

PLATFORM VALIDATION

D21

Report

IDEALVis Consortium

<http://idealvis.inspirecenter.org/>



European Union

European Regional
Development Fund



Republic of Cyprus



Structural Funds
of the European Union in Cyprus



**ΙΔΡΥΜΑ
ΕΡΕΥΝΑΣ ΚΑΙ
ΚΑΙΝΟΤΟΜΙΑΣ**

Executive Summary

This deliverable makes up an essential milestone of the IDEALVis development lifecycle, as it describes the testing and validation procedures applied on all low-level components of the platform, as well as on the whole platform. More specifically, the deliverable demonstrates the software quality assurance measures undertaken to guarantee that the platform delivers the expected operations efficiently and effectively while concurrently maintaining an acceptable level of quality in terms of performance (i.e., the speed at which the software responds to the user's requests) and security (i.e., securing the users personal data from adversaries). The contents of this deliverable include: (i) details regarding the project's testing and validation strategy; (ii) how the testing and validation strategy was implemented; and (iii) the results collected during testing and validation.

Table of Contents

EXECUTIVE SUMMARY	2
LIST OF TABLES	4
LIST OF FIGURES.....	5
1 Introduction	6
2 Platform Overview	7
3 Testing and Validation Strategy	8
4 Identifying the Primary User Roles	9
4.1 Administrative User Roles.....	9
4.2 Basic User Roles	9
4.3 Managerial User Roles	10
5 Backend Services Testing	11
5.1 Functionality Tests	11
5.2 Security Tests	12
5.3 Performance Tests	13
5.4 List of Tested API Endpoints.....	14
6 Activity-based Testing.....	21
7 Conclusions	27

List of Tables

Table 1 - Endpoint Performance Accepted Timeframes	13
Table 2 - All Tested Endpoints.....	14
Table 3 – All-purpose Functionality Test Cases.....	22
Table 4 – Activity-based Scenario Test Cases	24
Table 5 - Activity-based testing Findings	26

List of Figures

Figure 1 - IDEALVis Platform's Private GitHub Repository Page	26
---	----

1 Introduction

The deliverable aims to describe the testing and validation strategy used for the IDEALVis platform, and report corresponding results extracted. Initially, an overview of the operational platform is provided. Next, the overall testing and validation strategy used for IDEALVis is described. Additionally, the primary user roles that interact with the IDEALVis platform and their primary functions are characterized. Finally, the deliverable provides the infrastructure used for testing and validating the functionality and the security of the platform at a system level; as well as the results collected by users through scenario/activity-based testing, aiming to validate the key functionalities that are essential for the platform's operation.

The deliverable is structured as follows.

- Platform Overview
- Testing and Validation Strategy
- Identifying the Primary User Roles
- Backend Services Testing
- Activity-based Testing
- Conclusions

2 Platform Overview

Before explaining the methodology adopted for testing and validating the operational IDEALVis platform, it is essential to provide an overview of the platform and its objectives. The overarching goal of the IDEALVis platform is to provide the business data analyst user with adaptive data visualizations that facilitate a more efficient and effective data exploration analysis of complex and multivariate business datasets, in order to enable more effective decision making on critical business tasks. The IDEALVis platform is built using one of the latest technology stacks available in the industry (ASP.NET Core and Angular 10), enabling the platform to have competitive performance while operating in an environment that was built with the principles of security, privacy and data protection by design. For achieving its goals, the platform has several interconnected backend components (e.g., Service Manager, Adaptation Engine, Data Visualization Engine, User Model, Analytic Algorithms Repository, Analysis Tracking Mechanisms etc.) and frontend components (e.g., Dashboard, Visualizations, Analysis Wizard, User Profile) that gracefully operate in order to deliver the promised adapted data visualization experience to the end user.

For a comprehensive overview of the development phases and they description of key software components, we refer the reader to deliverable D20.

3 Testing and Validation Strategy

Since the early development steps of IDEALVis, the development team made sure to adopt an agile development methodology, consisting of multiple development, testing and review iterations (i.e., development cycles) that lead to the finalization of all the required functionality, essential to IDEALVis. Moreover, the team structured the set of development tasks in a way that enabled the implementation of test case design, test planning, test execution and the collection of data for test validation reasons. More inclusively, the strategy adopted for testing and validation is made up by the following iterative steps:

- 1. Identifying the Primary User Roles:** The first step of the process involves understanding (i) which are the primary user roles (i.e., stakeholders) that interact with the platform and (ii) how they interact with platforms to complete their daily tasks. Clarifying the user roles and their interactions is essential for producing appropriate activity-based and backend service testing scenarios and procedures.
- 2. Backend Services Testing:** The second step of the process is handled by the development and quality assurance teams of the project and the goal is to formally test and validate all low-level processes (e.g., APIs or backend services) of the platform.
- 3. Activity-based Testing:** The third step of the process involves building and validating test cases regarding all user roles. The software development and quality assurance teams are responsible for the creation of effective test scenarios that cover all important functionality expected by the stakeholders. Moreover, the goal of this step is to make sure that the platform's operations are efficient and effective.
- 4. Outcomes and Conclusions:** The fourth and final step of the testing and validation process concludes the platform validation process by demonstrating important conclusions that were extracted from the outcomes of the abovementioned steps.

