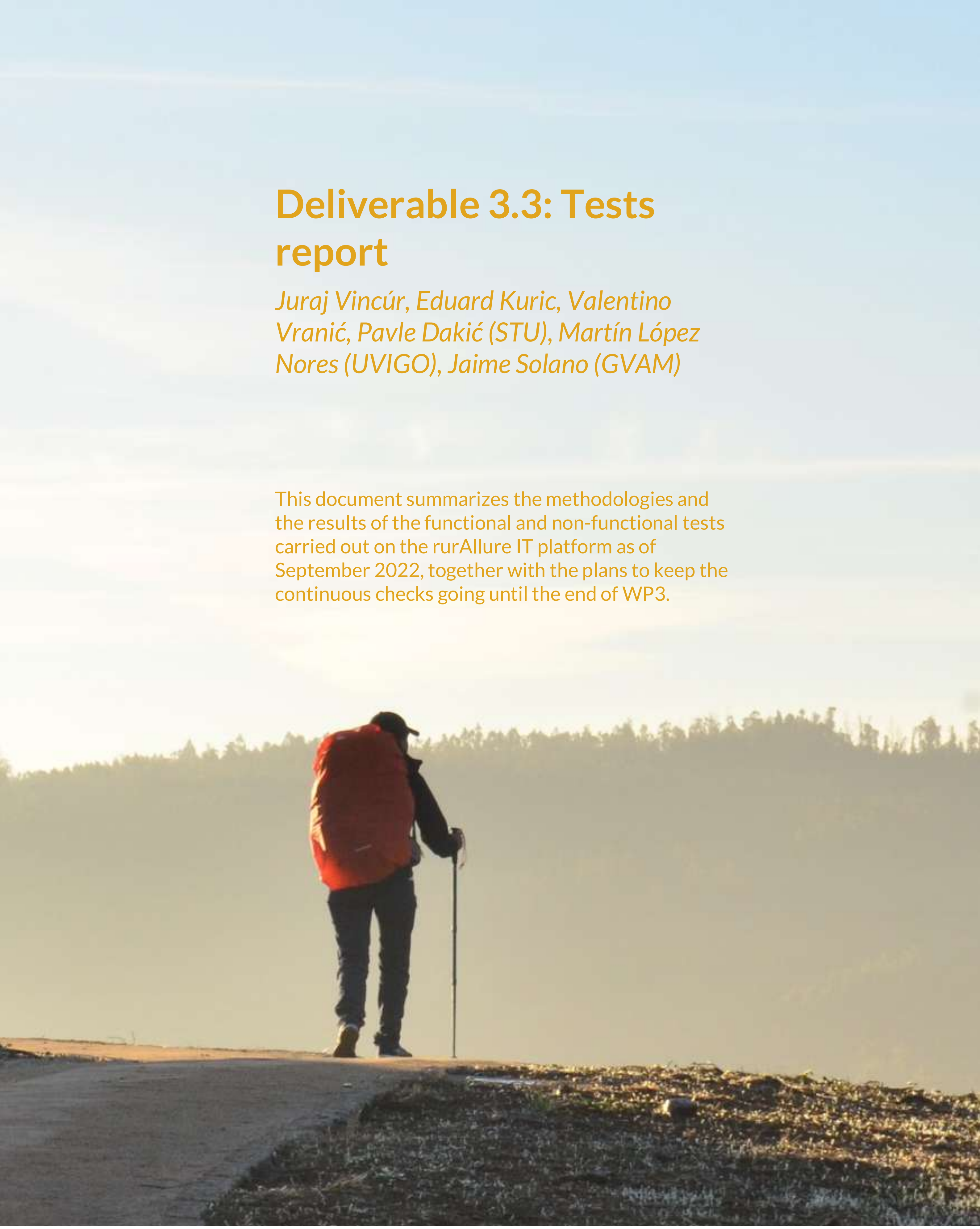


Deliverable 3.3: Tests report

Juraj Vincúr, Eduard Kuric, Valentino Vranić, Pavle Dakić (STU), Martín López Nores (UVIGO), Jaime Solano (GVAM)

This document summarizes the methodologies and the results of the functional and non-functional tests carried out on the rurAllure IT platform as of September 2022, together with the plans to keep the continuous checks going until the end of WP3.

This document is part of a project that has received funding from the **European Union's Horizon 2020** research and innovation programme under grant agreement No 101004887

rurAllure

reach out!

Project acronym	rurAllure
Full title	Promotion of rural museums and heritage sites in the vicinity of European pilgrimage routes
Grant agreement number	101004887
Funding scheme	Coordination and Support Action (CSA)
Work programme topic	SOCIOECONOMIC AND CULTURAL TRANSFORMATIONS IN THE CONTEXT OF THE FOURTH INDUSTRIAL REVOLUTION (H2020-SC6-TRANSFORMATIONS-2018-2019-2020)
Project start date	2021-01-01
Project duration	36 months
Work Package	WP3 – Technological platform
Deliverable lead organisation	STU
Authors	Juraj Vincúr, Eduard Kuric, Valentino Vranić, Pavle Dakić (STU), Martín López Nores (UVIGO) and Jaime Solano (GVAM)
Reviewers	Dészi Géza (KIFÜ), Sergio Arcay Mallo and Roi Martínez Portela (UVIGO)
Version	1.0
Status	For EU review
Dissemination level	Public
Due date	M21 (2022-09-30)
Delivery date	2022-09-30

Table of contents

1. Introduction

This document, Deliverable 3.3, is a report on the testing of the pilgrimage support platform being developed as a part of the rurAllure project. The work reported here was carried out within task 3.5 in WP3 since month 10 (November 2021) of the project realization, led by STU with UVIGO, GVAM and KIFÜ as participants. The task will continue in pursue of further improvements till the end of WP3 in M27 (March 2023), producing at least one update of this report.

The document is structured as follows:

- Section 2 reports on the extent of the implementation with respect to the use case model elaborated in the first months of the project.
- Section 3 describes the testing environment.
- Section 4 presents the functional testing performed and a summary of its results.
- Section 5 presents the non-functional testing, targeting cybersecurity, accessibility, usability, and performance.
- Section 6 presents the timeline of the testing.
- Section 7 presents a summary.

2. Review of Use Cases

The implementation of the rurAllure platform relied on the use case model presented in Deliverable 3.1 ("Platform requirements and design", M5). This model helped establish a common understanding of what the rurAllure platform should be and could be. The project partners strived for capturing what is feasible, but did not want to reject challenging opportunities. In addition, the original intent captured in the use cases evolved significantly during implementation, as is common in agile software development. Nevertheless, out of 40 use cases introduced in D3.1, the 22 deemed most important for the project pilots –and thus considered critical for the beta version released in Milestone 1 at M12– are fully implemented (some with minor changes). Implementation of 11 use cases is ongoing, 2 will be reviewed in light of the results of the first pilot gathering (documented in D2.2, "Conclusions and recommendations from pilot gathering" at M24), and only 5 were definitely cancelled as they were found to lack specific interest or to be achievable by means of integrations with third-party solutions.

It should be noted that, in many features, the implementation went beyond what was specified in use cases:

- The planner returns a list of the most relevant locations traversed in between two Points of Interest (POIs), to facilitate the user's navigation.
- The list of traversed locations is shown also in the cases that there are no POIs in the Content Management System between the origin and destination of a daily trip, to avoid the confusions that would arise when faced with empty lists.
- The users can browse and explore POIs on a map, rather than only through lists of form searches.
- The users can browse and search within lists of narratives for whatever contents and topics they may be interested in.

