TES_08: To create new incentive based services in the MDMS to increase the use of PT

Description of the measure and main outcomes expected

Thessaloniki's measure 08 aims to identify accessibility gaps, identify potential synergies between PT and shared mobility services and provide guidance about where shared mobility services should operate. TES_08 has two outputs: a) an accessibility analysis focused on the city of Thessaloniki, b) a neutral platform that is demonstrated for the city of Thessaloniki but it can be adopted also by other cities (if the required data are available). The purpose of the analysis was to identify areas in Thessaloniki with high travel demand where public transport provides inadequate services and micromobility can improve travel time and have commercial potential. The neutral platform aims to assist in evidence-based decision-making of local authorities and micromobility operators. More specifically, its goal is to demonstrate the potential that micromobility services have for attracting a reasonable modal share for specific Origin-Destination pairs.

Preparation of the measure

The first steps towards the preparation of the measure were to define the methodology for the accessibility analysis and to collect the required data. Following these, the accessibility analysis was conducted. Then, with regards to the neutral platform, the architecture and the methodology were initially defined. Then the development of the back-end and the user interface was carried out.

Accessibility analysis: The basis of the analysis is the Traffic Analysis Zones (TAZs) of the study area. TAZs are universally used in travel demand modelling to represent the spatial distribution of trip origins and destinations, as well as the population, employment and other spatial attributes that generate or otherwise influence travel demand. Except for the TAZs, the analysis also utilizes the Origin-Destination (OD) matrices of Thessaloniki's SUMP, which actually express the number of trips that are being conducted on a daily basis between each pair of TAZs. The analysis was carried out using QGIS 3.28.6.

First, isochrone maps[1] were created using as network the road network of Thessaloniki and as start points the bus stops. The considered mode of transport was walking and the time defined was 10 minutes. The purpose of this step was to identify any existing gaps, meaning areas of the city that are not accessible from a bus stop within 10 minutes of walking. With this step, the spatial coverage of the bus lines was essentially checked; no gaps were identified.

The trips considered were those either destinating to or originating from Thessaloniki's centre. Given the OD matrix the total number of trips of each TAZ from and to the centre area was calculated. In more detail, the following trips were calculated: a) trips from each TAZ outside of this area to the centre area, b) trips from the centre area to each TAZ outside of this area.

The number of calculated trips were divided by the area of each TAZ, for comparability purposes. Then the trips/km^2 were categorized in 4 classes of equal intervals. The class with the highest values included 17 TAZs. It was then checked if the selected TAZs were within the current operation zone of the main

[1] Isochrone maps, also known as travel time polygons or travel time maps, are maps that show all reachable locations within a specified limit by a specified mode of transport. They are mostly used to depict travel times, such as drawing a 30-minute travel time perimeter around a start location.

shared-mobility provider (HOP). Only 3 out of 17 selected TAZs were outside HOP's operation zone (Fig 51). The rest of the steps are focused on these three TAZs.

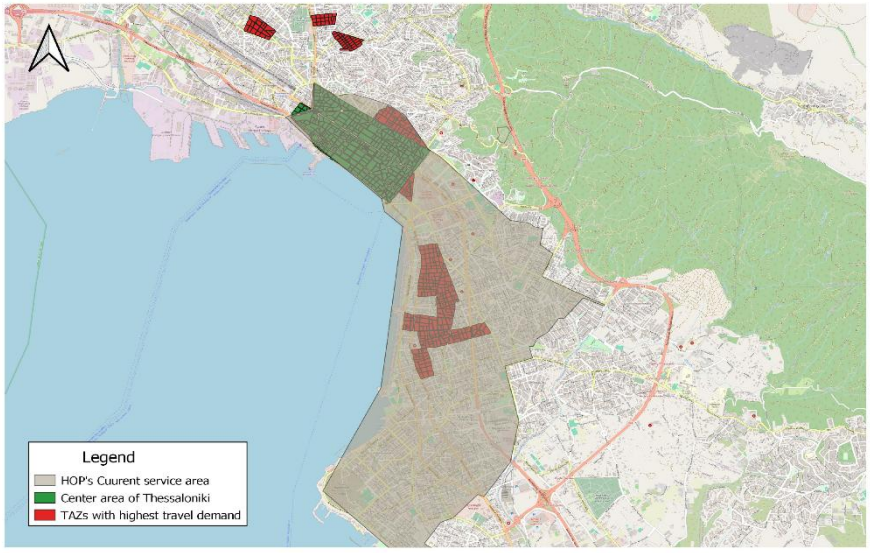

Fig 51. TAZs with the highest travel demand to/from the centre area of Thessaloniki and HOP's current operation zone.

