



UNIVERSIDADE
LUSÓFONA

Centro Universitário de Lisboa

Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação
Mestrado em Engenharia Informática e Sistemas de Informação

Management System for HCI Experiments

Orientador:

Prof Dr Nuno Maria Carvalho Pereira Fernandes Fachada

Co-Orientador:

Prof Diogo Nuno Dias Mesquita Gomes de Andrade

Marcelo Borges Alves | 21902415

2025

www.ulusofona.pt

Centro Universitário de Lisboa
Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação
Mestrado em Engenharia Informática e Sistemas de Informação

Management System for HCI Experiments

Dissertação defendida em provas públicas na Universidade Lusófona - Centro Universitário de Lisboa no dia 02/10/2024, nomeado pelo Despacho de Nomeação n.º: 885/2024, de 20 de junho, para obtenção do grau de Mestre em Engenharia Informática e Sistemas de Informação perante o júri com a seguinte composição:

Presidente: **Prof Dr Paulo Jorge Tavares Guedes**
Arguente: **Prof Dr Athanasios Vourvopoulos**
Vogal: **Prof Dra Maria Micaela Leal da Fonseca**
Orientador: **Prof Dr Nuno Maria Carvalho Pereira
Fernandes Fachada**

Marcelo Borges Alves | 21902415
2025

Agradecimentos

Gostaria de expressar minha profunda gratidão ao meu estimado professor Nuno Fachada, por sua orientação excepcional, sabedoria e apoio inabalável ao longo desta jornada de pesquisa. Sua mentoria inspiradora e insights valiosos foram fundamentais para o desenvolvimento deste trabalho. À Universidade Lusófona - Centro Universitário de Lisboa, agradeço pela infraestrutura e ambiente acadêmico enriquecedor que proporcionaram um espaço propício para o aprendizado e a pesquisa.

E minha mais sincera gratidão estende-se à minha amada família, cujo apoio incondicional foi determinante para conclusão do trabalho. Em especial, agradeço imensamente à minha esposa, Luci, cuja paciência, compreensão e amor foram o alicerce que sustentou cada desafio desta jornada. Sua presença constante e encorajamento incansável foi a luz que iluminou os momentos mais desafiadores.

Abstract

This paper presents the development of Xperis, an innovative system for managing and storing data from Multimodal Human-Computer Interaction (MHCI) experiments. The platform is designed to integrate with various data capture sources, ranging from questionnaires to 3D simulations, via a robust REST API that ensures secure and efficient communication between systems. One of Xperis' key strengths is its extensible architecture, which supports any data format through customisable plugins. These extensions enable flexible data visualisation and export, allowing adaptation to the specific needs of each experiment. To assess its effectiveness, integration tests were conducted in collaboration with the HEI-Lab team, using an interactive simulation environment developed with the Unity engine and .NET. The results demonstrated the system's flexibility and reliability in collecting, organising, and processing multimodal data. Thus, Xperis establishes itself as an essential tool for researchers and developers, offering a centralised and scalable solution for human-computer interaction experiments.

Keywords data management, systems development, MHCI, multimodal human-computer interaction, data types, extensible architecture, visualisation plugins, systems integration, API, REST

Resumo

Esta dissertação apresenta o desenvolvimento do Xperis, um sistema inovador para o gerenciamento e armazenamento de dados de experimentos em Interação Humano-Computador Multimodal (MHCI). A plataforma foi concebida para integrar-se a diversas fontes de captura de informações, desde questionários até simulações em 3D, por meio de uma API REST robusta, assegurando comunicação segura e eficiente entre os sistemas. Um dos diferenciais do Xperis é sua arquitetura extensível, que oferece suporte a qualquer formato de dados por meio de plugins customizáveis. Essas extensões possibilitam a visualização e exportação dos registros de forma adaptável às necessidades específicas de cada experimento. Para validar sua eficácia, foram conduzidos testes de integração em colaboração com a equipe do HEI-Lab, utilizando um ambiente de simulação interativa baseado na engine Unity e .NET. Os resultados evidenciaram a flexibilidade e confiabilidade do sistema na coleta, organização e processamento de dados multimodais. Assim, o Xperis se consolida como uma ferramenta essencial para pesquisadores e desenvolvedores, proporcionando um ambiente centralizado e escalável para experimentos de interação humano-computador.

Palavras-chave gerenciamento de dados, desenvolvimento de sistemas, MHCI, interação multimodal homem-computador, tipos de dados, arquitetura extensível, plugins de visualização, integração de sistemas, API, REST

Contents

Agradecimientos	iii
Abstract	v
Resumo	vii
Contents	ix
List of Figures	xi
List of Tables	xiii
1 Introduction	1
2 Background	3
3 Methodology	7
3.1 Definition of Application Requirements	7
3.1.1 Experiment Management	7
3.1.2 Data Storage	8
3.1.3 API	8
3.2 Data Model	9
3.3 Application Architecture	12
3.4 User Experience	14
3.5 Features and Functionality	15
3.5.1 User Profiles	15
3.5.2 User Management	17
3.5.3 Institution Management	17
3.5.4 Experiment Management	18
3.5.5 Templates Management	19
3.5.6 Plugin Management	19
3.5.7 Data Management	21
3.5.8 Application Programming Interface (API)	21
	ix

4	Results	25
4.1	Local Test	25
4.2	Remote Test	28
5	Discussion and Future Work	31
6	Conclusion	33
	Bibliography	35
A	User's Guide	37

List of Figures

3.1	UML Class Diagram (Step 1): The first 3 main classes: Experiment, Metadata and Data.	9
3.2	UML Class Diagram: Final version with all classes, interfaces and enums. . .	11
3.3	Model View Controller (MVC) diagram of the Xperis application, highlighting the main classes and their interactions.	12
3.4	Component diagram showing the web portal, the application, and the database server main components.	13
3.5	Web Portal's User Management first page listing all registered users and its functionalities.	16
3.6	Web Portal's Institution Management first page listing all registered institutions and its functionalities.	17
3.7	Web Portal's Experiment Management first page listing all registered experiments and its functionalities.	18
3.8	Web Portal's Experiment Metadata Management first page listing all registered metadata and its functionalities.	19
3.9	Web Portal's Template Management first page listing all registered templates and its functionalities.	20
3.10	Web Portal's Plugin Management first page listing all registered plugins and its functionalities.	20
3.11	Web Portal's Experiment Data Management first page listing all collected data and its functionalities.	21
3.12	Web Portal's API Documentation page listing all active endpoints and its parameters.	22
4.1	Web Portal's Experiment and Experiment Metada pages listing the inserted records for Questionnaire D	26
4.2	Postman application screen that shows a request to add data to the experiment through POST /metadata REST API endpoint.	27
4.3	Google Sheets screen that shows a CSV file generated by Xperis after registering data from 5 participants.	28
4.4	Unity Application screens related to sociodemographics, behavioral and experience questions.	29

4.5	Web Portal's Experiment Data Management first page listing all received data from Unity integration.	29
4.6	CSV File consolidated with all participants data.	30

List of Tables

4.1	Description, Type and Expected Values for the first questions of Sociodemographics Questionnaire D	25
-----	--	----

Chapter 1

Introduction

Over the past few decades, there has been a significant increase in computing power and network bandwidth. This has led to a shift in the computing industry from traditional local applications to services delivered through cloud computing, with a focus on making technology more human-centered (Pantic et al., 2008; Pantic and Rothkrantz, 2003). As a result, there has been a rise in research on multimodal human-computer interfaces (MHCI).

MHCI, according to Pantic et al. (2008), unlike traditional interfaces such as keyboards and mice, interacts with users using natural modalities such as facial expressions, body gestures, and verbal and non-verbal vocal cues. The goal is to simplify the use of the computer system, reduce user distraction, and enhance user satisfaction and productivity. Consequently, MHCI is seen as a necessary step towards future pervasive systems (Jaimes and Sebe, 2007; Pantic and Rothkrantz, 2003).