- The users can browse the whole contents of the narratives they select, regardless of their location.
- The planner can return a set of isochrones as polygons that help visualize the POIs reachable from the routes' official paths with 5-minute, 15-minute, 30-minute and 1-hour detours.[1]

We did not update the use case model as such since there was no need for this in further development. Consequently, producing such an artefact would not be lean. However, the initial use case model is still useful for following the status of the use cases, which follows below:

- Featured pilgrimage plans and places/activities of interest
    - UC Create a featured pilgrimage plan – implemented
    - UC Remove a featured pilgrimage plan – implemented
- Places/activities of interest
    - UC Manage POIs – implemented
    - UC Find a POI – implemented and extended by a map
    - UC Approve an activity/hospitality POI proposal – implemented, but not tested yet as platform interactions with vendors in the pilots are planned for 2023.
    - UC Propose an activity/hospitality POI – implemented, but not tested yet for the same reason as above.
    - UC Make corrections to a disapproved activity/hospitality POI – implemented, but not tested yet for the same reason as above.
- Basic pilgrimage planning
    - Getting an initial pilgrimage plan
        - UC Generate a pilgrimage plan – implemented
        - UC Adopt a pilgrimage plan – implemented
        - UC Find a featured pilgrimage plan – implemented
        - UC Find a pilgrimage plan – implemented
        - UC View a pilgrimage plan – implemented
    - Extending and adjusting pilgrimage plans
        - UC Extend a pilgrimage plan by POIs – implementation pending (to be reviewed once the input from the first pilot gathering has been collected and analysed, by M24)
        - UC Recommend POIs – implemented; recommending POIs only when asking for extra POIs in a given day of a plan

[1] *These visualizations are not used on the pilgrims' portal or the rurAllure mobile app in order not to clutter up the visualizations, but they can be used when embedding information in other websites (e.g., the update of the rurallure.eu website itself planned for January 2023).*

    - UC Make time adjustments to a pilgrimage plan – cancelled because of overly complex user interaction; no changes to plan features can be made for saved plans
    - UC Change the pilgrimage plan properties – cancelled because of lack of interest; no changes to plan features can be made for saved plans (rather, the same results can be attained more easily by requesting a plan with the intended features from scratch)
    - UC Change the pilgrimage plan accessibility – implementation pending (awaiting the termination of the analysis of quality and completeness of the route information available in open sources)
  - Adding services and narratives
    - UC Book an activity or service at a POI – implementation pending as vendor interactions in the pilots are planned in 2023 and some have expressed preference for existing booking solutions
    - UC Cancel a booked activity or service at a POI – implementation pending for the same reason as above
    - UC Pre-select multimedia narratives for a pilgrimage plan – implemented
    - UC Request joining a pilgrimage plan – implemented
    - UC Manage the pilgrimage plan members – implementation pending
    - UC Leave a shared pilgrimage plan – implemented
    - UC Transfer the pilgrimage plan ownership – cancelled for lack of interest
- Following pilgrimage plans and playing narratives
  - Following pilgrimage plans
    - UC Follow my daily pilgrimage plan – only viewing a plan implemented; the rest will not be implemented to keep user experience as simple as possible
    - UC Make a detour from a daily pilgrimage plan – cancelled to keep user experience simple
    - UC End up a daily pilgrimage plan with accommodations – cancelled to keep user experience simple (accommodations are treated as any other POIs, to be included during plan creation)
    - UC Rate a POI – implementation pending
  - Playing narratives
    - UC Play narratives following the daily pilgrimage plan – implementation pending
    - UC Play a pre-selected narrative – to be reviewed once the input from the first pilot gathering has been collected and analysed, by M24

- Communication
    - UC Send a message to the pilgrimage plan owner – implemented partially (microservice ready; integration pending with web portal and app)
    - UC Manage messages – implemented partially (microservice ready; integration pending with web portal and app)
    - UC Contact an emergency service – implementation pending

- Accounts
    - UC Register as a pilgrim – implemented
    - UC Register as a vendor – implemented
    - UC Log in – implemented
    - UC Update my account – implemented
    - UC Create a route administrator account – implemented
    - UC Manage user accounts – implementation pending
    - UC Find a user account – implemented only for the superadministrator; the rest will not be implemented as the feature was found to lack specific interest

3. Functional Testing

Unit Testing

The WP3 development and testing tasks have been conducted on a GitLab installation as explained in D1.1 ("Project handbook"). GitLab provides the GitLab CI/CD (Continuous Development, Continuous Integration) tool to implement various continuous methodologies such as test automation. After uploading new code to the repository of any microservice, the platform launches a new pipeline that automatically runs the test set and avoids publishing the new version if it detects any bugs in them. This method allows each new implementation to continue to meet minimum functional requirements and maintain the platform up and running.

Each microservice deployed in the environment has a /tests folder in its repository, with a set of unit tests that involve all the main functionalities within the corresponding microservice. Unit tests are small tests that check the functionality of individual components of the overall platform.

All the libraries and microservices defined here need a MongoDB database instance with sufficient information about points of interest, official routes, localities, etc. For the testing phase, there is a specific database, called "planner_test", which must be loaded and emptied before and after each unit test.

At the beginning of any test process, the function must, if necessary, load its own corresponding data into the test database. Each microservice has a set of dump files with different scenarios ready to be loaded into MongoDB. The following are some instances prepared for each one of the project's pilots:

- **dump_wp4_1_40_40:** This scenario is placed within the WP4 pilot. It contains a unique walking route from Verín to Santiago de Compostela (Spain) and includes 40 POIs reachable with a maximum detour of 40 km from the official paths of the Way of Silver.

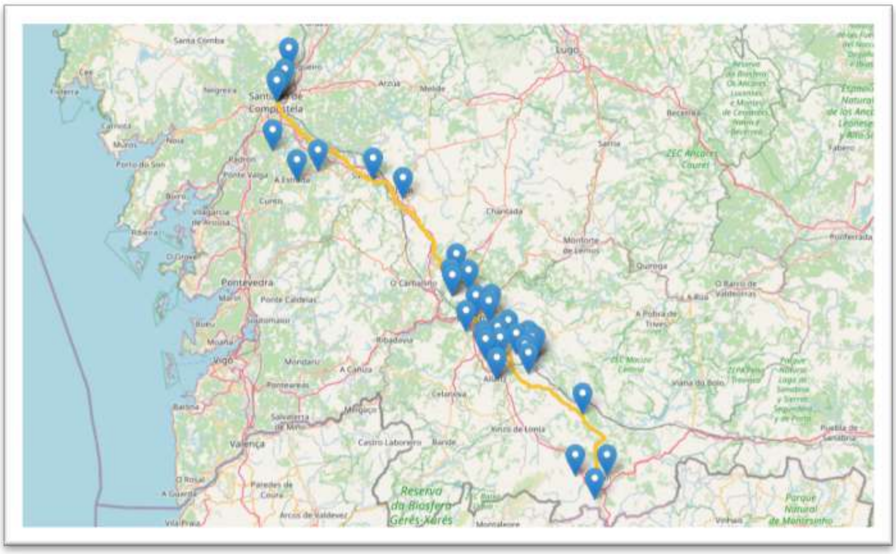


Figure 1. POIs on a dump scenario for WP4 ("Literary heritage on the ways to Santiago de Compostela").

- **dump_wp5_1_40_20:** This scenario is placed within the WP5 pilot. It contains a unique walking route from Ravenna to Arezzo (Italy) and includes 40 POIs reachable with a maximum detour of 20 km from the official paths of Via Romea Germanica.

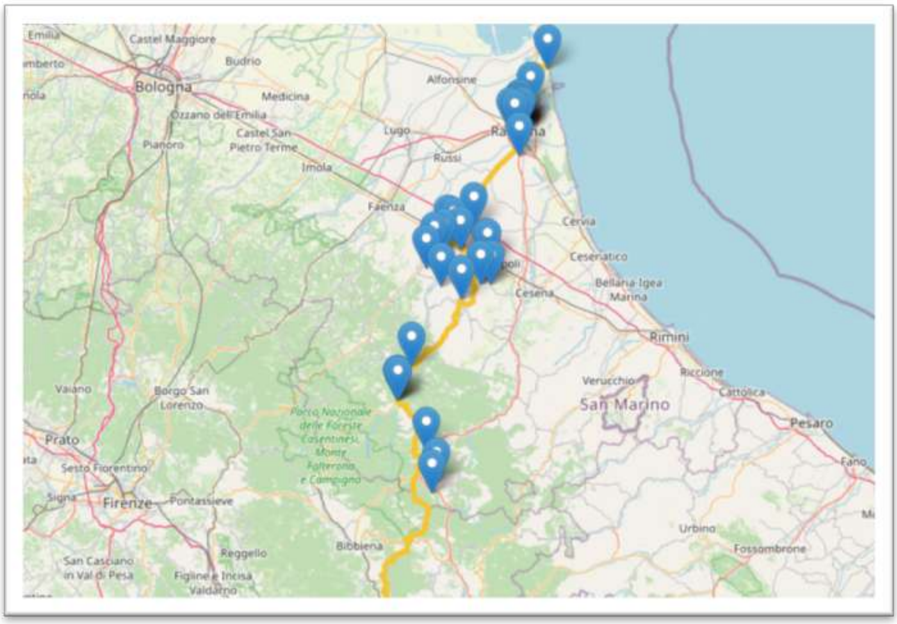


Figure 2. POIs on a dump scenario for WP5 ("Thermal heritage and others on the ways to Rome").

