D6.1 IMHOTEP Simulation and Decision Support Toolset

Deliverable ID:	D6.1
Dissemination Level:	PU
Project Acronym:	IMHOTEP
Grant:	891287
Call:	H2020-SESAR-2019-2
Topic:	SESAR-ER4-13-2019
Consortium Coordinator:	Nommon
Edition date:	15 June 2022
Edition:	01.00.00
Template Edition:	02.00.05

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Authoring & Approval

Authors of the document

Name / Beneficiary	Position / Title	Date
Juan Blasco Puyuelo (Nommon)	WP6 Leader	24/05/2022
Carmelo Alcaraz (Nommon)	Project Team	24/05/2022
Carolina Azevedo (Nommon)	Project Team	24/05/2022

Reviewers internal to the project

Name / Beneficiary	Position / Title	Date
Juan Blasco Puyuelo (Nommon)	WP6 Leader	31/05/2022
Oliva García Cantú (Nommon)	Project Team	31/05/2022
David Mocholí (Nommon)	Project Team	31/05/2022

Reviewers external to the project

Name / Beneficiary	Position / Title	Date
--	--	--

Approved for submission to the SJU By - Representatives of all beneficiaries involved in the project

Name / Beneficiary	Position / Title	Date
Ricardo Herranz (Nommon)	Project Coordinator	15/06/2022

Rejected By - Representatives of beneficiaries involved in the project

Name and/or Beneficiary	Position / Title	Date
--	--	--

Document History

Edition	Date	Status	Name / Beneficiary	Justification
00.00.01	24/05/2022	Draft	Juan Blasco Puyuelo	Initial draft for internal review
00.01.00	15/06/2022	Submitted to the SJU for approval	Juan Blasco Puyuelo	Internal approval for submission to the SJU
01.00.00	22/08/2022	Approved	Juan Blasco Puyuelo	Approved by the SJU

Copyright Statement

© 2022 – IMHOTEP Consortium. All rights reserved. Licensed to SESAR3 Joint Undertaking under conditions.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

IMHOTEP

INTEGRATED MULTIMODAL AIRPORT OPERATIONS FOR EFFICIENT PASSENGER FLOW MANAGEMENT

This document is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 891287 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document describes IMHOTEP Simulation and Decision Support Toolset. The document presents the integration workflow of the different simulation models developed by IMHOTEP to predict the evolution of airport passenger flows and describes the visualisation and decision support tool developed by the project, which will enables the user to visualise a set of selected KPIs for assessing the operational impact of different passenger flow management measures.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Table of Contents

EUROPEAN PARTNERSHIP

Co-funded by the European Union

List of Tables

List of Figures

EUROPEAN PARTNERSHIP

Co-funded by the European Union

EUROPEAN PARTNERSHIP

Co-funded by the European Union

List of Acronyms

Acronym	Meaning
ATD	Activity Travel Diaries
A-CDM	Airport Collaborative Decision-Making
CSV	Comma-Separated Values
CSS	Cascading Style Sheets
EIBT	Estimated In-Block Time
EU	European Union
HTML	HyperText Markup Language
HTTP/S	Hypertext Transfer Protocol / Secure
IMH	IMHOTEP
IMHOTEP	Integrated Multimodal Airport Operations for Efficient Passenger Flow Management
KPI	Key Performance Indicator
LCY	London City airport
ORDBMS	Object Relational Database Management System
ORM	Object-Relational Mapping
PMI	Palma de Mallorca airport
PT	Public Transport
REQ	Requirement
SIBT	Scheduled In-Block Time
SOBT	Scheduled Off-Block Time
SQL	Structured Query Language
TCP	Transmission Control Protocol
UC	Use Case
UK	United Kingdom
WP	Work Package
XML	Extensible Markup Language

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Executive summary

The main goal of IMHOTEP is to develop and evaluate the benefits of a concept of operations for the extension of airport collaborative decision-making to ground transport stakeholders. To support this goal, a set of data analysis methods and predictive models and decision support to have been developed and integrated into a decision support toolset that enables information sharing, common situational awareness and real-time collaborative decision-making between airports and ground transport stakeholders.

The development of IMHOTEP decision support toolset has comprised three main steps:

1. **Model integration**. IMHOTEP has developed two predictive models, which address the evolution of passenger flows in the ground access system and within the airport. These two models have been integrated to provide a holistic view of the whole passenger journey. Real-time integration of the models has been emulated offline. To demonstrate the capabilities of the integrated model, a set of scenarios will be simulated and stored in a database so they can later be accessed by the visualisation and decision support tool.

2. **KPI implementation**. Once the models have been integrated, we have implemented the calculation of the set of KPIs on which the proposed collaborative decision-making process is based. The KPIs initially defined in the IMHOTEP ConOps have been refined according to the needs of the different stakeholders involved and to the actual implementation of the predictive models.

3. **Development of IMHOTEP visualisation and decision support tool**. Once the model integration and the KPI calculation processes have been implemented, a prototype of the visualisation and decision support tool has been developed. This tool enables the user to visualise the different KPIs in order to analyse the expected performance in the short-term and assess the operational impact of different management measures.

The prototype decision support tool will enable the validation of the IMHOTEP ConOps. Prior to conducting the proposed validation exercises, the prototype tool has been presented in an internal workshop with the IMHOTEP partner airports and other stakeholders, in order to gather their feedback on how to improve the usability of the tool. This feedback will be integrated during the preparation of the validation exercises foreseen for the final stage of the project.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

1 Introduction

1.1 Scope and objectives

The general goal of IMHOTEP is to develop a concept of operations and a set of data analysis methods, predictive models and decision support tools that enable information sharing, common situational awareness and real-time collaborative decision-making between airports and ground transport stakeholders.

In this context, the activities of WP6 are mainly focused on the integration of the simulation models developed within WP4 and WP5 into a software tool that supports collaborative decision-making between airports and ground transport stakeholders based on a shared view of the full passenger trajectory. Within this view, the specific objectives to be completed are:

- To develop the integration and communication framework between the different simulation models developed in WP4 and WP5 so that the flow of outgoing information from each model can feed the other.
- To implement the set of Key Performance Indicators (KPIs) that will be used to capture different aspects of the passenger journey and evaluate the impact of passenger flow traffic management measures.
- To develop a prototype dashboard that enables the visualisation of the KPIs according to the different views relevant for the different stakeholders involved in the collaborative decision-making process.

1.2 Applicable documents

- Grant Agreement No 891287 IMHOTEP – Annex 1 Description of the Action.
- IMHOTEP D1.1 Project Management Plan, v01.01.00, March 2021.
- IMHOTEP D2.1 IMHOTEP ConOps, v02.00.00, February 2021.
- IMHOTEP D2.2 Specification of Case Studies v01.01.00, July 2021.
- IMHOTEP D3.1 Data Management Plan, v01.01.00, March 2021.
- IMHOTEP D4.1 Passenger Terminal Process Simulation Module, v01.00.00, December 2021.
- IMHOTEP D5.1 Airport Access and Egress Simulation Module, v01.00.00, December 2021.

1.3 Structure of the document

The rest of this document is structured as follows:

- Chapter 2 presents an overview of the general project concept and outlines the approach followed for its implementation, including the integration between the models and the communication with the decision support tool.
- Chapter 3 describes the approach for integrating the terminal and the access/egress models, detailing the data used and the flow of information between the models.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- Chapter 4 provides an overview of the KPIs implemented on the models and describes how the information will be presented to the final user through the visualisation tool.
- Chapter 5 describes the development of the visualisation tool, detailing the requirements of the system, the software design, the user manual, and the test and validations performed.
- Chapters 6 summarises the main conclusions obtained from the integration of the models and the development of the visualisation and decision support tool, discusses their further utilisation within the IMHOTEP project and provides a set of recommendations for the future improvement.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

2 General approach

2.1 General concept

IMHOTEP's approach is based on a passenger-centric view where the passenger journey becomes the backbone of a common and comprehensive situational awareness of the status of the airport processes, the ground transport system and the door-to-door passenger flows. This passenger-centric concept is sketched in Figure 1:

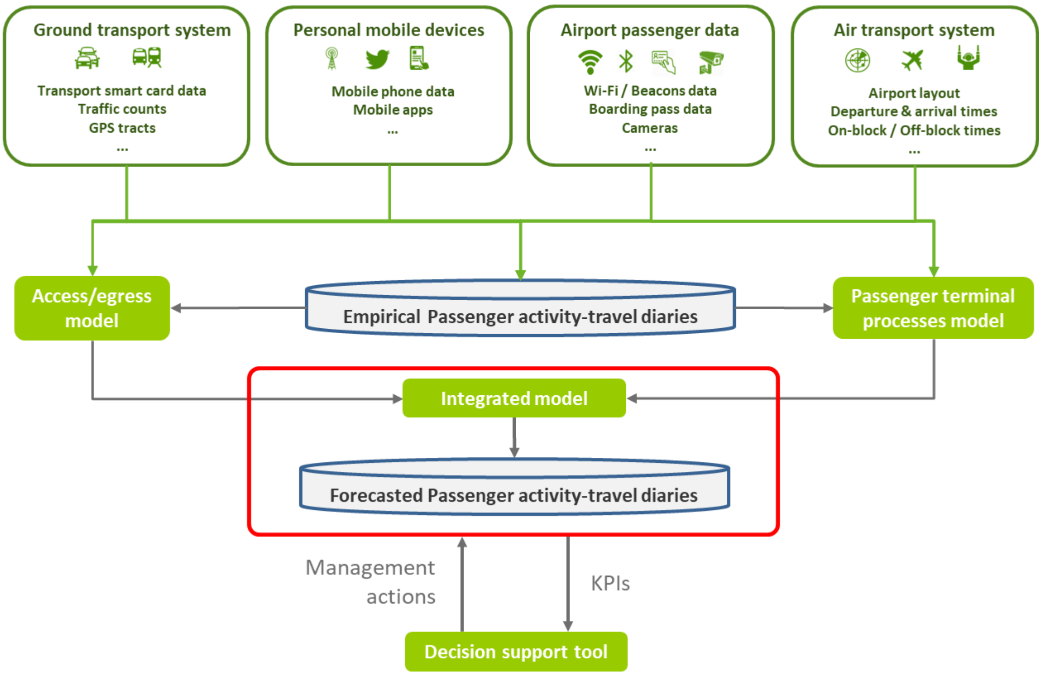

Figure 1. IMHOTEP concept

The implementation of the proposed concept entails the following steps:

1. The empirical Activity-Travel-Diaries (ATD) are reconstructed using historical data from relevant and heterogeneous data sources available during the passenger journey through different data fusion and analysis methods.
2. The empirical ATD as well as other available data sources are used to develop and calibrate a set of predictive models able to short-term forecast the passenger flows within the terminal and on the surface access and egress legs.
3. Once the models of the passenger terminal flows and the access and egress flows are developed, they are integrated in order to provide a comprehensive view of the full passenger journey. The integration process involves two main activities: (i) the development of the required interfaces so that each model can take as inputs some of the outputs produced by the other model, and (ii) the implementation of the KPIs on which the collaborative decisions will be based.
4. Finally, a newly-developed visualisation and decision support tool enables the users to interact with the integrated simulation model and perform what-if analyses, helping the user to decide the best course of action.

The methodologies and techniques developed to implement the concepts described in steps 1 and 2 are described in deliverables 'D4.1 Passenger Terminal Processes Simulation Module' and 'D5.1 Airport Access and Egress Simulation Module'. Hereinafter the discussion will be focused on steps 3 and 4.