However, the development of MHCI presents challenges. According to Pantic et al. (2008, 2007), the difficulty arises from two main issues. First, interpreting human behavior and modeling multimodal human-computer dialogue are closely tied to context sensing. Secondly, constructing MHCI systems involves a lengthy development and design process to identify requirements and transpose them into a useful and concise interface, as well as making it accessible to all experiment-evolving staff.

In turn, the process of designing and developing web systems is a complex and challenging endeavor, influenced by various factors with potentially conflicting needs, as mentioned by Cloyd (2001), Ivory and Hearst (2002), and Siegel (2003). Ensuring scalability is a critical consideration, addressing the system's ability to handle new requirements, increased content, higher usage, and security needs. However, the growth of web applications brings challenges such as reliability issues, usability problems, and security breaches that demand careful attention (Williams, 2000).

As Becker and Berkemeyer (2002) said, web development faces a shorter time frame to deliver applications with a focus on enhanced reliability, performance, and security, given the increasing mission-critical nature of web-based systems. Usability is a key concern, as meeting the diverse expectations of users with varying skills is a challenging task, involving factors such as design layout, consistency, accessibility, navigation, personalizing, and more. Additionally, making web content universally accessible, especially for disabled users, presents

a development challenge, with slow adoption of standards and technologies to support accessibility.

Localization, involving adapting web applications for different countries, adds another layer of complexity beyond simple language translation. Terms such as scalability, reliability, availability, maintainability, usability, and security are crucial to describe the architectural qualities of a web system (Williams, 2000). The challenge extends to ensuring continuous operation (24/7/365) and compatibility across various browsers and client devices. Furthermore, complexities increase when designing web applications for mobile and device-independent operations, requiring new testing and validation approaches (Murugesan and Venkatakrishnan, 2005). Stakeholder considerations, including system maintainers, organizations, and funders, add additional challenges to the design and development of sustainable web systems.

In this dissertation, we address the challenge of developing a full-stack web application to manage the content generated from MHCI experiments. These contents are of different types, such as answers to questionnaires, actions taken by users in the context of the experiments, biofeedback data (EEG data, for example), audio and/or video. It basically consists of an Application Programming Interface (API) so that the data from the experiments can be archived, a front-end for accessing the data, and a backend for managing experiments, responsibilities, users and data. In addition to being made available in open source form for use by the scientific community.

This dissertation is organized as follows. Chapter 2 provides an overview of the literature on experiment data management systems. Chapter 3 describes the methodology used in the development of the system, including the data model used in the system and the system's features and functionality. Chapter 4 presents the results of the system's implementation and evaluation. Finally, Chapter 6 concludes the paper and discusses future work.

Chapter 2

Background

In the contemporary software industry, effective management of experimental data poses a significant challenge. The use of web interfaces for data storage and management has emerged as a notable trend in response to this challenge. This chapter explores different strategies used by researchers and developers to streamline the handling of experimental data through Web interfaces. From user-friendly interfaces to robust backend architectures, these solutions share a common objective: to optimize the process of managing experimental data using Web technologies.

The examination commences with the work of Morrell et al. (2017), which introduces the Experiment Data Depot (EDD), an online tool designed to serve as a repository for experimental data and metadata within the realm of synthetic biology. In response to the nascent ability to predict the outcomes of biological designs despite enhanced efficiency in their production, EDD addresses the critical need for large quantities of high-quality data for the parameterization and testing of predictive models. The platform offers a user-friendly interface for uploading various data types, facilitating data visualization, and supporting standardized data export for use in predictive algorithms. The paper delineates the features of EDD and demonstrates its efficacy through three distinct use cases: storage of characterized synthetic biological parts, leveraging proteomics data to enhance biofuel yield, and utilizing extracellular metabolite concentrations to predict intracellular metabolic fluxes.

Schipor and Vatavu (2023) introduce GearWheels, a software tool designed for conducting studies on gesture input with wearables, encompassing devices such as smartwatches, rings, and glasses. Noteworthy for its event-based asynchronous software architecture, GearWheels is exclusively implemented with web standards, communication protocols, and data formats. This design choice renders GearWheels highly flexible, allowing seamless support for a variety of wearables through HTTP and WebSocket communications. Setting itself apart from previous tools dedicated to gesture acquisition, elicitation, recognition, and analysis, GearWheels adopts a web-based, wearable-oriented, and experiment-centered architectural approach. The paper provides a practical illustration of GearWheels' capabilities by demonstrating touch stroke-gesture and motion-gesture input acquisition using devices affixed to the index finger, wrist, and temple of glasses. Additionally, a technical evaluation in the form of a simulation experiment is conducted, reporting the request-response time performance of GearWheels'

software components when paired with off-the-shelf wearables. GearWheels is available as open-source software, aiming to support researchers and practitioners in the implementation of studies focused on gesture input with wearables.

In turn, Shen and Pantic (2013) introduce the HCI² Framework, a software system designed to advance development and research in the realm of MHCI systems. The framework adopts a publish-subscribe architecture, incorporating a shared-memory-based data transport protocol for message delivery and a TCP-based system management protocol to ensure runtime system structure integrity. With the incorporation of bridging modules, the framework achieves interoperability with other software frameworks, including Psychone (Machines, 2007) and ActiveMQ (Apache, 2009). This work not only focuses on the core communication middleware but also presents the framework’s integrated development environment (IDE). This IDE offers a comprehensive graphical environment to support all stages of a typical MHCI system development process, encompassing module development, debugging, packaging, and management, as well as overall system management and testing. Furthermore, the paper provides an example of a complete system developed using the framework — the CamGame system. This system, representing a computer game based on handheld markers and low-cost cameras, serves as a practical demonstration of the framework’s capabilities in facilitating the development of complex MHCI applications.

Fdez-Riverola et al. (2012) introduce AIBench, a Java (Oracle Corporation, 2012) desktop application framework developed by the SING group in Ourense, Spain, with a primary focus on scientific software development. Aimed at bolstering the productivity of research groups, AIBench follows the Model-View-Controller (MVC) design pattern, enabling programmers to construct applications utilizing only three key concepts: operations, data types, and views. The framework encompasses a wide array of functionalities commonly found in scientific applications, such as user parameter requests, logging facilities, multithreading execution, experiment repeatability, and the generation of graphical user interfaces. AIBench is designed with an emphasis on simplicity, offering a streamlined approach to application development within the scientific domain. AIBench adopts a plugin-based architecture, facilitating the incorporation of various functionalities but also allows for the assembly of new applications by reusing modules from prior development projects. This modular structure aligns with the evolving nature of scientific research, offering flexibility and adaptability for diverse research requirements.

Lastly, Soleymani et al. (2012) presents MAHNOB-HCI, a meticulously recorded multimodal database in response to affective stimuli, specifically designed to advance research in emotion recognition and implicit tagging. The database encompasses a multimodal setup, facilitating the synchronized recording of face videos, audio signals, eye gaze data, and physiological signals from the peripheral and central nervous systems. Twenty-seven participants, representing diverse genders and cultural backgrounds, took part in two experiments. In the initial experiment, participants viewed 20 emotionally charged videos, providing self-reported assessments of their felt emotions. Parameters such as arousal, valence, dominance, and predictability, along with emotional keywords, were used for this evaluation. The second experiment involved presenting short videos and images without any tags initially, followed by

their display with either correct or incorrect tags. The participants then assessed agreement or disagreement with the presented tags. The recorded videos and bodily responses underwent segmentation and were meticulously stored in a database. Accessibility to the academic community is facilitated through a user-friendly web-based system. The paper details the analysis of the collected data, presenting results for both single-mode and modality fusion in the context of emotion recognition and implicit tagging experiments. These results underscore the potential applications of the recorded modalities and underscore the significance of the emotion elicitation protocol used. The MAHNOB-HCI database stands as a valuable resource for researchers exploring emotion-related phenomena and implicit tagging mechanisms.