4 Identifying the Primary User Roles

The following subsections present all the end-user roles and how they interact with the platform. The user roles are categorized into three categories including: **administrative user roles** i.e., (user roles that are acquired by the consortium of the project for administration and data protection purposes), **basic user roles** i.e., (the most common users of the IDEALVis platform which are the business data analysts, which bare different levels of expertise and experience) and **managerial user roles** i.e., (users that have a managerial data analyst position entailing that they have subordinates). The testing and validation methodology's goal is to ensure that the provided functionality essential for the user-roles of the platform is effective and efficient.

4.1 Administrative User Roles

- **Admin:** This user role is responsible for i) viewing system statistics in a dashboard page for visually tracking key performance indicators (KPI), metrics and key data points; ii) managing other user accounts, i.e., setting the role and the user group of a newly registered user account; and iii) for creating user groups and organizations.
- **Researcher:** This user role is responsible for (i) creating research studies i.e., open or closed research studies; (ii) managing existing research studies i.e., publish, edit, and view research studies; (iii) inviting participants (i.e., data analysts) to participate in research studies for data collection i.e., the analyst participates and provides data in a research study, and the data they provide becomes a part of their user model; and (iv) for viewing statistics about research studies i.e., the researcher can see all data collected from user studies and export these data in a CSV format.
- **Data Protection Officer:** This user role is responsible for managing data related requests (i.e., regarding GDPR) that originate from all other user accounts of the platform. Specifically, this user role approves or denies (i) requests relating to the deletion of user accounts and (ii) requests relating to termination of data processing.

4.2 Basic User Roles

- **Data Analysts:** This user role can (i) participate in user studies and provide answers to questionnaires and/or solve psychometric tests as part of the user modeling process; (ii) respond to analysis tasks assigned to them by an team leader; (iii) execute data analytic reporting tasks (e.g., apply filters or descriptive models); (iv) apply and run multiple types of data analysis on a selected set of data; (v) report and view analysis results in different output formats (e.g., graphical and tabular) and export them in compatible formats such as PNG and CSV; (vi) pin analysis results on their personal dashboard or save the analysis

configuration privately or publicly for resuming later, or for sharing with other users; (vii) view analysis construction tracking reports for self-evaluation purposes (types of analysis and reporting tasks performed, including time spent for producing or for viewing each of these); and (viii) can view analysis report interaction tracking reports for self-evaluation, comparing the user's performance with the performance of other users.

- **Data Engineer:** This user role is focused on data preprocessing with the objective to improve the quality and trustworthiness of the data.

4.3 *Managerial User Roles*

- **Team Leader/Executive Analyst:** This user role is responsible and accountable for managing other analysts to monitor and assess the performance of specific items (e.g., brands, fraud patterns, risk factors). As such, they can (i) perform all functions that a data analyst can perform; (ii) define and assign analysis tasks to other data analysts.
- **Manager:** This user role can manage organizations, including the management of several team leaders, to monitor and assess the performance of the organization on multiple items, such as categories and sectors' performance, fraud detection performance, risk indices. As such, it coordinates team leaders to achieve specific business objectives. As such they have the same functionality as the Team Leader role.

Additionally, all the above user roles can (i) view or edit their profile information (e.g., name, profile picture etc.), and (ii) can change their settings for security (i.e., change a password) and privacy (i.e., request to download their data, request the deletion of their account or the termination of the processing of their data).

5 Backend Services Testing

Backend services testing aims to test all the low-level API endpoints of the platform. The goal is to validate all individual units or components of the platform to ensure that they work as expected and that each backend endpoint can be used in conjunction with other endpoints for forming larger business logic units and complex interactive business workflows. The tests that will be adopted at this phase implement a white box testing approach, meaning that the internal logic and structure of the component under validation/testing is known to the testers, which in this case are the software development and quality assurance teams.

Backend services testing was carried out in three consecutive phases as indicated below, using appropriate API testing/profiling tools:

- 1. Functionality Tests:** Verified that all endpoints deliver the appropriate and expected results according to the platform's requirements and specifications.
- 2. Security Tests:** Verified that each endpoint is secure and only accessible by authorized and/or authenticated users.
- 3. Performance Tests:** Verified that all endpoints of the system have adequate performance, and they do not negatively affect the user experience.

5.1 Functionality Tests

The goal of this phase is to verify that all endpoints deliver the appropriate and expected results according to the platform's requirements and specifications. The approach taken to validate the functionality for the different endpoint types is shown in the following steps:

1. For each endpoint using the POST or PATCH method, multiple requests were prepared, each with its payload (data in the form of JSON). The purpose of each different request required to test an endpoint below:
 - a. Provided an invalid payload (e.g., exclude a data field) to check the system resilience.
 - b. Provided a valid payload with an incompatible value in a data field (e.g., wrong user ID) to check the system resilience.
 - c. Provided a valid payload to check the system resilience.
2. For each endpoint using the GET or DELETE method, multiple requests were prepared as indicated below:
 - a. A request to a non-existent record, to check the system resilience and if the system throws an appropriate exception i.e., 404.
 - b. A request to an existent record, to check the system resilience.

More specifically the following actions/operations were tested against each applicable endpoint type as appropriate:

- **Update Operations:** This type of action regards testing for the successful deletion, creation, or update of a single existent record.
- **Read Operations:** This type of action regards testing for the successful retrieval of a single existent record from an endpoint e.g., the user's profile information.
- **List Operations:** This type of action regards testing for the successful retrieval of 0 (i.e., null) or more records. Such operations return the requested data in a list format and are mostly used for populating tables or dropdown lists with the appropriate information.
- **Custom Operations:** This type of action tests for the successful execution and response validity of an endpoint that has a very specific type of purpose. Such endpoints can perform multiple actions/operations at once (e.g., delete and insert a record) according to their purpose and specification. Examples of complex operation are the ones executed by the endpoints related to the Analysis Wizard, which receive multiple parameters through the request (defining an analysis) and return a response based on what was queried, analyzed, filtered, or aggregated etc. from the selected dataset.

