bEhaVioral Insights anD Effective eNergy policy acTions

Project acronym: EVIDENT

Project title:

bEhaVioral Insights anD Effective eNergy policy acTions

Deliverable 6.2

Crowdsourcing Tools of EVIDENT platform

Programme: H2020-LC-SC3-EE-2020-1

Start date of project: December 01, 2020

Duration: 36 months

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957117

Document Control Page

Deliverable Name	Crowdsourcing Tools of EVIDENT platform
Deliverable Number	Deliverable
Work Package	WP6
Associated Task	Task 6.2 System Development
Covered Period	M03-M20
Due Date	31st of July 2022
Completion Date	22nd of November 2022
Submission Date	22nd of November 2022
Deliverable Lead Partner	Sidroco Holdings Ltd.
Deliverable Author(s)	Anastasios Lytos, Antonios Sarigiannidis, Elisavet Grigoriou, Theocharis Saoulidis
Version	1.1

Dissemination Level		
PU	Public	**X**
CO	Confidential to a group specified by the consortium (including the Commission Services)	

Document History

Version	Date	Change History	Author(s)	Organisation
0.1	July 03, 2022	Initial version	Anastasios Lytos, Antonios Sarigiannidis, Elisavet Grigoriou, Theocharis Saoulidis	SID
0.2	July 12, 2022	Changes on structure	Paris-Aleksandros Karypidis	DUTH
0.3	July 03, 2022	Improve section 1, contribute to section 2 and section 3	Anastasios Lytos, Antonios Sarigiannidis, Elisavet Grigoriou, Theocharis Saoulidis	SID
0.4	July 18, 2022	Contribution in section 4	Paris-Aleksandros Karypidis	DUTH
0.5	July 19, 2022	Contribution in subsection 3.2 and in section 5	Anastasios Lytos, Antonios Sarigiannidis, Elisavet Grigoriou, Theocharis Saoulidis	SID

0.6	July 20, 2022	Proofreading	Panagiotis Sarigiannidis, Anna Triantafyllou, Georgios Fragulis, Stamatia Bibi	UOWM
0.7	July 22, 2022	Second proofreading	Anastasios Lytos, Antonios Sarigiannidis, Elisavet Grigoriou, Theocharis Saoulidis	SID
0.8	July 25, 2022	Final revisions	Panagiotis Sarigiannidis, Anna Triantafyllou, Georgios Fragulis, Stamatia Bibi	UOWM
0.9	July 29, 2022	Integrate changes from TCD	Anastasios Lytos, Antonios Sarigiannidis, Elisavet Grigoriou, Theocharis Saoulidis	SID
1.0	July 29, 2022	Integrate changes from CW	Anastasios Lytos, Antonios Sarigiannidis, Elisavet Grigoriou, Theocharis Saoulidis	SID
1.1	November 22, 2022	Add the footer with the EU funding in the first page. Change the URL in the provided example of the EVIDENT platform.	Anastasios Lytos, Antonios Sarigiannidis, Elisavet Grigoriou, Theocharis Saoulidis	SID

Internal Review History

Name	Institution	Date
Emma Delemere	TCD	July 29, 2022
Peter Rosengren	CW	July 29, 2022

Quality Manager Revision

Name	Institution	Date
Panagiotis Sarigiannidis	UOWM	July 29, 2022
Dimosthenis Ioannidis	CERTH	July 29, 2022

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Explanation
CD	Continuous Development
CI	Continuous Integration
DRY	Don’t Repeat Yourself
DX.X	Deliverable X.X
ERD	Entity Relationship Diagram
EU	European Union
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IT	Information Technology
JSON	JavaScript Object Notation
MVT	Model-View-View
MVC	Model View Controller
ORDBMS	Object-Relational Database Management System
RBAC	Role-Based Access Control
RDBMS	Relational Database Management System
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
XML	eXtensible Markup Language

Executive Summary

The record-high energy prices at the end of 2021 and the beginning of 2022 decreased the purchasing power of consumers but without increasing their ecological awareness. The goal of a European Union (EU) climate-neutral economy requires the adoption of new habits and routines from the EU citizens, which are both more eco-friendly and financial-wise. EVIDENT aims to study individuals' decision-making process on energy consumption decisions, understand their habits and based on the research's insight propose novel energy policy frameworks. The EVIDENT platform provides a series of crowdsourcing tools, such as polls, questionnaires, and serious games, trying to understand the behaviour of EU citizens on energy consumption issues and how their behavioural patterns correlate with their financial literacy level. The EVIDENT platform will also act as a point of reference for the research findings freely offering the generated knowledge.

The EVIDENT platform that is developed under the WP6 *"Prototyping and integration"* includes crowdsourcing tools such as surveys and serious games that collect information on the individuals' decision process. In this deliverable, *D6.2 "Crowdsourcing Tool of EVIDENT platform"*, technical details on the development of the EVIDENT platform are provided, describing the implementation of the platform's different components such as sessions, surveys, polls, and questionnaires. The deliverable is complementary to the software released within the context of the EVIDENT projects.

1. Introduction

1.1 Purpose of the Deliverable

The goal of this deliverable is to describe and illustrate technical details on the development of the crowdsourcing tools that are developed and integrated into the EVIDENT platform. The document includes technical information on the development process that is followed using the continuous integration/continuous deployment (CI/CD) paradigm providing insight into the development of the platform while presenting how different functional and non-functional requirements are met.

1.2 Relation with other Deliverables and Tasks

Task 6.2 receives as input the platform requirements resulting from the previous work packages, WP1 – *"Requirements and nature of behavioural biases"*, WP2 – *"Policy interventions and pilots design"*, WP3 – *"Intervention, preparation, and execution"*, and WP4 – *"Econometric analysis and policy evaluation"*. Deliverable 6.1 *"System Architecture and Design Specifications"* collects the requirements from the previous WPs and serves as the main input document. The system developed in D6.2 offers the main software basis for D6.3 *"Gamification Tools of EVIDENT"* and D6.4 *"Datahub services of EVIDENT platform"*. Finally, the platform will be used for the integration and testing plan that will be realised under Task 6.3 *"System Integration, Verification and Validation"* producing the Deliverable 6.5 *"Verification and Validation Report"* for the final version of the EVIDENT platform.

1.3 Structure of the Document

This rest of the deliverable is structured as follows:

- *Section 2 – Overview of the EVIDENT platform* provides information on the conceptualisation and implementation of the EVIDENT platform including information on the architecture and the Entity Relationship Diagram (ERD).
- *Section 3 – Technical details of the EVIDENT platform* presents an in-depth analysis of the components used in the EVIDENT platform, including information on the framework used, the models that are designed, and how the requirements are met.
- *Section 4 – User navigation* provides details on the browsing of the platform from non-expert users and their interactions with the crowdsourcing applications
- *Section 5 – Conclusion* concludes the document.

2. Overview of the EVIDENT platform

The EVIDENT platform offers a unified endpoint (system-as-a-whole) incorporating crowdsourcing tools, serious game applications, and datahub services tools. The crowdsourcing service is a core component of the platform enabling the active participation of external users in the lab experiments that are organised within the platform.

Subsection 2.1 summarises the EVIDENT platform architecture presenting how the different architectural components are connected. Subsection 2.2 provides an in-depth overview of the reasoning for the development of the platform through the presentation of the entity-relationship diagram scheme.

2.1 EVIDENT platform architecture

This subsection provides a short description of the platform architecture, while the extensive presentation of the architecture takes place in D6.1 *"Architecture, design, and integration documentation"*. The presentation of the EVIDENT platform architecture allows for a better understanding of the provided services and tools and justifies the use of specific methods and technologies.

The EVIDENT platform architecture takes into consideration the users' requirements and specifications collected during Task 6.1 *"System Architecture and Design Specifications"* through an open communication channel that was created to foster the conversation between the software development team and the end-users. Additionally, the developing team adopted the CI/CD paradigm allowing the collection of requirements in parallel with the actual software development, thus covering any need that might occur for the consortium.

A component-based architecture was selected for the EVIDENT platform, enabling the potential expansion of functionalities in order to host additional services. The architecture consists of five (5) components each one offering a specific service to the end-users, as follows:

- **Crowdsource component** collects information from the visitors to the platform. It includes the survey subcomponent (described in-depth in section 4.2 Session Component) and the serious game (described in-depth in the upcoming deliverable D6.3 *"Gamification Tools of EVIDENT platform"*).
- **Data collection component** compiles the information given from the users, proceeds in their anonymisation, and provides the interfaces for the external use of the collected data (described in-depth in the upcoming deliverable, D6.4 *"Datahub services of the EVIDENT platform"*).
- **Data extraction component** processes the collected data and produces useful insights that are used for both commercial and research purposes (described in-depth in the upcoming deliverable D6.3 *"Gamification Tools of EVIDENT platform"*).
- **User component** covers all the functionalities that are provided to the platform's users to interact with its services (described in-depth in section 4.1 User and User Profile Component).
- **Datahub services** provide a repository for all the datasets that are used in the context of the EVIDENT project and for future projects in the field of financial literacy (described in-depth in the upcoming deliverable D6.4 *"Datahub services of the EVIDENT platform"*).