The experiment data management software discussed in this dissertation, Xperis utilizes some of the ideas discussed in these papers but uses different tools and technologies. There are some similarities with EDD (Morrell et al., 2017), as both are web-based applications that allow users to upload data and metadata from different formats, both support API interface communication, and both provide a visualization option for recovery. However, the software presented in this thesis is designed for Human-Computer Interaction (HCI) experiments, while EDD is designed for synthetic biology experiments. GearWheels (Schipor and Vatavu, 2023) also presents a web-based application for data collection, but it is designed for study of gesture input with wearables, using an asynchronous software architecture based on events, while Xperis employs a NoSQL database (Redmond and Wilson, 2012) and a monolithic architecture (Fowler, 2002). Although GearWheels also supports human-computer interface experiments, its main focus is in being a framework for communication between different systems and an IDE for development. AIBench (Fdez-Riverola et al., 2012), although using a Java programming language as utilized in Xperis, is a desktop application framework that only allows programmers to construct applications using the functionalities commonly found in scientific applications. And finally, MAHNOB-HCI (Soleymani et al., 2012) is a multimodal database for affect recognition and implicit tagging, but it is a database and not software for experiment management.

Xperis stands out by integrating the strengths of the previously discussed systems into a unified platform designed specifically for Human-Computer Interaction (HCI) experiments. While EDD is tailored to synthetic biology, Xperis adapts its web-based architecture to accommodate the diverse requirements of HCI studies, offering robust tools for real-time data collection, organization, and visualization. By leveraging a NoSQL database, Xperis provides scalable support for heterogeneous data types, enabling researchers to seamlessly store and manage data from a variety of sources. This flexibility makes Xperis particularly suited for experiments that involve complex and multimodal datasets, a capability that sets it apart from systems like GearWheels, which emphasize event-driven architectures but lack extensive support for multimodal data integration.

Furthermore, Xperis enhances its functionality with a plugin-based approach, allowing researchers to customize and expand the platform to meet specific visualization needs. For instance, users can add specialized plugins to support advanced visualizations for unique data types, such as physiological signals or interaction logs, facilitating deeper analysis and insight. This modularity draws inspiration from AIBench’s framework but extends it with a focus on

ease of use and adaptability for non-programmers. Additionally, by incorporating multimodal synchronization capabilities akin to MAHNOB-HCI, Xperis provides researchers with an all-encompassing environment to analyze interactions across multiple channels. This combination of flexibility, scalability, and customization makes Xperis a comprehensive and versatile tool for advancing HCI research.

Chapter 3

Methodology

Xperis is a complete web platform for managing HCI experiments. Through this platform, HCI researchers and professionals can record experiments, manage the access of other collaborators, include new questionnaires and interactions, collect data of various formats and extract information in a simple and centralized way. The platform also provides a detailed API for connecting through other devices to collect and retrieve recorded data.

In this chapter, in Section 3.1 we will outline the outcomes of requirement gathering meetings with stakeholders categorized into Experiment Management, Data Storage, and API. Section 3.2 presents the data model that has been implemented to fulfill the HEI-Lab requirements. This model serves as the foundation for managing and organizing data effectively. We also provide a comprehensive overview of the design of the application, as detailed in Section 3.3, addressing the technologies employed in this design process, focusing on their advantages and the motivations behind their selection. Additionally, we discuss the rationale for opting for a NoSQL database and its specific advantages within the project. Finally, in Section 3.5, we introduce the system's features and functionality, including the graphical user interface — Xperis HCI Portal — and the API. This section will provide an in-depth look at the web-based administrative portal and the service that facilitates the transmission of data from experiment applications to the central database.

3.1 Definition of Application Requirements

During the first half of 2021, several requirements gathering meetings were held with stakeholders. These meetings were attended by the supervisor, co-supervisor and the HEI-Lab team. The requirements were divided into 3 main distinct functionalities: Experiment Management, Data Storage, and API.

3.1.1 Experiment Management

The application must provide an authentication system where it is possible to control which users and institutions have access. For this, it must consider 3 profiles: Administrators, Researchers and Assistants. Administrators should be able to:

- Log in;

- Create credentials for researchers and assistants. These will have an expiration date that is also defined by the administrators, and at any time, the administrator can select a researcher and change the expiration date of their credential;
- Recover passwords for researchers if they lose them;
- Add new applications and institutions to the list or even remove them;
- Associate or disassociate a researcher with an application;
- Associate or disassociate a researcher with an institution;
- Know how many people are using each application.

Researchers and assistants should be able to:

- Log in;
- Change their passwords;
- Manage participants: create participants, change their passwords, and assign them an institution and/or application (that the experimenter are themselves assigned to);
- Download experiment data.

On the other hand, data collectors will not have any interaction with the front end of the web page. They will only use a researcher-provided credential and can only log in if the researcher's credential is still valid. This is done through a unique ID stored with the rest of the application data.

3.1.2 Data Storage

The system should allow storing different types of experiment data.

- Applications should be able to send data to the server;
- Different types of data may be stored in a database (such as biofeedback data that is produced in large quantities per second, audio, or even video);
- One application can use only one type of data or multiple types of data.

3.1.3 API

The API allows access to experiment data (via the authentication system). Its requirements are as follows:

- There are data that can only be accessed by the researcher accompanying the participant;
- The researcher can export data from any institution and application with which they are associated. They can export data from one or more participants, and specify a time frame of data to export;

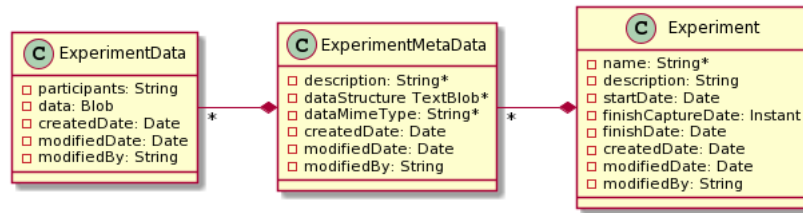


Figure 3.1 – UML Class Diagram (Step 1): The first 3 main classes: Experiment, Metadata and Data.

- The administrator knows which researchers exported what data and when. The researchers then assume responsibility for the security and privacy of the records they have downloaded.

3.2 Data Model

The data model created to address the requirements of managing experiments relies heavily on a structure encompassing the essential components of an HCI experiment, namely: a detailed description of the experiment, the specification of data that will be collected, information about the participants involved, and the actual data collected during the experiment.

Initially, to define an experiment, a detailed description of what it is intended to evaluate and which researchers are involved is necessary. In addition, the researchers also have a forecast of when they will start and finish the data collection phase. This data was incorporated into the `Experiment` class.

However, an experiment does not always have only one type of data that will be collected. One can, for example, collect personal data, biometric data, and responses to a behavioral questionnaire. And for this reason it was necessary to create a new intermediary class for this purpose, `ExperimentMetadata`.

In compliance with the General Data Protection Regulation (GDPR), all participant data stored in the system is anonymized and associated only with a unique, non-identifiable key. Personally identifiable information (PII) is never stored within the Xperis platform, and it is not possible to trace the data back to a specific individual. Consent to participate in the study is obtained directly through the Unity-based applications, which present the terms of participation prior to data entry. If a participant later chooses to withdraw their consent, the system allows the responsible researcher to locate and permanently delete the corresponding data using the participant's identifier. These measures ensure that ethical standards and data protection requirements are upheld throughout the data lifecycle.