The bus lines that connect the three selected TAZs with the centre area of Thessaloniki were identified. For each bus line the Level of Service (LoS) was calculated using data of Thessaloniki's Public Transport Authority (TheTA). The LoS of public transport is defined by the calculation of a set of indicators considering frequency, travel time etc. The indicators calculated in this analysis are the following: a) frequency during peak hours, b) hours of operation, c) percentage of completed vehicle kilometres to planned. The purpose of this step was to identify which of the selected TAZs are not sufficiently served by PT and thus could benefit from micromobility. According to the results (Fig 52):

TAZ 513 – LoS B and C

TAZ 516 – LoS C

TAZ 615 – LoS D

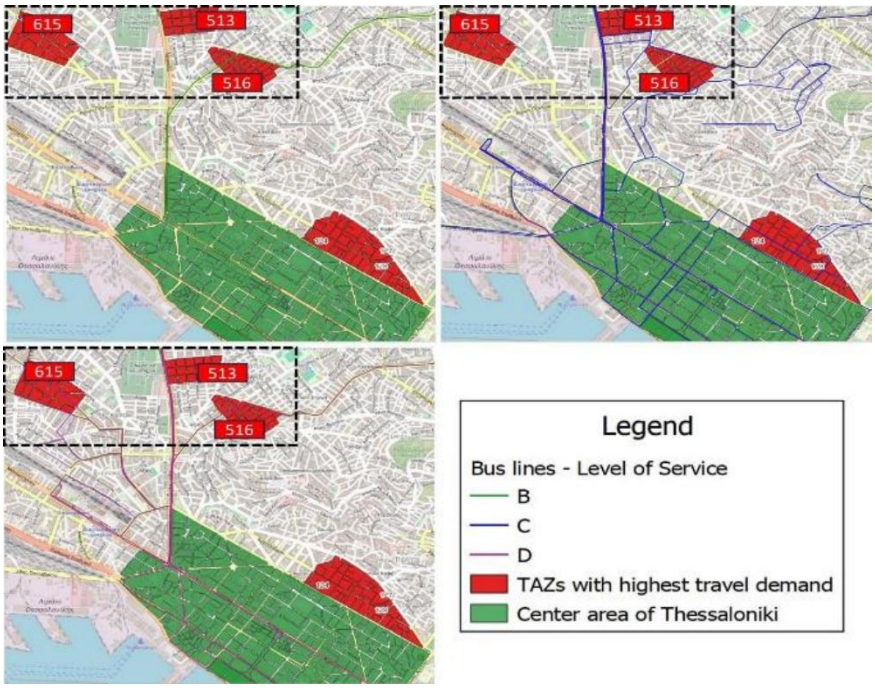

Fig 52. Level of service of bus lines that serve the selected TAZs outside HOP's operation zone (TAZ 513, 516 and 615).

The mean slopes of these three zones were checked to verify if travel by electric scooters is suitable; all TAZs are considered suitable for electric scooters. Also, for identifying whether the combination of PT with shared mobility forms a competitive alternative, the following were calculated:

Travel time by PT from each zone's centroid to three main points of interest (POIs): In order to calculate travel times, the average speed per bus line was used as provided by the TheTA. For each TAZ the bus line with the highest Level of Service was considered. The walking time to/from the nearest bus stop was estimated as well as the waiting time. The waiting time was calculated as half the frequency. If, for example, there is a bus every 20 minutes, the waiting time was estimated 10 minutes. The total travel time by 'PT' is the sum of walking time, waiting time as well as the bus travel time.

Travel time by scooter & PT from each zone's centroid to three POIs: A bus stop served by many bus lines and is within the average scooter travel distance (2500m) was selected. The bus stop selected is KOLOMVOU on Egnatia Street. Then the shortest path from the centroid of each zone to KOLOMVOU was calculated using the average scooter speed based on historic micromobility data that are available from the city of Thessaloniki (13.75km/h). Then the bus line with the highest Level of Service was selected and given the average speed, the time travel by bus was calculated. Waiting time as well as walking time from the nearest bus stop to the final destination were calculated. The waiting time was calculated as half the frequency. The total travel time by 'scooter & PT' is the sum of the above calculated times. It is worth noting that the time needed to find an available scooter is not considered and it is assumed that the user will find an e-scooter close to his/her place.

The results are presented in Table 14.

Table 14. Comparison of travel times by 'PT' and by 'scooter & PT' from selected TAZs to 3 POIs.