Finally, all responses (i.e., returned data, status codes and messages) received from the endpoints were recorded and were validated according to the platform's specification. For example, if a request was invalid (e.g., by requesting a non-existent record) the platform had to respond with a 404 error and an appropriate error message.

5.2 Security Tests

Once the functionality aspect of the platform was tested and verified the second phase was to perform security tests. The goal of this phase is to verify that each endpoint is secure and only accessible by authorized and/or authenticated users according to the endpoint's specification. Several security metrics were tested against each endpoint making sure that all endpoints are protected from unauthorized and/or unauthenticated users or malicious actors. Each of the platform's endpoints adheres to specific access control rules i.e., there is only a predefined set of user roles allowed to call a specific endpoint/or an endpoint is open for access irrespectively of authentication, authorization, or user role (e.g., login endpoint). A simple security rule is that no user should be able to process a data related request (i.e., GDPR request) using the `/_api/data-protection-requests/change-request-status` endpoint, other than a user that bares the Data Protection Officer role. Accordingly, two sets of requests were created for each endpoint; on one hand the first set of requests was trying to validate that only the "appropriate" users can access each endpoint, and on the other hand the second set of requests included penetration testing requests that were trying to break the system's security e.g., by letting a non-authorized request execute successfully.

5.3 Performance Tests

With the completion of functionality and security testing phases the last phase of testing for the backend services was performance testing (i.e., covering non-functional requirements such as execution speed). The goal of this phase was to verify that all endpoints of the system can achieve adequate performance so that the user experience is not negatively affected. One of the major performance tests applied to the platform was the execution of various analysis requests, each resulting in more than 1000 records of data. This process verified that the execution time for each analysis was under the accepted timeframe of 5 seconds. All endpoints of the system were tested and found to be within the acceptable execution timeframe depending on their operation type (e.g., Create Operation, Data Analysis Operation). Table 1 summarizes all accepted endpoint performance timeframes for different types of operations.

Table 1 - Endpoint Performance Accepted Timeframes

Type of Operation	Description	Accepted Timeframe (In Seconds)
CRUD i.e., create, read, update, and delete operations	This operation encompasses all requests related to simple create, read, update, and delete operations of a specific database entity (e.g., creating a new Analysis Task).	0-2s
Data Analysis operations	This operation encompasses all analytical requests made from the user via the Analysis Wizard and other analytical operations that run within dashboards.	0-5s
Search operations	This operation encompasses all information retrievals, where users are searching for data in a datable.	0-2s
Load Page operations	This operation considers the loading time of all pages (this does not include components within a page that load data asynchronously e.g., data tables)	0-1s

5.4 List of Tested API Endpoints

This subsection presents all API Endpoints tested using the above test phases. Endpoints are categorized by their parent controller in the following table:

Table 2 - All Tested Endpoints

Controller Name	Controller Description	Endpoints	Endpoint Description	Required Input for Testing	# Of Tests	Resulting Outcome
UserGroupsController	Responsible for managing the available user groups that exist in the platform	POST/_api/user-groups	Create a User Group	Organization Id, Name	5	Success
		GET/_api/user-groups	Get all User Groups	None	5	Success
		GET/_api/user-groups/for-dropdown	Get all User Groups for Dropdown List	None	5	Success
		GET/_api/user-groups/{id}	Get a specific User Group	User Group Id	5	Success
		PATCH/_api/user-groups/{id}	Update a User Group	User Group Id, Organization Id, Name	5	Success
		DELETE/_api/user-groups/{id}	Delete a User Group	User Group Id	5	Success
		GET/_api/user-groups/{id}/for-update	Get a specific User Group's details for the update page	User Group Id	5	Success
ReflectiveAnalyticsController	This controller is responsible for producing the Analysis Tracker Dashboard	GET/_api/reflective-analytics/get-analysis-tracker-dashboard	Returns all information required for the statistics shown on the Analysis Tracker Dashboard	None	5	Success
		POST/_api/reflective-analytics/save-analysis-result-tracking-record-for-reflection	Saves an analysis report specification for analysis exploration tracking	Analysis report specification and Start and End Time of user's focus on the report	2	Success
OrganizationsController	Responsible for managing the available organizations that exist in the platform	POST/_api/organizations	Create an Organization	Organization Name	5	Success
		GET/_api/organizations	Get all Organizations	None	5	Success
		GET/_api/organizations/for-dropdown	Get all Organizations for Dropdown List	None	5	Success
		GET/_api/organizations/{id}	Get a Specific Organization	Organization Id	5	Success
		PATCH/_api/organizations/{id}	Update Organization	Organization Id, Name	5	Success
		DELETE/_api/organizations/{id}	Delete Organization	Organization Id	5	Success
		GET/_api/organizations/{id}/for-update	Get specific Organization's details for the update page	Organization Id	5	Success
AccountController	Responsible for handling all the user profile (i.e., data and security) related aspects for all user accounts	GET/_api/account/download-user-information	Downloads a JSON file with all user's personal information maintained by the platform	None	5	Success
		POST/_api/account/confirm-data-protection-request	Allows the user to verify a data protection request	Data Protection Request Type, One Time Password	5	Success