In the `ExperimentData` class, in addition to the identifier and data collected, it was also necessary to include the format and date of obtaining this data. The Class Diagram presented in Figure 3.1 shows core relations between this three classes.

The experimenters involved in the experiment are represented by `User` class. The researcher who owns the experiment must be mapped in the `user` field in the `Experiment` class. To differentiate the various members of the experiment, two fields are required: `members` and

assistants. An experiment member is another researcher from the same institution who can create new experiments and collaborate with experiment owners. An experiment assistant assists researchers in capturing and analyzing the data. They are not allowed to modify the basic data of the experiment. These roles will be discussed in more detail in Section 3.5.1.

The complete class diagram contains 13 classes and is shown in Figure 3.2. The presented classes have the following responsibilities:

User Class that represents a user in the system. It stores personal data such as password and email and allows one to control access to the system.

Role Enum that represents the role of a user in the system (e.g., admin, master, user).

Institution Class that represents an institution with associated researchers.

ExperimentStatus Enum that represents the status of an experiment (e.g., created, confirmed, started, finished).

Experiment Class that represents an experiment. Through this class, it is possible to list all the users involved in the experiment as well as record the progress of the phases and their current status.

ExperimentMetaData Class that represents metadata associated with an experiment. It represents the type of data that will be stored in the experiment, the type of media, and the expected structure.

ExperimentData Class that represents data associated with an experiment. This is really the centrepiece of the system and is responsible for storing the data collected in the experiment encoded into base64 format.

DataTemplate Class that represents a data template. It is a metadata model that can be reused in other experiments.

IViewer Interface that represents a generic viewer for different media types. Any new plugins that are added to the system must be implemented using this interface.

CSVViewer Class that represents a viewer for Comma Separated Values (CSV) media type. Extends **IViewer** for displaying and exporting data stored in CSV format

XDFViewer Class that represents a viewer for Extensible Data Format (XDF) media type. Extends **IViewer** for displaying and exporting data stored in XDF format

ActionLog Class that represents an action log related to experiments. This class stores all user actions. These actions can be UPDATE, DOWNLOAD DATA, UPDATE DATA.

AbstractAuditable Class that represents an abstract auditable entity. It contains the audit fields to control when each system record is created and changed.

Classes marked with «Auditable» are auditable entities, which means they track creation and modification information through the attributes inherited from the **AbstractAuditable** class.

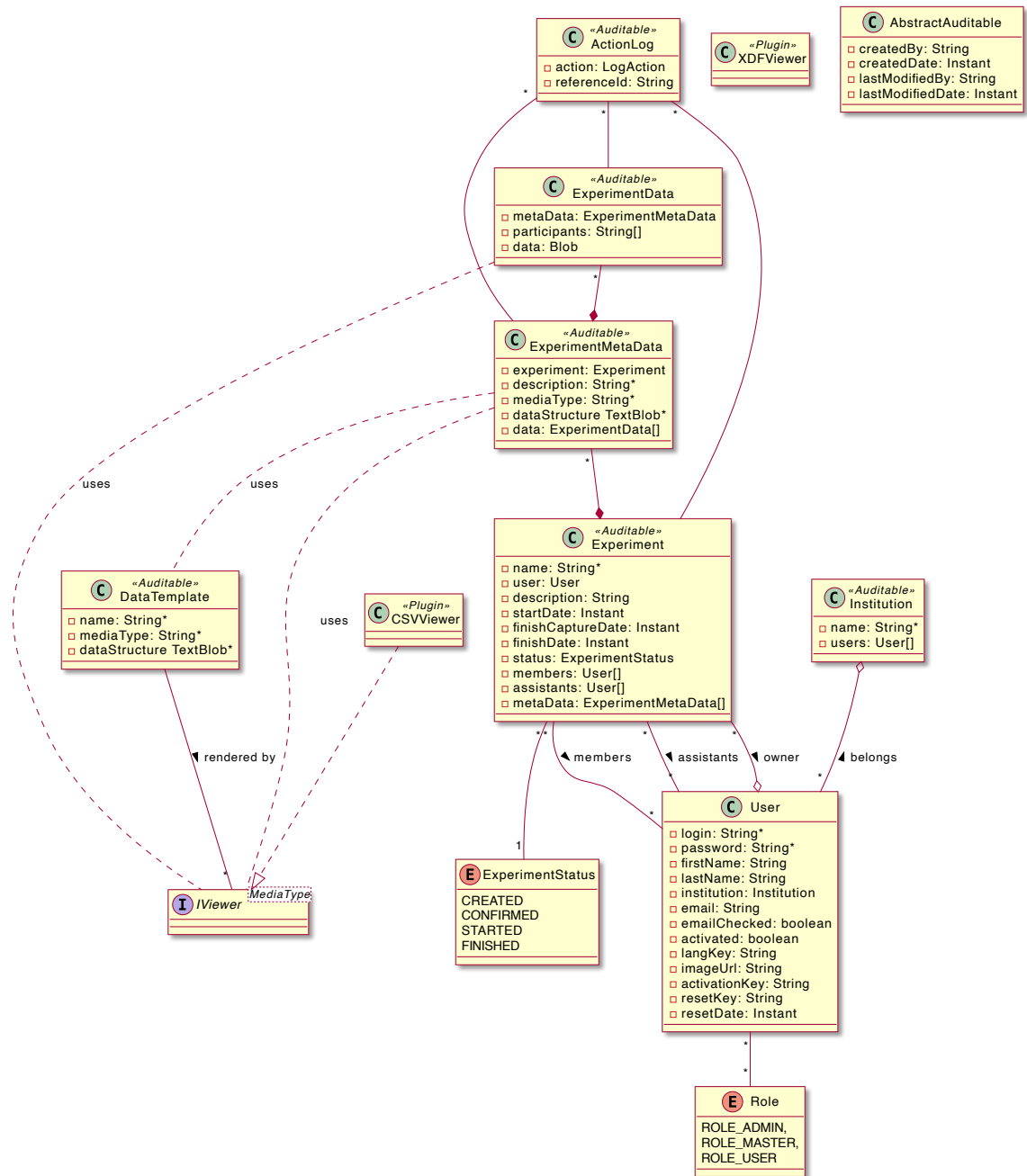


Figure 3.2 – UML Class Diagram: Final version with all classes, interfaces and enums.

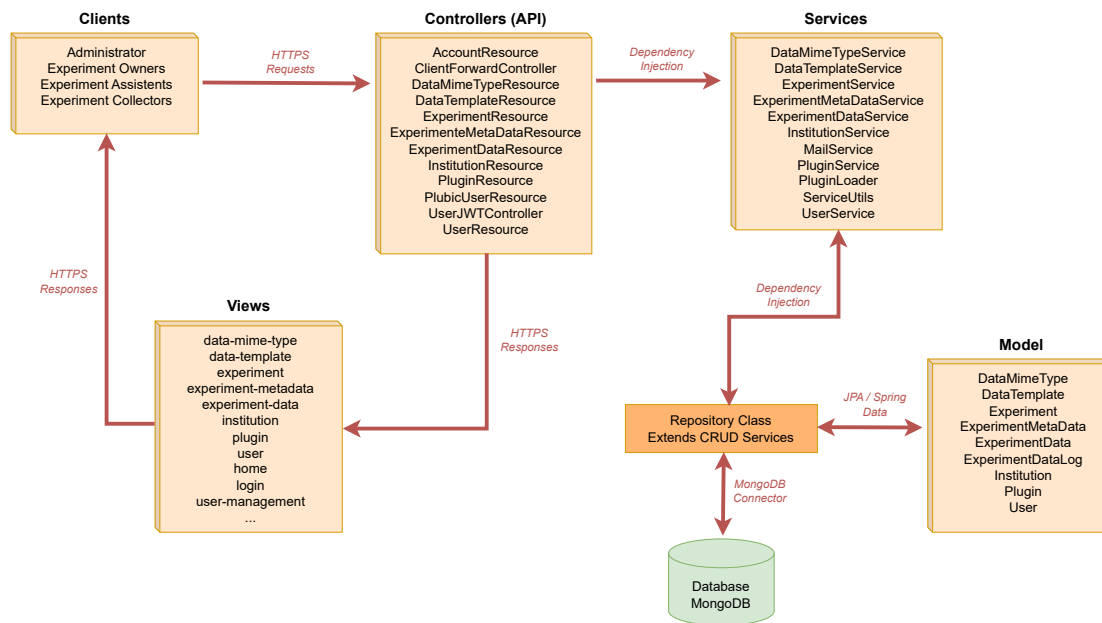


Figure 3.3 – Model View Controller (MVC) diagram of the Xperis application, highlighting the main classes and their interactions.