Figure 1 presents the architecture that is followed for the development of the EVIDENT platform. It illustrates the flow of information and the connection between the different software components. A short explanation of the architecture's components follows, while their extensive description can be found in D6.1 *"System Architecture and Design Specifications"*.

The EVIDENT platform follows the modular architecture paradigm, allowing the addition of new features if it is requested, and it has been designed to have two types of users: organisations and participants. The former set up a lab experiment using the services of the platform (questionnaires, polls, etc.) and the latter participate in these lab experiments. The users that desire to organise a lab experiment and use the services of the crowdsource component are authorised by the EVIDENT consortium to act as organisations, after the related request with a short justification via email.

The crowdsource component is one of the cornerstone concepts in the EVIDENT architecture, extracting visitors' current behavioural biases on energy consumption habits via surveys, polls, questionnaires, and serious games. The crowdsource component consists of two subcomponents, namely the survey and serious game subcomponents. Through the survey subcomponent, users registered as an organisation can create and edit surveys that can be used within the lab experiments. If published, the new survey can be selected as a distinct step in the context of a lab experiment. The SurveyJS library is used to develop the survey feature in the platform providing an integrated survey creator for in-house data storage. The organisation users will be able to leverage the full functionalities of the SurveyJS library creating dynamic surveys and using them in their lab experiments. Similarly, the organisations can use the serious games subcomponent to upload and edit serious games as a web app in the context of a lab experiment both as a website and a mobile application.

The subcomponent "S*essions"*, which enables users to participate ins and engage with the lab experiments, ensures the platform's primary functionalities run without a hitch. The "Data collection component" is a system for organizing and storing the data obtained from users' participation in the lab experiments. The "Data extraction component" can be used by organisations and visitors to extract the collected data from a session or their participation, respectively. The "User component" includes the actions a user can perform regarding her/his account. Based on use cases, a user might want to extract all her/his data the EVIDENT platform has collected for her/him, update her/his personal data processing consent, etc. The component implements the user dashboard and extends the user profile subcomponent and the data extraction component. The "User profile component" provides the mechanism for the user to edit her/his profile such as personal information, platform password, etc. and to leverage the functionalities of the data extraction component to extract the data collected during her/his participation in a session. The datahub services tool will be used as a hub that will host all the data used for the intervention trials (in anonymised form, without exposing any private user information), the econometric analytical tools that will be developed during the project lifetime, and the reports and insights derived from results.

The EVIDENT platform offers a comprehensive method for planning and executing online lab experiments, as it is built on established technology solutions and resources. While the interconnections between the various subcomponents ensure the consistency of the acquired data, the crowdsourcing component coordinates the primary interactions in the platform acting as a consolidated tool for the organisations. To make it simple for each platform actor to export needed data, all developed apps (surveys, serious games, and sessions) and their acquired data should be organised and securely stored in the central EVIDENT database. For instance, if the session includes a survey and a serious game in addition to two different applications, each participant's behaviour and responses should be identified by their respective unique ID within the lab experiment. The difficulty that comes with the crowdsourcing component is

creating a development and integration strategy that will enable the platform to materialise quickly and smoothly.

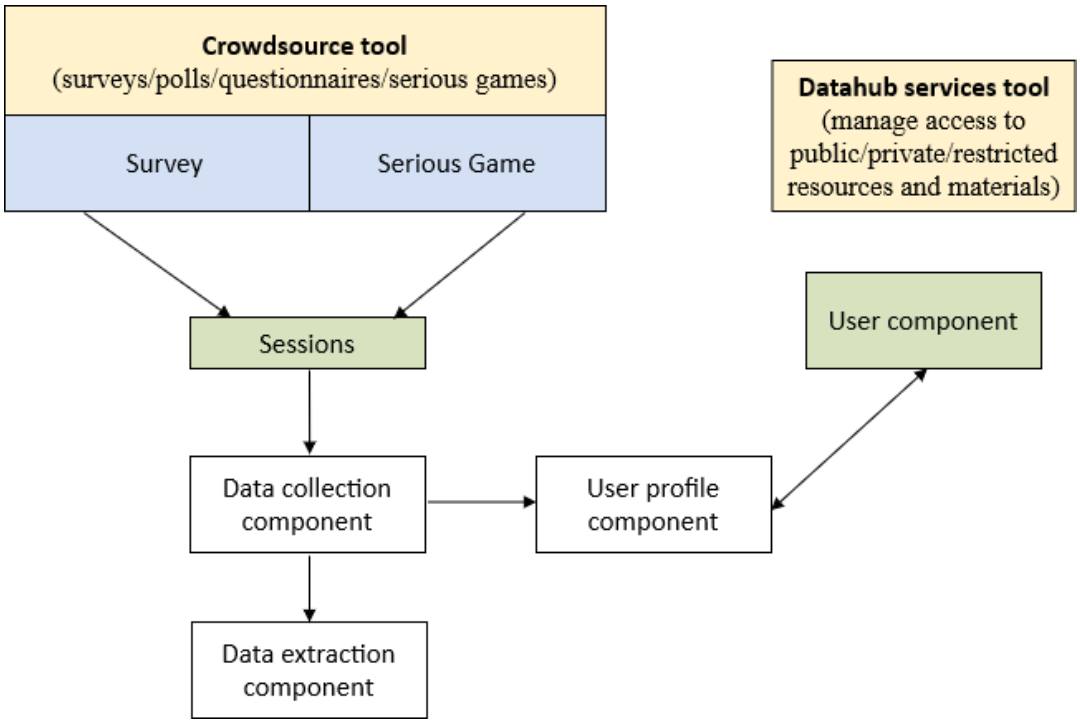


Figure 1: EVIDENT platform architecture

2.2 ERD Scheme

An ERD or an entity-relationship model is a graphical representation that shows relationships between individuals, things, locations, concepts, or events within an information technology (IT) system. An ERD employs data modelling techniques to represent the business processes and prepares the basis for a relational database. It offers a visual starting point for database architecture and depicts the flow of information within the information system.

Figure 2 presents the ERD scheme that was followed for the conceptualisation of the EVIDENT platform and used as a basis for the construction of the models. There are seven (7) main entities: user profile, user, session, session reply, survey, serious game, and serious game configuration. Each entity represents an essential feature for the platform's operation.

- User profile – it includes all the fields that are related to the profile of a user such as profile image, deletion date of profile, location, primary language, banned flag, a short bio of the user, consent flag, and user type. It relates to the entity *User* which has as a foreign key the *User*'s ID.
- User – it is one of the two main entities in the system, and it includes information related to the user. It is connected with the entities *User profile*, *Session Reply,* and *Session.*
- Session – it is the second main entity of the system and is the backbone of the platform. It represents the main functionality of the crowdsource component implementing the lab

experiments for the different organisations that are enrolled in the platform. It is connected with the entities *User, Session Reply, Survey, Serious game,* and *Serious game configuration.*

- Session reply – it collects the responses for the lab experiments that take place, and it associates them with specific sessions and users.
- Survey – it represents one of the two crowdsourcing applications alongside the serious game. It has four fields: *title*, *description*, and *q_n_a*, and it relates to the entity *Session*.
- Serious game – it is the second crowdsourcing application, and it is connected with the entities *Session* and *Serious game configuration*.
- Serious game configuration – it represents the configurations that every serious game receives on every session. It is connected with the entities *Serious game* and *Session*.