	GET/_api/account/data-protection-status	Loads data protection request status	None	5	Success
	POST/_api/account/cancel-data-protection-request	Cancels a confirmed data protection request	Data Protection Request Id	5	Success
	POST/_api/account/create-data-protection-request	Submits a new data protection request	Data Protection Request Type	5	Success
	POST/_api/account/change-password	Changes the user's password	Current and New Password	5	Success
	POST/_api/account/create-password	Creates a password for an account that only uses an external login provider	New Password	5	Success
	GET/_api/account/user-has-password	Checks if the user has a password set for the account	None	5	Success
	GET/_api/account/two-factor-status	Loads two factor authentication setup status	None	5	Success
	GET/_api/account/two-factor-key	Returns required keys and URLs for setting up two factor authentication	None	5	Success
	POST/_api/account/enable-two-factor	Enables two factor authentication for the user's account	2FA Code	5	Success
	POST/_api/account/disable-two-factor	Disables two factor authentication for the user's account	Current Password, 2FA Recovery Code	5	Success
	POST/_api/account/get-new-recovery-codes	Provides the user with new two factor authentication codes	2FA Code and Recovery Code	5	Success
	POST/_api/account/two-factor-forget-device	Removes a device used for two factor authentication	None	5	Success
	POST/_api/account/two-factor-remember-device	Add a new device for 2FA	2FA Code and Recovery Code	5	Success
	GET/_api/account/external-login-status	Retrieves the associated external logins for the user	None	5	Success
	POST/_api/account/delete-external-login	Removes an external login from the user's account	Login Provider Name, Provider Key	5	Success
	POST/_api/account/send-message-to-support	Allows the user to contact IDEALVis support	Message Body, Message Subject	5	Success
	GET/_api/account/user-model	Returns an object containing the user's user model information	None	5	Success
	POST/_api/account/update-personal-information	The user can update their personal information (currently only first and last name are supported)	User's First and Last name	5	Success
	POST/_api/account/mark-tutorial-as-seen	Marks the user's profile as having completed the tutorial that appears	None	5	Success

AuthController	Responsible for handling the authorization of a user on the platform	GET/_api/auth/status	Retrieves the authentication status of the user	None	5	Success
		POST/_api/auth/login	Authenticates the user with credentials	Username, Password and Remember Me	5	Success
		POST/_api/auth/login-two-factor	Authenticates the user with 2FA	2FA Code, Remember Me and Remember Device	5	Success
		POST/_api/auth/login-two-factor-recovery	Authenticates the user with 2FA recovery	2FA Code	5	Success
		POST/_api/auth/logout	Logs the user out of the system	None	5	Success
		POST/_api/auth/register	Allows the user to register an account on the platform	Email and Password	5	Success
		POST/_api/auth/external-provider-register	Allows the user to register an account on the platform using an external provider	Email	5	Success
		POST/_api/auth/resend-email-confirmation	Manual trigger for sending an email confirmation email	Email	5	Success
		POST/_api/auth/confirm-email	Enables the user to confirm their new account's email	Email and Confirmation Code	5	Success
		POST/_api/auth/request-password-reset	Allows the user to request a password reset	Email	5	Success
		POST/_api/auth/complete-password-reset	Allows the user to reset their password	Email, Confirmation Code and Password	5	Success
ExternalLoginController	Responsible for enabling users to login/register using an external provider	GET/external-login/Microsoft	Used for authenticating with Microsoft as the external login provider	Return URL	5	Success
		GET/external-login/callback	Endpoint called by Microsoft's server for authenticating the user	Return URL and Remote Error (This is called automatically by Microsoft)	5	Success
EditableUsersController	Responsible for providing an administrator with information about the registered users for management purposes	GET/_api/editable-users	Retrieves all users	None	5	Success
		GET/_api/editable-users/{id}	Retrieves a specific user's details	User's Id	5	Success
		PATCH/_api/editable-users/{id}	Used for updating a user's role, email, and user group	User's Id, Email, Role Id and Group Id	5	Success
		GET/_api/editable-users/{id}/for-update	Get specific User's details for the update page	User's Id	5	Success
DataProtectionRequestsController	Enables the data protection officer to manage data protection requests	GET/_api/data-protection-requests	Lists all available data protection requests	None	5	Success
		POST/_api/data-protection-requests/change-request-status	Updates the status of a protection request	Request Id and New Status	5	Success
DashboardController	Responsible for	POST/_api/dashboard/card_entries	Adds a new analysis	Card Name and Analysis Report	5	Success