Origin	Destination	PT travel time (1)	Scooter + PT travel time (2)	Difference (2-1)

TAZ 513	POI 1 (Dikastiria)	19,3 min	10,6 min*	-8,7 min
	POI 2 (Agia Sofia)	26,3 min	24,2 min	-2,1 min
	POI 3 (Confex park)	33,7 min	32,7 min	-1 min
TAZ 516	POI 1 (Dikastiria)	16 min	8,8 min*	-7,2 min
	POI 2 (Agia Sofia)	20,2 min	22,2 min	+2 min
	POI 3 (Confex park)	35,5 min	30,9 min	-4,6 min
TAZ 615	POI 1 (Dikastiria)	24,3 min	9,9 min*	-14,4 min
	POI 2 (Agia Sofia)	33,4 min	24,9 min	-8,5 min
	POI 3 (Confex park)	47,4 min	33,3 min	-14,1 min

*Travel only by scooter - distance is less than 2500m.

Based on the calculated travel times and the LoS of bus lines, it is concluded that TAZ 615 has both high potential for the operators and contributes to enhanced accessibility. However, the other two TAZs can also have potential for the operators. Thus, two alternative extension scenarios were formed, one covering only TAZ 615 and one covering TAZs 615, 513 and 516. For both alternatives the area and population coverage was calculated (Table 15).

Table 15. Area and population served currently (by HOP) and after the two alternative extensions.

	Area	Population
Currently operating zone	15,932 km^2	364640
Extended operating zone a (TAZ 615)	16,229 km^2	373870
Extended operating zone b (TAZs 615, 513, 516)	16,905 km^2	398140

Neutral platform:

Architecture

The high-level platform architecture (Fig 53) consists of the frontend component, a Node.js backend component and a Python backend component for the algorithm.

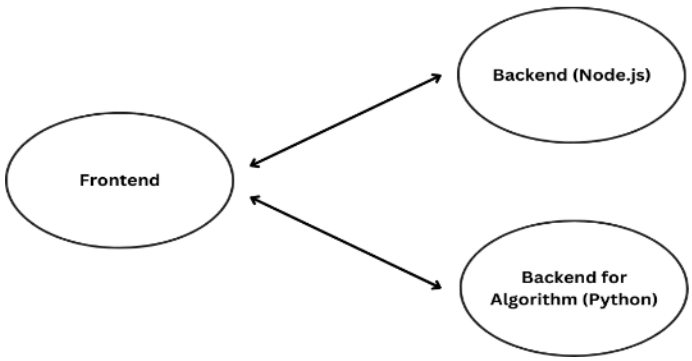

Fig 53. TES_08 high-level platform architecture.

Frontend: The frontend component is written in Angular[2] and provides all the functionalities needed in a dashboard view. So, the user has to make a registration (for the first time) and login in order to have access to the dashboard.

Backend (Node.js): It is the backend component (written in Node.js[3]) that serves the dashboard needs. For example, it is used for functionalities such as registration and login. For this reason, it is connected with a MySQL[4] database.

Backend for algorithm (Python): It is a backend component for the needs of the algorithm. Except for the algorithm code that is written in Python[5], Flask[6] is used in order to allow communication with the frontend component.

Inputs

The user has to fill a form with the algorithm inputs, so that the algorithm delivers the desired results. The inputs are the following ones:

Origin and destination coordinates: the neutral platform provide results at an OD level.

Traffic analysis zone: Traffic analysis zone (TAZ) file (in .geojson or .shp format) contains detailed geographical data and attributes used for analyzing travel demand. It outlines zone boundaries and includes information about the TAZ I, the coordinates of the zones and the name of the area.

OD matrix: The OD matrix file (in .xls or xlsx format) provides information about travel demand between specific OD pairs. It includes three columns; From, To, Trips.

Available micromobility stations: This station data file (in .geojson or .shp format) has the locations of the micromobility stations.

Unlocking cost of micromobility: The unlocking cost is the initial fee (in euros) a person has to pay to start using the micromobility vehicle.

Rental cost of micromobility per minute: Rental cost per minute is the fee charged for each minute that the micromobility vehicle is in use.

Available stations of shared cars: This station data file (in .geojson or .shp format) has the locations of the shared car stations.

Unlocking cost of shared cars: The unlocking cost is the initial fee (in euros) a person has to pay to start using the shared car.

Rental cost of shared cars per minute: Rental cost per minute is the fee charged for each minute that the shared car is in use.

Micromobility vehicles real-time availability: It is an API containing detailed information about dockless vehicles (both micromobility vehicles and shared cars). The details include unique vehicle IDs, type of vehicle and precise GPS coordinates.

Bus stations**:** Bus stations (in .geojson or .shp format) contain geographic data about the locations of bus stations.