2.2 Implementation approach

According to the IMHOTEP concept, the integrated model and the decision support tool are seamlessly integrated, meaning that the modification of a parameter in the decision support tool will trigger a simulation in both the terminal and the access and egress model, providing the new KPI values. This process is depicted in Figure 2 and involves the following steps:

1. Initially, the set of departing and arriving passengers using the airport, which are an input to both the access/egress and the terminal model, is generated. This set of passengers contains all the information required for the models to run, e.g., flight number, airport of origin /destination, origin/destination of the trip to/from the airport, purpose of the trip, nationality.
2. Once the information is fed to the models, the simulations are performed and the following outputs are produced:
    - Departing passengers' arrival time to the airport, in the case of the access/egress model.
    - Arriving passengers' airport exit time, in case of the terminal model.
    - Traffic information and terminal processes from which to extract the KPIs on which the collaborative decision-making process are based.
3. The models are again fed with the outputs calculated in the previous step. In this case, the departing passengers' arrival time to the airport is an input for the terminal model, while the arriving passengers' airport exit time information is an input for the access/egress models. The simulations are then re-run providing outputs for the other direction of the passenger flow. Because we have two different models and the passenger flow affects these models in both directions, this step is executed iteratively, which means that simulations need to be performed until both models converge to a solution.
4. Once a solution is reached, the outputs are postprocessed and the KPIs are extracted. These KPIs are presented to end users through a visualisation tool that enables the evaluation of their evolution in the short term.
5. The final user can use the visualisation and decision tool to simulate different what-if scenarios to better understand the performance impact of several management decisions and choose the best course of action. This functionality is carried out through the following process: (i) the management actions are translated into modifications of the model inputs (such as the number of open lanes at the security control or the frequency of the different public transport options); (ii) new simulations are executed until the process converges; (iii) finally, the new results are displayed in the dashboard.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

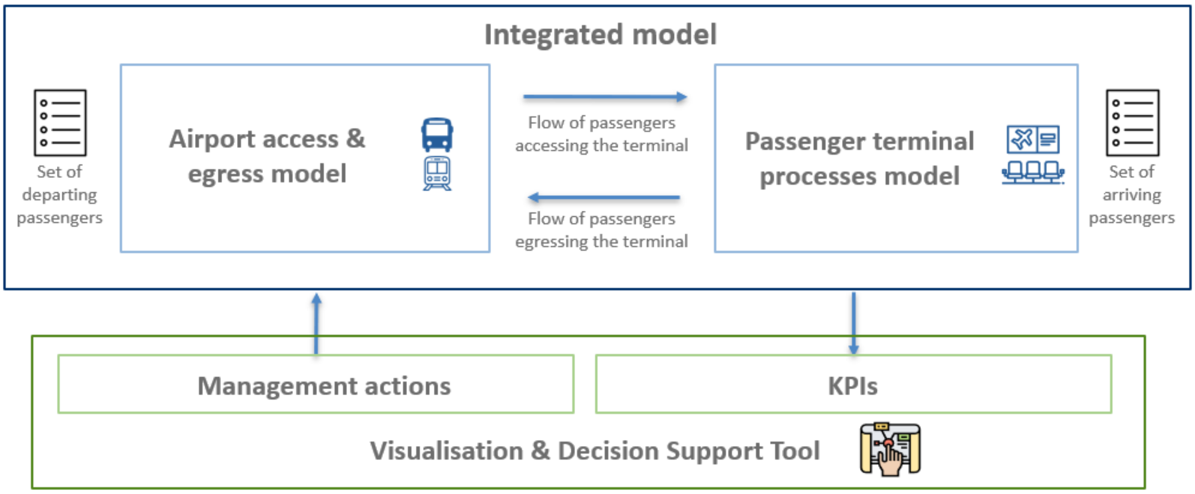

Figure 2. Implementation approach

2.3 Prototype implementation

The implementation approach presented in the previous section envisions a seamless integration between the simulation models with the decision support tool. However, when applying this approach within the project, a series of limitations had to be tackled that forced the implementation approach to be slightly adapted:

- The simulation time of the models is too high. To run the models in real time, an optimisation of the computational performance of the solution would be required.
- The integration of both models requires an iterative process. Although the flow of passengers in both directions (arriving vs departing passengers) within the terminal do not significantly affect each other, this has a relevant effect on the surface transport where an event or a modification of some input in one direction is affecting the other direction. This implies that the simulations need to be performed in an iterative way until the results converge.

With this in mind and aiming at proving the validity of the tool, a simplified approach has been implemented for this first prototype. The key differences with respect to the initial implementation approach are the following:

- The integrated model and the decision support tool are not directly connected. Instead, an intermediate layer has been included between both in order to store the results obtained by the integrated model. This implies that the number of scenarios that the user can visualise is limited and must be simulated previously.
- The implemented approach does not consider an iterative simulation process between both models; instead it has been assumed that both directions (arriving and departing passengers) are completely independent.

The prototype implementation process is depicted in Figure 3, which includes a database as an intermediate layer for storing the already calculated scenarios. The adapted implementation starts from the first two steps of the original process (see section 2.2) and then adds the necessary simplifications, as listed below:

EUROPEAN PARTNERSHIP
Co-funded by the European Union

1. The set of departing and arriving passengers containing all the necessary information is generated.
2. Once the information is fed to the models, the simulations are performed and, analogously to the original process, the required outputs are produced (see section 2.2).
3. The values for the departing passengers' arrival time to the airport and the arriving passengers' airport exit time are sent to the models in order to run again the simulations and provide the outputs for the other direction of the passenger flows.
4. The outputs of the models are postprocessed and the selected KPIs are extracted and stored in a database.
5. The visualisation and decision support tool accesses the database and displays the stored KPIs to the final users.

Steps 1 to 4 are repeated for all the different management actions to be evaluated in the case studies. In this case, these management decisions are defined in advance and after running the corresponding simulations their results are stored in the database so we can later visualise them through the decision support tool.

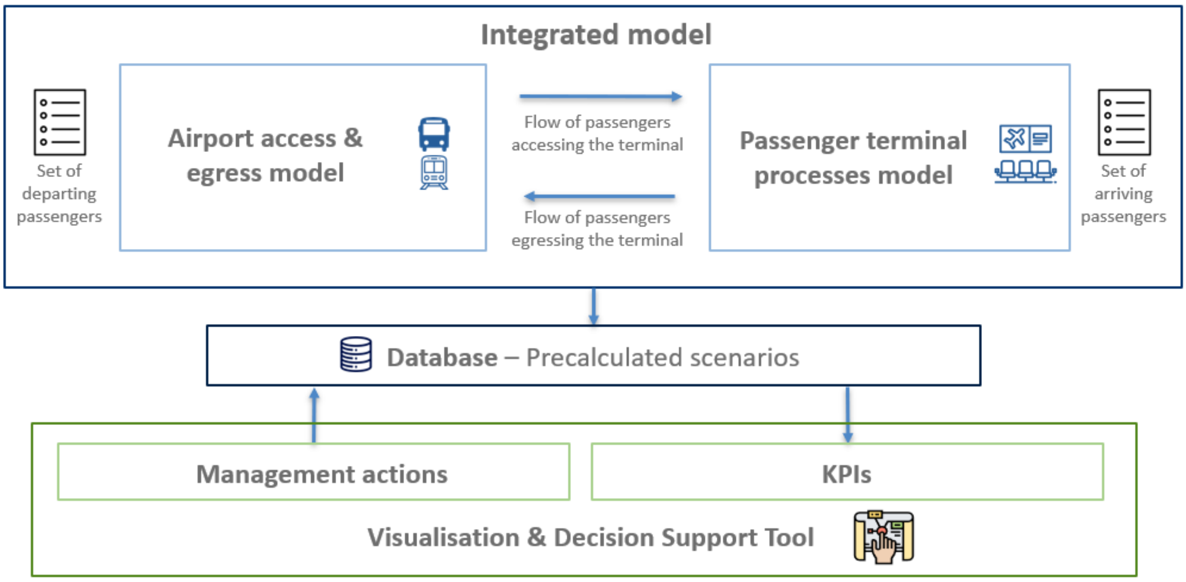

Figure 3. Final implemented approach

EUROPEAN PARTNERSHIP
Co-funded by the European Union

3 Model integration

The integration process involves the exchange of a series of files as well as different merging and formatting processes. This section focuses on the integration of both models, emphasising two main points: the flow of information between the models and the formats of the data files necessary for the models (along with the information included in each of them).

3.1 Integration workflow

As the integration procedures are slightly different for arriving and departing passengers, in the subsequent subsections each process is detailed separately.

3.1.1 The case of departing passengers

For departing passengers, the flow of information is depicted in Figure 4 and described below.

1. The input for the access/egress model is generated from the data of the airport users contained in the ATD. This file includes all the information available on the different types of airport users (passengers, workers, professional drivers and other types of short-term visitors) who access the terminal and the information required by both models to correctly run the simulations (the information contained in this file is further described in Section 3.2.1).

2. The access/egress model simulates the airport users access to the airport and provides the arrival time to the airport terminal. The model output file is similar to the input file but with the additional information of arrival time at the terminal.

3. A formatting process is performed in order to prepare the output file from the access/egress model to a suitable format readable by the terminal model (the information contained in this file is further described in Section 3.2.2). It should be noted that for the terminal model only passengers will be considered, as the model cannot simulate other types of airport visitors, which are in any case less relevant than in the access leg.

4. The terminal model simulates the passengers' trajectory within the airport terminal. The outcome of the process is a detailed terminal itinerary for all the passengers departing from the airport of study.

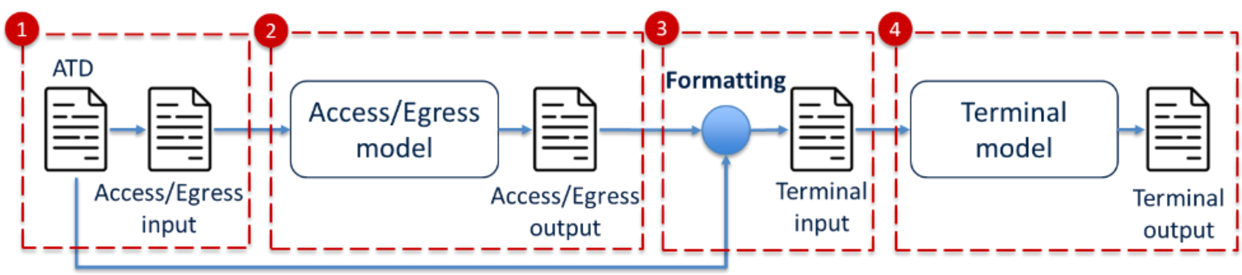

Figure 4. Flow of information for simulating passengers' door-to-gate journey

EUROPEAN PARTNERSHIP
Co-funded by the European Union

3.1.2 The case of arriving passengers

The case of arriving passengers is slightly different. Here, the terminal model creates the passengers based on flights and not directly on passenger flow. Therefore, the workflow needs to be modified. The resulting process is depicted in Figure 5 and described below.

1. The terminal model generates the passengers from the flight schedule file and simulates the passengers exit itinerary until the airport kerb, thus providing the flow of passengers exiting the airport terminal.

2. The terminal model output information is formatted so that it can be sent to the access/egress model. This output contains the passengers' exit time from the terminal and some information related to the flight (such as flight number, gate, airport of origin, etc.). However, some relevant data required by the access and egress model is missing (such as egress mode, final destination, etc.), as well as information on the rest of airport users. This information is complemented with the list of airport users contained in the ATD and later transformed into the appropriate format.