3.3 Application Architecture

In this section, we present the Xperis application architecture, which integrates Java Spring Boot (Walls, 2016) as the backend framework and Angular (Fain and Moiseev, 2018) as the frontend framework. We start by providing an overview of a web application architecture standard called model-view-controller (MVC) and how it was used in the project.

The Model-View-Controller (MVC) architecture is a widely adopted pattern for web application development. It divides the application into three components: Model, View, and Controller. The Model represents the data and business logic, the View handles the user interface, and the Controller acts as an intermediary, handling user input and updating the Model and View accordingly. This separation of concerns promotes modularity, maintainability, and code reusability. The MVC architecture of Xperis is presented in Figure 3.3.

In a Java Spring Boot and Angular application, the Model consists of entities or data transfer objects (DTOs) that interact with the database. The View is implemented using HTML templates, CSS stylesheets, and Angular components to render the user interface. The Controller, in Spring Boot, handles HTTP requests, invokes business logic, and returns responses, while in Angular, it is implemented through services that communicate with backend APIs and manage data flow. The MVC architecture facilitates independent development, testing, and modification of components, leading to cleaner code, easier maintenance, and improved collaboration among development teams.

Spring Boot, a popular Java framework for web development, streamlines the process by offering a rich set of features and a convention-over-configuration approach. Its architecture, centered on starters and preconfigured dependencies, enables rapid application setup while leveraging the Spring ecosystem’s modules like Spring Data and Spring Security. Following the

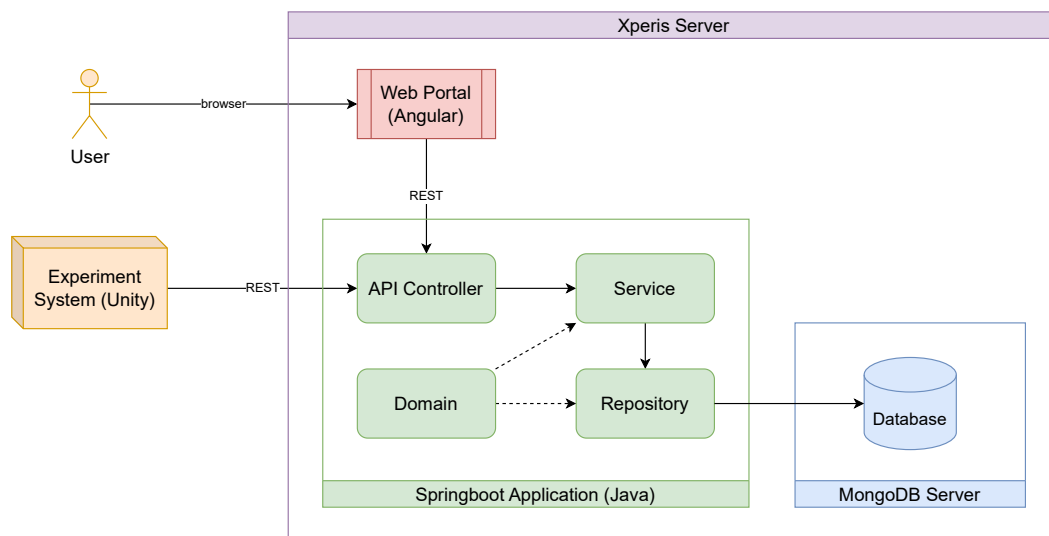


Figure 3.4 – Component diagram showing the web portal, the application, and the database server main components.

Model-View-Controller pattern, Spring Boot simplifies data handling, client interactions, and request handling through auto-configuration and annotations, allowing developers to focus on business logic rather than complex setups.

Conversely, Angular, a frontend framework, utilizes TypeScript for dynamic web application development. It employs a component-based architecture that divides applications into reusable components, thus facilitating easier maintenance. Angular’s MVC pattern separates concerns: the Model handles data, Views render UI with HTML and CSS, and Controllers manage logic via Angular services. Features include routing, form validation, and integration with Angular CLI for efficient project management.

After those definitions we focus in the specific details of the application. The Xperis component diagram, depicted in Figure 3.4, highlights the interactions between the application, the user, and other systems. Within the “Xperis Server” box, it is possible to observe three different components: the application server, the user interface, and the database.

The graphical interface or web portal, as previously described, uses the Angular framework and runs within a static embedded container server. It includes all main functionalities, such as user management, experiment creation, experiment data upload, and even configuration of plugins to display data. It has a top menu bar with four items. The Entity menu includes options such as Institution, Experiment, Metadata, Data, and Templates. For each of them, there are options to create, view, edit, and delete items. The Administration menu includes User Management, Metrics, Configuration, Logs, Data Types, and Plugins. These options are available according to the user’s profile, which can be administrator, master, or user. The Language menu shows 2 options: English and Portuguese, although it is possible to add more languages using Java i18n, which provides internationalization support through resource bundles. The Account menu includes User Settings, Password Change, and Logout. A detailed description of each functionality is provided in Section 3.5.

The main application, written in Java JDK 11 using the Spring Boot framework, also runs in this container. With Spring Boot, the application is packaged as an executable JAR file that includes the Undertow embedded server, making the application self-contained and easy to distribute. When we run the Spring Boot application, the embedded server starts up and listens for incoming HTTP requests on a specified port. It then routes the requests to the appropriate Representational State Transfer (REST) endpoint based on the defined mappings. The endpoint methods execute the necessary business logic, access data from databases or external services, and produce the desired response, which is then sent back to the client. Provides a REST API interface that contains all the services needed to serve the graphical interface, as well as for data input and view performed by external systems. In this context, the only external system considered is a prototype developed by a HEI-Lab researcher in the Unity game engine (Unity Technologies, 2005), as will be discussed in the next chapter. The main application is divided into four major packages, namely the controller, service, domain, and repository, as shown in Figure 3.4. The first deals with the external connection using the HTTP protocol and the REST standard. The service package contains the business rules related to each of the functionalities. The domain package describes the entities. Finally, the repository deals with the connection to the database. These packages are supported by others such as security, configuration and utility.

The last component is the MongoDB server (Chodorow, 2013). It is an independent application running on the same machine that is responsible for managing the database system. MongoDB is a powerful, flexible, and scalable NoSQL database system designed to handle vast amounts of data with ease. It follows a document-oriented model, storing data in JSON-like documents that can vary in structure (Crockford, 2006), allowing dynamic and agile data management. With its intuitive query language, MongoDB offers efficient data retrieval and manipulation, making it ideal for applications with evolving data requirements. Its distributed architecture and automatic sharding enable seamless horizontal scaling, ensuring high availability and performance. MongoDB's robust features, including built-in replication, indexing, and support for geospatial data, empower developers to build modern, data-driven applications that can adapt to the ever-changing demands of the digital landscape.