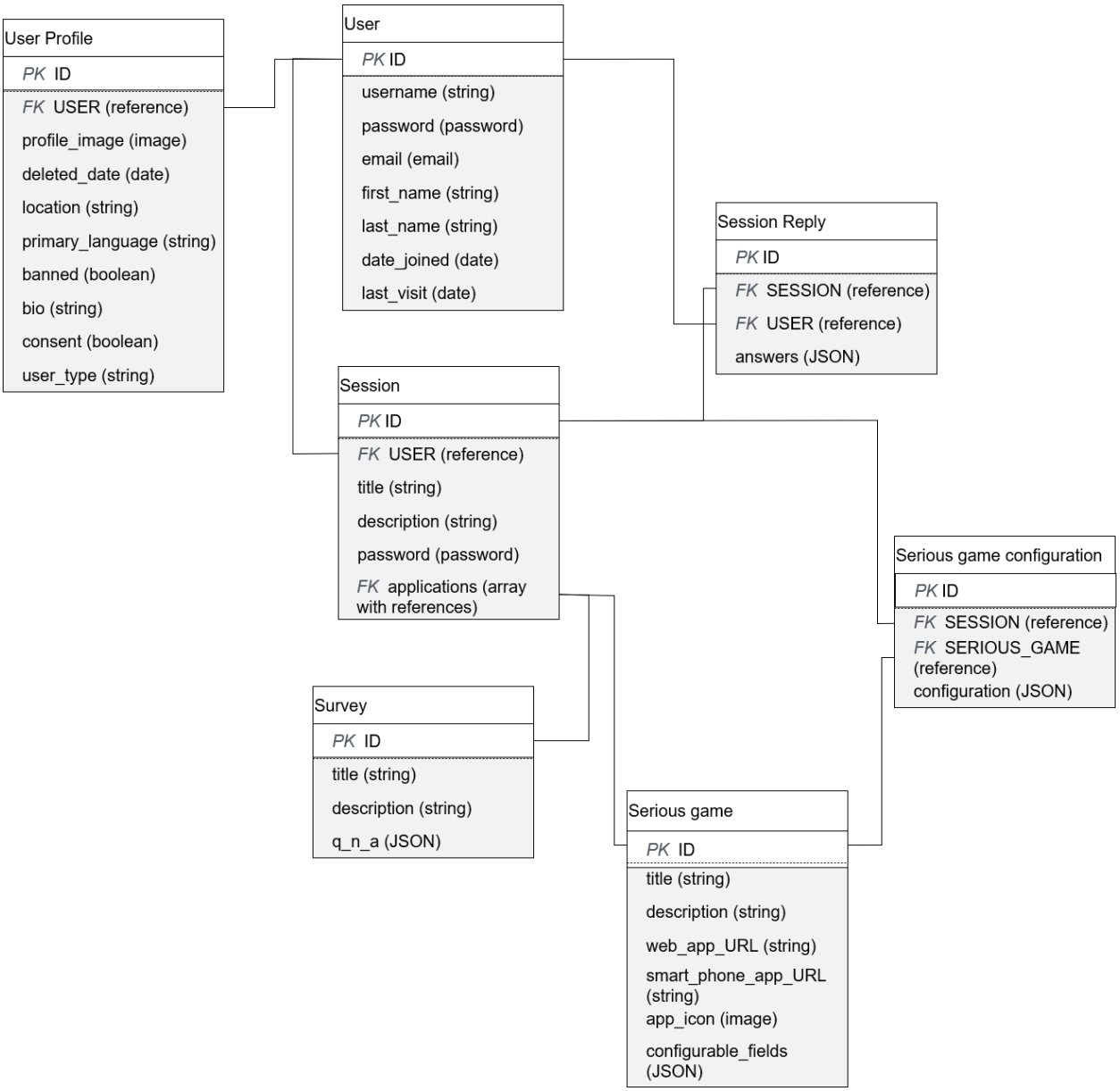


Figure 2: EVIDENT platform ERD scheme

The ERD scheme provides a good overview of the main entities in the EVIDENT's platform including information on the fields the models will have, the connection between the different entities, and the significance of every entity. Overall, it visualises important information on the nature of the system, provides a good overview of the system at a glance, and shortly describes the fields of every entity.

3. Technical details of the EVIDENT platform

The design and development of the EVIDENT platform followed established best practices in creating a web application that can be extended, modified, and used by external users. The decisions on the approaches and technologies that were taken to support the system are explained in this section.

Subsection 3.1 provides a short description of the decision for the use of the Django framework, while subsection 3.2 presents the implementation of the models that materialise the platform's entities. Finally, subsections 3.3 and 3.4 respectively describe the process for the UI/UX and the platform's template and how the requirements of the platform are met in its implementation.

3.1 Django Framework

The EVIDENT platform was developed using the Django framework, a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It is one of the most popular web frameworks with increasing popularity over time and an active developing community [1].

Django follows the model-view-template (MVT) architectural pattern including three components:

- The Model manages the data and is represented by a database. Every model is linked to a database table.
- The View receives HTTP requests and sends back HTTP responses. The view connects the model with the template to complete a response.
- The Template is the front-end layer of the platform representing the dynamic HTML component of a Django application.

Figure 3 presents the MVT structure that is followed by every Django application [2]. The left part of the figure illustrates the user that interacts through requests with the platform and replies with responses. The right part of the figure represents the interaction between the different building blocks of a Django application.

The entity URL includes the platform's URL patterns, for example, for the domain name *https://platform.evident-h2020.eu/* a URL pattern is *https://platform.evident-h2020.eu/sessions/* which redirect (through a web request) to a function of the platform. After receiving the web request from the user through a web browser, Django seeks the proper response activating the MVT pattern.

A view is a python function (or a python class) implementing the logic of the response. It is the corresponding response to a URL request fetching data from models and rendering a given template. For example, when a request for *https://platform.evident-h2020.eu/sessions/participate_session/46d18b8b-87f4-4508-8e52-9738c74cfdbb* is given, Django seeks to connect the specific URL with a specific view responding with action.

The URLs that are used in the example above redirect to the temporary domain name that hosts the EVIDENT platform. The final domain name of the EVIDENT platform is going to be finalised in *WP7 - Dissemination and exploitation* in Task *7.1 Communication, dissemination and premarketing activities*.

A model is a python class that is linked to a database table. Django supports the creation of models, instead of database tables, allowing the web developers to focus on the functionality of the platform and

not on the construction of the database. Django builds automatically the database without interference from the developers. A model offers the necessary data to a view to serve a user request.

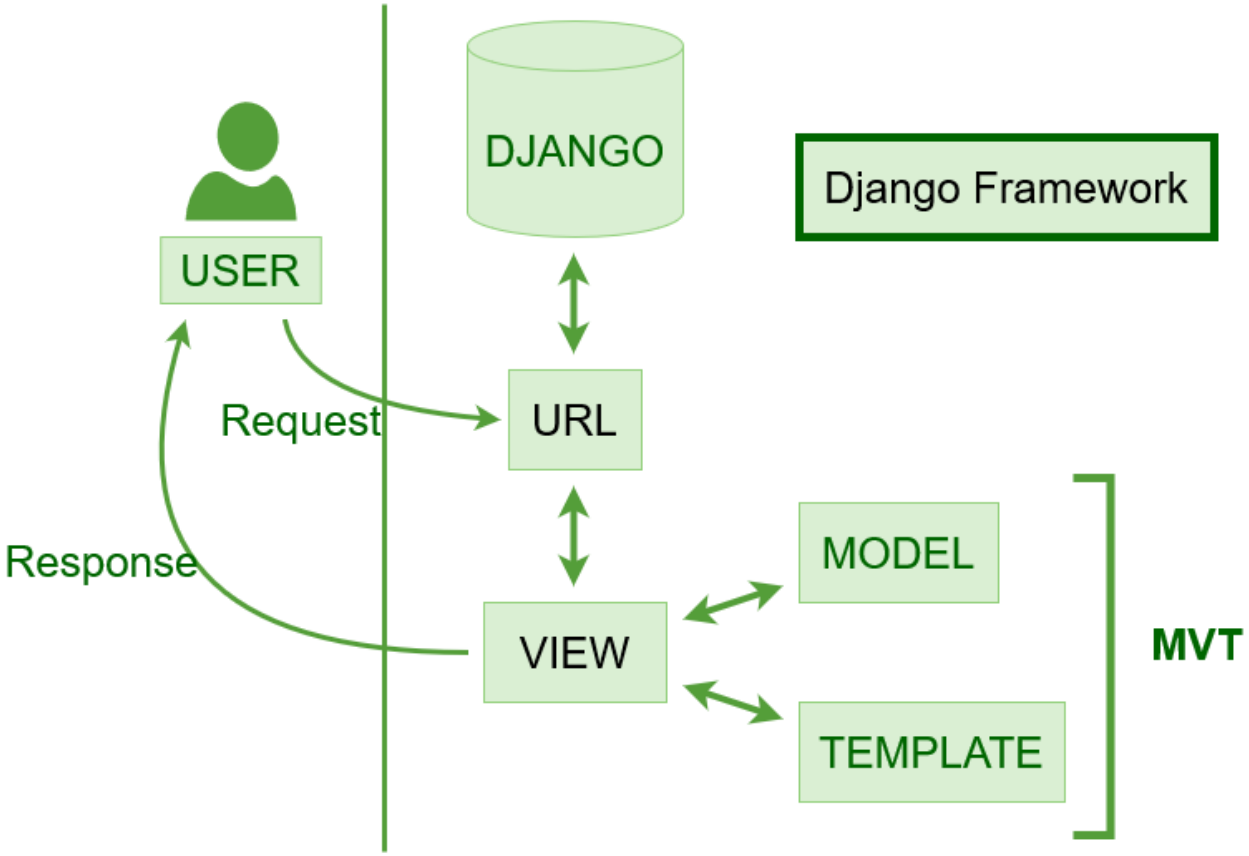


Figure 3: MVT structure of a Django application

A template is the front-end component of a Django application and it contains the static HTML and the dynamic functions implemented in JavaScript. A view renders a specific template, and dynamically shows the data from the models within the rendered template.