3. The access/egress model is executed, generating the simulated egress leg of the airport users and providing as outcome of the process the final arrival time to the users' destination.

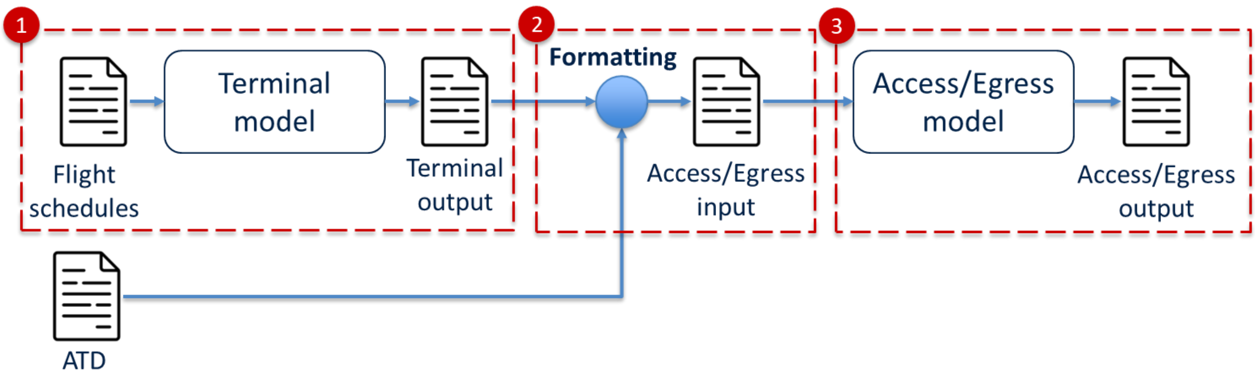

Figure 5. Flow of information for simulating passengers' gate-to-door journey

3.2 Main integration files

The integration workflow described in the previous section is based on a common view of passenger flows. To achieve such integration between the models, it is necessary to track individual passengers across both the access/egress model and the terminal model. The piece that enables this is the Activity-Travel-Diaries (ATD) file.

The ATDs have been constructed in earlier stages of the project (for further information on the development of the ATD, please refer to deliverables 'D4.1 Passenger Terminal Processes Simulation Module' and 'D5.1 Airport Access and Egress Simulation Module') and contain all the information required about the airport users, including a unique airport user identifier that enables the tracking of the users along both models. This information provides the necessary inputs for each model so that all passenger information is properly transmitted through the simulation workflow.

The next subsections describe the information contained in each of the input files as well as their format.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3.2.1 Access and egress model input

To create the necessary input file for the access/egress model, the information contained in the ATD is transformed into a 'csv' file containing the following fields:

- **id:** user identifier
- **date:** study date
- **user_type:** type of airport visitor (passenger, worker, professional driver or other kind of short-term visitor)
- **od_location:** origin/destination of the ground transport leg of the trip, depending if the passenger is departing or arriving
- **trip_init_time:** starting time of the ground transport leg of the trip (from the od_location to the airport in case of departing passengers; and from the airport to the od_location in case of arriving passengers)
- **mode:** ground transport mode used
- **scheduled_flight_date:** Scheduled Off-Block Time (SOBT)
- **flight_number:** flight number, including airline (e.g., RYR7965)
- **gate:** airport gate allocated
- **od_airport:** airport of origin/destination depending if the passenger is arriving/departing
- **flight_type:** od_airport grouped into the categories: Inter-islands, Domestic, Schengen, No Schengen or International
- **residence:** place of residence of the user
- **age:** age of the user
- **gender:** gender of the user
- **purpose:** purpose of the trip (business vs leisure)
- **stay_duration:** duration of the passenger trip in days (days at destination)
- **group_id:** group identifier
- **group_size:** total number of passengers travelling together
- **entry_point_id**: identifier of the point of entry/exit to/from the access and egress model.
- **entry_time:** arrival time at entry_point_id (just for departing passengers)
- **external_travel_time:** travel time from the origin of the ground trip to the access and egress model boundaries (or the other way round in the case of arriving passengers)
- **bags:** number of checked-in bags
- **check_in_type:** type of check-in method (online, kiosk or counter) used by the passenger (just for departing passengers)

This information is later enriched by the access and egress model by adding the following additional fields:

- **arrival_time**: user arrival time to the destination of their ground transport leg of the trip (airport for departing passengers, entry_point_id for arriving passengers)

id	date	user_type	hinterland_location	trip_init_time	mode	scheduled_flight_date	flight_number	gate	od_airport	flight_type	residence	age	gender	purpose	stay_duration	group_id	group_size	entry_point_id	entry_time	external_travel_time	bags	check_in_type
20190803_outbound_1	20190803	pax	704203004	2019-08-03T08:24:50	Rental car	2019-08-03T11:45	RYR2327	A22	LBA	No Schengen	UK	None	None	leisure	8	4763	2	533399	2019-08-03T09:06:12	2638,2	0	Online
20190803_outbound_2	20190803	pax	704203004	2019-08-03T08:24:50	Rental car	2019-08-03T11:45	RYR2327	A22	LBA	No Schengen	UK	None	None	leisure	8	4763	2	533399	2019-08-03T09:06:12	2638,2	0	Counter
20190803_outbound_3	20190803	pax	701105006	2019-08-03T10:56:01	Rental car	2019-08-03T12:25	TUI2715	D93	STR	Schengen	Rest of Europe	None	None	leisure	5	35761	2	533401	2019-08-03T10:40:24	937,9	0	Counter
20190803_outbound_4	20190803	pax	701105006	2019-08-03T10:56:01	Rental car	2019-08-03T12:25	TUI2715	D93	STR	Schengen	Rest of Europe	None	None	leisure	5	35761	2	533401	2019-08-03T10:40:24	937,9	0	Online
20190803_outbound_5	20190803	pax	703305002	2019-08-03T15:39:56	Private bus	2019-08-03T20:10	EZY6896	A18	GLA	No Schengen	France	None	None	leisure	2	28976	2	533399	2019-08-03T16:27:59	3492,5	0	Online
20190803_outbound_6	20190803	pax	703305002	2019-08-03T15:39:56	Private bus	2019-08-03T20:10	EZY6896	A18	GLA	No Schengen	France	None	None	leisure	2	28976	2	533399	2019-08-03T16:27:59	3492,5	0	Counter
20190803_outbound_7	20190803	pax	705102002	2019-08-03T07:00:24	Rental car	2019-08-03T10:00	EXS190	A26	GLA	No Schengen	Ireland	None	None	leisure	14	16010	1	533399	2019-08-03T07:53:29	3806,5	1	Online
20190803_outbound_8	20190803	pax	700302001	2019-08-03T19:40:34	Rental car	2019-08-03T20:50	IBS3975	D86	MAD	Domestic	Spain	25-44	Female	leisure	5	12673	1	533389	2019-08-03T18:53:22	2832,5	0	Online
20190803_outbound_9	20190803	pax	700302001	2019-08-03T19:40:34	Rental car	2019-08-03T20:50	IBS3975	D86	MAD	Domestic	Spain	25-44	Female	leisure	5	12674	2	533389	2019-08-03T18:53:22	2832,5	0	Online
20190803_outbound_10	20190803	pax	700302001	2019-08-03T19:40:34	Rental car	2019-08-03T20:50	IBS3975	D86	MAD	Domestic	Spain	25-44	Female	leisure	5	12674	2	533389	2019-08-03T18:53:22	2832,5	0	Online
20190803_outbound_11	20190803	pax	700302001	2019-08-03T19:40:34	Rental car	2019-08-03T20:50	IBS3975	D86	MAD	Domestic	Spain	25-44	Female	leisure	5	12675	2	533389	2019-08-03T18:53:22	2832,5	0	Online
20190803_outbound_12	20190803	pax	703104009	2019-08-03T03:37:06	Private bus	2019-08-03T05:25	EWG581	C54	CGN	Schengen	Netherlands	None	None	leisure	7	19830	2	533425	2019-08-03T03:44:06	1160,2	0	Kiosk
20190803_outbound_13	20190803	pax	703104009	2019-08-03T03:37:06	Private bus	2019-08-03T05:25	EWG581	C54	CGN	Schengen	Netherlands	None	None	leisure	7	19830	2	533425	2019-08-03T03:44:06	1160,2	0	Online
20190803_outbound_14	20190803	pax	703104009	2019-08-03T03:37:06	Private bus	2019-08-03T05:25	EWG581	C54	CGN	Schengen	Netherlands	None	None	leisure	7	19831	4	533425	2019-08-03T03:44:06	1160,2	0	Online
20190803_outbound_15	20190803	pax	703104009	2019-08-03T03:37:06	Private bus	2019-08-03T05:25	EWG581	C54	CGN	Schengen	Netherlands	None	None	leisure	7	19831	4	533425	2019-08-03T03:44:06	1160,2	1	Kiosk
20190803_outbound_16	20190803	pax	703104009	2019-08-03T03:37:06	Private bus	2019-08-03T05:25	EWG581	C54	CGN	Schengen	Netherlands	None	None	leisure	7	19831	4	533425	2019-08-03T03:44:06	1160,2	1	Online
20190803_outbound_17	20190803	pax	703104009	2019-08-03T03:37:06	Private bus	2019-08-03T05:25	EWG581	C54	CGN	Schengen	Netherlands	None	None	leisure	7	19831	4	533425	2019-08-03T03:44:06	1160,2	0	Counter
20190803_outbound_18	20190803	pax	700302001	2019-08-03T07:46:04	Rental car	2019-08-03T11:10	TOM1675	A26	EDI	No Schengen	UK	None	None	leisure	10	22604	3	533389	2019-08-03T08:20:08	2538,6	0	Counter
20190803_outbound_19	20190803	pax	700302001	2019-08-03T07:46:04	Rental car	2019-08-03T11:10	TOM1675	A26	EDI	No Schengen	UK	None	None	leisure	10	22604	3	533389	2019-08-03T08:20:08	2538,6	0	Online

Figure 6. Access and egress model input file template

3.2.2 Terminal model input

In order for the terminal model to be able to process the information on passenger flows, the data contained in the ATD and the output data from the access/egress model are integrated into an 'xml' file. Figure 7 shows the 'xml' template of the input file for the terminal model.

```
<object FirstObject="" LastObject="">
<optimed time="">
  <newobject classname="" name="">
    <prop name="Age" val=""/>
    <prop name="AirlineCode" val=""/>
    <prop name="AirportCode" val=""/>
    <prop name="Bags" val=""/>
    <prop name="CheckInType" val=""/>
    <prop name="EntryPoint" val=""/>
    <prop name="ExitPoint" val=""/>
    <prop name="FlightNumber" val=""/>
    <prop name="Gender" val=""/>
    <prop name="Nationality" val=""/>
    <prop name="Passport" val=""/>
    <prop name="PaxType" val=""/>
    <prop name="SchengenStatus" val=""/>
    <prop name="TSchedule" val=""/>
    <prop name="TravelClass" val=""/>
    <prop name="TravelPurpose" val=""/>
  </newobject>
</optimed>
</object>
```

Figure 7. Terminal model input file template

The meaning of the different fields is described below:

- **FirstObject**: date and time of first object (passenger) created
- **LastObject**: date and time of the last object (passenger) created
- **time**: object creation time (passenger entry time to the airport terminal)
- **name**: passenger identifier
- **Age**: age of the passenger
- **AirlineCode**: airline IATA code (e.g., RYR)
- **AirportCode**: IATA code of the destination airport
- **Bags**: number of checked-in bags by the passenger

- **CheckInType**: check-in method used (Counter, Online, Kiosk)
- **EntryPoint**: entry point to the terminal
- **ExitPoint**: gate number
- **FlightNumber**: flight number (just the numeric code, without the airline)
- **Gender**: gender of the passenger
- **Nationality:** nationality of the passenger
- **Passport:** type of passport used (EU, non-EU, UK for PMI; and UK; non-UK for LCY)
- **SchengenStatus:** passenger destination grouped into the categories: Inter-islands, Domestic, Schengen, No Schengen or International
- **TSchedule:** Scheduled Off-Block Time (SOBT)
- **TravelClass:** Class (Economy vs Business). It has been assumed that business passengers travel in Business class while leisure passenger travel in Economy class.
- **TravelPurpose:** purpose of the trip (business vs leisure)

EUROPEAN PARTNERSHIP

Co-funded by the European Union

4 KPI implementation

The execution of the integrated model presented in the previous section generates all the information available about the door-to-gate and gate-to-door passengers' journey in both the ground transport access and egress legs and in the terminal processes. These outputs need to be postprocessed in order to extract a series of KPIs which enable the final user to assess the short-term evolution of the ground transport system and the airport performance.