3.4 User Experience

sec:userexperience The development of Xperis utilized an existing theme and design template from JHipster (JHipster Team, 2013) to streamline the creation process while maintaining a strong focus on user experience (UX). The template provided a consistent and visually appealing layout, which was customized to meet the specific requirements of researchers. Tailoring the color scheme, typography, and branding elements ensured the application appeared professional and approachable, aligning with its purpose of facilitating efficient experiment management and data collection.

To enhance usability, the template's navigation structure was modified to prioritize features critical to HCI researchers, such as experiment creation, participant management, and experiment data visualization. Initial usability testing with a sample of potential users re-

vealed that the default layout included redundant components that could distract from core tasks. Based on this input, the interface was streamlined by removing irrelevant widgets and optimizing the placement of essential tools, ensuring tasks could be completed efficiently.

Accessibility was a key focus during the adaptation of the pre-existing design. While the original theme adhered to basic accessibility standards, it was reviewed against WCAG guidelines, and additional improvements were implemented. These included enhancing color contrast for readability, providing descriptive alt text for graphical elements, and enabling full keyboard navigation to support users with diverse abilities. Performance optimization further refined the application, reducing load times and ensuring smooth functionality even on devices with limited processing power.

By customizing the pre-existing theme with a user-centered approach, the application effectively supports the unique needs of HCI researchers. The clean design, intuitive interface, and accessibility features were well-received during usability evaluations, demonstrating that thoughtful adaptation of pre-designed elements can result in a highly effective and user-friendly tool.

3.5 Features and Functionality

In this section we will present the main features and functionality of the Xperis web application. We begin by providing an overview of the user profiles and their respective roles and responsibilities. Then we discuss the user management functionality, which is available to administrators. Next, we describe the institution management functionality that is available to administrators and researchers. The experiment management functionality, available to researchers, is then explored. Finally, we discuss the template management functionality, also available to researchers. More detailed information on the features and functionality can be found in the User's Guide in Appendix A.

3.5.1 User Profiles

Xperis is designed to accommodate a diverse user base, each with specific roles and responsibilities tailored to their involvement in the research process. The system categorizes users into three primary profiles: administrators, researchers, and participants. These distinct profiles help create a structured and efficient environment for experiment management.

Administrators wield comprehensive control and authority over the system. They play a critical role in the setup of the application, the configuration of permissions and the management of user accounts. Administrators are responsible for ensuring the system's integrity and overseeing its smooth operation. They have the ability to create, modify, and delete experiments, institutions, and user accounts. Additionally, they can define data types, integrate plugins, and access the system's configuration settings. This profile is crucial for the overall governance of Xperis, ensuring its security and functionality. The name of this profile is ADMIN.

The screenshot displays the 'Users' management interface. At the top, there are navigation links for Home, Entities, Administration, Language, and Account. The main content area features a table of users with the following data:

Login	Email	Language	Profiles	Created date	Modified by	Modified date	Actions
user	user@localhost	en	ROLE_MASTER	08/02/25 19:47	system	08/02/25 19:47	View, Edit, Delete
admin	admin@localhost	en	ROLE_MASTER, ROLE_ADMIN	08/02/25 19:47	system	08/02/25 19:47	View, Edit, Delete
marcelo	marcelo.balves@outlook.com	en	ROLE_ADMIN	08/02/25 20:12	admin	08/02/25 20:14	View, Edit, Delete
nuno	nuno.fachada@ulusofonapt.pt	en	ROLE_MASTER	08/02/25 20:14	admin	08/02/25 20:14	View, Edit, Delete
pedrojtortes	pedrojuantortres@gmail.com	en	ROLE_MASTER	08/02/25 20:16	admin	08/02/25 20:16	View, Edit, Delete

Below the table, a pagination bar shows 'Showing 1 - 5 of 5 items' with navigation arrows. A copyright notice at the bottom reads: 'Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License'.

Figure 3.5 – Web Portal’s User Management first page listing all registered users and its functionalities.

Researchers form the core user group of Xperis and are responsible for creating, conducting, and analyzing experiments. This profile caters to researchers engaged in various scientific disciplines, from psychology to social sciences. Researchers can create experiments, define metadata structures, and collaborate with colleagues within the system. They possess the ability to invite collaborators to their experiments and collect data effectively. Although researchers have substantial control over their experiments, they operate within the confines of administrator-defined system settings. This ensures data security and integrity while granting researchers flexibility in the execution of their experiments. Researchers are the primary beneficiaries of the features of Xperis, as they can focus on their research objectives without being encumbered by the administrative aspects of experiment management. The name of this profile is MASTER.

Assistants are individuals who contribute to experiments as data collectors. They have a more limited scope of actions within the system, aligning with their specific role. They are responsible for entering data and contributing to the success of experiments. Assistants have read-only access to the experiment data, ensuring the preservation of data integrity. This profile is invaluable for researchers conducting large-scale experiments that require the involvement of numerous participants. The name of this profile is USER.

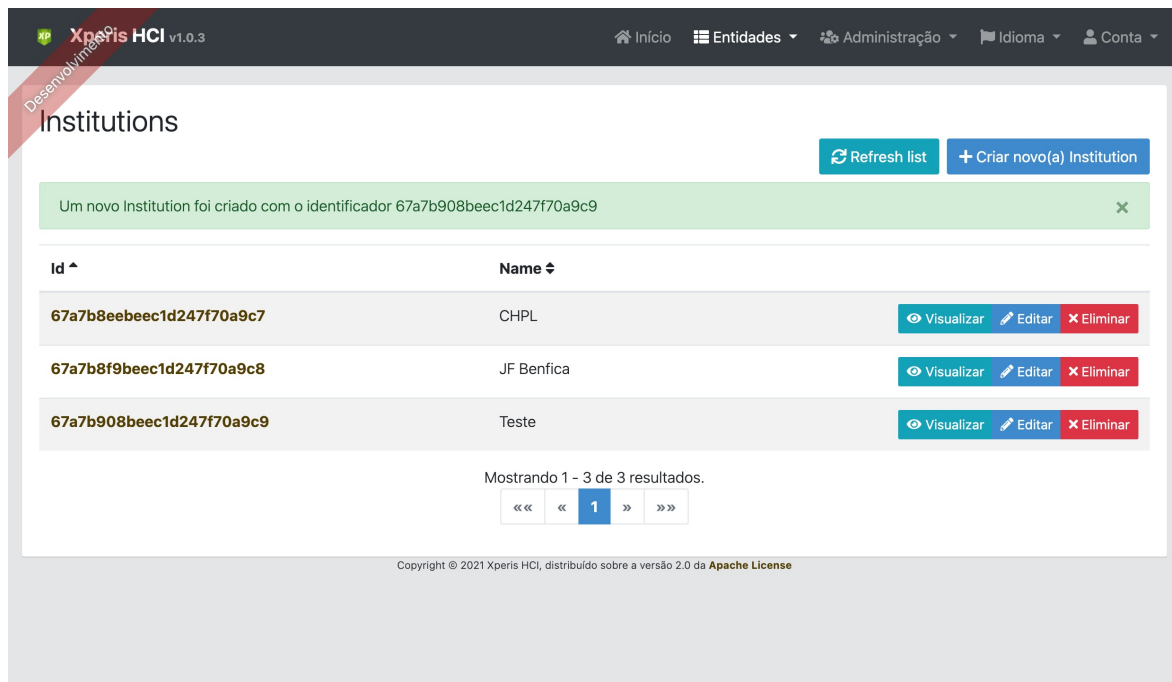


Figure 3.6 – Web Portal’s Institution Management first page listing all registered institutions and its functionalities.

3.5.2 User Management

The administrator profile is responsible for managing user accounts. Administrators can create, modify, and delete user accounts, as well as assign roles to users. The administrator can also view a list of all users and their respective roles. This functionality is available through the User Management option in the Administration menu. Figure 3.5 shows the user management page, where administrators can view a list of all users and their respective roles. Administrators can also create new users, edit existing users, and delete users. The administrator can also assign roles to users, granting them access to specific features and functionality: ADMIN, MASTER or USER.