	allowing the user to manage their personal dashboard configuration		report to the user's personal dashboard	Specification		
		GET/_api/dashboard/card_entries/{currentAnalysisTaskId}	Returns all analysis reports that are saved on the user's personal dashboard for the current analysis task	The user's currently selected Analysis Task Id	5	Success
		PATCH/_api/dashboard/card_entries	Updates the configuration of all card entries on the user's personal dashboard	List of card entry objects	5	Success
AnalysisTasksController	Responsible for enabling a Team Leader to manage data analysis tasks and for allowing all users to see and respond to available tasks	POST/_api/analysis-tasks	Create an Analysis Task	Name, task question, Users Ids, is response text-based?, correct text answer, complete by date, and list of possible multiple-choice answers	5	Success
		GET/_api/analysis-tasks	Get all Analysis Tasks	None	5	Success
		GET/_api/analysis-tasks/{id}	Get a specific Analysis Task	Analysis Task Id	5	Success
		PATCH/_api/analysis-tasks/{id}	Update an Analysis Task	Analysis Task Id, Name, task question, Users Ids, is response text-based?, correct text answer, complete by date, and list of possible multiple-choice answers	5	Success
		DELETE/_api/analysis-tasks/{id}	Delete an Analysis Task	Analysis Task Id	5	Success
		GET/_api/analysis-tasks/{id}/for-update	Get a specific Analysis Task's details for the update page	Analysis Task Id	5	Success
		GET/_api/analysis-tasks/users-dropdown				
		GET/_api/analysis-tasks/list-daily-tasks	Provides the user with the set of Analysis Tasks they need to perform	None	5	Success
		POST/_api/analysis-tasks/daily-task-respond	Stores the user's response to an Analysis Task	Text Answer, Selected Answer Id and Analysis Task Id	5	Success
		GET/_api/analysis-tasks/current-analysis-task	Retrieves the user's current analysis task characteristics (i.e., question, instructions etc.)	None	5	Success
		POST/_api/analysis-tasks/set-current-analysis-task	Sets a new Analysis Task as the current task for the user	The Id of a specific Analysis Task	5	Success
AnalysisFlowController	Responsible for enabling all users to manage their saved analysis flows i.e., wizard configurations	GET/_api/analysis-flow/{analysisFlowId}	Retrieve a Wizard Configuration	Analysis Flow Id	15	Success
		POST/_api/analysis-flow	Save a Wizard Configuration	Wizard Configuration, Name and Is Public?	15	Success
		PATCH/_api/analysis-flow/{id}	Update a Wizard Configuration	Analysis Flow Id, Wizard Configuration, Name and Is Public?	15	Success
		DELETE/_api/analysis-flow/{id}	Delete a Wizard Configuration	Analysis Flow Id	15	Success

		GET/_api/analysis-flow/for-dropdown	Provides the user with a list of available Wizard Configurations	None	15	Success
AnalysisEngineController	Responsible for providing the Analysis Wizard interface with the appropriate data/information	GET/_api/analysis-engine/datasets-for-dropdown	Get a list of available datasets	None	25	Success
		GET/_api/analysis-engine/data-attributes-for-selector/{datasetId}	Get a list of available data attributes for a given dataset	Dataset Id	25	Success
		GET/_api/analysis-engine/dataset-info/{datasetId}	Get information for a given dataset	Dataset Id	25	Success
		GET/_api/analysis-engine/dataset-descriptive/{datasetId}	Get descriptive statistics for a given dataset	Dataset Id	25	Success
		GET/_api/analysis-engine/attribute-descriptive	Get descriptive statistics for a given data attribute	Dataset Id and Attribute Id	25	Success
		GET/_api/analysis-engine/attribute-unique-values	Returns a list of unique attribute's values. Used for loading possible values when filtering a specific attribute	Dataset Id and Attribute Id	25	Success
		POST/_api/analysis-engine/produce-analysis-guidance	Given the analysis specification the user is constructing this returns a set of instructions guiding the user on what are the minimum inputs required for the selected analysis method	Analysis report specification	25	Success
		POST/_api/analysis-engine/execute-analysis-flow-configuration	Executes a Wizard Configuration and returns an analysis report	Dataset Id, list of Attribute Ids, analysis method and list of filters	50	Success
BUSINESS_ROLE_EXPERTISE_TESTController	This controller is part of the user modeling system and is responsible for enabling the user to manage their responses regarding the Business Role Expertise Questionnaire	POST/_api/BUSINESS_ROLE_EXPERTISE_TEST/InsertBUSINESS_ROLE_EXPERTISE_TEST	Add a new response to this questionnaire	Various form fields of several data types, Research Test Id and Research Study Id	5	Success
		POST/_api/BUSINESS_ROLE_EXPERTISE_TEST/UpdateBUSINESS_ROLE_EXPERTISE_TEST	Update a previous response to this questionnaire	Various form fields of several data types, Research Test Id and Research Study Id	5	Success
		POST/_api/BUSINESS_ROLE_EXPERTISE_TEST/DeleteBUSINESS_ROLE_EXPERTISE_TEST	Delete a previous response to this questionnaire	Various form fields of several data types, Research Test Id and Research Study Id	5	Success
		POST/_api/BUSINESS_ROLE_EXPERTISE_TEST/GetBUSINESS_ROLE_EXPERTISE_TEST	Retrieve a previous response to this questionnaire	Research Test Id and Research Study Id	5	Success
DATA_ANALYSYS_TASKS_TESTController	This controller is part of the user modeling system and is responsible for enabling the user to manage their responses regarding the Data Analysis	POST/_api/DATA_ANALYSYS_TASKS_TEST/InsertDATA_ANALYSYS_TASKS_TEST	Add a new response to this questionnaire	Various form fields of several data types, Research Test Id and Research Study Id	5	Success
		POST/_api/DATA_ANALYSYS_TASKS_TEST/UpdateDATA_ANALYSYS_TASKS_TEST	Update a previous response to this questionnaire	Various form fields of several data types, Research Test Id and Research Study Id	5	Success
		POST/_api/DATA_ANALYSYS_TASKS_TEST/DeleteDATA_ANALYSYS_TASKS_TEST	Delete a previous response to this	Various form fields of several data types, Research Test Id and Research	5	Success