In deliverable 'D2.1 IMHOTEP ConOps' and 'D2.2 Specification of Case Studies', a preliminary analysis of the KPIs to be implemented was outlined. However, as the project has progressed, the definition of these KPIs has been fine-tuned according to:

1. The needs and requirements expressed by the different stakeholders (airport and ground transport operators) as well as the ground transport and terminal modelling experts involved in the project.
2. The capabilities of the different models to calculate the different available indicators.

In Table 1 and Table 2, the KPIs implemented for the ground transport side and the terminal processes are presented, respectively. Tables also include information on how the KPI is measured, which kind of temporal aggregation is considered and which kind of segmentation/filtering is available.

Table 1. Implemented KPIs on the ground transport side

KPI	How to measure it?	Temporal aggregation	Segmented by:
Modal share	Percentage of users choosing each transport mode	Total time window	– Pax/workers – Business/Leisure – Resident/Non-resident
Travel times	Probability distribution of the door-to-gate/gate-to-door travel time broken down by terminal and access/egress	Total time window	– Destination/airline – Flight time
CO2 emissions	Total CO2 emissions	Total time window	N/A
Occupancy on the PT alternatives	Number of users in the PT vehicles every hour.	Total time window	– Bus time – Pax/workers – Business/Leisure – Resident/Non resident
Productivity of the ground side transportation	Daily totals of the passengers-km travelled by mode.	Total time window	– Transport mode – Pax/workers – Business/Leisure – Resident/Non resident
Waiting time in the PT alternatives	Probability distribution of the waiting time for the PT alternatives.	Total time window	N/A

Co-funded by the European Union

Table 2. Implemented KPIs on the terminal side

KPI	How to measure it?	Temporal aggregation	Segmented by
Queuing time at the airport facilities	Evolution of the waiting time at the airport facilities	15 minutes	– Destination – Business/Leisure
Facilities throughput	Evolution of the throughput at the airport facilities	15 minutes	
Occupancy at the airport areas	Evolution of the occupancy in the different airport areas. Occupancy is defined as: $\sum_{p=0}^{n\_pax} \frac{time\ spend\ in\ the\ area_{p}}{time\ interval}$ where *n_pax* is total number of passengers (*p*)	15 minutes	– Destination – Business/Leisure
Dwelling times at the airport areas	Evolution of the dwelling time in the different airport areas	15 minutes	– Destination – Business/Leisure
Missed flights	Number of passengers that missed their flights	Total time window	– Cause (flight lost at check-in or at gate) – Flight number/Airline – Transport mode

5 Visualisation and decision support tool

This section contains all the documentation required to describe the development and operation of the IMHOTEP visualisation and decision support tool.

5.1 Software requirements

5.1.1 Requirements management methodology

Requirements management is one of the first tasks that takes place in software development projects. Each requirement must correspond to a physical or functional need to which the system must be able to respond. After their definition, those requirements serve as an input in the software design phase.

The quality of requirements consists of four fundamentals:

- Review: requirements must be reviewed by different roles in the project (at least software analysts and software engineers) to ensure their feasibility and usability.
- Verification: requirements must be verifiable to ensure that the system meets all the needs identified in them.
- Validation: the requirements must be validated with the end users of the system to ensure that the system produces the desired results.
- Agreement: once all stakeholders agree on a stable version of the requirements specification, the first baseline of the specification is established.

During the project lifecycle, a series of processes take place to define and refine the requirements according to the above baselines:

1. Requirements capture
2. Requirements analysis
3. Change management

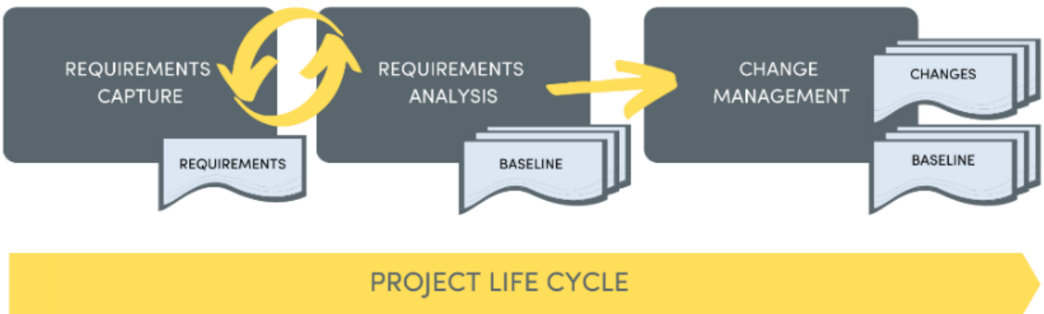

Figure 8. Requirements management process

EUROPEAN PARTNERSHIP
Co-funded by the European Union

5.1.1.1 Requirements capture

During this phase, the needs from different sources are captured and formulated as system requirements. These requirements are compiled and documented in a specific artefact together with their associated assumptions, exclusions, and constraints. In the IMHOTEP project, the following techniques have been used to perform this task:

- Semi-structured meetings with the different project stakeholders.
- Brainstorming
- User stories
- Prototyping
- Analysis of the system environment:
  - Existing technologies
  - Available data
  - Security constraints
  - Applicable regulations

The main artefact produced in this phase is a set of system requirements, which represents the starting point for the requirements' analysis phase. In that phase, new requirements may be discovered that have to be returned to requirements capture. This process must be repeated iteratively until a requirements' specification is completed that satisfies all parties involved in the development of the system.

5.1.1.2 Requirements analysis

The requirements' analysis phase aims to determine whether the requirements are clear, complete, atomic, and consistent, resolving any existing doubts or conflicts. As a consequence of this exercise, requirements may change from their original statement or new requirements may emerge. It should be noted that these changes are not yet recorded as change proposals, as this is the normal evolution of requirements during this phase.

The requirements must be reviewed, verified, validated and agreed upon by all project stakeholders prior to implementation. These four steps, which are described as a linear sequence, are often executed iteratively until the set of requirements is considered mature enough to successfully achieve the project objectives. Figure 9 shows this process.

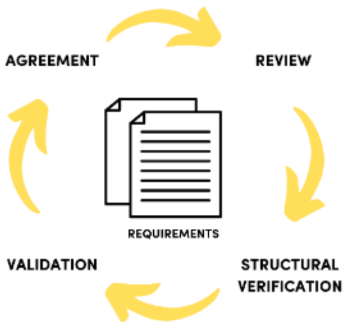

Figure 9. Requirements analysis process.

1. Review: the requirements are gathered and analysed by different project profiles, at least one operational concept analyst and one software engineer, to ensure a broad view of the problem. In this way, we can ensure that a final set of viable requirements is generated for implementation.
2. Structural verification: by verifying the requirements, we ensure that they are properly documented. Some activities carried out at this point are:
   - Reformulate them so that they are: specific, measurable, achievable, realistic and time bound. These characteristics are commonly grouped under the term SMART (Specific, Measurable, Attainable, Realistic and Time bound)
   - Add attributes such as source, responsible party, status, and rationale for the need to include them in the specification
   - Define dependencies between the different requirements to ensure their traceability.
3. Validation: is the mechanism used to ensure that the implementation of the requirements produces the system expected by the user. Validation requires the participation of an operational expert with whom regular meetings are held, discrepancies are anticipated and doubts are resolved.
4. Agreement: to formalise the approval of requirements it is important to reach agreement, ideally, with all stakeholders in the system. It is this set of requirements that forms the first baseline of the requirements' specification, which can be modified throughout the project through change management mechanisms.

5.1.1.3 Change management

Once the first baseline of the requirements' specification is established and the design phase begins, the change management phase also begins. At this point, the different stakeholders in the project share a common vision of the problem and the proposed solution, and all parts of the system are properly interlinked with each other. Any change to a requirement could change the common vision of the system or impact another part of the system under development. The only way to ensure that the system remains consistent is to establish a formal process for change management. Under this methodology, requirements could be changed when the following issues arise:

- Errors in requirements
- Inconsistency between requirements
- Requirements evolution
- Technical problems
- Management problems
- New rules and regulations
- Contractual changes

When changes occur to the requirements, a version of the requirements sheet is released. The numbering of the requirements sheets follows the format A.B, where A and B are integers. When major changes take place, the digit A is increased by one and the digit B is set to 0. When minor changes happen, the digit B is increased by one unit.

5.1.2 Software requirements specification

This section presents the requirements specification for the visualisation platform of the IMHOTEP project obtained from the analysis of use cases, existing technologies, security constraints, applicable regulations, etc. The specification presents a complete vision of the system agreed by the different stakeholders of the project.

The collection of requirements is in tabular format (see Table 3), where each column corresponds to a requirement. In addition, the requirements are grouped and organised under different sections for ease of reference and maintenance. Each requirement in the specification contains the following fields:

- **ID:** unique identifier composed of an alphanumeric prefix and a three-digit sequential (e.g., IMH-REQ-001).
- **Definition:** text describing the requirement.
- **Level:** we distinguish the following levels of requirements:
  - **High level**: High-level requirements describe the overall functionality of the system. They specify the main interfaces, functionalities, and performance of the system. This level contains information only about what the system does, not how it does it.
  - **Low level**: Low-level requirements are derived from the requirements of the previous level. In addition to details of functionality, this level includes specific information on how the system performs operations.