When a new user is created in the system, an email is sent to the user’s email address with a link to activate the account. The user must click on the link to activate the account and set a password. This process ensures that only authorized users can access the system. The administrator can also reset a user’s password if necessary.

After the user is created and activated, they can log in to the system using the credentials provided during the registration process. The user can then access the system’s features and functionality according to their assigned role. The administrator can also deactivate a user’s account if necessary. This prevents the user from logging in to the system and accessing its features and functionality. The administrator can reactivate the user’s account at any time.

3.5.3 Institution Management

The administrator or researcher are responsible for managing institutions. Institutions are organizations that conduct research experiments. Institutions can be universities, research

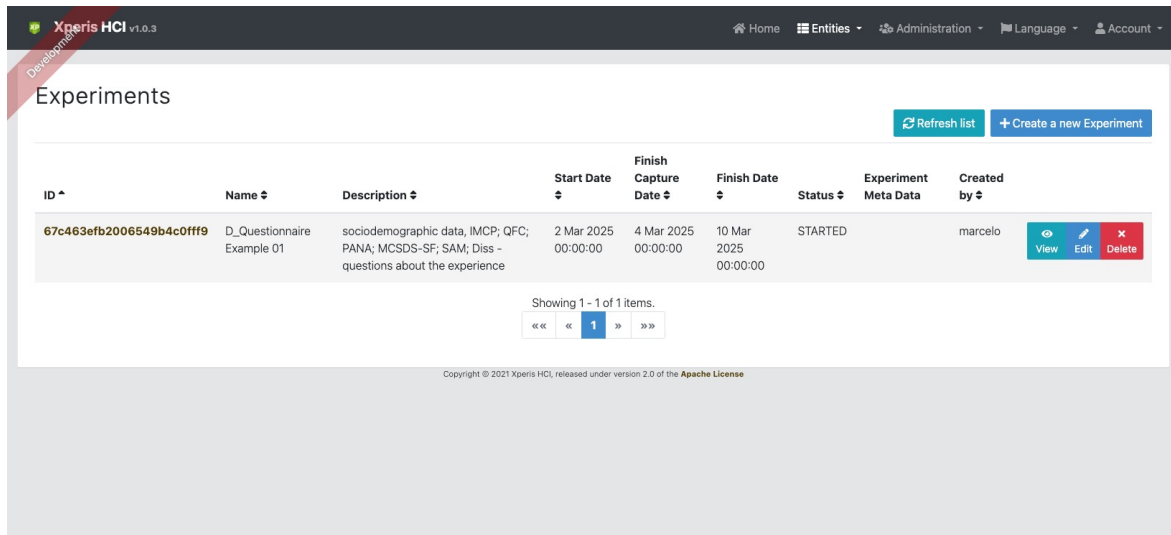


Figure 3.7 – Web Portal’s Experiment Management first page listing all registered experiments and its functionalities.

centers, or private companies. Institutions are associated with users, and each user belongs to one institution. The administrator/researcher can create, edit, and delete institutions, as well as view a list of all institutions. This functionality is available through the Institution option in the Administration menu. Figure 3.6 shows the institution management page, where administrators/researchers can view a list of all institutions. For each institution, administrators/researchers can view the institution’s name and the users associated with it. They can update the institution’s name and add/remove users from it.

3.5.4 Experiment Management

Only researchers can manage experiments. Experiments are the core entities in the system, representing the research projects conducted by researchers. Experiments are associated with researchers, and each researcher can create multiple experiments. The researcher can create, edit, and delete experiments, as well as view a list of all experiments that belong to them. This functionality is available through the Experiment option in the Entity menu. Figure 3.7 shows the experiment management page, where researchers can view a list of all experiments. For each experiment, researchers can view/edit the experiment’s name, description, start date, finish capture date and finish date. The status is calculated automatically.

They can also add/remove members and assistants to/from the experiment. Members are other researchers that also contribute to experiment, and are thus also able to manage it. Assistants are individuals who contribute to experiment as data collectors. They have read-only access to experiment data, ensuring the preservation of data integrity.

After creating an experiment, the researcher can add metadata to it. Metadata describes the data that will be collected during the experiment. The researcher can define the metadata structure and respective data type. This functionality is available through the Metadata option in the Entity menu. Figure 3.8 shows the metadata management page, where researchers

ID	Description	Data Structure	Experiment	Data Mime Type	Experiment Data	Plugins	Created by
67c4685db2006549b4c0fffb	Example 01 Metadata	1.Idade; 2.Gênero; 3.Nacionalidade; 4.Escolaridade; 5.Estatuto Civil; 6.Situação Profissional; 7.01.IMCP - Não posso ficar satisfeito senão fizer as coisas de modo perfeito; 7.02.IMCP - Tenho que ser perfeito a qualquer custo; 7.03.IMCP - Quanto mais elevado o objetivo, melhor; 7.04.IMCP - É uma vergonha cometer um erro; 7.05.IMCP - É para meu benefício que estabeleço padrões elevados; 7.06.IMCP - Sinto-me péssimo se cometer um erro; 7.07.IMCP - Não posso sentir-me satisfeito, a não ser que as coisas sejam feitas na perfeição; 7.08.IMCP - Vou procurar atingir os padrões mais elevados; 7.09.IMCP - Irei culpar-me se cometer um erro; 7.10.IMCP - Quanto mais elevado o objetivo, maior o desafio; 7.11.IMCP - "Fazer algo na perfeição" faz sentido; 7.12.IMCP - Se não conseguir fazer isto bem, significa que estou abaixo da média; 7.13.IMCP - As coisas não deviam ser imperfeitas; 7.14.IMCP - É importante estabelecer padrões elevados para mim próprio; 7.15.IMCP - Sentir-me-ia sem valor se falhasse;	D_Questionnaire Example 01	text/csv	0 item(s)	CSVViewer	marcelo

Figure 3.8 – Web Portal’s Experiment Metadata Management first page listing all registered metadata and its functionalities.

can view a list of all metadata associated with an experiment. For each metadata, researchers can view/edit the metadata’s description, data mime type, and data structure. They also can download all data collected. All updates and downloads are logged in the system and can be viewed in metadata’s view page.