- **dump_wp6_40_10:** This scenario is placed within the WP6 pilot. It contains a unique walking route from Oppdal to Trondheim (Norway) and includes 40 POIs reachable with a maximum detour of 10 km from the official paths of Gudbrandsdalsleden.

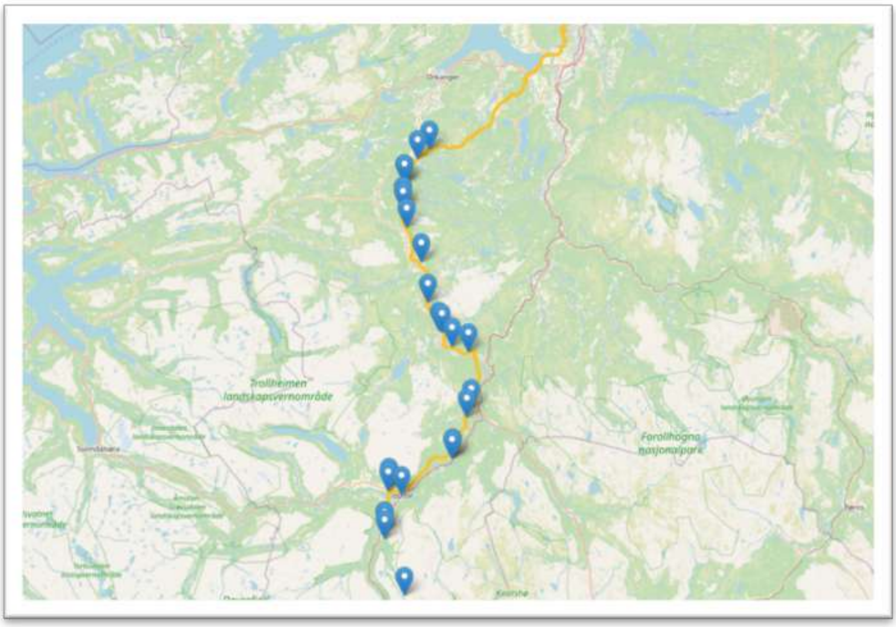


Figure 3. POIs on a dump scenario for WP6 ("Ethnographic heritage and the ways to Trondheim").

- **dump_wp7_1_40_40:** This scenario is placed within the WP7 pilot. It contains a unique walking route from Banská Bystrica (Slovakia) to Budapest (Hungary) and includes 40 POIs reachable with a maximum detour of 40 km from the official paths of Mária Út.

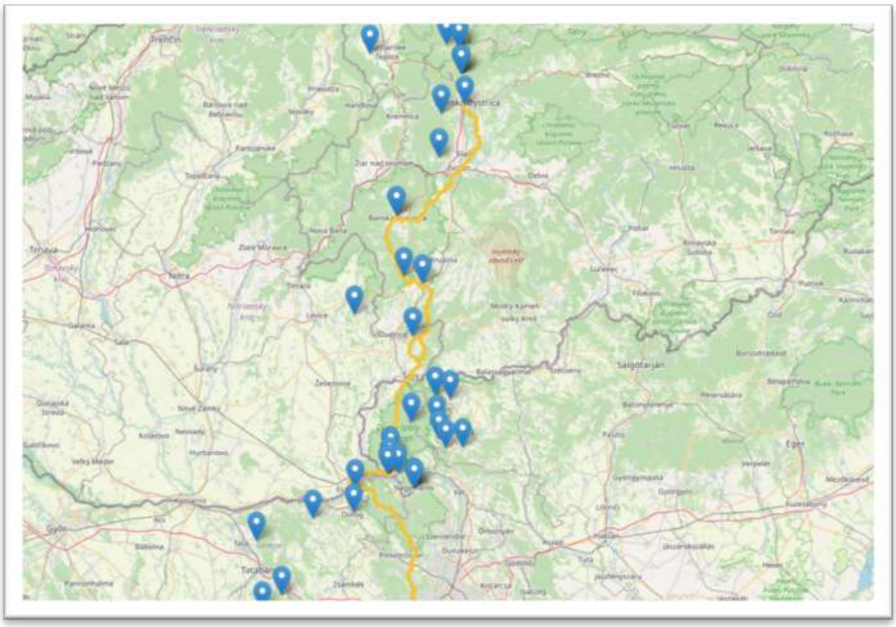


Figure 4. POIs on a dump scenario for WP7 ("Natural heritage and the ways to Csíksomlyó").

The choice of one dump or another depends on the specific purpose of the unit tests: the developers may choose to use one, several or none.

The following subsections contain samples of test documentation from the platform's microservices whose correct implementation depends more on unit testing.

Test Cases for the Planner Microservice

Table 1: UNIT TEST reachable POIs from partnership route.

TESTS.TEST_FUNCS.TESTREACHABLEPOISONROUTE.TEST_WP4_1_40_40	
Scenario	Scenario: <dump_wp4_1_40_40>
Function description	During the plan generation process, all the reachable points of interest along the user's traveling segment become in potential detours for the final plan. The results must be always equal for the same scenario, as the detour distances to reach a POI are the same.
Proposal	This test checks this functionality to find reachable POIs according to a given traveling distance.
Input	• route: preloaded section of the partnership route from Verín to Silleda.
Expected output	• The function returns a list with 14 POIs reachable from different locations of the provided segment. • The execution does not generate any exceptions.

Table 2: UNIT TEST new walking route from user's location to partnership route.

TESTS.TEST_FUNCS.TESTROUTETOPARTNERSHIP.TEST_WP4_1_40_40	
Scenario	Scenario: <dump_wp4_1_40_40>
Function description	At the starting phase of the plan generation process, the Planner algorithm calculates the best route from the initial user's location and the nearest point of the partnership route. The results may be slightly altered between different versions and implementations, but always close to the expected output.
Proposal	This test checks this functionality to find a valid route from the given location, or returns the appropriate alert when it is not possible to meet the user's requirements.
Input	• initial_position: (-7.8120, 42.3755) • transport: HIKING • maximum_distance: 40 km

Expected output	- The function returns a GeoJSON LineString from the location (-7.8120, 42.3755) to any point close to the end of the stage “ES16a-11a-ourense-cea_part001”, with a traveling distance lower than 40 kilometers. - The execution does not generate any exceptions.

Table 3: UNIT TEST segment of partnership route between two points.

TESTS.TEST_FUNCS.TESTROUTEONPARTNERSHIP.TEST_WP4_1_40_40	
Scenario	Scenario: <dump_wp4_1_40_40>
Function description	At the starting phase of the plan generation process, the Planner algorithm calculate and choose the best section of the partnership route between several possible origin and destination stages. The results may be slightly altered between different versions and implementations, but always close to the expected output.
Proposal	This test checks this functionality to find the best route between two locations of the partnership route, or returns the appropriate alert when it is not possible to meet the user’s requirements.
Input	- potential_origins: [“ES16a-09a-laza-xunqueira-de-ambia_part001”, “ES16a-09a-laza-xunqueira-de-ambia_part002”] - potential_destinations: [“S16a-15a-ponte-ulla-santiago_part001”, “ES16a-15a-ponte-ulla-santiago_part002”] - transport: HIKING - maximum_distance: 160
Expected output	- The function must return a new segment from the stage “ES16a-09a-laza-xunqueira-de-ambia_part002” to the stage “S16a-15a-ponte-ulla-santiago_part001”. - The execution must generate no exceptions.

Table 4: UNIT TEST generate a new pilgrim plan from user requirements, WP4 data.