Table 3. Visualisation and decision support tool requirements

Id	Definition	Level
01	NON-FUNCTIONAL REQUIREMENTS	
01.01	Hardware	
IMH-REQ-001	The system shall be installed in a centralised server.	High
IMH-REQ-002	The system shall have access through Internet connectivity 24/7.	High
IMH-REQ-003	The server shall have an uptime availability of at least 99%.	High
01.02	Platform Software	
IMH-REQ-004	The operative system shall provide shell access to the operative system.	High
IMH-REQ-005	The operative system shall provide a user with administrative rights.	High
01.03	Software	
IMH-REQ-006	The frontend shall be coded using Dash, HTML, CSS and bootstrap.	High
IMH-REQ-007	The backend shall provide the responses to the requests issued by the frontend.	High
IMH-REQ-008	The backend shall be coded using Python, Django and Plotly.	High
IMH-REQ-009	The database shall be a Postgres database.	High
IMH-REQ-010	The backend shall serve the frontend using a web server Apache2.	High
02	Backend	

EUROPEAN PARTNERSHIP

Co-funded by the European Union

IMH-REQ-011	The backend shall have an endpoint to load the visualisation indicators.	High
02.01	**Data Storage**	
IMH-REQ-012	The backend shall store the necessary data to create visualisations.	High
02.02	**Authentication**	
IMH-REQ-013	The backend shall allow only registered users to access data.	High
IMH-REQ-014	The backend shall have a database with a table containing the users' credentials.	High
IMH-REQ-015	The users' password shall be encrypted.	High
IMH-REQ-016	The backend shall have an endpoint to perform the user's login.	High
03	**FRONTEND**	
03.01	**Airport selection view**	
IMH-REQ-017	The view shall have a number of buttons that match the airports available in the tool.	High
IMH-REQ-018	The view shall be the first view the user encounters after authentication.	High
IMH-REQ-019	The view shall redirect the user to the visualisations concerning the airport selected.	High
03.02	**What-if visualisations**	
IMH-REQ-020	The view shall have a user control with three selectors.	High
03.02.01	**Selectors**	
IMH-REQ-021	The user shall select a scenario to visualise.	High
IMH-REQ-022	The user shall select a management action, contained into a management plan to visualise.	High
IMH-REQ-023	The user shall select a view to visualise.	High
IMH-REQ-024	The user shall have all the selectors with a selected option to see visualisations.	High
IMH-REQ-025	The visualisation shall reload automatically after one of the values of one of the selectors has changed.	High
03.02.02	**Visualisations**	
IMH-REQ-026	The view shall present all the visualisations of the selected options at once.	High
IMH-REQ-027	The view shall have a scroller on the visualisation area for the user to see all the visualisations.	High
03.02.02.01	**Graphs**	
IMH-REQ-028	The indicators shall be represented through graphs.	High
IMH-REQ-029	The graphs shall be one of the following types: - Bars plot - Lines plot with quartiles - Stacked bars plot	Low

EUROPEAN PARTNERSHIP

Co-funded by the European Union

	- Histogram	
IMH-REQ-030	The graphs shall have filter selectors according to the type of indicator.	High
IMH-REQ-030.01	The types of filters shall be the following: - Destination - Airline - Purpose - User Type - Residence - Flight Hour	Low
IMH-REQ-031	The filters shall be defined using two selectors, one for the type and other for the value of the filter.	High
IMH-REQ-032	The filter values shall be fetched automatically from the data.	High
IMH-REQ-033	The filter values shall be unique.	High
IMH-REQ-034	The graphs shall reload the data automatically when a filter is selected.	High
IMH-REQ-035	The visualisations shall provide a pair of filter selectors for each graph that is fitted for filtering.	High
IMH-REQ-035.01	The visualisations shall allow only one filter per graph.	Low
03.02.02.02	**Heatmap**	
IMH-REQ-036	The occupancy indicator shall be represented through a heatmap.	High
IMH-REQ-037	The map shall represent the areas of the airport with colours according to the occupancy level.	High
IMH-REQ-038	The map shall be available for an airport that has geographic data of the zones.	High
IMH-REQ-039	The map shall have a tooltip when hovering an airport zone.	High
IMH-REQ-039.01	The tooltip shall include: - Name of the airport zone - Value for the occupancy of the airport zone hovered - Percentage of the occupancy discriminated by purpose - Percentage of the occupancy discriminated by destination	Low
IMH-REQ-040	The map shall have zoom in/zoom out controllers.	High
IMH-REQ-041	The map shall have below a selector to filter.	High
IMH-REQ-041.01	The selector shall be used to filter the data from the map according to interval hours	Low
IMH-REQ-041.02	The selector shall have a default value to show all hours when the map is loaded.	Low

5.2 Software design

5.2.1 Architectural design

During software development, the architectural design has the purpose of explaining how the software is structured and organised. Such organisation includes all components, how they interact with each other, the environment in which they operate, and the principles used to design the software.

The IMHOTEP visualisation and decision support tool have been developed as a web application, so that it is accessible from multiple devices by different users concurrently.

5.2.1.1 Component structure

In web applications, the system architecture is distributed between the user interface itself (commonly known as “frontend”) running on the user device and the components executing the system logic (“backend”) running on a server. This is meant to prevent the user device from storing the data and executing the algorithms, which would require high computational resources. The diagram presented in Figure 10 represents the components of the visualisation and decision support tool.

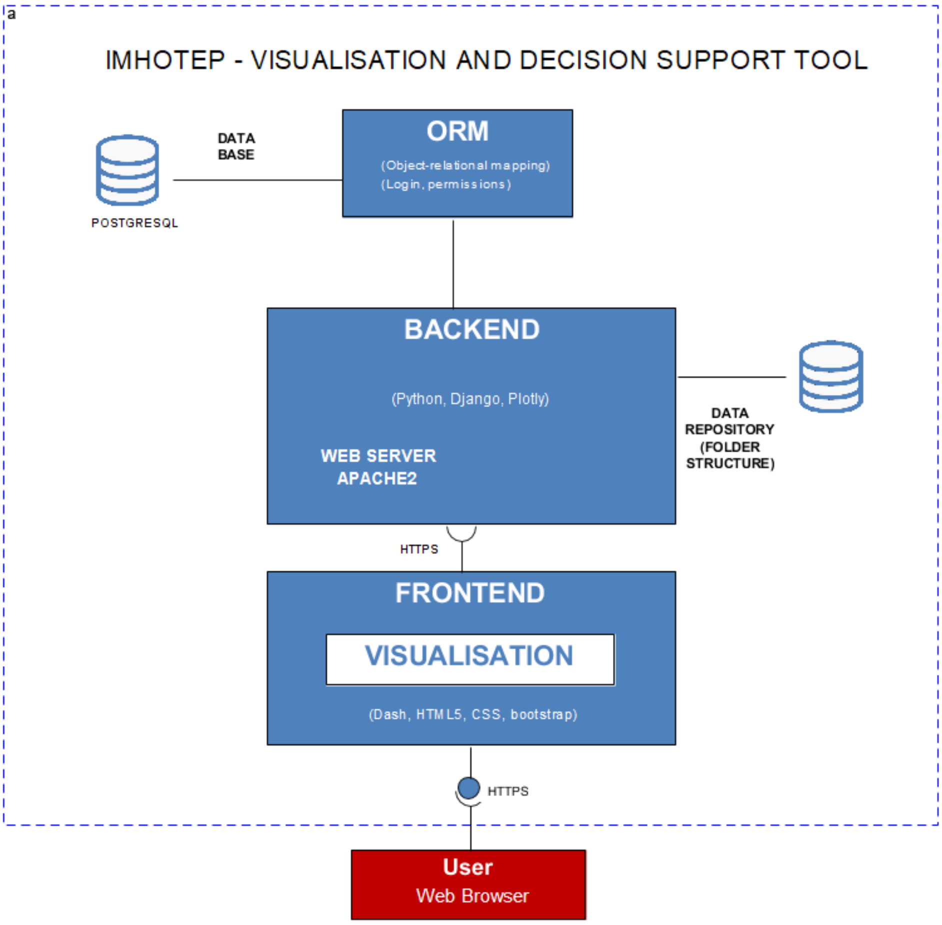

Figure 10. Visualisation and decision support tool high-level architecture

EUROPEAN PARTNERSHIP
Co-funded by the European Union

- **Data repository**: folder and file system where the airports, scenarios, management actions and indicators used to generate the visualisations are stored.
- **ORM**: Object-Relational Mapping is an interface that facilitates communication between an application and the database. This abstraction layer encourages good software practices such as reusability and low coupling between components. This component is part of the Django framework. In addition, Django provides the “auth” library, which uses the ORM component to safeguard user credentials and sessions in the database.
- **Database**: this is used to store session data and metadata associated with the visualisation of indicators. PostgreSQL will be used as the engine of this database. Its main advantages are: open-source licence, stability and performance when executing transactions.
- **Visualisation**: the visualisation represents the user’s access point to the dashboard, through which the following actions can be performed:
  - Login to the visualisation tool;
  - Consult the different types of visualisations generated by the choice of a scenario, management action and view;
  - Visual exploration of the indicators presented.

5.2.1.2 Data structure

This section presents the data structure defined in this tool. In this case, data structure refers to the way of storing and organising data in the computer so that it can be used efficiently. This section intends to present the data structure for this visualisation tool. Figure 11 represents the folder structure used to develop this visualisation tool.

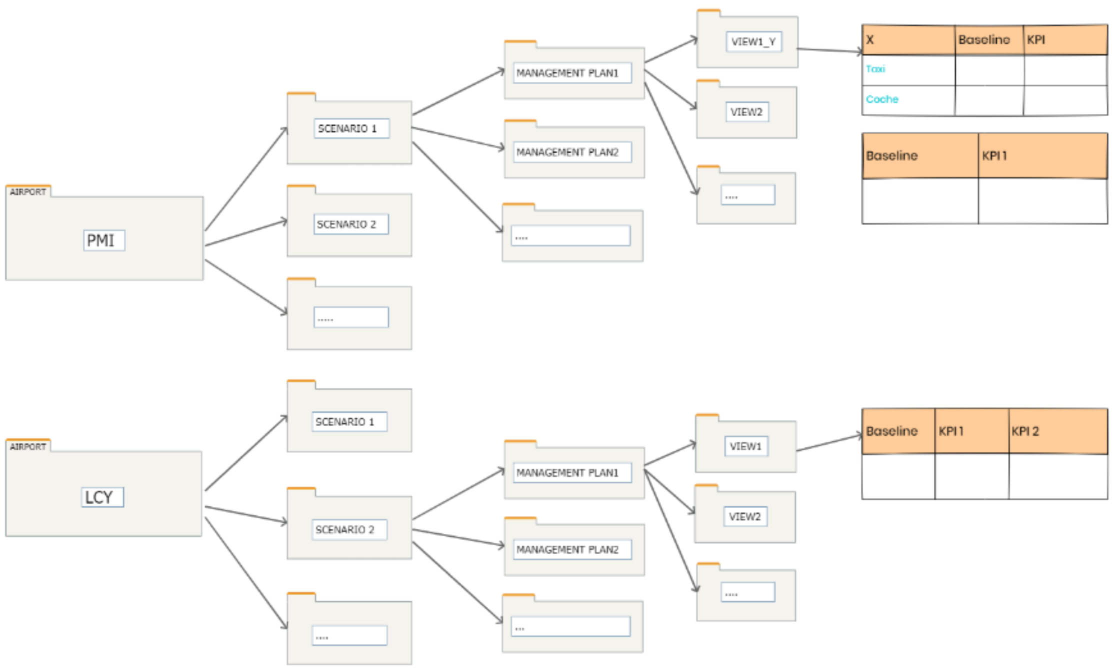

Figure 11. Visualisation and decision support tool folder structure

The image reflects the current data in the visualisation tool, although, if we generalise this structure, the sequence of the folders should be:

- Level 1: Airport folders. Each airport should have its own folder for when the user chooses which information wants to see concerning the airport.
- Level 2: Scenario folders. Each scenario should have its own folder inside the airport folder.
- Level 3: Management plan folders. Each management plan concerning each scenario should have its own folder. Likewise, the baseline data should be included at this level.
- Level 4: View folders. Each view should have its own folder according to the type of management plan.
- Level 5: CSV data file. At this point, each view folder will have all multiple CSV files and each data file corresponds to the variables of the graphs/heatmap the user will visualise.

Using this approach, the visualisation tool will fetch information that is filtered level by level through folders. When a visualisation is loaded, it will be represented based on one specific file. As the user chooses the airport, the scenario, the management plan, and later the view, the tool will be excluding the other paths that do not match these choices.