---

[2] Angular: https://angular.dev

[3] Node.js: https://nodejs.org

[4] MySQL: https://www.mysql.com

[5] Python: https://www.python.org

[6] Flask: https://flask.palletsprojects.com

PT time schedule**:** GTFS files (in .txt format) contain information about the bus schedule including stop times, ID of the bus, routes, calendar, and frequency of the bus.

PT price for one-way tickets: This parameter is defined as the price of a one-way ticket, representing the cost for a single bus travel.

Average fuel price**:** The average price of fuel is measured in euros.

Average fuel consumption for cars: Fuel consumption (L/100 km) is the average consumption of a car.

Methodology

The platform includes six different transport options (micromobility, shared car, bus, walking, car, and multimodal) and calculates the optimal routes for each option. With regard to the routing data that are necessary for the platform, we use the API provided by OSRM[7] (Open Source Routing Machine).

Micromobility (bicycles, scooters): The user has to enter API (for dockless micromobility) and upload a file for micromobility stations. So, there are two types of vehicles that the algorithm takes into consideration (dock-based and dockless micromobility). The average speed is assumed equal to 15km/h and the user has to determine two parameters: the unlocking cost and the rental cost per minute. The algorithm initially identifies the nearest micromobility vehicle to the origin coordinates of the trip and calculates the walking distance that the person has to travel from the origin point. Then, it calculates the micromobility distance from the location of the nearest micromobility to the final destination and incorporates the cost parameters (unlocking cost and rental cost per minute) that the user has initially determined to calculate the outputs.

Shared car: The user enters an API (for dockless cars) and has to upload a file for shared car stations. The average speed is assumed equal to 30km/h and the user has to determine: the unlocking cost and the rental cost per minute. Based on these, the algorithm selects the nearest shared car to the origin coordinates of the trip which the person has to take by walking. So, the algorithm calculates the walking distance from the origin point to the nearest shared car. Then, the algorithm calculates the car distance from the location of the nearest shared car to the final destination and incorporates the cost parameters (unlocking cost and rental cost per minute) that the user has initially determined to calculate the outputs.

Bus: The average speed is assumed equal to 15km/h and the user has to determine the price of a one-way ticket. The user uploads a GTFS file and the bus routes files to provide the platform with accurate data on bus operations. The algorithm utilizes the uploaded files in order to identify the appropriate bus that the person should take. The person typically has to walk from the origin to the bus stop, travel with the bus to the bus stop which is nearest to the final destination and also has to walk from the bus stop to the final destination. If a second bus is required, the algorithm calculates the walking distance from the bus stop where the person got off to the nearest bus stop for the second bus. Then, when the person leaves the second bus, the algorithm calculates the walking distance to the final destination. So, the algorithm provides with the following information: which bus(es) the person needs to take to reach the destination, information about the bus stops, the distance the person will have to walk to the bus stop, the distance the person will have to walk after getting off the bus, either to catch a second bus or to reach the final destination.

Walking**:** Regarding walking, the user does not upload any files and it is assumed that the average speed is 4.5km/h. The algorithm calculates the optimal walking distance and time it will take to move from the origin coordinates to the destination coordinates.

Car: The average speed is assumed equal to 30km/h and the user has to determine the average fuel price and the fuel consumption for the car. The algorithm calculates the optimal car distance and time.

[7] OSRM: https://project-osrm.org

Multimodal: With regard to multimodal, the optimal solution is based on the minimum generalized cost. So, the algorithm has to calculate each combination separately. The currently examined multimodal combinations are: “bus-shared car”, “shared car-bus”, “micromobility-bus”, “bus-micromobility”, “shared car -micromobility”, “micromobility-shared car”. The algorithm compares all the combinations and selects the one with the lowest generalized cost. These combinations may include walking as well. The algorithm presents results only for this optimal solution.

Results

The results of the algorithm are:

- Generalized cost: Generalized cost (GC) is calculated separately for each transport mode based on specific parameters, such as the distance travelled, the time taken, the value of time (VOT) and costs parameters (unlocking cost and rental cost for micromobility and shared car, one-way ticket for bus, average fuel price and fuel consumption for car).
- Modal share: Modal share is also calculated for each transportation mode separately. Each value is between 0 and 1, while the sum of the modal shares must be equal to 1 (or 100%). The modal share is calculated based on the generalized cost and specifically using the following equation:

$$\text{MScar} = \frac{e^{-GCcar}}{(e^{-GCcar} + e^{-GCbus} + e^{-GCwalk} + e^{-GCmicro} + e^{-GCmulti} + e^{-GCshcar})}$$

- Number of daily trips: The number of daily trips is calculated specifically for micromobility, shared car, bus, and multimodal. It is derived from the Origin-Destination (OD) matrix file that the user uploads. This calculation involves multiplying the modal share of each transport solution by the daily travel demand between specific zones as indicated in the OD matrix.
- Daily income: Daily income depends on the number of daily trips. It is applicable exclusively to micromobility, shared car, bus, and multimodal (in which the daily income is calculated separately for each mode).