Django offers the framework to develop complex, database-driven websites without requiring time investment in database design and development while prioritising component reusability, pluggability, rapid development, low coupling and less code following the DRY (Don't Repeat Yourself) and explicitness principles [3]. Even though Django uses the term MVT, the framework can be seen as a Model-View-Controller (MVC) architecture which is the term widely used in software engineering frameworks [4].

Additionally, the Django framework includes some core functionalities that assist in the rapid development of web applications such as:

- a lightweight and standalone web server for development and testing, allowing shorter cycles of design, development, release, and feedback
- a form serialisation and validation system that automatically translates HTML forms and values to database tables and fields
- a template system that utilises the concept of inheritance enabling object-oriented design and development

- an automated caching framework that saves time by temporarily storing the outcome of different functions
- an internationalisation system, including translations of Django's components into a variety of languages allowing multi-language content
- a serialisation system that can produce and read XML and/or JSON representations of Django model instances assisting the communication on a component-based architecture

3.2 Database and Models

Django developers design the entities of the platform in the format of models. Django automatically converts these models into database tables, as explained in subsection 3.1. In this subsection, an in-depth analysis of the models used in the EVIDENT platform takes place and the choice of the PostgreSQL database is supported.

A model represents the related information of the available data. It contains the essential fields and behaviours of the stored data and maps to a single database table. Django officially supports the following databases [5]:

- PostgreSQL
- MariaDB
- MySQL
- Oracle
- SQLite

The default option for Django applications is PostgreSQL, a powerful, open-source object-relational database system that is characterised by its reliability, feature robustness, and performance. A PostgreSQL's aspect is that is an Object-Relational Database Management System (ORDBMS) providing the same relational capabilities as a Relational Database Management System (RDBMS), but additionally has some object-oriented features [6].

Table 1 presents the model *Session* having four columns: Field, Description, Type, and Example. The session model is correlated with the model *User* via a foreign key and includes all the necessary fields to provide the necessary information for the platform's functionalities. The model *Session* includes different types of fields including characters, DateTime, boolean, and JSON objects. More specifically, the JSON object includes the list of applications (e.g. questionnaires, serious games) within the specific session.

Table 1: Model *Session*

Field	Description	Type	Example
User	A foreign key that maps the session X with the user Y.	ForeignKey	Jon_Doe
title	The title of the session.	CharField	DUTH session
description	The description of the session.	CharField	Demo description

password	The password to enter the session (optional).	CharField	Session_password
application_list	List of applications under a specific session.	JSONField	[{"type": "questionnaire", "primary_key": "2"}, {"type": "questionnaire", "primary_key": "3"}]
created_date	The date of the session is created.	DateTimeField	2022-02-22 09:59:09
published	The date of the session is published.	BooleanField	Yes
total_answers	The number of answers received in the session	IntegerField	33

Table 2 presents the model *SessionReplies* having the same four columns (Field, Description, Type, Example). The model *SessionReplies* is connected with the model *Session* and the model *User* using them as foreign keys. It also includes the fields that have related information such as corresponding answers, current application, and current application type.

Table 2: Model *SessionReplies*

Field	Description	Type	Example
Session	A foreign key that maps the session reply Z with the session X.	ForeignKey	Demo session
User	A foreign key that maps the session reply Z with the user Y.	ForeignKey	Jon_Doe
answers	The answers provided from a specific user to a specific session	JSONField	[{"question1": "item1", "question2": "item3", "question3": "item2"}, {"question1": {"It is important to respect nature": "6", "It is important to be in unity with nature.": "6", "It is important to protect the environment.": "6", "It is important to prevent environmental pollution.": "6"}}]
current_application	An identifier for the application the user currently runs.	IntegerField	1

current_application_type	Type of application the user currently runs.	CharField	questionnaire

Table 3 presents the model *Survey* It is connected with the model *User* having it as a foreign key while it also includes the related information such as title, description, created date, published date, and a JSON object including the questions and answers within the survey.

Table 3: Model *User Profile*

Field	Description	Type	Example
User	The username of the user that creates the survey.	ForeignKey	Jon_Doe
title	The title of the survey.	CharField	Assessing the financial literacy level
description	The description of the survey.	CharField	Details on the motivation of the survey
created_date	The date the survey is created.	DateTimeField	2022-02-22 10:49:22
Published	The date the survey is published.	BooleanField	Yes
q_n_a	The questions and answers within the survey	JSONField	{"pages": [{"name": "page1", "elements": [{"name": "question1", "type": "text", "title": "What is your name?"}]}], "title": "First"}

Table 4 presents the model *User Profile*. It is connected with the User model having it as a foreign key. It also includes all the necessary fields for the user to have a complete profile on the platform. The fields include information such as the type of the user (organisation, participant), the name of the organisation (if applicable), the position in the organisation, (if applicable), and a justification for her/his role in the organisation (if applicable), a profile image, date of her/his last visit, date of account termination (if applicable), the preferred language, a flag that indicates if the user is banned, a short bio, the consent date of the user, and the timezone the user is located.

Table 4: Model *User Profile*

Field	Description	Type	Example
user	The username of the user that creates (or participates in) the survey.	ForeignKey	Jon_Doe
User_type	The type of the user (organisation, participant).	CharField	Organisation
user_organisation	The name of the organisation (optional).	CharField	DUTH

user_position_in_organisation	The user's position in the organisation.	CharField	Director
user_organisation_justification	A short description of the organisation.	CharField	Responsible to execute experiments
image	The profile image.	ImageField	User_avatar.png
last_visit	The date of the last visit.	DateField	2022-02-22
deleted_date	The date of the account deletion (optional).	DateTimeField	2022-02-22
language	The language the user prefers.	CharField	English
banned	The flag indicates if the user is banned.	BooleanField	No
bio	A short text providing information on the user.	TextField	Short introduction of Jon Doe
consent	The date the user provided consent on the platform.	DateTimeField	2022-02-22 09:04:32
timezone	The timezone the user is located.	TimeZoneField	UTC

Table 5 presents the model *User Demographics*. It is connected with the model *User* having it as a foreign key. It also includes details on the demographics of the user that will be used in the analysis of the lab experiments. Demographics include details such as age, sex, yearly income, race, employment status, location, home ownership, education level, and date of the latest profile update.

Table 5: Model *User Demographics*

Field	Description	Type	Example
user	The username of the platform's participant.	ForeignKey	Jon_Doe
age	The age of the participant.	CharField	22
sex	The sex of the participant.	CharField	male
yearly_income	The yearly income of the participant.	CharField	22000
race	The race of the participant.	CharField	Asian
employment	The employment status of the participant.	CharField	Full-time
location	The location of the participant.	CharField	Ireland

homeowner	The flag indicates if the participant owns a house.	BooleanField	Yes
education	The level of the participant's education.	CharField	Bachelor's or equivalent
updated	The flag indicates if the participant has updated at least once her information on the platform.	BooleanField	Yes

3.3 Meet the requirements

In D6.1 "System Architecture and Design Specifications", the requirements of the platform were collected and used as a guidebook for the development of the crowdsourcing applications.

Requirement ID	Requirement title	Coverage
Func_01	User Authentication	Implemented 100%. Django comes with a user authentication system. It handles user accounts, groups, permissions and cookie-based user sessions [7].
Func_02	User Authorisation	Implemented 100%. Django authentication system covers the needs of authorisation determining what an authenticated user is allowed to do [7].
Func_03	Organisation Validation	Implemented 100%. Every request for an organisation account is automatically sent to the platform's administrators.
Func_04	Organisation grant access	Implemented 100%. When the request is reviewed by the platform's administrators an email is sent confirming the rights of the user as an organisation.
Func_05	User login	Implemented 100%. The users can login after the platform's authentication/authorisation.
Func_06	User roles	Implemented 100%. The EVIDENT platform provides different roles for the users (super-admin, organiser, participant).
Func_07	User logs out	Implemented 100%. The EVIDENT platform offers the log-out functionality to its logged-in users independently from their role.
Func_08	Complete profile information	Implemented 100%. The users can provide their profile information on their registration.
Func_09	Update profile information	Implemented 100%. The users can provide their profile information on demand.
Func_10	Update user demographics	Implemented 100%. The users can update additional information regarding her/his demographic details.
Func_11	Delete account	Implemented 100%. The user can delete her/his account
Func_12	Forgot password	Implemented 100%. The platform provides the option "forgot password" to the users.