TESTS.TEST_PLANNING_V4.TESTBUILDPLAN.TEST_WP4_1_40_40	
Preconditions	Scenario: <dump_wp4_1_40_40>
Function description	This function generates a new pilgrim plan according to the parameters required by the user.
Proposal	• This test verifies the whole process which the Planner algorithm uses to generate a new pilgrim plan from scratch. • Verifies the correct deployment of the pilot WP4 data.
Input	• Origin: Verín - (-7.4404, 41.9395) • Destination: Silleda - (-8.2471, 42.6984) • Start date: 20/01/2022 • End date: 20/02/2022 • Transport: HIKING • Traveling km per day: 40 • Preferences: [ ]
Expected output	The execution must generate no exceptions.

Table 5: UNIT TEST generate a new pilgrim plan from user requirements, WP5 data.

TESTS.TEST_PLANNING_V4.TESTBUILDPLAN.TEST_WP5_1_40_20	
Preconditions	Scenario: <dump_wp5_1_40_20>
Function description	This function generates a new pilgrim plan according to the parameters typed by the user.
Proposal	• Verifies the whole process which the Planner algorithm uses to generate a new pilgrim plan from scratch. • Verifies the correct deployment of the pilot WP5 data.
Input	• Origin: Ravenna - (12.2003, 44.4220) • Destination: Arezzo - (11.8707, 43.457) • Start date: 20/01/2022 • End date: 20/02/2022 • Transport: HIKING • Traveling km per day: 40 • Preferences: [ ]

Expected output	The execution must generate no exceptions.

Table 6: UNIT TEST generate a new pilgrim plan from user requirements, WP6 data.

TESTS.TEST_PLANNING_V4.TESTBUILDPLAN.TEST_WP6_1_40_10	
Preconditions	Scenario: <dump_wp6_1_40_10>
Function description	This is the main functionality of the Planner service. This function generates a new pilgrim plan according to the parameters typed by the user.
Proposal	• Verifies the whole process which the Planner algorithm uses to generate a new pilgrim plan from scratch. • Verifies the correct deployment of the pilot WP6 data.
Input	• Origin: Oppdal - (9.6926, 62.5908) • Destination: Trondheim - (10.3820, 63.4234) • Start date: 20/01/2022 • End date: 20/02/2022 • Transport: HIKING • Traveling km per day: 40 • Preferences: [ ]
Expected output	The execution must generate no exceptions.

Table 7: UNIT TEST generate a new pilgrim plan from user requirements, WP7 data.

TESTS.TEST_PLANNING_V4.TESTBUILDPLAN.TEST_WP7_1_40_40	
Preconditions	Scenario: <dump_wp7_1_40_40>
Function description	This is the main functionality of the Planner service. This function generates a new pilgrim plan according to the parameters typed by the user.
Proposal	• Verifies the whole process which the Planner algorithm uses to generate a new pilgrim plan from scratch. • Verifies the correct deployment of the pilot WP7 data.
Input	• Origin: Szob - (18.8700, 47.8179) • Destination: Budapest - (19.0447, 47.4986) • Start date: 20/01/2022 • End date: 20/02/2022

	- Transport: HIKING - Traveling km per day: 40 - Preferences: [ ]
Expected output	The execution must generate no exceptions.

Table 8: UNIT TEST recalculate the pilgrim plan information required by the algorithm from CMS model.

TESTS.TEST_PLANNING_V4.TESTREBUILDPLAN.TEST_WP4_1_40_40	
Preconditions	Scenario: <dump_wp4_1_40_40>
Function description	The Planner service needs very specific information for each POI about time and distances on the plan route. This data needs to be re-calculated before processing any request related to edits, localities or map display. The resulting pilgrim plan can be slightly different from the original one, but always maintaining the following fields unchanged: - Origin - Destination - List of selected POIs - Start date - End date - Transport
Proposal	This test verifies the process which the Planner algorithm uses to re-cover the physical route and the time and distance values uses during the generation phase.
Input	- base_plan: pilgrim plan between Ourense and Santiago de Compostela with 18 auto-selected POIs.
Expected output	- The function must return a new pilgrim path with exactly the same set of selected points of interest, origin, destination and number of days. - The execution must generate no exceptions.

Table 9: UNIT TEST obtain the list of localities the route pass through during the plan.

TESTS.TEST_PLANNING_V4.TESTBUILDLOCALITIES.TEST_WP4_1_40_40	
Preconditions	Scenario: <dump_wp4_1_40_40>
Function description	Besides the point of interests along the plan, the Planner service consults and returns the list of localities which the user pass through along the route.
Proposal	This test verifies the process which the Planner uses to retrieve and sort the set of localities within the plan.
Input	• base_plan: pilgrim plan between Ourense and Santiago de Compostela with 18 auto-selected POIs.
Expected output	The function must return a list of 5 localities, which correspond to the positions the user passes through while following the route between Ourense and Santiago: • Vilamarín • San Cristovo de Cea • Silleda • Sta Eulalia • Santiago de Compostela The execution must generate no exceptions.

Test Cases for the Recommender Microservice

Table 10: UNIT TEST calculate affinity value for a list of Points of Interest.

TESTS.TEST_RECOMMENDING. TESTFILTERPOIS.TEST_1	
Function description	This function filters and sorts a list of POIs according to a value of affinity with the user's requirements, calculated from three fields: • transport_method: means of transport used by the user. • suitable_for: what type of group the plan is intended for. • preferences_list: type of points of interest that are mostly contained in the plan (ex. Cultural, Religious, etc.) The results must be always equal, as the three fields defined below of the input POIs are the same.
Proposal	Verifies the Recommender service is able to calculate a correct affinity value for a Point of Interest.
Input	• Transport: HIKING

	• Suitable For: ALONE • Preferences: [“Culture”, “Heritage”] • Points of Interest: [{"id": "e14e9cef-da51-4e09-91d5-9b0c7118eb17", ...}, {"id": "a0ffff4b-8ef5-494d-a4af-245d006a7dc8", …}, {"id": "f49ba399-4c96-4039-990e-772e84feb7a4", …}]
Expected output	The function must return a list with all the provided points of interest, sorted by the affinity value calculated: • "e14e9cef-da51-4e09-91d5-9b0c7118eb17": 0.48 • "f49ba399-4c96-4039-990e-772e84feb7a4": 0.40 • "e14e9cef-da51-4e09-91d5-9b0c7118eb17": 0.08 The execution must generate no exceptions.

Test Cases for the Matchmaker Microservice

Table 11: UNIT TEST compare two pilgrim plans by searching for POI coincidence.

TESTS.TEST_FUNCS.TESTCOMPAREPLANSBYPOIS.TEST_1	
Function description	This function calculates a coincidence percentage between two pilgrim plans, according to their lists of points of interest. The results must be always equal, as the pilgrim plans are the same.
Proposal	Verifies the functionality for comparing plans via points of interest.
Input	• base_pilgrim_plan: pilgrim plan between Ourense and Santiago de Compostela. • compare_pilgrim_plan: pilgrim plan between Lalín and Silleda.
Expected output	• The function must return a coincidence value of 5.5%. • The execution must generate no exceptions.

Table 12: UNIT TEST compare two pilgrim plans by searching for dates coincidence.

TESTS.TEST_FUNCS.TESTCOMPAREPLANSBYDATES.TEST_1	
Function description	This function calculates a coincidence percentage between two pilgrim plans, according to their start and end dates. The results must be always equal, as the pilgrim plans are the same.
Proposal	Verifies the functionality for comparing plans via their dates.
Input	• base_pilgrim_plan: pilgrim plan between 13/03/2022 and 16/03/2022. • compare_pilgrim_plan: one-day pilgrim plan for 15/03/2022.
Expected output	• The function must return a coincidence value of 25%. • The execution must generate no exceptions.

Test Cases for the Narratives Microservice

Table 13: UNIT TEST Validate the format of the narrative Excel file.

TESTS.TEST_NARRATIVES.TESTNARRATIVESVALIDATION.TEST_1	
Function description	Before uploading a new narrative, this function verifies the excel file is well formed. The results must be always equal, as the file is the same.
Proposal	Verifies the functionality for checking the narrative file format.
Input	• narrative_file: path to an Excel file with errors.
Expected output	• The function must return a negative value. • The execution must generate no exceptions.

Table 14: UNIT TEST Extract narrative information from the Excel file.