User Interfaces

Initially, the user has to fill in a registration form. After a successful registration, the user has to fill in the login form in order to have access to the platform. After the user’s login, the user views a dashboard that consists of the pages “Trip Planner” and “Account”. On the “Account” page, the user can view the account details or change password. For the needs of this document, we will present in detail only the “Trip Planner” page which is the page that includes the main functionalities of the platform.

On the “Trip Planner” page (Fig 54), the main part of the page consists of the form for the basic scenario, while on the right side there is a big map. The user selects the origin and the destination by adding two markers (blue for the origin and red for the destination) to the map. With regard to the form, the user has to upload files and enter values. The user inputs are presented in the following categories: Shared micromobility, Shared car, Public transport, Car, Travel demand, Traffic analysis, Shared mobility (dockless vehicles). The required inputs have the “*” symbol.

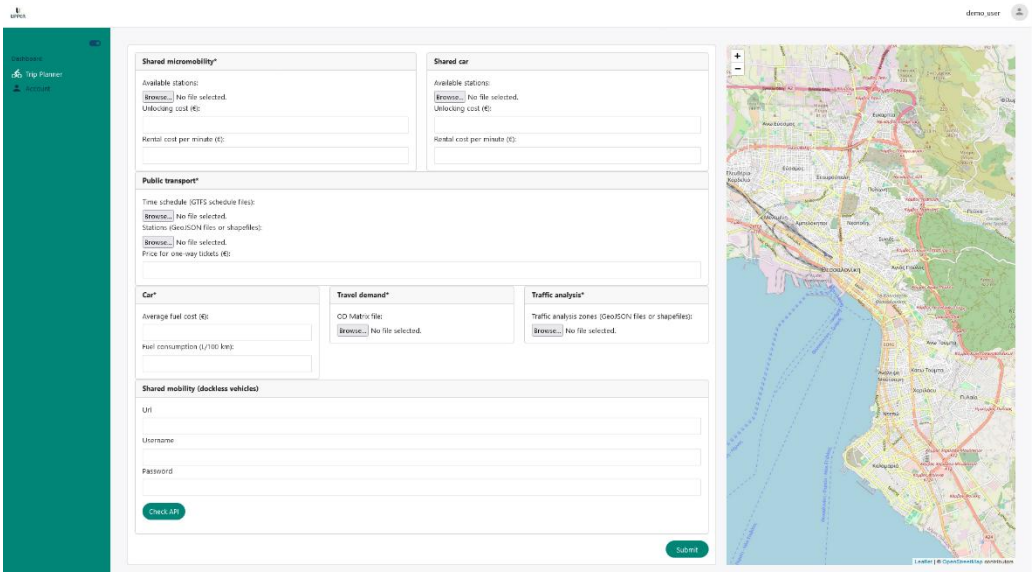

Fig 54. The form for the basic scenario.

After adding the Origin-Destination markers, filling in all the required fields and clicking on the “Submit” button, the results for the basic scenario are presented (Fig 55). There is a result for each supported transported mode: Micromobility, Shared car, Bus, Walking, Car and Multimodal. For each transport mode there is a card with results and a map. The map presents the optimal solution for each transport mode. Each route is depicted as a line. A map can have lines with different colours. For example, the map for Micromobility has a green and a purple line. As it is explained in the bottom part of the card, the green line symbolizes the walking route, while the purple line the micromobility route. There is information about the distance and the time for each route as well. Moreover, each card presents in its main part the following information: Generalized cost, Modal share, Number of daily trips, Daily income, Total distance, Total time.