Func_13	Update password	Implemented 100%. The users can update their passwords.
Func_14	Export user profile data	Implemented 100%. The user can export her/his activity from the platform.
Func_15	Create survey	Implemented 100%. The users registered as an organisation can create surveys.
Func_16	Update survey	Implemented 100%. The user/organisation who already created a survey can update it on request.
Func_17	Delete survey	Implemented 100%. The user/organisation can delete the survey he/she created.
Func_18	Overview of existing serious games	To be implemented in M22 in D6.3 "Gamification Tools of EVIDENT platform" according to the timeplan.
Func_19	Upload a new serious game	To be implemented in M22 in D6.3 "Gamification Tools of EVIDENT platform" according to the timeplan.
Func_20	Update a serious game	To be implemented in M22 in D6.3 "Gamification Tools of EVIDENT platform" according to the timeplan.
Func_21	Overview of existing lab experiments	To be implemented in M22 in D6.3 "Gamification Tools of EVIDENT platform" according to the timeplan.
Func_22	Create a lab experiment	Implemented 100%. The user/organisation can create a new lab experiment.
Func_23	Setup lab experiment wizard	Implemented 100%. The lab experiment wizard shows up when the user creates a new lab experiment.
Func_24	Edit an existing lab experiment	Implemented 100%. The user/organisation can edit existing lab experiments.
Func_25	Delete an existing lab experiment	Implemented 100%. The user/organisation can delete existing lab experiments.
Func_26	Serious game manager	To be implemented in M22 in D6.3 "Gamification Tools of EVIDENT platform" according to the timeplan.
Func_27	Configure a serious game	To be implemented in M22 in D6.3 "Gamification Tools of EVIDENT platform" according to the timeplan.
Func_28	Attach a serious game to lab experiment	To be implemented in M22 in D6.3 "Gamification Tools of EVIDENT platform" according to the timeplan.
Func_29	Participate in a lab experiment	Implemented 100%. The user/participant can participate in published lab experiments.

Func_30	Inspect participation in a lab experiment	Implemented 100%. The user/participant can review her/his answers on a lab experiment previously participated.
Func_31	Export answers from a lab experiment	Implemented 100%. The user/organisation that has set up and published a lab experiment, has the option to export the answers to the lab experiment.
Func_32	Share a lab experiment	Implemented 100%. The user/participant that has completed her/his participation in a lab experiment has the option to share it with her friends via social media.
Func_33	User scoreboard	To be implemented by M22 in D6.3 "Gamification Tools of EVIDENT platform" according to the timeplan.
Non_Func_01	Password encryption	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.
Non_Func_02	Data anonymisation	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.
Non_Func_03	Availability	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.
Non_Func_04	Scalability	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.
Non_Func_05	Reliability	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.
Non_Func_06	Recoverability	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.
Non_Func_07	Maintainability	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.
Non_Func_08	Serviceability	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.
Non_Func_09	Usability	To be implemented by M34 in D6.5 "Verification and Validation Report for the final version of EVIDENT platform" according to the timeplan.

4. User Navigation

As described in D6.1 "*Architecture, design, and integration documentation*", four different user roles are identified regarding the EVIDENT platform: super-admin, organiser (also referred to as "organisation"), participants, and non-registered users (also referred to as "visitors"). The EVIDENT platform implements a Role-Based Access Control (RBAC); thus, the user role determines the actions each user can implement on the platform. Besides the actions a registered user can perform, non-register users can simply view the platform information and a list of the available sessions. However, to participate in the available sessions they must first register.

The sitemap of the EVIDENT platform is presented in Figure 4.

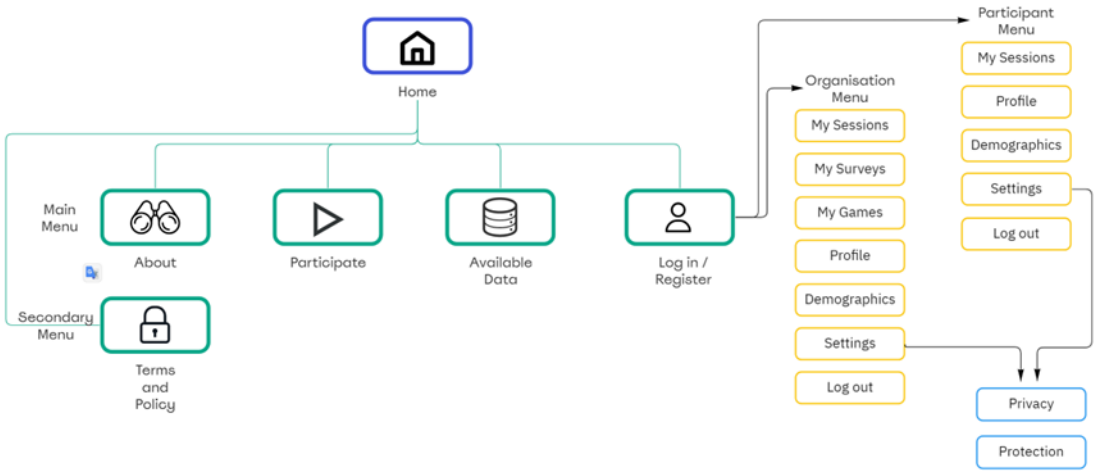


Figure 4: EVIDENT platform sitemap

A registered user can be either a participant or an organiser. Each user type is provided with a different user menu, thus different actions that they can perform. Finally, the super-admin role is not available for registration, thus only existing super-admins can register this kind of user. Table 6 briefly presents the actions each user role can implement in the EVIDENT platform.

Table 6: User permission per user role

User permissions / User role	Non-registered users (Visitors)	Participants	Organisers	Super-admins
Visit the website and see static information	✓	✓	✓	✓
Registration / Log in / Log out / Reset password	✓	✓	✓	✓
Manage user profile		✓	✓	✓
Delete user account		✓	✓	✓
Export user profile data		✓	✓	✓

Update user demographics		✓	✓	✓
Create a new survey / Update a survey / Delete a survey			✓	✓
Upload a new serious game / Update a serious game / Delete a serious game			✓	✓
Create a new lab experiment / Edit an existing lab experiment / Delete a lab experiment			✓	✓
Configure a serious game for a Lab Experiment			✓	✓
Participate in a lab experiment		✓	✓	✓
Inspect participation in a lab experiment		✓		✓
Export answers from a lab experiment		✓		✓
Collect answers from a lab experiment			✓	✓
Share a lab experiment	✓	✓	✓	✓
Full access to every information				✓

Each user role that visits the EVIDENT platform is landed on the homepage where information about the EVIDENT platform is presented (as depicted in Figure 5). The user can also visit the “About” page where more information about the EVIDENT project is presented.