TESTS.TEST_NARRATIVES.TESTREADEXCELFILE.TEST_1	
Function description	Before uploading a new narrative, this function processes the narrative Excel and extract all the data about the chapters. The results must be always equal, as the file is the same.
Proposal	Verifies the functionality for obtaining all the data contained in a narrative Excel file.
Input	• narrative_file: path to an Excel file with a valid format.

Expected output	The function must return a list with the narrative chapters defined in the Excel file. Each element must contain the existing data about: • Coordinates • Narrative text • Paths to the image files • Paths to the video files • Paths to the audio files • URLs to different sources The execution must generate no exceptions.

End-to-End (E2E) Testing

E2E testing has been employed to verify complete system flow and to increase test coverage of subsystems. A series of automatic tests that simulate user interaction with the pilgrim portal has been implemented. For these purposes, a browser automation framework named Selenium (www.selenium.dev) was employed. The E2E tests cover 9 use cases crucial from the pilgrim's perspective. The collection is being extended with the aim of covering all implemented use cases in which pilgrim is the primary actor.

A short description of the test cases is provided below, with the coverage shown in Table 15 below.

Table 15: E2E tests coverage.

USE CASE	TESTS COVERING THE SCENARIO
Generate a pilgrimage plan	TC01
Find a featured pilgrimage plan	TC02
Find a pilgrimage plan	TC03
Extend a pilgrimage plan by pois	TC04
Log in	TC01, TC05, TC06
View a pilgrimage plan	TC02, TC03, TC04
Leave a shared pilgrimage plan	TC05
Request joining a pilgrimage plan	TC05
Manage the pilgrimage plan members	TC06

Table 16: Test Case 01 Create and save a pilgrimage plan.

TC01_CREATESAVEPILGRIMAGEPLAN	
Preconditions	TC starts on loaded landing page. For the selected route there is sufficient number of POIs stored in the database. User account is prepared.
Description	Following user actions are being simulated: • Select route on the landing page • Fill and send planner form • Browse plan returned by system • Login • Save plan • Delete saved plan
Asserts	• Nonempty plan has been generated. • Plan has been saved. • Plan has been deleted.
Status	Passing with limitations
Comments	• Several form inputs generate empty plan. Although this behavior is expected, no reasonable response is provided to the user. • Origin and destination input fields are diacritic-sensitive, but several cities are stored with incorrect diacritics. • Sending form without all mandatory fields breaks planning functionality and page needs to be reloaded.

Table 17: TC02 Find a featured plan.

TC02_FINDFEATUREDPLAN	
Preconditions	TC starts on loaded landing page. Number of featured plans stored in DB is greater than 0.
Description	Following user actions are being simulated: • Select route on the landing page • Open view of featured plans • Open detail of the first plan • Close detail
Asserts	• Nonempty list of featured plans is returned. • The requested number of featured plans is returned. • Plan detail is not empty.
Status	Passing
Comments	

Table 18: TC03 Find a pilgrimage plan.

TC03_FINDPILGRIMAGEPLAN	
Preconditions	TC starts on loaded landing page. Number of pilgrim plans stored in DB is greater than 0.
Description	Following user actions are being simulated: • Select route on the landing page • Open view of pilgrim plans • Open detail of the first plan • Open map • Close map
Asserts	• Nonempty list of pilgrim plans is returned. • The requested number of featured plans is returned. • Plan detail is not empty. • Map for the plan loaded successfully.
Status	Passing
Comments	

Table 19: TC04 Extend pilgrimage plan by POIs.

TC04_EXTENDPILGRIMAGEPLAN	
Preconditions	TC starts on loaded landing page. For the selected route there is sufficient number of POIs stored in the database.
Description	Following user actions are being simulated: • Select route on the landing page • Fill and send planner form • Open list of additional POIs • Add selected POIs to the plan • Remove selected POIs from the plan • Clear plan for selected day • Revert changes to the plan
Asserts	• Number of additional POIs offered by system is greater than 0. • Added POI appears in the plan. • Added POI is removed from the list of additional POIs. • POIs removed from the plan are added to the list of additional POIs. • If plan is reverted, list of additional POIs is reverted as well. • If plan is cleared, all POIs from the plan are added to the list of additional POIs.
Status	Passing
Comments	Duplicate pilgrim plans are provided in results.

Table 20: TC05 Join and leave pilgrim plan.

TC05_JOINLEAVEPLAN	
Preconditions	Two user accounts are prepared. TC starts on page of the public plan of User 2 that allows other users to join. User 1 is logged in.
Description	Following user actions are being simulated: 1. User 1 clicks on "I want to join" 2. User 1 fills and send join request form 3. User 1 tries to fill and send join request second time 4. User 1 logs out 5. User 2 logs in 6. User 2 opens notifications and accepts the request 7. User 2 logs out 8. User 1 logs in 9. User 1 navigates to My plans section 10. User 1 switches to tab "I appear as a participant" 11. User 1 leaves the plan 12. User 1 load URL of the initial plan 13. User 1 clicks on "I want to join" 14. User 1 fills and send join request form
Asserts	• Sending second request in step 3 yield system error – joining request already exists. • User 2 received notification. • User 1 can rejoin plan in step 14.
Status	Failing
Comments	User 1 is not able to rejoin the plan in step 14, system thinks that join request already exists.

Table 21: TC06 Accept and remove pilgrimage plan member.

TC06_ACCEPTREMOVEPLANMEMBER	
Preconditions	Two user accounts are prepared. TC starts on page of the public plan of User 2 that allows other users to join. User 1 is logged in.
Description	Following user actions are being simulated: 1. User 1 clicks on "I want to join" 2. User 1 fills and send join request form 3. User 1 logs out 4. User 2 logs in 5. User 2 opens notifications and accepts the request 6. User 2 loads URL of the plan that this request belongs to 7. User 2 opens list of members 8. User 2 deletes the member (confirmation required) 9. User 2 opens list of members
Asserts	• User 2 received notification. • User 1 is in list of members after step 7. • User 1 is not in list of members after step 9.
Status	Passing
Comments	–

A list of identified issues that are pending is provided in Table 22. Follow-up of these issues will be provided in subsequent versions of this document.

Table 22: Pending issues of E2E testing.

DESCRIPTION	STATUS	SEVERITY
Sending form without all mandatory fields breaks planning functionality and page needs to be reloaded.	Pending	High
Origin and destination input fields are diacritic-sensitive, but several cities are stored with incorrect diacritics.	Pending	Medium
Several form inputs generate empty plan. Although this behavior is expected, no reasonable response is provided to the user.	Pending	Medium
Duplicate pilgrim plans are provided in results.	Pending	Low
User is not able to rejoin the plan.	Pending	Low

4. Non-functional Testing

Cybersecurity

Cybersecurity is a major concern in modern information systems, particularly for a platform like that of rurAllure as it is intended to provide service to hundreds or thousands of users, of whom some personal data are kept. Accordingly, the WP3 development works have been conducted by adopting recommended practices since the beginning. Nevertheless, a cybersecurity audit was started in M18 (June 2022) in order to obtain a thorough assessment of threats and vulnerabilities, as in commonly done before a software system goes into production.

There are multiple possible methodologies for conducting a cybersecurity audit. Most references fit the scenario of jointly auditing a web portal and a mobile app that operating against a common backend. The Open Web Application Security Project (OWASP) methodology has been chosen for having specific guidelines on how to realize and exploit vulnerabilities.

The testing strategy combines white-box testing of the elements for which the source code is available and of which the infrastructure is fully known and black-box testing of the CMS, as it has aspects that cannot be disclosed for being protected as background knowledge. The tests are being conducted against a mirror deployment of the platform to avoid interference with the operation of the web portal and the mobile app.

The following subsection explain the cybersecurity threats that are planned to assess in the cybersecurity audits. The report of the tests and the ensuing correcting actions (if any) will be documented in a subsequent update of this document.

Web portal analysis

The analysis of the web portal started out with a static scrutiny of the source code using the SonarQube tool (www.sonarqube.org), with the aim of detecting possible vulnerabilities at the code level. This will be followed by dynamic analyses carried out through the OWASP ZAP and Nikto tools, in order to identify vulnerabilities that are frequent in this type of application, such as Path Traversal, Local File Inclusion, Remote File Inclusion, etc.