	Tasks Questionnaire		questionnaire	Study Id		
		POST/_api/DATA_ANALYSYS_TASKS_TEST/GetData_ANALYSYS_TASKS_TEST	Retrieve a previous response to this questionnaire	Research Test Id and Research Study Id	5	Success
DATA_VIS_EXPERIENCE_TESTController	This controller is part of the user modeling system and is responsible for enabling the user to manage their responses regarding the Data Visualization Experience Questionnaire	POST/_api/DATA_VIS_EXPERIENCE_TEST/InsertDATA_VIS_EXPERIENCE_TEST	Add a new response to this questionnaire	Various form fields of several data types, Research Test Id and Research Study Id	5	Success
		POST/_api/DATA_VIS_EXPERIENCE_TEST/UpdateDATA_VIS_EXPERIENCE_TEST	Update a previous response to this questionnaire	Various form fields of several data types, Research Test Id and Research Study Id	5	Success
		POST/_api/DATA_VIS_EXPERIENCE_TEST/DeleteDATA_VIS_EXPERIENCE_TEST	Delete a previous response to this questionnaire	Various form fields of several data types, Research Test Id and Research Study Id	5	Success
		POST/_api/DATA_VIS_EXPERIENCE_TEST/GetData_VIS_EXPERIENCE_TEST	Retrieve a previous response to this questionnaire	Research Test Id and Research Study Id	5	Success
PARTICIPANTS_DASHBOARDController	This controller is part of the user modeling system and is responsible for providing the participant user with various information regarding their progress of providing all the required data related to their user model	POST/_api/PARTICIPANTS_DASHBOARD/ListPARTICIPANTS_RESEARCH_STUDIES	Lists available Research Studies the user can attend	None	5	Success
		POST/_api/PARTICIPANTS_DASHBOARD/ListRESEARCH_STUDYS_TESTS	Lists available Research Tests for a selected Research Study	Research Study Id	5	Success
QAController	This controller is part of the user modeling system and is responsible for enabling the users to solve cognitive tests and respond to likert-scale questionnaires	POST/_api/QA/GetNextQuestion	Provides the next available task in a given Research Test	Research Test Id and Research Study Id	5	Success
		POST/_api/QA/SaveAnswer	Saves the answer to a given task or likert-scale question	Question Id, Research Study Id, Start and End Time, answer string, answer string	5	Success
		POST/_api/QA/UpdateAnswer	Updates the answer to a given likert-scale question	Question Id, Research Study Id, Start and End Time, answer string, answer string	5	Success
QuestionnairesController	This controller is part of the user modeling system and is responsible for constructing likert-scale questionnaires	POST/_api/Questionnaires/GetQuestionnaire	Gets all likert-scale questionnaire questions and metadata	Research Test Id	5	Success
		POST/_api/Questionnaires/GetQuestionnaireAnswers	Get all answers provided by the user for this questionnaire	Research Test Id and Research Study Id	5	Success
RESEARCH_STUDYController	This controller is part of the user modeling system and is responsible for enabling a Researcher user to manage Research	POST/_api/RESEARCH_STUDY/InsertRESEARCH_STUDY	Create a Research Study	Research Study Id, Study Name, Description, Status, Publication and Expiration Dates, is study open?, list of research tests and allowed participants emails	5	Success
		POST/_api/RESEARCH_STUDY/UpdateRESEARCH_STUDY	Update a Research Study	Research Study Id, Study Name, Description, Status, Publication and	5	Success

	Studies			Expiration Dates, is study open?, list of research tests and allowed participants emails		
		POST/_api/RESEARCH_STUDY/DeleteRESEARCH_STUDY	Delete a Research Study	Research Study Id	5	Success
		POST/_api/RESEARCH_STUDY/GetRESEARCH_STUDY	Get a specific Research Study	Research Study Id	5	Success
		POST/_api/RESEARCH_STUDY/ListRESEARCH_STUDY	List all Research Studies	None	5	Success
		POST/_api/RESEARCH_STUDY/ListRESEARCH_STUDY_CARDS	Lists all Research Studies and some statistical indicators for each	None	5	Success
		POST/_api/RESEARCH_STUDY/ListRESEARCH_STUDY_ALLOWED_PARTICIPANTS	Provides a list of participants allowed to complete a given Research Study	Research Study Id	5	Success
RESEARCH_STUDY_STATISTICSController	This controller is part of the user modeling system and is responsible for enabling a Researcher user to view statistics about Research Studies (i.e., monitor the user modeling process)	GET/_api/RESEARCH_STUDY_STATISTICS/GetRESEARCH_TEST_RESULT_CSV/{studyId}/{testId}	Downloads a CSV file of results for a given Research Test under a specific Research Study	Research Test Id and Research Study Id	5	Success

6 Activity-based Testing

Whereas the previous section dealt with the lowest level of units in the platform i.e., API endpoints, this section will focus on more complex scenarios that involve the orchestration of several endpoints. Moreover, in the previous section the testers were the development and quality assurance teams, testing endpoints with API testing/profiling tools, whereas now the system will be tested with the actual actors by utilizing the actual platform's user interface. Several test cases in the form of activities were created for testing the system, covering all functionality available to all user roles. For that reason, a single end user from each user role was selected for carrying out the corresponding test cases. Moreover, while building the test cases, specific emphasis was given to the three major phases of the project, all of which encompass the main functionality of the IDEALVis platform. The following tables provide information about the activity-based testing scenarios. Table 3 starts by describing all-purpose functionality test cases (e.g., authentication). Next, Table 4, provides information about the various activity-based scenarios that were tested.