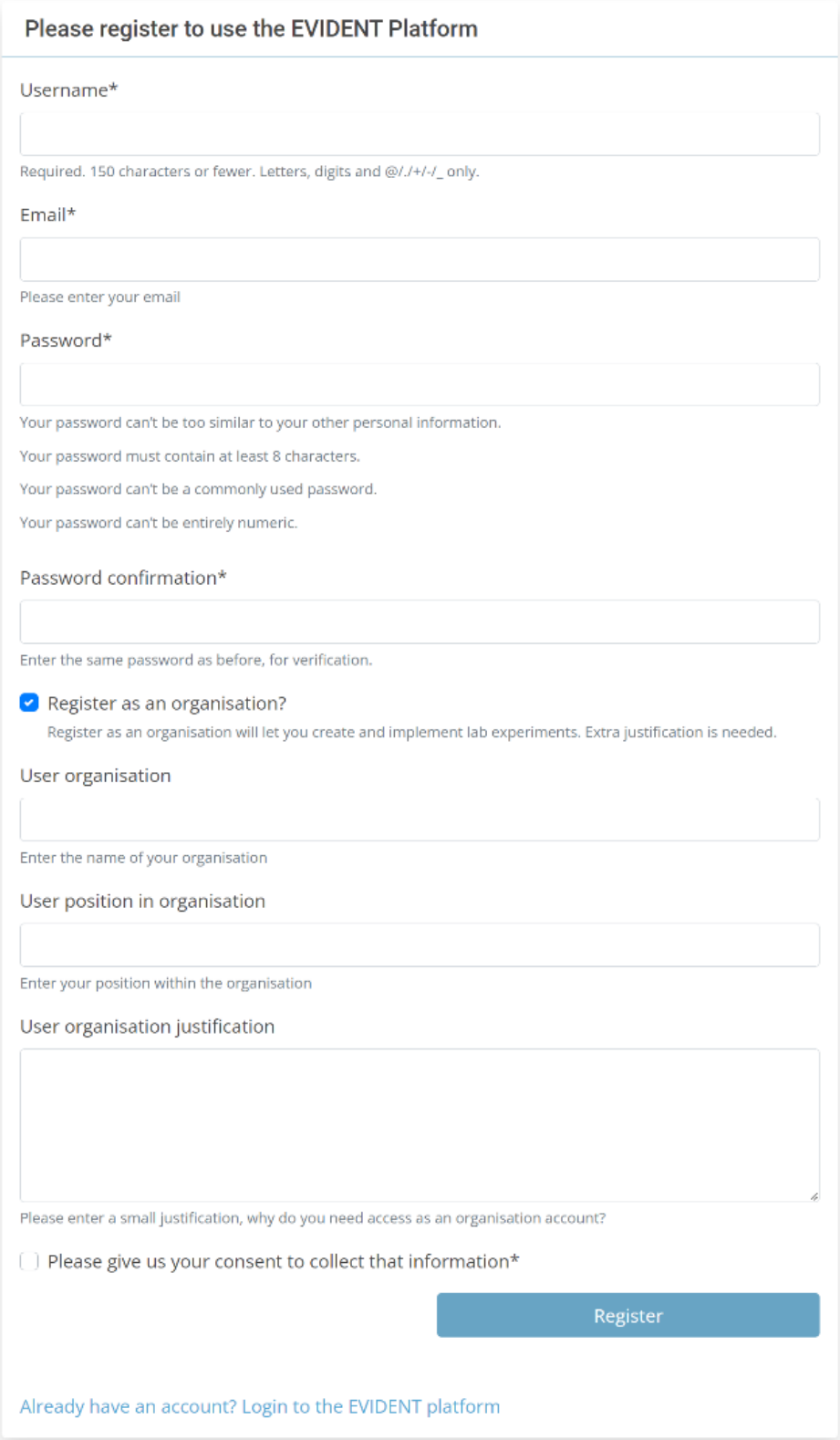
Figure 5: EVIDENT platform homepage

4.1 User and User profile component

In the following section, the user permissions regarding the user and the user profile component are presented.

User Registration

Each visitor can register for the EVIDENT platform by following the steps described in the registration form, which is shown in Figure 6. Each user can register as an organiser, however extra information is required. When a user registers as an organiser, additional information about the representative organisation, the user's position in the organisation and a small justification about the reason an organiser account is requested. If this is the case, an external verification procedure will validate the user's identity and the super-admin will grant the permissions to the user.



Please register to use the EVIDENT Platform

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

Please enter your email

Password*

Your password can't be too similar to your other personal information.

Your password must contain at least 8 characters.

Your password can't be a commonly used password.

Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

Register as an organisation?

Register as an organisation will let you create and implement lab experiments. Extra justification is needed.

User organisation

Enter the name of your organisation

User position in organisation

Enter your position within the organisation

User organisation justification

Please enter a small justification, why do you need access as an organisation account?

Please give us your consent to collect that information*

Register

Already have an account? Login to the EVIDENT platform

Figure 6: User registration form

Profile & Settings

Each registered user can log in to the EVIDENT platform and edit her/his profile information. On the profile page, presented in Figure 7, the user can update information such as their first and last name, the language the platform uses, the time zone, and the avatar while providing a small bio. The user can additionally change her/his email and password. In the settings page, presented in Figure 8, the user can apply for an organiser role (if not already), find more information about the way her/his personal data are used and being protected in the platform, and provide consent for processing her/his personal data, view or download the collected personal information and finally delete her/his account.

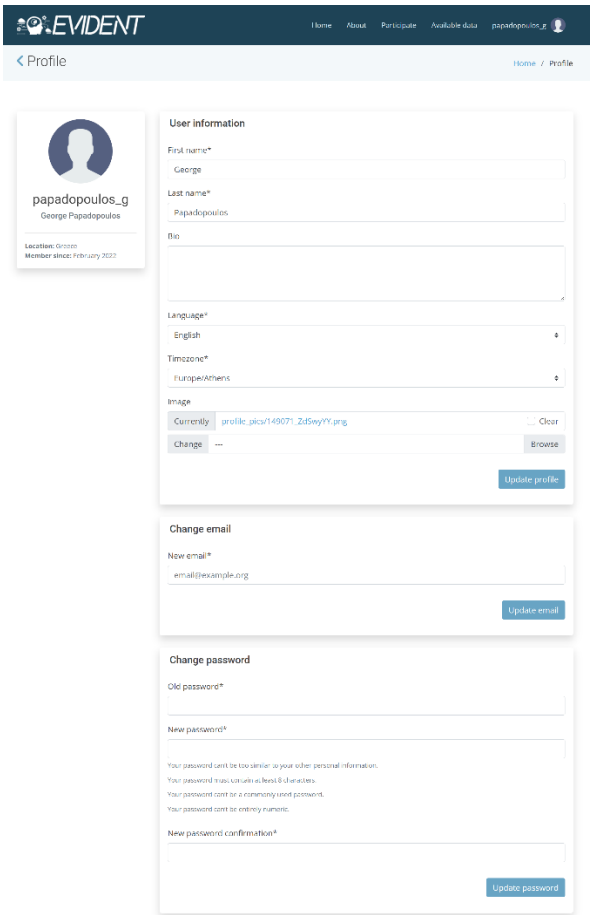


Figure 7 - User profile page

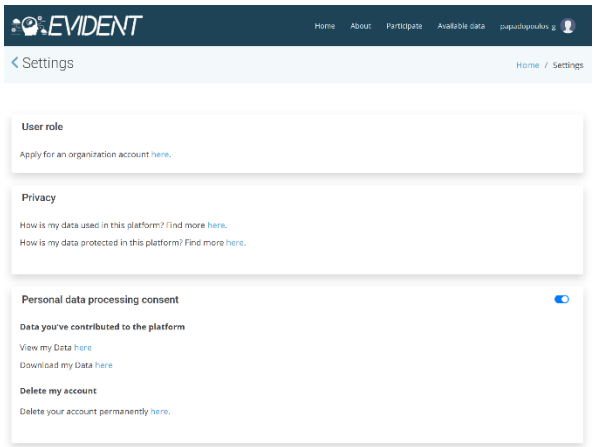


Figure 8 - User settings page

Demographics

Every user to participate in the available sessions should first fill in her/his demographic data. This can be done on the demographics page, presented in Figure 9. The questions about the user's age, gender, yearly income, race, employability, location, whether the user is a homeowner or not, and the level of education is mandatory.

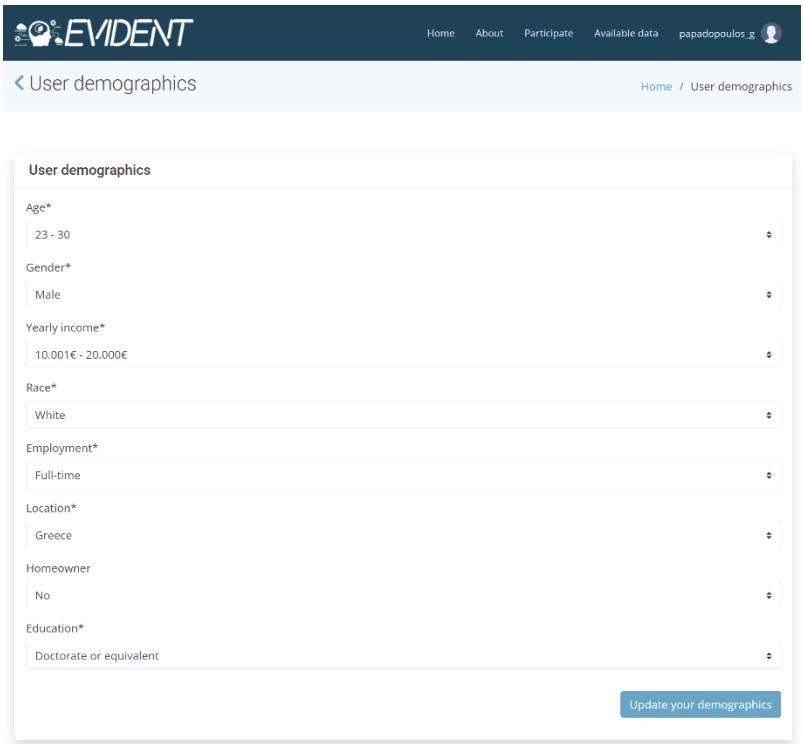


Figure 9: User demographics page

4.2 Survey Component

Section 4.2 is devoted to the user navigation of the survey component. In this section, the main views a user can see are presented.

Survey creation/edit/deletion

Based on Table 6, an organiser can create surveys. This is feasible through the user menu and more specifically the option “My Surveys”, as shown in Figure 10. On this page, all surveys the organiser has created are presented. The user can select a specific survey from the table and either publish, edit or delete it. Through the publish option the user characterizes a specific survey as finalised, meaning that the survey is ready to be used as a concrete step/application in a lab experiment. Expect from these actions, that the user can create a new survey by selecting the “Create new” button.