Finally, if another data set intends to be included in the tool, it should have this structure and no further alterations are needed in order to be functional.

5.2.2 Use cases

This section regards the use cases of the visualisation and decision support tool of IMHOTEP. The use cases describe the functionalities that the tool offers from the user's point of view. Essentially, it is a way to summarise the details and the users of a system.

Figure 12 represents the interactions among the elements of the visualisation tool. In this case, it is possible to identify two different kinds of roles:

- Registered user: actor registered in the IMHOTEP visualisation and decision support tool. Therefore, the user has a username and a password to authenticate in the tool. In the next sections this actor will be referred to as the user since it will be the main actor of the system.
- Administrator: actor in charge of the management of the system.

These actors interact with the system, which will be reflected in the next subsection, where the user stories are presented and described.

5.2.2.1 IMH-UC-1

- **User Story**: Login
- **User**: Registered User
- **Description**: The user login into the application
- **Pre-conditions**: The user must have been register

- **Event Flow**:
  - The user enters the username
  - The user enters the password
  - The user clicks in the “Login” button
  - The system checks that the username exists and that the password matches it
  - The user successfully logs into the system
- **Post-conditions**:
- **Alternative Event Flow**:
  - The username does not exist, or the password does not match
  - The system will display an error message
  - The user can repeat the first step

5.2.2.2 IMH-UC-2

- **User Story**: Accessing the airport selection page
- **User**: Registered User
- **Description**: The user wants to select one airport
- **Pre-conditions**: The user must have been registered and logged in
- **Events Flow**:
  - The user clicks on one of the possible airports (in this case, two)
  - The user will follow to the page of the possible visualisations for the airport selected
- **Post-conditions**:
  - The user can click on the “Return” button in the header to return to the airport selection page

5.2.2.3 IMH-UC-3

- **User Story**: Selecting user control options
- **User**: Registered User
- **Description**: The user wants to select a what-if set of options
- **Pre-conditions**: The user must have been registered and logged in
- **Events Flow**:
  - The user clicks on one of the possible scenarios
  - The user clicks on one of the options under “Management Action”
  - The user clicks on one of the views
  - The user will visualise all the graphs and/or heatmap with data that concerns the options selected
  - The graphs or/and heatmap are displayed successfully

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- **Post-conditions**:
  - The user can modify the information displayed on the visualisation
  - The user can select different scenarios
  - The user can select different management plans
  - The user can select different views
  - The user can filter the graphs/heatmap with the data available for each one
  - The user can click on the "Return" button in the header to return to the airport selection page
- **Alternative Event Flow**:
  - The graphs or/and heatmap are still being uploaded

5.2.2.4 IMH-UC-4

- **User Story**: Logout
- **User**: Registered User
- **Description**: The user wants to leave the visualisation tool
- **Pre-conditions**: The user must have be logged in
- **Events Flow**:
  - The user clicks the "Logout" button in the header
  - The system closes the user session
- **Post-conditions**:
  - The user will be able to log back into the system

5.2.2.5 IMH-UC-5

- **User Story**: Register User
- **User**: Administrator
- **Description**: The administrator created a new user who will be able to log into the system
- **Events Flow**:
  - The administrator begins the process of creating a new user
  - The administrator enters the username
  - The administrator enters the password
  - The administrator enters a full name
  - The administrator enters an email
  - The administrator completes the user creation
  - The system stores the credentials in the encrypted database
- **Post-conditions**:
  - The system stores the credentials in the encrypted database

- **Alternative Event Flow**:
  - The username already exists
  - The system will display an error message indicating that a user with that username
  - already exists
  - The password is not strong enough
  - The system will display an error message indicating that the password is not secure

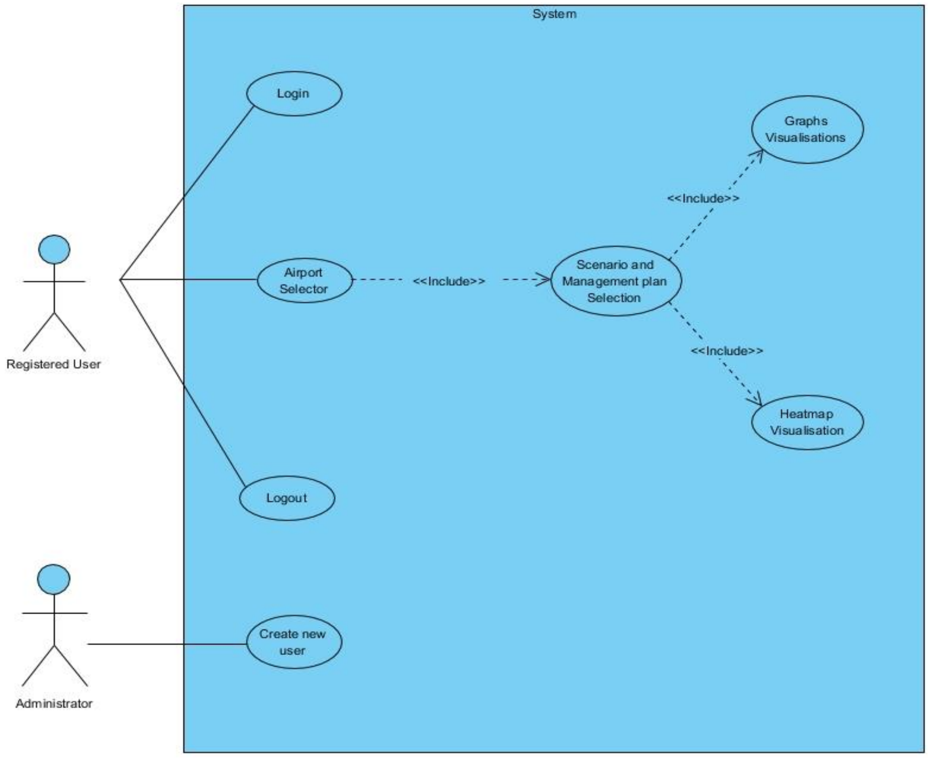

Figure 12. Interactions among the elements of the visualisation tool

5.2.3 Technological stack

This section presents the technological stack of the developed tool. It is the collection of technologies chosen to build, run and, later, deploy the visualisation tool. Commonly, it is divided into two components, the client-side component, best-known as the front-end, and the server-side component, also well-known as the back-end.

5.2.3.1 Frontend

The following technologies and frameworks have been selected to implement the frontend component of the system:

- **HTML:** HyperText Markup Language (HTML) is the set of markup symbols or codes inserted into a file intended for display on the browser. The markup tells web browsers how to display a web page's words and images.

- **CSS:** CSS is the language for describing the presentation of the web pages, including colours, layout, and fonts. It allows the developer to adapt the presentation to different types of devices, such as large screens, small screens, or printers.
- **Bootstrap**: Bootstrap is a front-end open-source framework used to feature multiple HTML and CSS templates for user interface elements such as buttons and forms.
- **Dash**: Dash is a Python framework meant to build web applications running in the browser. It is ideal to develop data visualisation apps with highly custom user interfaces in pure Python.

5.2.3.2 Backend

The backend refers to any part of a software that the user does not visualise. This means that everything that happens or needs to happen before a visualisation is displayed concerns the backend. The following frameworks and technologies will be used to implement the backend component of the system:

- **Python**: Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.
- **Django**: Django is a high-level and open-source framework that provides database-driven websites. It is based on the Model-View-Controller (MVC) architecture, which promotes reusability and facilitates components to plug-in easily. Furthermore, Django comes with an interface to set user permissions, where administrators can add and remove users easily. Django is preferred by experienced python users because it has many built-in features to develop complex web apps.
- **PostgreSQL**: PostgreSQL is a general purpose and object-relational database management system (ORDBMS). It is used as the primary data store or data warehouse for web, mobile, geospatial, and analytics applications. This framework can also store structured and unstructured data in a single product. Finally, it is designed to be extensible, and you can define your own data types, index types, functional languages, etc.
- **Apache HTTP Server**: The Apache HTTP Server is cross-platform web server software and open-source also. It serves to communicate over networks from client to server using the TCP/IP protocol. Despite being used for a wide variety of protocols, the most commonly used is HTTP/S.

5.3 User manual

This section provides an overview of the IMHOTEP visualisation and decision support tool, exposing the possible interactions that the user can perform, as well as explaining the different views.

The tool presented has the aim to allow the users to interact with the integrated simulation model and perform what-if analyses, by simulating the impact of different decision alternatives in terms of the KPIs implemented in T6.2. A more detailed description of the tool's development can be found in Section 5.2 regarding the Software Design.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

5.3.1 Login

The first interaction the user has with the tool concerns the login, the process of authentication. Such is important to prevent unwanted access to the platform. In this view, the user should introduce a username and a password provided by the platform administrator.

5.3.2 Airport selection

After the authentication, the user will encounter the airport selection view which presents the airports available in the tool (see Figure 13). At the current moment there are only two options available: the London City airport and the Palma de Mallorca airport. By selecting one of the buttons, the user will move on to the next view that will contain information exclusive to that airport.

Figure 13. Airport selection view

Note that in future implementations this selection option will not be necessary. The tool is expected to be implemented in a specific airport, therefore all the involved stakeholders, once they have logged, will directly access the information on that specific airport. However, to properly evaluate the case studies selected for the project, this airport selection view is required.

5.3.3 What-if selection

Here we describe the set of views and options that the user may manipulate to carry out 'what-if' analyses for evaluating the impact of the different management actions included in the dashboard.

5.3.3.1 Selection: scenario, management action, view

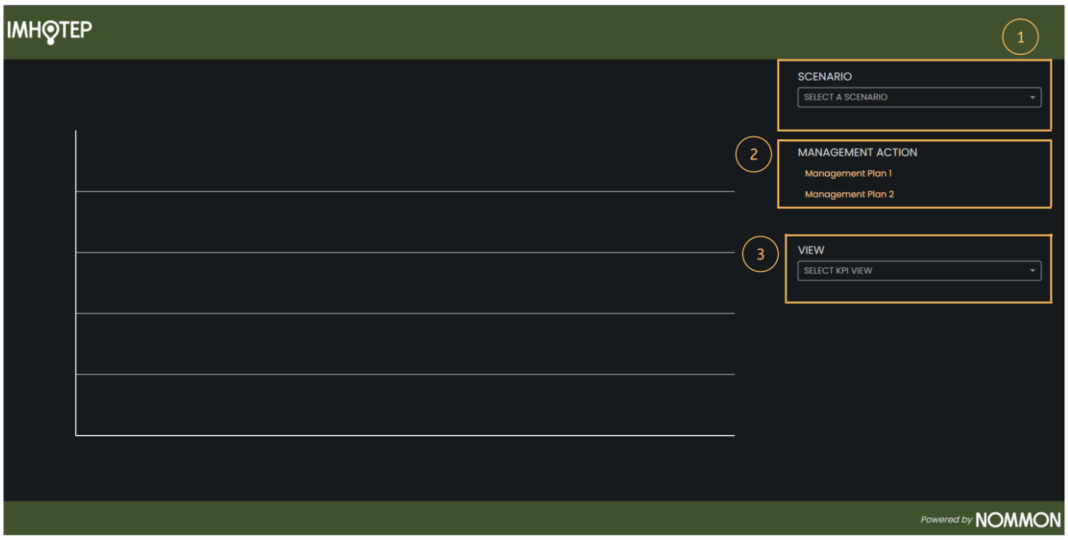

Figure 14. Initial selection view

After selecting an airport, the user will face a blank graph with a control panel on the right. To start the visualisations the user must select three different types of options (see Figure 15):

1. A scenario. The user must select the scenario, which is basically the use case to be studied and which represents an adverse situation for the airport, for example, several aircrafts arriving late or a traffic jam in the airport access. The scenarios have been defined in deliverable 'D2.2 Specification of Case Studies' and will be fine-tuned in WP7.
2. A management action. The different management actions applied to a certain adverse condition have been grouped into management plans. The management plans will include a set of predefined management actions that may or may not have an impact on the scenario, for example, increasing the number of security control lanes or increasing the frequency of the public transport alternatives.
3. A view. In order to emulate the final implementation of the tool, a view-based approach has been implemented to visualise the KPIs. The idea is that the different views will group the KPIs relevant for the different stakeholders. Initially, two views have been defined, one for the Terminal and one for the Access/Egress KPIs. This provides the information required by the two stakeholders represented in the project, the airport operator and the ground transport operator, respectively. However, as many views as desired can be created for the rest of stakeholders or one common view which includes all the KPIs can also be included.