Having cleared those checks, the audit of the web portal plans to look for the existence of OWASP Top 10 vulnerabilities:

- The first to be analyzed were vulnerabilities in the handling of input data, starting with SQL injections, which belong to the third most frequent group of vulnerabilities according to the latest OWASP report, A03:2021 Injection [OWA21c]. To address this analysis, the SQLMap tool (www.sqlmap.org) configured with multiple heuristics will be used to detect the possible types of injections present on the web platform, such as Boolean-based, error-based, union-query based, stacked queries, time-based blind, etc.
- The next step was to verify JavaScript code injections, which, like the vulnerabilities in the previous point, belong to the group of injections, A03:2021 Injection [OWA21c]. Specifically, the existence of Stored XSS, Reflected XSS and DOM XSS is being checked. To undertake this analysis, both manual payloads and automated tools, such as XSS Strike, will be used.
- Additionally, the security of the authentication mechanisms of the web portal will be analyzed, which covers the vulnerabilities positioned in the seventh most frequent position according to the list of OWASP, A07:2021 Identification and Authentication Failures

[OWA21f]. Specifically, the use of weak/default passwords, the effectiveness of brute force attacks, and the management of session identifiers will be evaluated.

- Related to the previous point, the second most frequent vulnerability, A02:2021 Cryptographic Failures [OWA21b], will also be evaluated. The aspects that will be studied related to this heading are the transmission of unencrypted information, the use of weak cryptographic keys and the use of outdated cryptography algorithms.
- As a complement to these tests, the authorization mechanisms will be checked. These tests focus on detecting A01:2021 Broken Access Control [OWA21a] vulnerabilities. Specifically, tests will be carried out for vulnerabilities such as Session Hijacking, URL Rewriting or Manipulation of Access Tokens, among others.
- Next, the state of the monitoring systems will be analyzed, with the aim of studying whether the logging of the information is sufficient or whether, on the contrary, there are vulnerabilities related to A09:2021 Security Logging and Monitoring Failures [OWA21g].
- Finally, the design and configuration of the platform will be checked, corresponding to vulnerabilities A04:2021 Insecure Design [OWA21d] and A05:2021 Security Misconfiguration [OWA21e].

Mobile application analysis

Following the approach of the web portal, in addition to certain basic aspects, the vulnerabilities collected in the OWASP Top 10 Mobile will be checked:

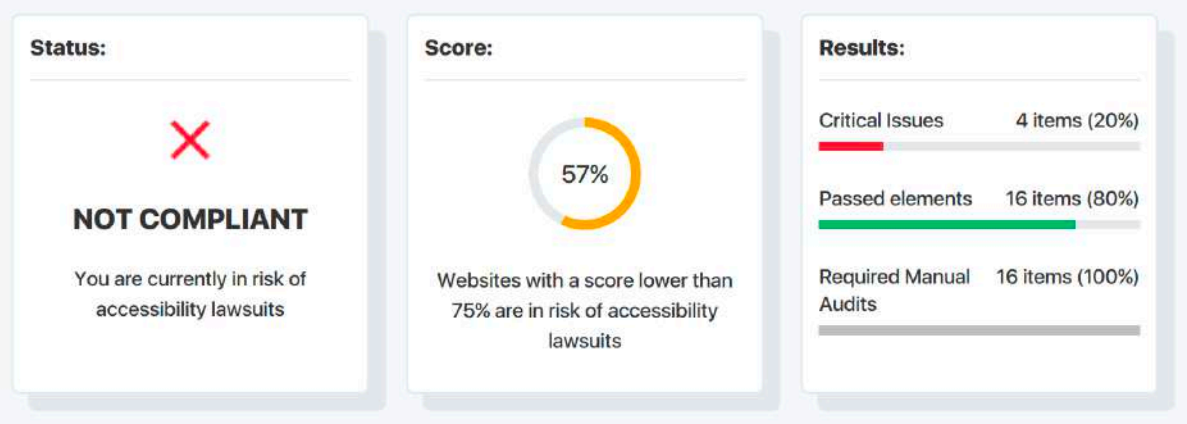
- First, the most frequent vulnerability, M1: Improper Platform Usage [OWA22a], will be analyzed, which is related to those collected in the OWASP Top 10 (SQL Injections, XSS, etc.). The verification of this type of vulnerabilities will be carried out by manual exploitation.
- The next item to audit are problems related to the configuration of the application. Specifically, it will be verified that information is being stored securely (M2: Insecure Data Storage [OWA22b]) and that communications with the rest of the platform are carried out using secure protocols (M3: Insecure Communication [OWA22c]).
- In the same way as in the case of the web portal, the authentication schemes used, M4: Insecure Authentication [OWA22d] and the security and robustness of the cryptography applied in the application, M5: Insufficient Cryptography [OWA22e] will be evaluated. This group of tests also includes the verification of authorization mechanisms [M6: Insecure Authorization], by checking the existence of login-bypass or mismanagement of access tokens [OWA22f].
- The last block of analysis will be the one related to the code of the mobile application itself. Specifically, three aspects will be studied: the quality of the code M7: Poor Code Quality, the possibility of carrying out M8: Code Tampering and the protection against reverse engineering M9: Reverse Engineering.

Microservices and 3rd party libraries

Regarding microservices, most of the security tests are already implicit in the tests carried out on the web portal and the mobile app. Therefore, in addition to the vulnerabilities that can be detected from the previous tests, the known vulnerabilities of third-party libraries and services that use these elements will also be verified.

Accessibility

Compliance with Web Content Accessibility Guidelines (WCAG) 2.1 has been checked using online tool Accessibility Checker (www.accessibilitychecker.org), followed by manual inspection. As shown in Figure 5, 4 critical issues were identified in M18 (June 2022) that must be assessed and potentially removed in the future:



Status:

✗

NOT COMPLIANT

You are currently in risk of accessibility lawsuits

Score:

57%

Websites with a score lower than 75% are in risk of accessibility lawsuits

Results:

Critical Issues	4 items (20%)
Passed elements	16 items (80%)
Required Manual Audits	16 items (100%)

Figure 5. Summary of the analysis performed on the pilgrims' portal by Accessibility Checker in June 2022.

- Background and foreground colors do not have a sufficient contrast ratio.
- Heading elements are not in a sequentially-descending order.
- Some image elements do not have [alt] attributes.
- Links do not have a discernible name.

The only complex issue is that of the first point, which may not be solvable because it relates to the interactive mapping libraries, which are provided by 3rd party services. Thus, WCAG level AA may not be reachable, and the goal has been set to ensure WCAG level A before M24 instead.

Usability

This section contains design of the user study aimed to evaluate scenarios related to the rurAllure pilgrims' portal. The study will be realized in the laboratory of user experience at Faculty of Informatics and Information Technologies of the Slovak University of Technology in Bratislava.

Methodology

The designed usability study is formative, with a qualitative character and targeting a low number of respondents. During the study following techniques were employed:

- direct observation of respondent (screen and gaze has been recorded),
- verbal protocol with eye tracking,
- verbal questionnaire with open-ended questions,
- informal conversation between respondent and moderator,
- SUS questionnaire.

Participants will take part in the experiment one at the time. In all tasks, participants of the experiment assume that they are in the role of a person, whose intention is to fulfill the goals of the given task. Tasks were defined to reflect the use cases.

The moderator of the experiment will guide each participant through the entire scenario, while other observers will be watching the experiment from the observation room. The moderator will introduce the participant to the room where the experiment will be performed. The observation will be based on a predetermined sequence of steps that the moderator had to follow for each of the participants:

1. Welcome the participant.
    a. Explain the purpose of the experiment (see moderator instructions in Annex I).
    b. Explain that we are not testing them, but the application.
2. Describe the process of testing.
    a. Participant performs several tasks (5 total).
    b. Length of the session is estimated to 45 minutes.
3. Explain the use of eye tracking technology.
    a. During the study eye tracking camera would be used, that would allow us to determine what are you looking at, which would be valuable information in terms of evaluation.
4. Note that recordings would be used for the purposes of this evaluation only.
5. Ask if participant has any questions.
6. Introductory questionnaire (answers are noted down by observers).
    a. How would you rate your experience in area of information and communications technology on scale 1-5, where 1 means no experience and 5 means extensive experience.
    b. How often do you go on pilgrimages? Who are you traveling with? What kind of activities do you prefer? How long routes do you prefer?
7. Calibration of the eye tracking device.
8. Start the recording.
9. The participant performs the tasks – with each task following steps are repeated:
    a. Instructions are read by moderator.
    b. The participant receives materials required to perform the task (paper form).
    c. Any questions from the participant are answered by the moderator.
    d. The participant performs the task. Due to applied verbal protocol (think-aloud) the participant comments how he perceives and process information in the application, what kind of actions does he perform and for what purpose.
    e. Additional information is obtained from the participant during the informal conversation. In this section, there is also space for answering possible questions from observers.