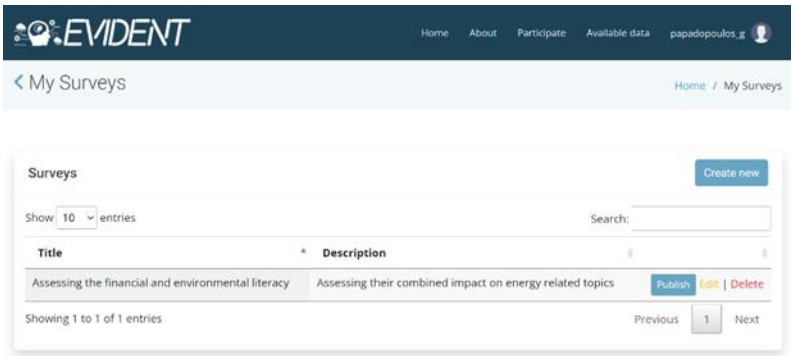


Figure 10: My surveys page

The "Create new" button will redirect the user to the survey creator page. The survey creator page, presented in Figure 11, provides an extensive toolbox and building blocks for the user to create advanced, including a powerful logic engine, surveys.

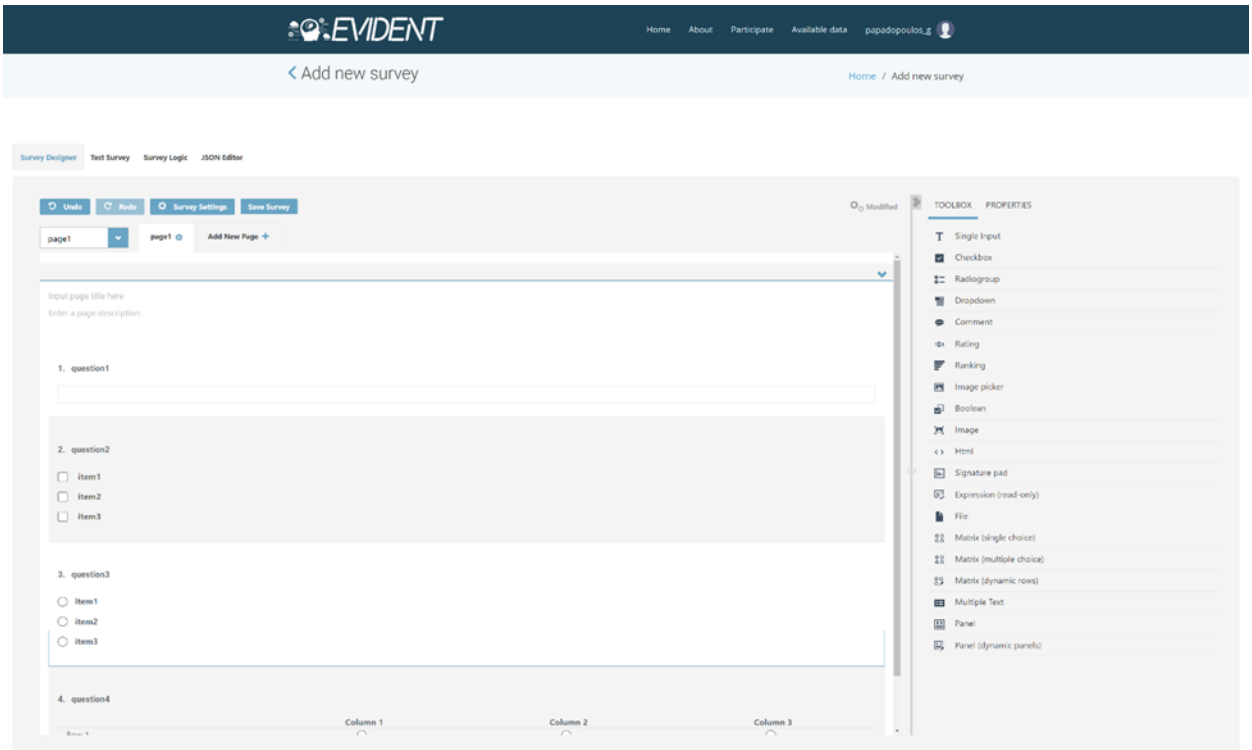


Figure 11: Create a new survey

The survey creator is based on the SurveyJS[1] JavaScript library which has been configured to undercover the needs of the EVIDENT platform. Right now, 20 different input types are supported, providing advanced features that would be useful for the extension of the platform.

4.3 Session Component

Each session or lab experiment consists of different steps or applications. There are two application types provided by the EVIDENT platform, surveys and serious games. The surveys are based on the survey component described in section 4.2 of the deliverable. The serious games component will be described in *D6.3 Gamification Tools of EVIDENT platform*.

Session creation/edit/deletion

Each organiser can create, edit, delete and track her/his sessions. The user can navigate to the menu and more specifically the option "My Sessions", as depicted in Figure 12. On this page, all sessions the user has created are presented. The user can select a specific session from the table and either publish, edit, share or delete it.

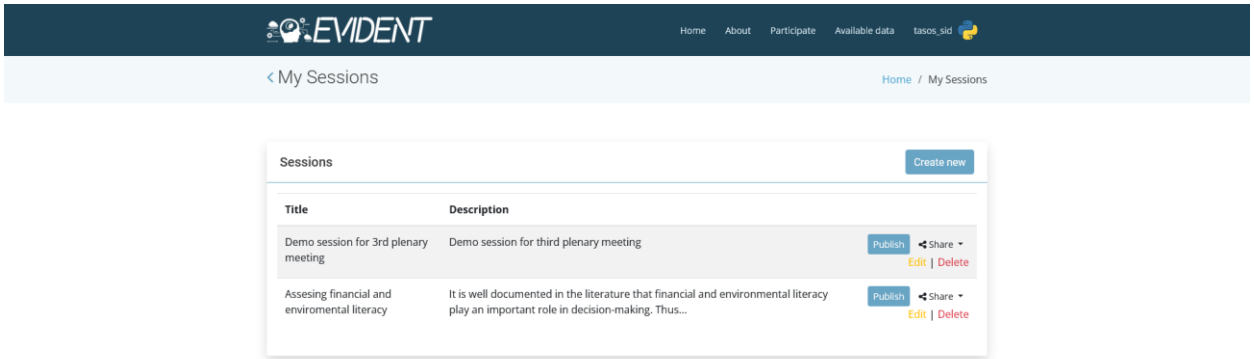


Figure 12: My sessions page

If the user selects to publish a session, then the session, as well as the connected applications (surveys or serious games), are locked for editing. This mechanism will protect the data collection mechanism and guarantee the proper operation of the EVIDENT platform. The user should unpublish a session to be able to edit either the session or the applications.

To create a new session, the organiser selects the "Create new" button. Figure 13 presents the session creator page. The user inserts a title, a description and the applications of the session. In case the organiser wishes to "lock" the session she/he can choose a password. This feature is quite useful in case the organiser wishes to share her/his session with specific participants (e.g., through an email).

[1] https://surveyjs.io

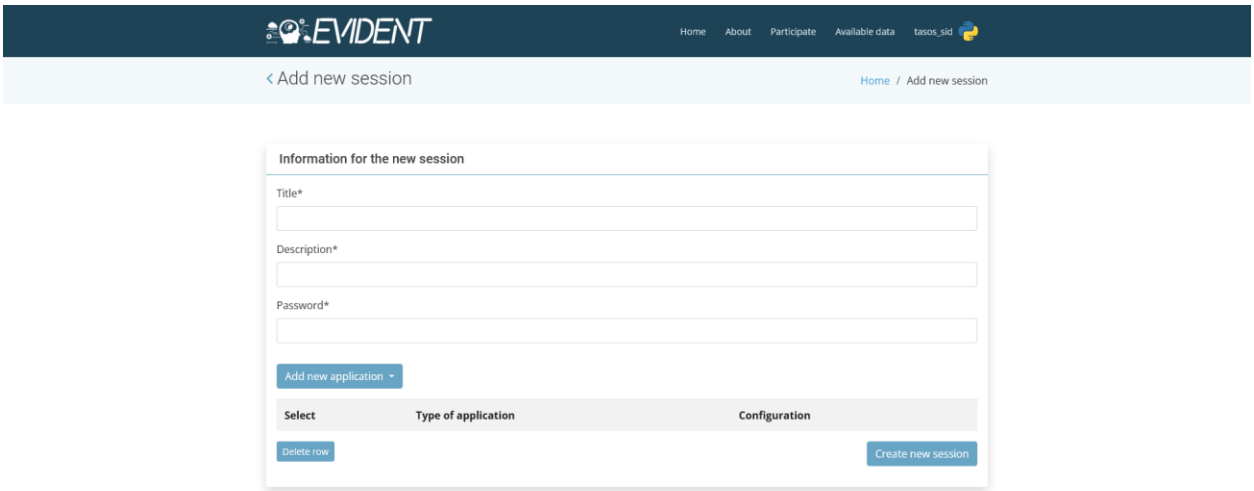


Figure 13: Session create page

In Figure 14, a full example of a session created in the EVIDENT platform is depicted.

