city-specific schemas are named according to the format *ZZZZ_[in/out]_name*. In these tables, the scenario_id is included as a separate column in case the data relates to a specific scenario. The following structure shows examples of tables in these schemas. The tables in the schemas toolkit_config and public are fixed.

Abbreviations:

- uc = usecase
- param(s) = parameter(s)
- cfg = configuration

Server:
postgresql-01 nuMIDAS
Database:
numidas
Schemas:

- **toolkit_config**
  Tables:
  - **cities**
    Columns:
    - city_id character (6)
    - city_name character varying
    - country_name character varying
  - **use_cases**
    Columns:
    - uc_id character (4)
    - uc_name character varying
    - uc_description character varying
    - city_id character (6)
    - page_cfg json

```
{
    “template”: <1…4>,
    “panels”: [
        {
            "id": <1..4>,
            “info”: {
                “name”: “<panel_name>”,
                “description”: “<panel_description>”,
                “image”: “<panel_description_image>”
            },
            "type": "<bar_chart/line_chart/map/kpi>",
            "axis_names": {
                "x": "<x_axis_name>",
                "y": "<y_axis_name>"
            },
            “indicators”: [
                {
                    "id": <1..N>,
                    "name": "<name>",
                    "objecttype": "<type>",
                    "description": {
                        "main": "<main description>",
                        "sub": "<sub description >",
                    },
                    “benchmark”: “<benchmark_value>”,
```

```
                                        “color”: ”color_value”
                                    },
                                    {
                                    …
                                    }
                                ],
                            },
                            {
                                …
                            }
                        ]
                    }
```

- params_in_cfg json

```
[
    {
        "id": <1..N>,
        "name": "<name>",
        "objecttype": "<type>",
        "description": "<description>",
        "constraint": {
            "min": "<parameter minimum value>",
            "max": "<parameter maximum value>",
        }
        },
        {
        …
        }
    ]
```

- datasets_in_cfg json

```
- [
    {
        "_id": <1..N>,
        "description": “<description>”,
        "type": "<s3/postgresql>",
        "path": "<bucket name/table name>",
        “columns”: [
            {
                “column_name”: "<column_name>”,
                “column_ objecttype”: "<column_type>”
            },
            {
                …
            }
        ]
    },
    {
        …
    }
]
```

- datasets_out_cfg json

```
- [
    {
        "id": <1..N>,
        "description": “<description>”,
        "type": "<s3/postgresql>",
        "path": "<bucket name/table name>",
        “columns”: [
            {
                “column_name”: "<column_name>”,
```

```
                                                    “column_ objecttype”: "<column_type>”
                                                }, 
                                                {
                                                    …
                                                }
                                            ]
                                        },
                                        {
                                            …
                                        }
                                    ]
```

- **scenarios**
  Columns:
  - city_id character (6)
  - uc_id character (4)
  - scenario_id integer
  - scenario_name character varying
  - params_in_value json
    - [
      {
        "id": <1..*N*>,
        "value": "<value>"
      },
      {
        …
      }
      ]
  - indicators_out_value json
    - [
      {
        "id": <1..*N*>,
        "value": "<value>",
      },
      {
        …
      }
      ]
  - started_at timestamp without time zone
  - percentage_completed integer
  - completed_at timestamp without time zone
  - failed_at timestamp without time zone
  - edited timestamp without time zone
  - created_by character varying

- **[XX_YYY]**
  Tables:
  - **[ZZZZ]_[in/out]_<name>**
    Columns:
    - <name> <type>
    - uc_id character (4)
    - scenario_id integer

- **public**
  Tables:
  - **spatial_ref_sys**

The primary place for storing the results of a scenario (in other words: the KPI that are shown in the presentation layer of the toolkit) is the indicators_out_value column in the toolkit_config.scenarios table. For clarity and optimisation reasons, a secondary approach using a separate results table has also been implemented for a selection of use cases based on the data size and structure of its KPI results. In these cases, a separate results table has been created in the city-specific schema with a custom table name. This approach allows the toolkit to benefit from the inherent benefits of table-based data storage where needed. A scenario_id column is included to join these separate results back to the corresponding toolkit_config.scenarios record.

7.3 S3 buckets

Files within the buckets xx-yyy are named according to the format *[zzzz]/[in/out]/name[-scenario_id]*. The *scenario_id* in the file name can be omitted if the output data does not differ by scenario or if the input data does not specifically relate to a scenario.

The following structure shows examples of files within the buckets.

Buckets:

- **[xx-yyy]**
  Objects:
  - **[zzzz]**
    Objects:
    - **[in]**
      Objects:
      - **<name><-scenario_id>**
    - **[out]**
      Objects:
      - **<name><-scenario_id>**

8 Conclusions

During the project, the toolkit was designed, set up, and populated with use cases. The partners and end users extensively evaluated all use cases. Most use cases in the toolkit are at a TRL 7 at the end of the project with Use cases 3 and 6 more on TRL 6. During the development of the toolkit and the implementation of the six use cases, the building process speeded up when more use cases came to a usable and test-worthy version. More generic code came available and by following the same scenario setup and approach the creation of the use case became easier and quicker. This gives the nuMIDAS toolkit and its partners a starting point for further exploitation.