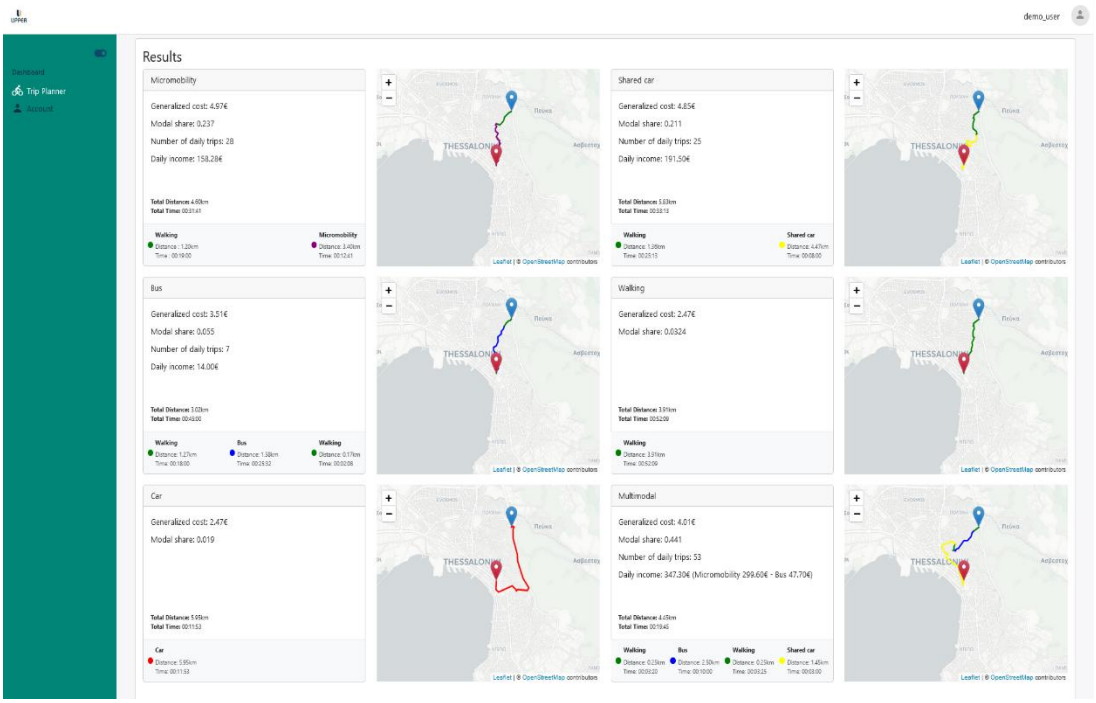

Fig 55. The results for the basic scenario.

Below the results there is the “Test scenario” button. After clicking on this button, a new window with a form about the test scenario is presented (Fig 56). With regard to the test scenario, the user can change only the inputs concerning Micromobility or Shared car. For each transport mode the user can upload a file with the available stations. After uploading the file with the stations, the multiselect component presents all the stations as checked (activated). The user can uncheck one or more stations in order to deactivate them. Moreover, the user can enter new values for the unlocking cost and the rental cost per minute. We suppose that the user makes the following changes in the test scenario:

The user adds Micromobility station “M1003”.

Micromobility unlocking cost changes from 0.90 to 1.00.

Micromobility rental cost per minute changes from 0.15 to 0.20.

The user deactivates Shared car station “C1002”.

Shared car unlocking cost changes from 1.00 to 1.10.

Shared car rental cost per minute changes from 0.20 to 0.25.

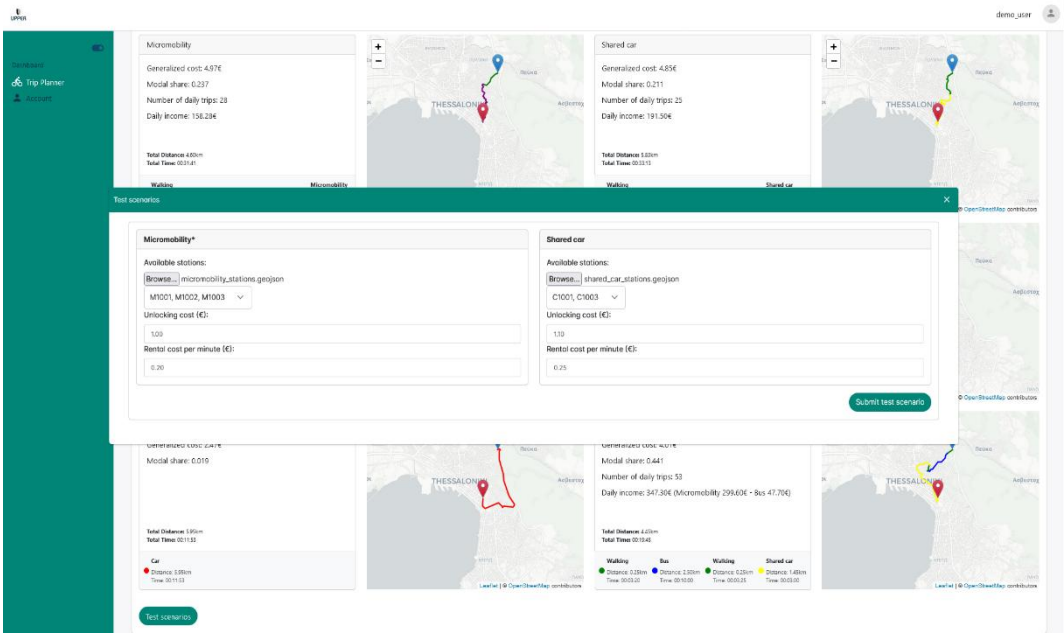

Fig 56. The form for the test scenario.