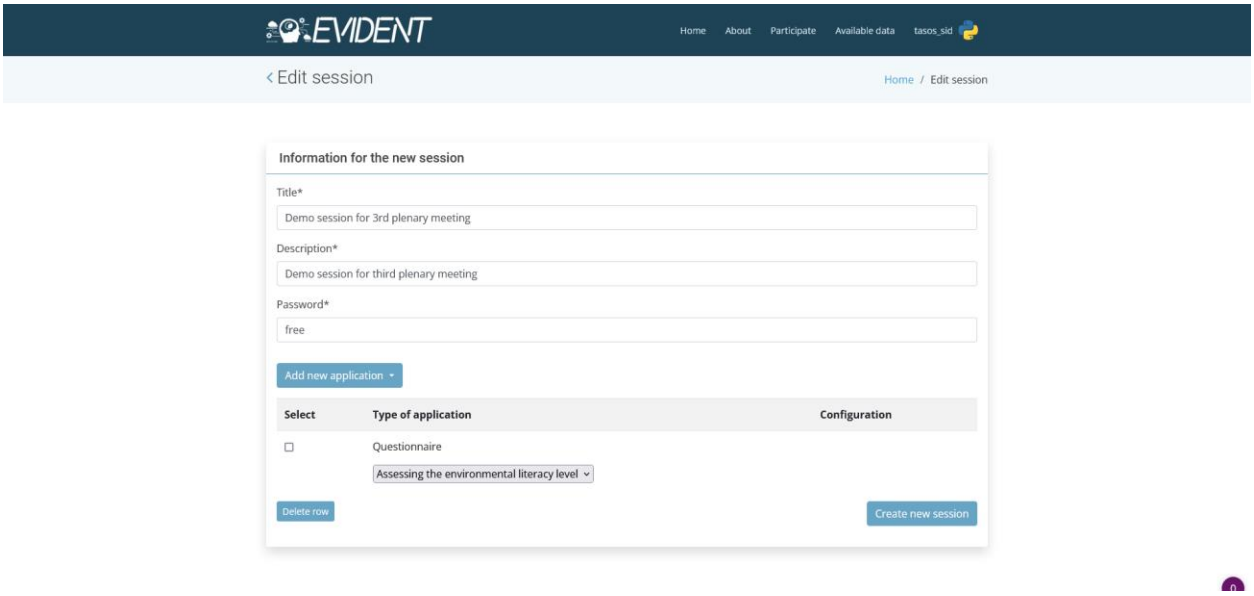


Figure 14: Example of a session

Session Participation

The users can participate in the available sessions by visiting the participate page in the main menu. On this page, a table with all available sessions is presented. For a user to be able to participate in a session she/he must first insert her/his demographic data. Right after the participant can select a session, a pop-up window with the relative information such as the title, the description and the application list will appear. If the session is “locked”, the participant should first enter the appropriate password to be able to start her/his participation. Based on the session, the participant should answer all available applications

(surveys and/or serious games) to complete her/his participation. Figure 15 presents an instance of user participation in an example session.

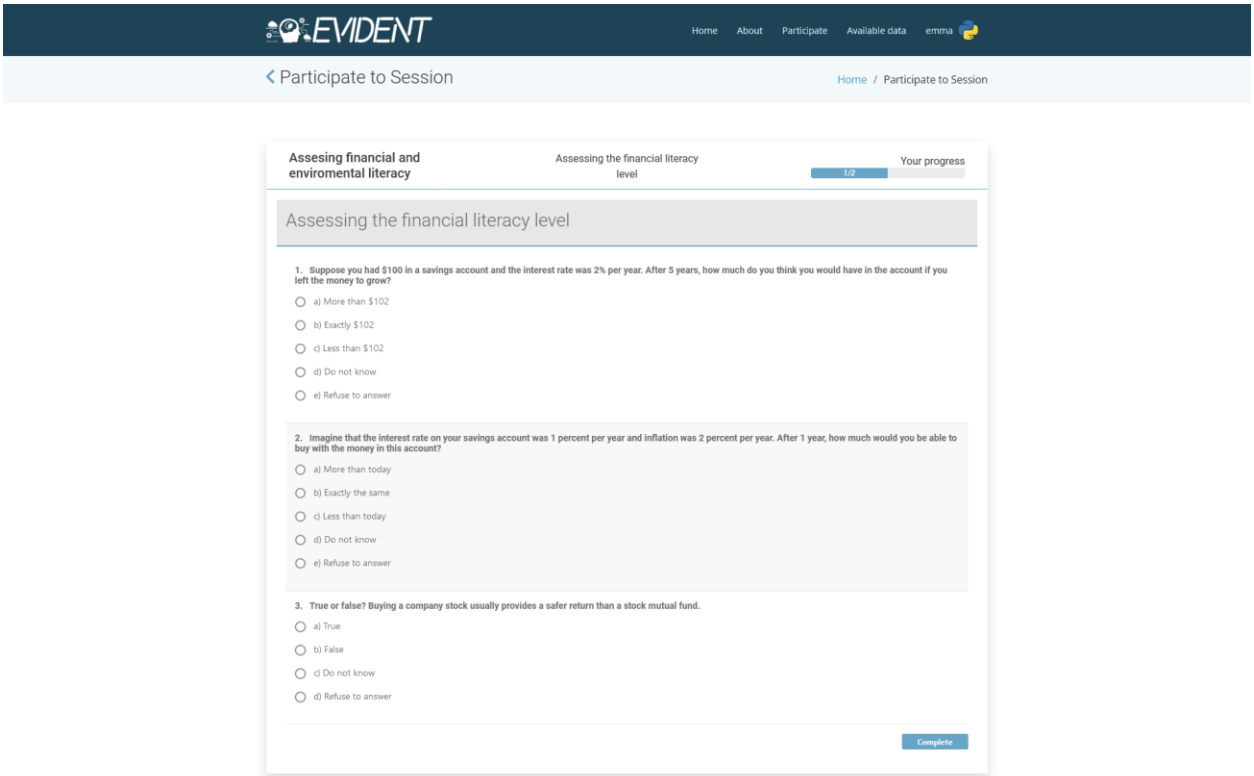


Figure 15: User participation in a session

5. Conclusion

This document describes the activities performed in the context of Task 6.2 "*System Development"* of the EVIDENT project providing information on the development of the crowdsourcing tools of the EVIDENT platform. The deliverable is a supporting document to the software developed in the context of Task 6.2 and provides information on the technical details in the development of the platform and information on the user interface and navigation to the platform. Additionally, it presents how the functional requirements regarding the development of the crowdsourcing applications have been met.

The development of the serious game application for the EVIDENT platform among the data extraction component (e.g., extraction of anonymised data) will be described in D6.3 "*Gamification Tools of EVIDENT platform*". The final feature of the EVIDENT platform, the development of the datahub services will be developed until Month 30 of the project and it will be described in D6.4 "*Datahub Services of EVIDENT platform*". The final version of the platform will be ready on M34 of the project and it will be presented in D6.5 "*Verification and Validation Report for final version of EVIDENT platform*".

The development of the crowdsource application is a major step towards the completion of the EVIDENT platform using as a foundation the third milestone of the project "*Architecture, design and integration documentation*" and paves the way towards milestone 9 (MS9) "*EVIDENT System Validation*". Overall, the deliverable provides a good synopsis of the progress that took place in the development of the software and it can be used as a future point of reference for best practices in future projects.

References

[1] JetBrains, “Django Developers Survey 2021,” 2021. [Online]. Available: https://lp.jetbrains.com/django-developer-survey-2021-486/.

[2] F. Shadhin, “The MVT Design Pattern of Django,” plainenglish.io, 4 May 2021. [Online]. Available: https://python.plainenglish.io/the-mvt-design-pattern-of-django-8fd47c61f582. [Accessed 18 July 2022].

[3] D. Rubio, “Django Design Principles,” Webforefront, [Online]. Available: https://www.webforefront.com/django/designprinciples.html. [Accessed 19 July 2022].

[4] Mozilla Foundation , “MDN Web Docs Glossary: Definitions of Web-related terms > MVC,” Mozilla Foundation , 18 February 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/MVC. [Accessed 20 July 2022].

[5] Django Software Foundation, “Django > Documentation > Datbases,” Django Software Foundation, [Online]. Available: https://docs.djangoproject.com/en/4.0/ref/databases/. [Accessed 20 July 2022].

[6] The PostgreSQL Global Development Group, “PostgreSQL 14.4 Documentation,” 2022.

[7] Django Foundation, “Djago Project,” Django Foundation, [Online]. Available: https://docs.djangoproject.com/en/4.0/topics/auth/. [Accessed 19 Jully 2022].