Table 3 – All-purpose Functionality Test Cases

User Role Responsible	Test Case Category	Test Case Action	Result
All User Roles	Account and Authentication:	Register an Account	Success
		Login using the Account	Success
		Verify account via email	Issue Detected (Activation emails arrive late)
		Use the Remember Me Functionality	Success
		Use Forgot Password Functionality	Success
		Edit profile information and save	Success
		Try logging out	Success
		Enable Two Factor Authentication	Success
		Enable an External Login i.e., Using a Microsoft Account	Success
	Data Protection/Privacy Settings	View/Download all user account data	Success
		Request the termination of data processing (verify using OTP)	Success
		Request the deletion of the account (verify using OTP)	Success
	Other Settings	Change the default UI theme	Issue Detected (Some of the colors in the themes need tweaking – contrast issue)
		Send a query to the support team using the built-in contact form	Success
	Analysis Tasks	View all available analysis tasks. One by one set them as current tasks	Success
		Provide an answer to all analysis tasks	Success
	Dashboard	Resize the card of a pinned analysis	Success
		Move the pinned analysis into a different position	Success
		Remove a pinned analysis from the dashboard	Success
	Data Protection Officer	Data Protection Requests	Browse the list of all active data protection requests
Mark some requests as approved			Success
Mark some requests as denied			Success

Admin	Organizations	Create new organizations	Success
		Manage existing organizations i.e., list, edit, delete	Success
	User Groups	Create new user groups	Success
		Assign an organization to a new user group	Success
		Manage existing user groups i.e., list, edit, delete	Success
	Manage Users	Browse the list of all registered users of the platform	Success
		Edit some users and assign them to a user role	Success
		Edit some users and assign them to a user group	Success
	Team Leader	Analysis Tasks	Create new analysis tasks (use both multiple choice and textual response questions)
Manage existing analysis tasks i.e., list, edit, delete			Success
For a given analysis task view the completion status of all users assigned to that task			Success

Table 4 – Activity-based Scenario Test Cases

User Role Responsible	Test Case Category	Test Case Action	Additional Test Case Description	Result
All User Roles	User Studies	Use the available URL to join a Research Study	The user should try to join a Research Study with an existing account and also using a new account that is created through the process of joining the Research Study	Success
		Complete all questionnaires in the Research Study	A set of Research Studies were provided to the user. The user should complete all the Research Tests (regarding questionnaires) provided in these. Moreover, upon completion the user should check that their dashboard provides the correct “successful completion” notifications	Success
		Complete all psychometric tests in the Research Study (view performance and accuracy statistics on the finish page)	A set of Research Studies were provided to the user. The user should complete all the Research Tests (regarding psychometric tests) provided in these. Moreover, upon completion the user should check that their dashboard provides the correct “successful completion” notifications. The user should also verify that at the end of each psychometric test the appropriate performance and accuracy statistics are available	Success
Researcher	Research Studies	Create new Research Studies (both open and closed studies)	The user should try and create 5 different Research Studies 2 of which need to be closed user studies. Each Research Study should contain 2 questionnaires and 2 psychometric tests	Success
		Manage existing Research Studies i.e., list, edit, delete	-	Success
		Retrieve the invitation URL for a Research Study	The user should test the copy button found next to each Research Study in the list of available Research Studies. The user should browse the copied URL and verify that when loaded in a browser the correct study information shows up	Success
		View the results of a specific Research Study	-	Success
		Export the results of a Specific Research Study	-	Success
All User Roles	Analysis Wizard	Select one analysis task and create a new analysis (use all analysis types)	The user should create a new Analysis Wizard Session by first selecting an analysis task and then using that session generate multiple analysis reports with the various analysis types	Success
		Use multiple datasets, attributes, and filters	For each analysis created the user is required to select different data datasets, attributes, and filters to thoroughly test the underlying query engine	Success
		View descriptive statistics of both datasets and attributes	While selecting a dataset or an attribute the user is required to also test the descriptive statistics functionality	Success
		Save a new analysis (test both public and private access)	Once the user is at the last step of the Analysis Wizard (i.e., viewing an analysis report), the user is further required to save the wizard configuration i.e., analysis flow under a given name	Issue Detected (Cannot save an analysis with a name that is already taken by another saved analysis)
		Pin a new analysis on the dashboard (provide a name)	The user is required to create an analysis report using the Analysis Wizard and save the analysis produced on their personal dashboard	Success
		Download the analysis resulting data and export the visualization	For a given analysis report generated using the Analysis Wizard the user is required to use the buttons for saving the visualization PNG and for exporting the underlying report data	Success

		Try leaving the Analysis Wizard page without saving your analysis	Once the user is about to leave the Analysis Wizard page without saving their wizard configuration (i.e., analysis) an appropriate exit prevention prompt should appear	Success
		Load one analysis that you created at a previous point in time	The user is required to load a previously saved wizard configuration and progress until the last step of the Analysis Wizard so the result can be loaded	Success
		Load one analysis that belongs to another user, edit the analysis, and save it publicly	The goal here is for the user to browse publicly available analyses (i.e., wizard configurations) load one and try to modify it and finally save it again	Issue Detected (For some users the resulting visualizations were impossible to understand – bad rendering)
	Analysis Tracker Dashboard	Explore Analysis Construction Tracking Statistics	After the user has created multiple analysis reports via the Analysis Wizard interface, the user is further required to explore their analysis construction statistics via this dashboard. Moreover, the user should compare their analysis construction performance with other users using the appropriate dashboard tracking results. Moreover, the user is required to identify the analysis task where they spent the most time selecting attributes when compared to other users.	Issue Detected (Some users did not utilize the Analysis Wizard before visiting this dashboard and this caused the dashboard to crash)
		Explore Analysis Report Interaction Tracking Statistics	After the user has explored multiple analysis reports in the system (i.e., through the Analysis Wizard's last step and via their personal dashboard), the user is further required to explore their analysis report interaction tracking statistics via this dashboard. Moreover, the user should identify the analysis task report where they spent the most time processing when compared to other users.	Success