10. Final questionnaire:
    a. Would you recommend the portal to your friends? Answer on a scale of 1-10, where 1 is definitely not and 10 is definitely yes.
    b. How easy was to plan a route? Answer on a scale of 1-10, where 1 is very complicated and 10 is very easy.
    c. Could you name one thing that you considered as the most complicated? And one thing that you liked the most?
11. The participant fills out SUS questionnaire.

Identified usability issues will be reported per task. Three levels of severity will be assigned to each issue: low, medium, and high. When determining the degree of severity, it will be taken into account how many participants were slowed down by the issue or how many participants failed the task because of it. Issues will be organized into groups according to the task in which they occurred. Reporting format is provided in Table 23.

Table 23: Reporting format for usability issues.

TASK X – NAME OF THE TASK
Name of the issue
Description
Recommendation
Severity

Once the participants finish all tasks, they will answer questions from a standardized questionnaire that serves to evaluate the overall usability of the system. The name of the selected questionnaire is System Usability Scale (SUS) and it contains 10 statements about the tested system:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The participants will indicate the degree to which they agree with each statement on a 5-point scale, ranging from strongly disagree to strongly agree. The app's usability is reflected by 8 statements and 2 statements are dedicated to learnability. The SUS score is a value in a numerical interval between 0 and 100, where the result:

- less than 50 is unacceptable,
- between 50 and 70 is on the border of acceptability and
- higher than 70 is acceptable.

Based on the results of the questionnaires, we will obtain the average value of the SUS score. For completeness, we will also list the highest and lowest achieved SUS scores.

Tasks and Respondents

Next, Tables 24 to 28 specify the tasks devised for the usability study.

Table 24: Task 1 User registration.

Assignment	Create a new rurAllure web portal account.
Expected steps	1. The respondent clicks on the button “Sign in”. 2. The respondent fills out required fields and confirms the form.
Additional questions	

Table 25: Task 2 Family pilgrimage on a way to Csíksomlyó.

Assignment	Imagine that you are planning a pilgrimage with your family on a way to Csíksomlyó, while you also looking for a wide range of activities. You would like to start on October 1st in Banská Štiavnica, Slovakia, and plan to end your pilgrimage on October 3rd in Šahy.
Expected steps	1. On the landing page, the respondent choose tile “Ways to Csíksomlyó”. 2. In the form with headline “Start planning your route...”, the respondent set the field “Origin“ to Banská Štiavnica and “Destination” to Šahy. 3. The respondent sets “Date from” to October 1st and “Date to” to October 3rd. 4. In the group of radio buttons labeled “Who are you traveling with?”, the respondent chooses option “Family”. 5. In the group of checkboxes labeled “Activities preferences”, the respondent chooses several options. 6. The respondent submits the form by clicking on button “See your plan”.
Additional questions	

Table 26: Subtask 2a Pilgrimage plan modification.

Assignment	Imagine that you want to modify the pilgrimage plan returned by web portal. On the second day, you want to visit the Roman Catholic Church of St. George. instead of chapel Babba.
Expected steps	Following steps are being performed after finishing Task 2: 1. The respondent opens the detail of the second day of the plan by clicking on the button "2" in the section "Days". 2. The respondent remove point of interest "Chapel Babba" from the plan by clicking on the button "Delete point of interest". 3. The respondent clicks on the button "Add point of interest". 4. The respondent selects option St. George Roman Catholic Church, by clicking on button "Select". Note: Removal and addition of a point of interest can be performed in reversed order.
Additional questions	

Table 27: Task 3 Recommended pilgrimages on the Ways to Csíksomlyó.

Assignment	Imagine that you would like to travel by bike and you would like to inspire your pilgrimage plan by the recommended routes to Csíksomlyó.
Expected steps	1. On the landing page, the respondent chooses tile "Ways to Csíksomlyó". 2. The respondent selects in the main menu item "Prepare your trip" > "Recommended plans". 3. In the left section with label "Transport", the respondent selects radio button "Bicycle". 4. The respondent opens the detail of a recommended route by clicking on its name within the list of filtered routes.
Additional questions	

Table 28: Task 4 Joining existing pilgrimage plan.

Assignment	Imagine that you would like to join someone on a planned pilgrimage on the route Banská Štiavnica - Šahy, while his nationality should be Slovak.
Expected steps	1. On the landing page, the respondent chooses tile "Ways to Csíksomlyó". 2. The respondent selects in the main menu item "Prepare your trip" > "Pilgrim plans". 3. In the left section with label "Pilgrim nationality", the respondent selects option "Slovakia". 4. The respondent opens detail of the plan with name "Banská Štiavnica – Šahy". 5. The respondent clicks on the button "I want to join". 6. [User not signed in] The respondent sign in using the form that pops out. 7. Within modal window "Send join request", the respondent write short text message to the owner of the plan and clicks on the button "Send".
Additional questions	

The planned number of respondents is 6 (plus 2 substitutes). Respondents will be carefully chosen so they fairly represent the target group - selected respondents must have experience with pilgrimage planning. Expected duration of each session is 45 minutes and additional 15 minutes is reserved for debriefing and for the preparations for the next session (1 hour testing slots).

Performance

Performance testing was started by the time the beta version of the platform was available (M12, Milestone 1). The reference measurements for baseline testing were made under conditions of minimal background workload, while the system is handling the activity of a single user. As of M18, the plan was updated to measure average load times (client perspective) and resource utilization (server perspective: CPU utilization, memory utilization, disk usage). Thus, testing procedures were defined for the following:

- Load testing: performance metrics are measured under anticipated user loads (number of concurrent users).
- Stress testing: user load will be simulated to extreme values to identify maximum workload that can be handled by the system.

To simulate the user load, several workstations will be employed to act as injectors, where each workstation would emulate selected number of concurrent users by executing automatic E2E tests. The following performance metrics will be measured:

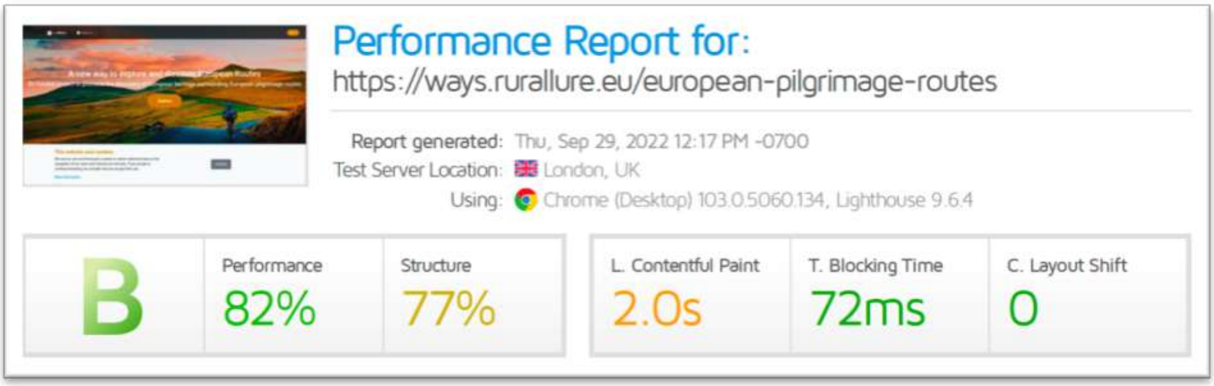
- First contentful paint: how quickly content like text or images are painted onto a page.
- Largest contentful paint: how long it takes for the largest element of content to be painted on a page.
- Time to interactive: how long it takes for a page to become fully interactive.
- Processor usage: amount of time processor spends executing non-idle threads.
- Memory usage: amount of physical memory available to processes.
- Disk time: amount of time disk is busy.
- Committed memory: amount of virtual memory used.