In future real implementations the scenario selector will not be included. In that case, the final user will visualise the short-term forecast of the transport system in real-time and, in case of an adverse situation arises, the user will explore different what-if scenarios by modifying the different management actions available in order to assess the best course of action.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

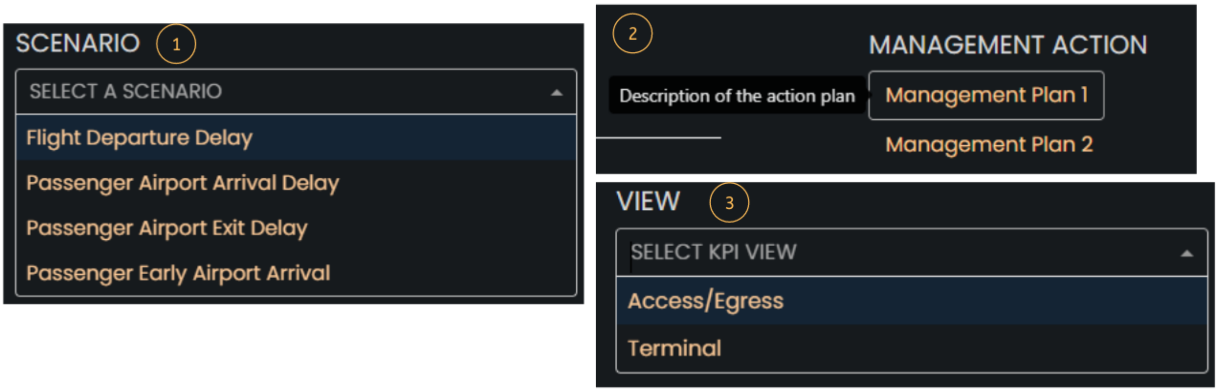

Figure 15. Different types of selectors available in the visualisation tool

5.3.3.2 Graphs

Once all the necessary options have been selected in the control panel, the space where there was previously an empty chart will be loaded with a variety of plots, each representing a different KPI, as presented in Figure 16. This view has a scroll to the right so that the user has all the different graphs displayed at once and uses the scroll bar to see them all.

Figure 16.Example of KPI visualisation

5.3.3.3 Filters

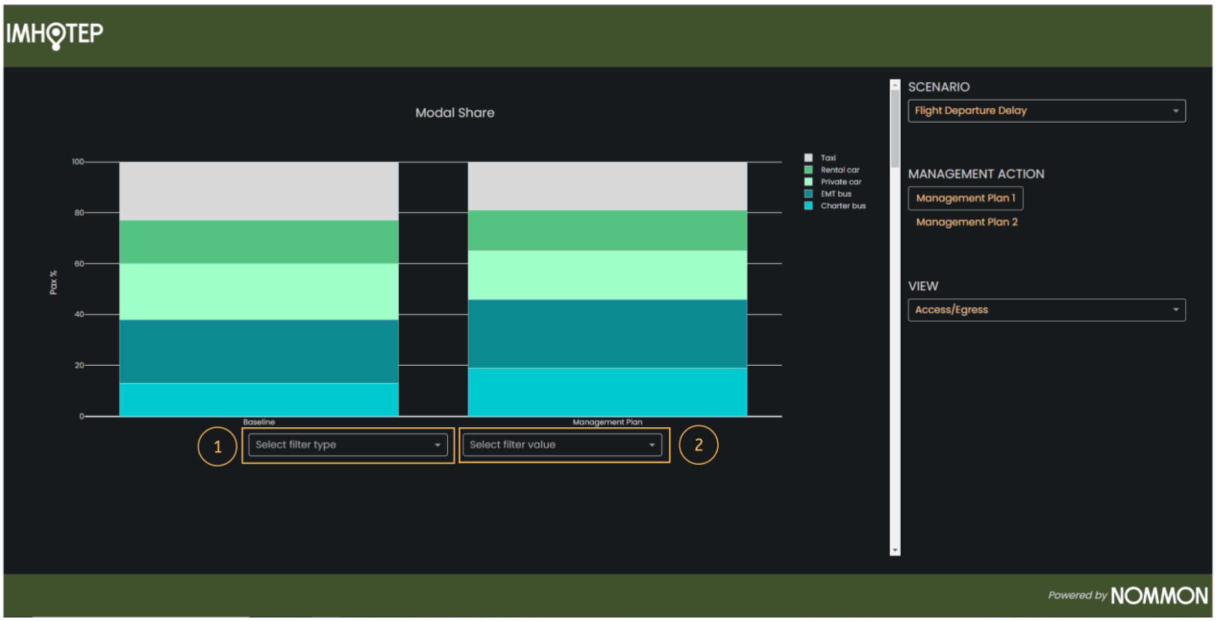

Figure 17. Example of KPI visualisation with selectors for the filter type (1) and the filter value (2)

Some of these charts contain two filter selectors below the x-axis, as displayed in Figure 17. In order to filter the information, the user must choose the type of filter first, which indicates the categories available to filter in that particular graph, for example, the type of users. After setting the filter type, the user must decide the value of that filter. Figure 18 shows an example for the "User Type" filter type, where the value can be, for example, a passenger or a worker.

After selecting both options, the graph will automatically refresh and represent the data that matches with that filter type and value. Since the graph can only be filtered once at a time, if the user selects the option "All" in the values of the filter, the graph would be loaded to the initial state.

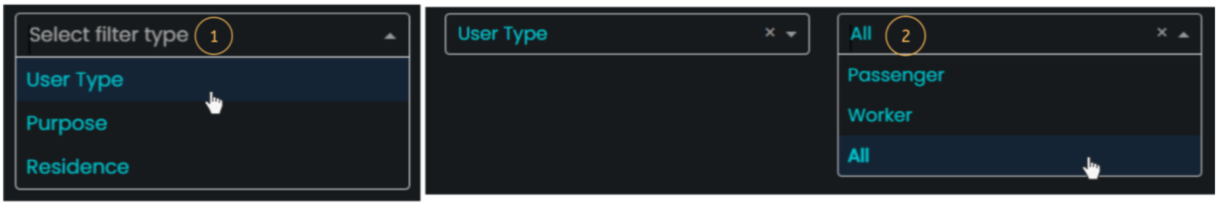

Figure 18. Example for the "User Type" filter

5.3.3.4 Heatmap

In this view, the user is able to visualise a heatmap, which shows the occupancy information available for each sector of the airport (see Figure 19). Such information is presented resorting to colours and tooltips. At the moment, this visualisation is only available for the Palma de Mallorca airport. The reason is that geographic information regarding all the airport areas needs to be available in order to properly represent the data through a map.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

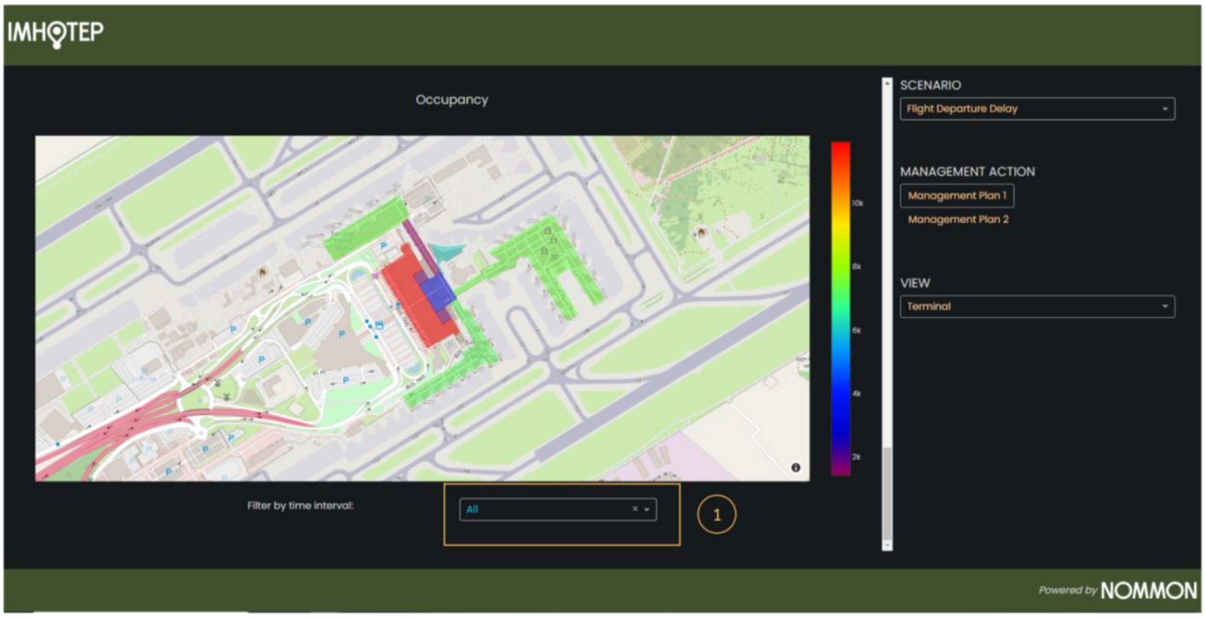

Figure 19. Occupancy heatmap implemented for PMI airport

By hovering over an airport area, the map has a tooltip available that presents more detailed information regarding the area (see Figure 20). This tooltip includes:

- Name of the airport area
- Value for the total passengers in that airport area
- Percentage of the occupancy discriminated by purpose
- Percentage of the occupancy discriminated by destination

Figure 20. Tooltip showing more detailed information on the airport occupancy

Under the map, the user may find a filter selector. This enables the user to filter the information according to an interval of hours pre-defined in the data. So, by default, when the map is loaded for the first time, all hours are displayed, but then there is a possibility to choose just a single interval of these hours.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

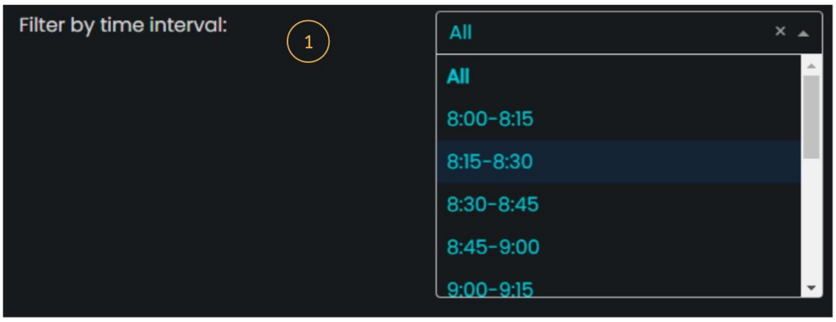

Figure 21. Time interval selector for the occupancy heatmap

5.4 Validation and testing

This section concerns the validation and testing process of the visualisation tool.