All the important findings recorded during this testing phase are reported in the table below:

Table 5 - Activity-based testing Findings

No	Severity	Problem	Resolution
1	Minor	Minor unaddressed UI issues e.g., contrast of colors	All UI issues were resolved
2	Minor	Minor bugs and improvements	All bugs and suggested improvements were resolved
3	Minor	Activation emails were arriving late and were marked as junk	The SMTP issue was resolved by switching the email sender implementation to SendGrid ¹
4	Major	Analysis wizard had an issue with saving an analysis with a name that is already used by another analysis	The analysis ID was set as the primary key since analyses with the same names can coexist
5	Major	When the platform does not have enough collected interaction data the Analysis Tracker Dashboard was crashing during loading	An appropriate interface was set in place for informing the user that they need to perform further data exploration in order to generate tracking data for the Analysis Tracker Dashboard
6	Major	Visualizations for some users were impossible to understand since there was a conflict in the Adaptation Engine	Instances where two adaptation rules collided were resolved accordingly

Moreover, throughout the IDEALVis development lifecycle the codebase for the platform was maintained online on GitHub² which is a leading tool/platform that provides versioning control and project management capabilities. All bugs that were found during development (and during testing as indicated in Table 5) were tracked via GitHub issues and were addressed accordingly through multiple git commits. Figure 1 presents the IDEALVis platform's private GitHub repository page.

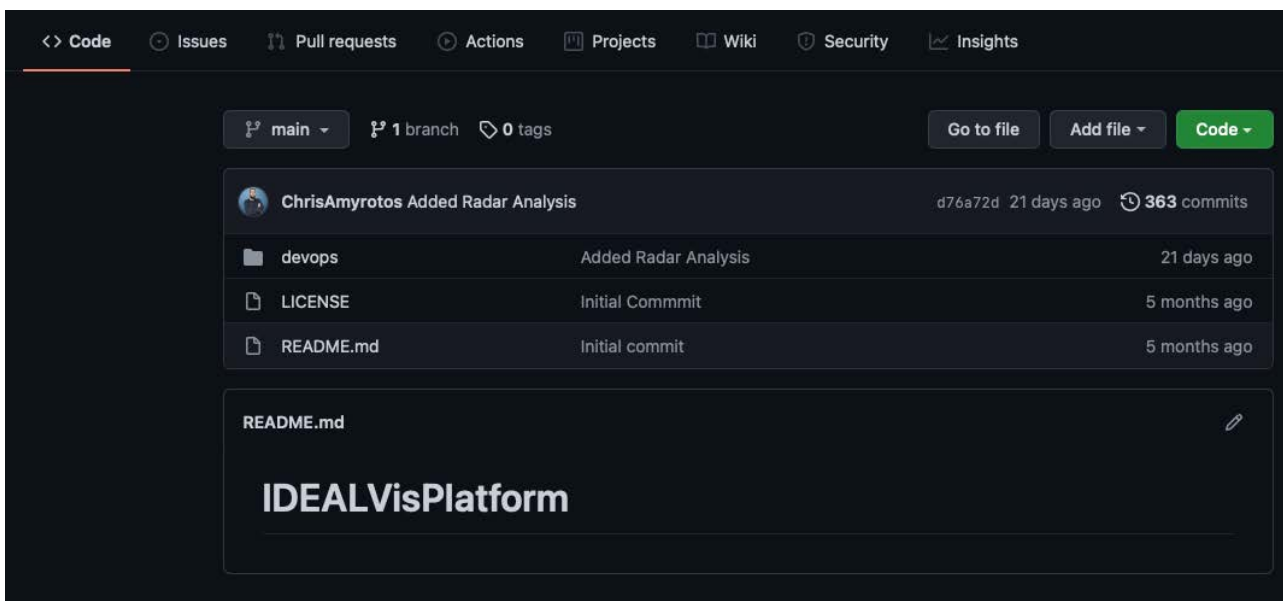


Figure 1 - IDEALVis Platform's Private GitHub Repository Page

¹ <https://sendgrid.com>

² <https://github.com>

7 Conclusions

This deliverable outlined the approach taken for testing and validating the IDEALVis platform and all its underlying components and functionality. Focus was given on testing and validating the platform's (i) functionality and the acceptance by the users, according to the requirements of the platform; (ii) performance; and (iii) security. Initially, all the low-level endpoint components of the platform were tested and validated through multiple phases (i.e., functionality, security, and performance tests), followed by thorough testing and validation of the platform's overall functionality (through multiple well-tested, role targeted test cases performed through the user interface of the application). During the testing and validation process several minor bugs and issues were collected and were resolved accordingly. The testing and validation procedure that was followed ensured that the platform can serve what is promised while satisfying all main requirements that correspond to the needs of the targeted users, in an effective, efficient, and secure manner.