After clicking on the “Submit test scenario” button, the comparison results of Fig 57 are presented. On the left side there are the results of the basic scenario, while on the right side the results for the test scenario are presented. As the user can make changes in the test scenario only regarding Micromobility and Shared Car, there are three transport modes with possible changes in their routes: Micromobility, Shared car and Multimodal. So, the user can compare the results of the basic scenario with the results of the test scenario for Micromobility, Shared car and Multimodal.

It is out of scope to describe all the changes in the results. As an example, we could mention that the addition of a Micromobility station (“M1003”) reduced the walking distance and time in the Micromobility test scenario. The reason is that the algorithm found a closer micromobility vehicle (at the station added) to the origin. Furthermore, the deactivation of a Shared car station (“C1002”) increased the walking distance and time in the Shared car test scenario, as the distance between the origin and the closest shared car increased.

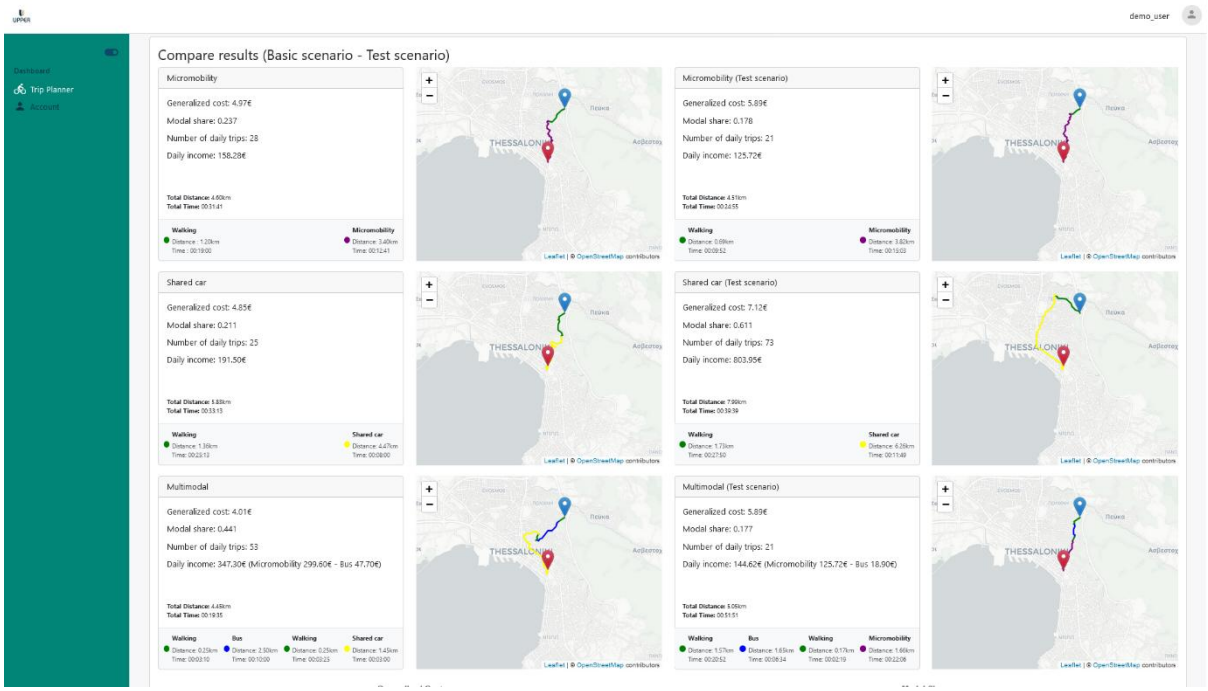

Fig 57. Comparison of results (basic scenario on the left side - test scenario on the right side).

Moreover, there are four bar charts for the comparison between the basic scenario and the test scenario (Fig 58). There is a bar chart for each output of the algorithm: Generalized cost, Modal share, Number of daily trips and Daily income. In each bar chart there are blue bars for the basic scenario and orange bars for the test scenario. For each transport mode there is the bar of the basic scenario next to the bar of the test scenario. So, the comparison is easy and helps the decision making.