Initial measurements from the client perspective have been obtained by tools GTmetrix (www.gtmetrix.com) and Lighthouse (www.lighthouse-metrics.com). The results are provided in the table below, along with excerpts from the reports provided by both tools. A full update will be included in a subsequent version of this document before M27.

Table 29: Initial results provided by GTmetrix and Lighthouse.

METRIC	MEASURED VALUE LIGHTHOUSE	MEASURED VALUE GTMETRIX	RECOMMENDED VALUE
First contentful paint	3.3s	1.5s	0.9 or less
Largest contentful paint	4s	2s	1.2s or less
Time to interactive	3.8s	2.1s	2.5s or less

GTmetrix benchmark:

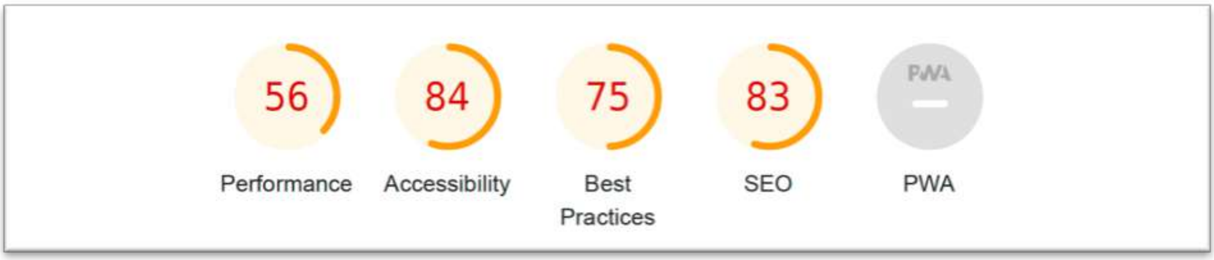


Performance Report for:
https://ways.rurallure.eu/european-pilgrimage-routes

Report generated: Thu, Sep 29, 2022 12:17 PM -0700
Test Server Location: London, UK
Using: Chrome (Desktop) 103.0.5060.134, Lighthouse 9.6.4

B	Performance	Structure	L. Contentful Paint	T. Blocking Time	C. Layout Shift
	82%	77%	2.0s	72ms	0

Lighthouse benchmark:



56	84	75	83	PWA
Performance	Accessibility	Best Practices	SEO	PWA

The extensive reports provided by both GTmetrix and Lighthouse include substantial hints for improvements, that will be fully considered until the end of WP3 in March 2023.

5. Timeline

M1 (Jan 2021, Project launch)

GitLab environment set up

M5 (May 2021)

Platform specification ready (D3.1)

- CI/CD tools set up + first set of unit tests defined.

M12 (Dec 2021, Milestone 1)

Beta version of the platform available (D3.2)

- Pilgrims' portal and apps + administrator and vendor portals + backend services.
- Performance baseline defined by first measurements.

M19 (July 2022)

Detailed plans for non-functional tests

- Cybersecurity
- Accessibility
- Usability
- Performance

M24 (Dec 2022)

New internal milestone

- Remaining use cases implemented + unit-tested.
- Solved issues of E2E testing. Complete analysis of threats and vulnerabilities.
- Accessibility level WCAG A achieved.
- Formative usability study completed.
- Completed performance assessment + revised goals.

M27 (Mar 2023)

End of WP3

- Updated test report (D3.3 v1.1)
- Solved cybersecurity and usability pitfalls.
- Microservices integrated with external websites and apps; CMS delivering content to third parties. Completed performance profiling as input for final exploitation plan (D2.6).

6. Summary

The work conducted to develop the rurAllure IT platform started out from the specification of use cases, that served to consolidate common understanding among the WP3 partners about the functionality to be provided. The implementation of the use cases was intertwined from the beginning with a CI/CD approach facilitated by the GitLab development environment. The functional tests (both unit and end-to-end) show that the implementation has been making progress steadily since the release of the first beta version in December 2021. The plans for comprehensive non-functional tests (addressing cybersecurity, accessibility, usability and performance aspects) were fully defined in M19 and will be addressed until the end of WP3 in M27 (March 2023). The results will ensure improved support to the pilots during 2023, as well as relevant input (particularly, in relation to performance) for the final exploitation plan, to be communicated as D2.6 ("Final strategy for exploitation of results") at the end of the project in M36 (December 2023).

7. References

- [OWA21a] OWASP Foundation. A01:2021 Broken Access Control. 2021. URL: https://owasp.org/Top10/A01_2021-Broken_Access_Control/ (vid. p. 25).
- [OWA21b] OWASP Foundation. A02:2021 Cryptographic Failures. 2021. URL: https://owasp.org/Top10/A02_2021-Cryptographic_Failures/ (vid. p. 25).
- [OWA21c] OWASP Foundation. A03:2021 Injection. 2021. URL: https://owasp.org/Top10/A03_2021-Injection/ (vid. p. 25).
- [OWA21d] OWASP Foundation. A04:2021 Insecure Design. 2021. URL: https://owasp.org/Top10/A04_2021-Insecure_Design/ (vid. p. 25).
- [OWA21e] OWASP Foundation. A05:2021 Security Misconfiguration. 2021. URL: https: //owasp.org/Top10/A05_2021-Security_Misconfiguration/ (vid. p. 25).
- [OWA21f] OWASP Foundation. A07:2021 Identification and Authentication Failures. 2021.
- URL: https://owasp.org/Top10/A07_2021-Identification_and_Authentication_ Failures/ (vid. p. 25).
- [OWA21g] OWASP Foundation. A09:2021 Security Logging and Monitoring Failures. 2021. URL: https://owasp.org/Top10/A09_2021-Security_Logging_and_ Monitoring_Failures/ (vid. p. 25).
- [OWA22a] OWASP Foundation. M1: Improper Platform Usage. 2022. URL: https://owasp. org/www-project-mobile-top-10/2016-risks/m1-improper-platform-usage (vid. p. 26).
- [OWA22b] OWASP Foundation. M2: Insecure Data Storage. 2022. URL: https://owasp.org/www-project-mobile-top-10/2016-risks/m2-insecure-data-storage (vid. p. 26).
- [OWA22c] OWASP Foundation. M3: Insecure Communication. 2022. URL: https://owasp. org/www-project-mobile-top-10/2016-risks/m3-insecure-communication (vid. p. 26).
- [OWA22d] OWASP Foundation. M4: Insecure Authentication. 2022. URL: https://owasp. org/www-project-mobile-top-10/2016-risks/m4-insecure-authentication (vid. p. 26).
- [OWA22e] OWASP Foundation. M5: Insufficient Cryptography. 2022. URL: https://owasp.org/www-project-mobile-top-10/2016-risks/m5-insufficient-cryptography (vid. p. 26).

- [OWA22f] OWASP Foundation. M6: Insecure Authorization. 2022. URL: https://owasp.org/www-project-mobile-top-10/2016-risks/m6-insecure-authorization (vid. p. 26).

Annex I: Moderator Instructions for the Usability Tests

Hello,

you probably already know why we are here today, but just to be sure; I will summarize all the essential information. BRIEFLY ABOUT RURALLURE. Today we would like to test some of the features, so we would need you to perform several tasks (X in total), while we will record your interaction with the application using a camera that can also record the direction of your gaze on the screen. We will use all recorded data only for the purpose of evaluation of the usability of the application.

We estimate that the entire session will take 45 minutes in total, but nothing will happen if the session would take longer or shorter. It's important to remember that we're not testing you, we're testing the app. Nothing you do can be wrong - on the contrary, your feedback will help us improve the app. Therefore, please try to behave as naturally as possible.

I will read each task to you, but you will be also able to read the instructions on a piece of paper. If the instructions are not clear to you, ask questions before you start working on the task.

If you will have questions during the task, please ask them after finishing the task, as we want to verify how you would interact with the application in a real situation, where you do not have the opportunity to ask someone how to proceed. If you don't know how to proceed with the task, please let me know and we will move to the next task. Also, let me know when you consider the task to be finished.

At the beginning, we would be interested in answers to the questions provided in a following short questionnaire.

reach out!

www.rurallure.eu