Once the development of the tool is finished and after its subsequent deployment in a real environment, a formal functional validation of the platform's operation must always be carried out. Such validation assesses the system's performance from an operational point of view, taking into consideration the following dimensions:

- **Functionality**: the system must function correctly, being able to collect user's inputs and interactions, showing relevant results without errors, warnings or unexpected behaviour.
- **Usability**: the developed tool must provide an intuitive interface that allows interaction with the system in a fluid way and without the need for any help other than the tool's user manual.
- **Fluidity**: the changes in the tool should be fluid, without abrupt stops, so as to enable continuous interaction that facilitates the use of the system.
- **Consistency of data**: visualisation results provided by the system should be consistent and meaningful to the user.

This section describes the validation process carried out, as well as the users involved in the validation process and the conclusions drawn about the platform once the process is completed.

5.4.1 Unit testing

The tool's validation process consists of defining a set of systematic tests specific to each validation dimension that enable a qualitative assessment of the degree of compliance of the platform with basic quality and functionality standards.

To carry out these tests we chose an open-source, unit testing platform, that assists the developer in reaching high coverage of code tests. The idea of unit testing is based on testing small parts of the tool's code. Normally, this kind of testing should test individual components so that it is ensured that they behave as expected.

Ultimately, unit testing serves to validate the functionalities of the tool and it can either be done manually or automatically. For this visualisation tool, we used an automated testing tool named Testim, which is a software testing tool that provides developers a fast way to automate tests, resorting to Artificial Intelligence.

Before running the validation tests, it is necessary to define the desired set of user interactions that the tool should be able to run without error. For this, the following points have been defined, covering all the possible actions that a user can carry out on the visualisation tool:

- The tool is able to present to the user the buttons which correspond to the airports available.
- The tool is able to redirect the user to the view of the visualisations after clicking one of the buttons of the airports.
- The tool is able to load the options in each selector in the visualisation view and the user able to select those options.
- The tool loads the graphs and map corresponding to the options selected by the user.
- The tool is able to update the graphs while the user uses the filter selector to filter the data.
- The tool is able to present a heatmap and the user is able to filter the information in the heatmap resorting to the filter selector.

Below is the list of tests made and performed in the testing platform, where the "Last Runs" column represents the status of the tests. If it is green, it means that passed, as we can see in Figure 22.

NAME	OWNER	LAST RUNS
Graphs filters This tests intends to test the graphs filters. Select a filter type and a value to see if the graph realoads with new information.	Carolina Vasconcelos	
Map and filters This test intends to assure that the map is loaded properly and that it is realoaded when the filters are used as well.	Carolina Vasconcelos	
Select Airport This test intends to check if the tool redirects the user to the visualisation page. Select and airport and see the page load with that airport on the URL.	Carolina Vasconcelos	
Select graphs This test intends to verify if the graphs appear as expect upon the selection of the options "Scenario", "Managment Plan", "View"	Carolina Vasconcelos	
Select Visualisation Options This test intends to check the visualisation options. Select a "Scenario", "Managament Plan" and "View" properly.	Carolina Vasconcelos	
Show Airport Buttons This test intends to check if the airports buttons are loaded.	Carolina Vasconcelos	

Figure 22. Test performed and results obtained

The platform also provides a more detailed report of each test if selected, as can be observed in Figure 23. In particular, it provides a percentage success value, which represents whether the test was fully successful or partially successful. On the other hand, it also includes a counter for the number of times that test has been executed. Finally, it shows the average duration of the test, which is a very useful information if the developer wants to do a performance analysis.

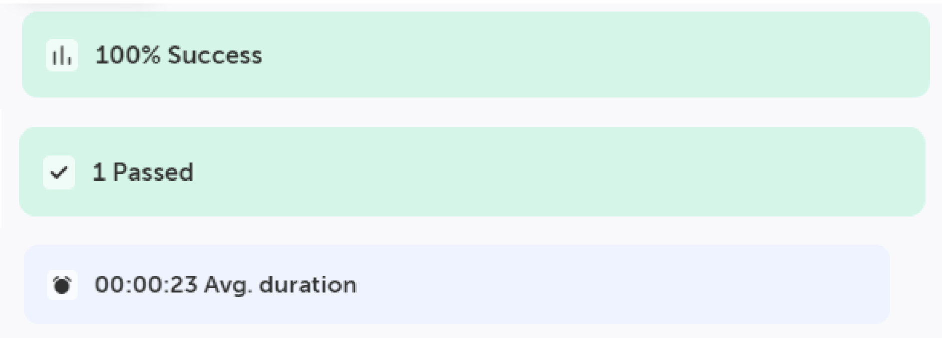

Figure 23. Detailed report information on the test performed

5.4.2 Testing and validation results

The final results of the validation tests are shown in the following images:

1. Show Airport Buttons Test

100% Success | 2 Passed | 00:00:08 Avg. duration

TEST NAME	BRANCH	BROWSER	STARTED	DURATION	RESULT LABELS	FAILURE TYPE	STATUS
Show Airport Buttons	master		an hour ago	7 seconds			✓
airport buttons	master		a day ago	8 seconds			✓

Figure 24. Show airport buttons results

2. Select Airport Test

100% Success | 1 Passed | 00:00:15 Avg. duration

TEST NAME	BRANCH	BROWSER	STARTED	DURATION	RESULT LABELS	FAILURE TYPE	STATUS
Select Airport	master		a day ago	15 seconds			✓

Figure 25. Select airport test results

3. Select Visualisation Options Test

100% Success | 1 Passed | 00:00:10 Avg. duration

TEST NAME	BRANCH	BROWSER	STARTED	DURATION	RESULT LABELS	FAILURE TYPE	STATUS
select visualisation options	master		8 hours ago	10 seconds			✓

Figure 26. Select visualisation options test results

4. Select Graphs Test

100% Success | 2 Passed | 00:00:27 Avg. duration

TEST NAME	BRANCH	BROWSER	STARTED	DURATION	RESULT LABELS	FAILURE TYPE	STATUS
Select graphs	master		7 hours ago	13 seconds			✓
Select graphs	master		7 hours ago	41 seconds			✓

Figure 27. Select graphs test results

5. Select Graphs Filters Test

100% Success | 1 Passed | 00:00:23 Avg. duration

TEST NAME	BRANCH	BROWSER	STARTED	DURATION	RESULT LABELS	FAILURE TYPE	STATUS
Graphs filters	master		7 hours ago	23 seconds			✓

Figure 28. Select graph filters test results

EUROPEAN PARTNERSHIP

Co-funded by the European Union

6. Load Map and Select Filters Test

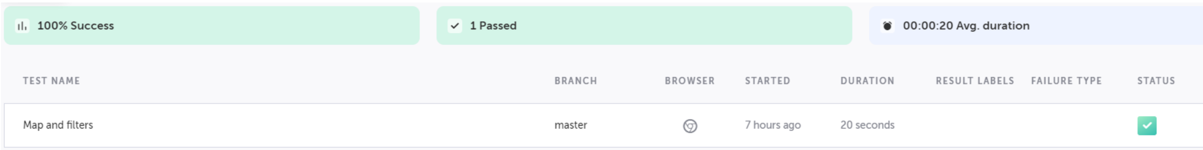

Figure 29. Load map and select filters test results

6 Conclusions and recommendations for future improvement

6.1 Conclusions

The purpose of this document was to present and describe the methodology to integrate two heterogeneous simulation models and to develop a visualisation and decision support tool that enables information sharing, common situational awareness and real-time collaborative decision-making between airports and ground transport stakeholders.

The implementation approach described in this document has demonstrated that it is possible to integrate the two different simulation models, thus being able to reconstruct the full door-to-gate and gate-to-door passenger journey. This enables the generation of a passenger-centric vision and creates the basis to implement the case studies described in 'D2.2 Specification of Case Studies' with the objective of validating and assessing the viability of the new concept of operations defined in deliverable 'D2.1 IMHOTEP ConOps', which proposes the extension of the A-CDM to the ground transport modes.

A first prototype of the visualisation and decision support tool has been developed which enables the user to visualise different KPIs that capture the most relevant aspects of the passenger journey. The tool also enables the user to assess the impact of different management actions on the airport and ground transport system performance.

6.2 Recommendations for future improvement

Within this document it has been demonstrated that integrating two heterogeneous simulation models able to reconstruct the full door-to-gate and gate-to-door passenger journey is feasible. In future stages, real-time integration between both models should be researched. This will enable the final user to freely test potential management actions that improve the transport system performance and visualise the results in real-time. Different methodologies can be investigated in order to reduce the computational time of the simulations, such as using meta-modelling techniques.

Additionally, the decision support tool should be validated by the stakeholders involved in the collaborative decision-making process. Significant work has been done in this respect. An internal workshop took place in the Palma de Mallorca airport on 19-20th of May 2022, where a demo of the prototype of the visualisation and decision support tool was presented. During the workshop, relevant feedback was provided by:

- The Palma de Mallorca ground transport operator (EMT)
- The Palma de Mallorca airport operator (AENA)
- Terminal modelling experts (Amsterdam University of Applied Sciences)
- Academic experts on digital aviation technology (Cranfield)

The possibility to include some of the collected suggestions in further stages of the project prior to the evaluation of the case studies will be analysed.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

The main suggestions collected at the workshop are the following:

- Some comparison between the results for the baseline scenario and the scenario after applying the management action are represented in different graphs, which makes comparative evaluation difficult. The reasoning behind this design decision was to avoid producing an information overload which could make the visualisation confusing for the user. However, some visualisation techniques can be explored in order to enable the inclusion of results in the same graph without penalisation for the final user.
- In some of the KPIs it would be interesting to include the modification rate between the baseline scenario and the scenario after applying the management action, such as in the CO_2 emissions, where the percentage reduction of the emissions could be a valuable indicator.
- The occupancy indicator should be measured in passengers per m^2 as it provides more relevant information on the status of the different airport areas.
- The choropleth map showing the occupancy displays a static image of the airport status, but sometimes it is more important to analyse the evolution of the performance. Therefore, it would be interesting to prepare a hybrid visualisation consisting of a map showing the occupancy at a precise time and a diagram showing the evolution of the occupancy during the complete simulation time.
- The choropleth map showing the occupancy should improve the visibility as it may confuse the user. Some of the ideas were:
    - Outline the different airport areas.
    - Highlight the occupancy colours so it is not mixed with the background colour.
    - Divide the colour range into: lower than the maximal occupancy (ramp of green colours) and over the maximal occupancy (ramp of red colours), so it will be easier for the user to see the conflict areas.
- Include a slicer for selecting the time in the choropleth map instead of a list selector.
- Include the possibility to visualise the total number of passengers who lost their flight or the percentage.
- It would be interesting that the final users could create their own customised view with the desired set of KPIs.
- The Scenario, Management action and View selectors could be hidden after the selection so the user will have more space in the screen to visualise the KPIs.
- Include an option to download the data as the involved stakeholders may want to personally analyse the data.

Although the feedback received is relevant for the improvement and the future real implementation of the tool, the work done will be shared with a wider audience in order to collect external feedback. A new demo session is expected to take place during IMHOTEP 2nd Stakeholder Workshop, where the tool will be presented to relevant stakeholders external to the project such as airport operators and airport organisations, airlines, ground transport operators, technology providers and academic experts.

EUROPEAN PARTNERSHIP

Co-funded by the European Union