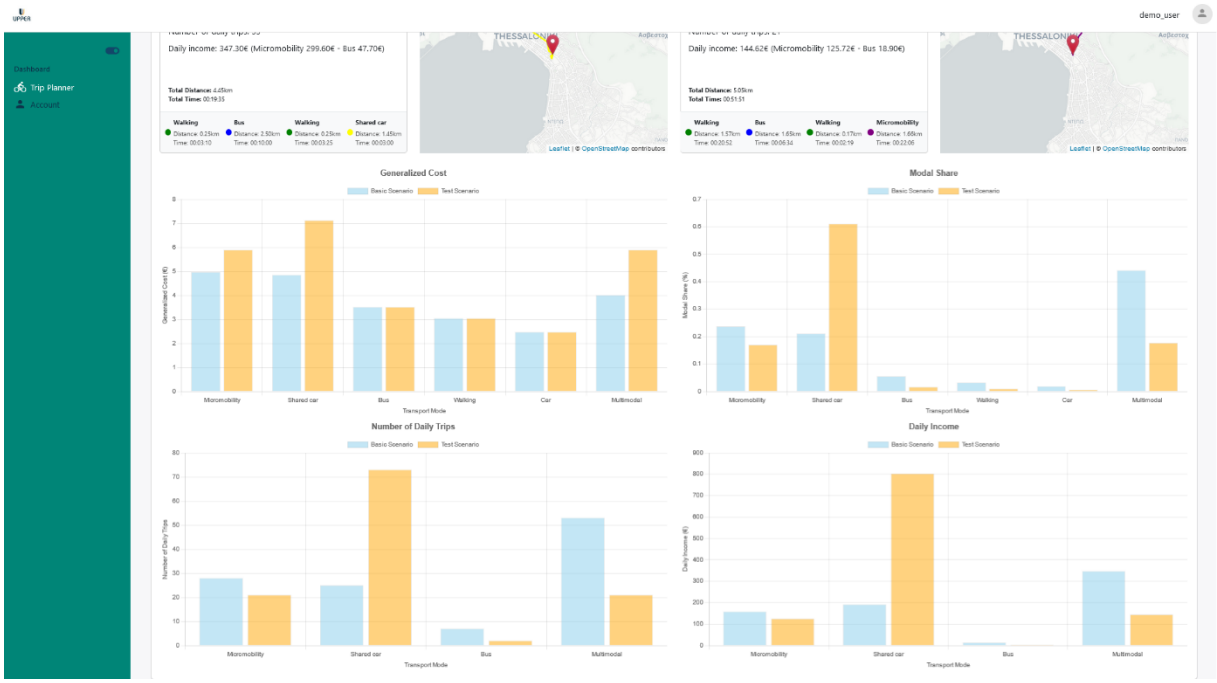

Fig 58. Comparative analysis of generalized cost, modal share, number of daily trips, and daily income for basic and test scenario across different transport modes.

Challenges & Mitigations

The challenges that were identified along with the respective solutions are the following.

Bus trip: One of the most challenging aspects of ensuring the algorithm functions correctly involves integrating various data sources, such as GTFS data, bus routes and Transportation Analysis Zone (TAZ) files. This integration is crucial for accurate and effective algorithm performance for the bus. Moreover, the calculations can become too complicated if the algorithm has to take any combination of buses into account. For this reason, the maximum number of different buses that the algorithm will examine needs to be determined in order to find the optimal bus trip solution. With regards to Thessaloniki, a person can move from east to west with a maximum combination of three different buses. So, the algorithm should examine the combination of two or three buses maximum.

Multimodal trip: The integration of multiple transportation modes in a single journey introduces significant complexity to the calculation of generalized costs and modal shares. Unlike unimodal transportation, multimodal journeys require the combination of different cost structures, travel times, and value of time parameters for each mode. This necessitates a sophisticated algorithm capable of accurately calculating and optimizing these combined metrics. For these reasons, we have determined which combinations of multimodal trips should be examined by the algorithm, based on relevant research.

Generic: An important requirement was that the platform code should be as generic as possible. The aim is to use the same platform for scenarios of other cities, except for Thessaloniki, in the future. So, the platform could be used for other cities, by making small changes in the code or even without any changes needed. For this purpose, the user uploads all the necessary files and enters the API about the dockless vehicles.

Next steps towards implementation

The next steps for TES_08 are the following:

- The results of the accessibility analysis have been already presented to the micromobility operator (HOP) and there are currently discussions regarding the possibility of extending their operation area.
- Internal testing of the neutral platform will continue for identifying possible bugs and fixing them. Real big datasets that cover the whole city of Thessaloniki will be used in this process.
- The neutral platform will be presented both to local authorities and micromobility providers for getting their feedback and optimizing the user experience based on it.
- The bus trip algorithm currently provides solutions that combine the use of maximum two buses, but it will support the combination of three buses in the future. Moreover, the platform will take waiting time at bus stations into consideration as well.
- The algorithm will take more possible multimodal solutions into account in order to find the optimal multimodal trip. For example, the combination “micromobility-bus-micromobility” will be added to the possible multimodal solutions.