3.5.5 Templates Management

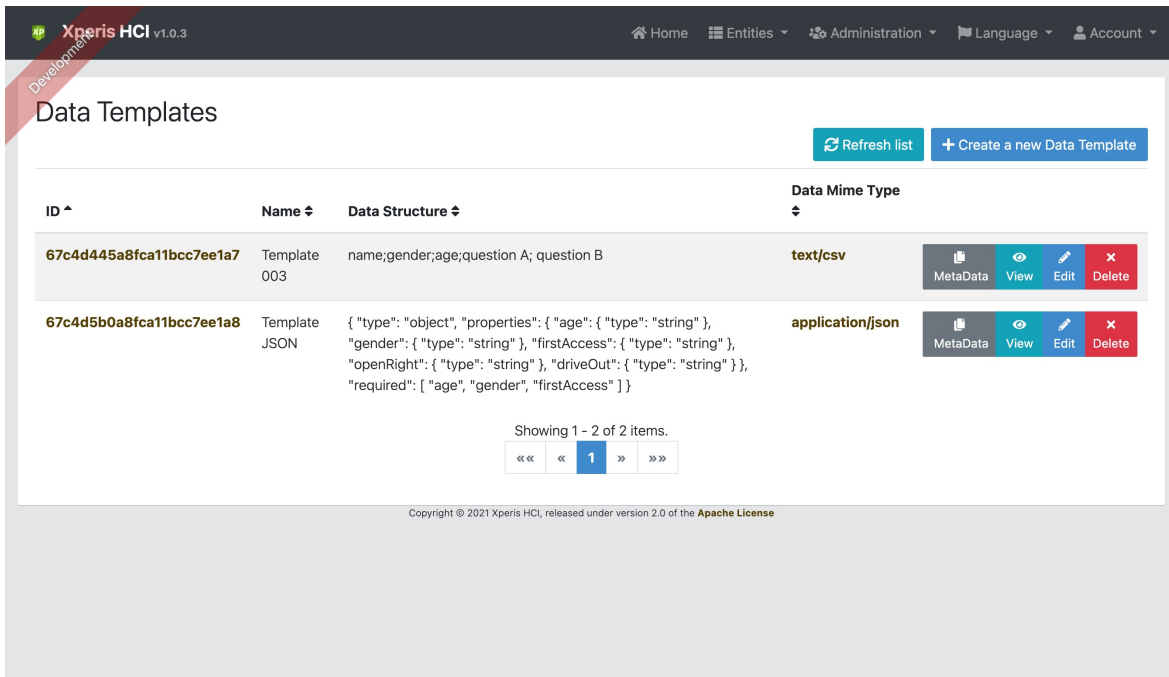
The researcher can also define data templates, which are predefined metadata structures that can be reused in multiple experiments. This functionality is available through the Data Type Templates option in the Entity menu. Figure 3.9 shows the template management page, where researchers can view a list of all templates. For each template, researchers can view/edit the template’s name, data mime type, and data structure.

After creating a data template, researchers can add it to an experiment. This functionality is available through the Metadata button in the Data Templates List showed in Figure 3.9. In that page, researchers can choose an experiment to use the chosen template.

3.5.6 Plugin Management

In the experiment metadata list, shown in Figure 3.8, researchers and assistants can also see the amount of data already collected and access the data viewer. The data viewer is a plugin that allows researchers to visualise the data collected during the experiment. The data visualiser plugin is specific to each mime type.

Only the administrator can add new plugins to the system. This functionality is available through the Plugins option in the Administration menu. Figure 3.10 shows the plugin



Xperis HCI v1.0.3 | Home | Entities | Administration | Language | Account

Data Templates

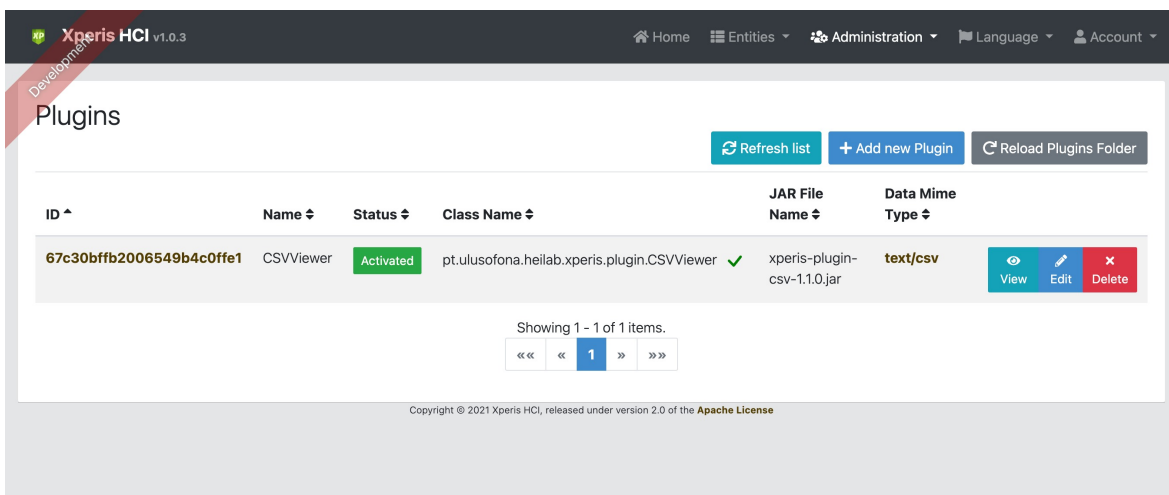
Refresh list | + Create a new Data Template

ID ^	Name ⇅	Data Structure ⇅	Data Mime Type ⇅
67c4d445a8fca11bcc7ee1a7	Template 003	name;gender;age;question A; question B	text/csv
67c4d5b0a8fca11bcc7ee1a8	Template JSON	{ "type": "object", "properties": { "age": { "type": "string" }, "gender": { "type": "string" }, "firstAccess": { "type": "string" }, "openRight": { "type": "string" }, "driveOut": { "type": "string" } }, "required": ["age", "gender", "firstAccess"] }	application/json

Showing 1 - 2 of 2 items.

Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License

Figure 3.9 – Web Portal’s Template Management first page listing all registered templates and its functionalities.



Xperis HCI v1.0.3 | Home | Entities | Administration | Language | Account

Plugins

Refresh list | + Add new Plugin | Reload Plugins Folder

ID ^	Name ⇅	Status ⇅	Class Name ⇅	JAR File Name ⇅	Data Mime Type ⇅
67c30bffb2006549b4c0ffe1	CSVViewer	Activated	pt.ulusofona.heilab.xperis.plugin.CSVViewer ✓	xperis-plugin-csv-1.1.0.jar	text/csv

Showing 1 - 1 of 1 items.

Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License

Figure 3.10 – Web Portal’s Plugin Management first page listing all registered plugins and its functionalities.

ID	Participant(s)	Experiment Meta Data	Data	Created date	Created by	
67c4d798a8fca11bcc7ee1a9	D01	Example 01 Metadata (text/csv)	text/csv, 26 bytes	02/03/25 22:11:36.264	marcelo	View Edit Delete
67c4d7aaa8fca11bcc7ee1aa	D02	Example 01 Metadata (text/csv)	text/csv, 26 bytes	02/03/25 22:11:54.595	marcelo	View Edit Delete
67c4d7aea8fca11bcc7ee1ab	D03	Example 01 Metadata (text/csv)	text/csv, 26 bytes	02/03/25 22:11:58.444	marcelo	View Edit Delete
67c4d7b2a8fca11bcc7ee1ac	D04	Example 01 Metadata (text/csv)	text/csv, 26 bytes	02/03/25 22:12:02.791	marcelo	View Edit Delete
67c4d7b7a8fca11bcc7ee1ad	D05	Example 01 Metadata (text/csv)	text/csv, 26 bytes	02/03/25 22:12:07.342	marcelo	View Edit Delete

Figure 3.11 – Web Portal’s Experiment Data Management first page listing all collected data and its functionalities.

management page, where administrators can view a list of all plugins. For each plugin, administrators can view/edit the plugin’s name, class name, JAR file, and data mime type. They can also add a new plugin to the system by uploading its JAR file. This plugin needs to be developed using the plugin core library implementing the interface `PluginInterface` and the method `executePlugin()`. The plugin core library is available in the Xperis repository.

Administrators can also deactivate a plugin, preventing it from being used in experiments. The administrator can reactivate the plugin at any time. This functionality is also available in the plugin list page showed in Figure 3.10 through the button status. The status can be Deactivated, Activated or Invalid. The status Invalid means that the plugin uploaded is not compatible with the system.

3.5.7 Data Management

Users can access the data management functionality. This functionality is available through the Data option in the Entity menu. Figure 3.11 shows the data management page, where users can view a list of all data associated with an experiment metadata. For each data, users can view/edit the data’s metadata, participants, and data file. They can also download the data file. All updates and downloads are logged in the system and can be viewed in data’s view page.

3.5.8 Application Programming Interface (API)

Xperis is not merely a standalone web application; it serves as a versatile platform that fosters seamless integration with other systems, allowing for efficient data exchange and collaboration. The heart of this interoperability lies in its robust API. The Xperis API al-

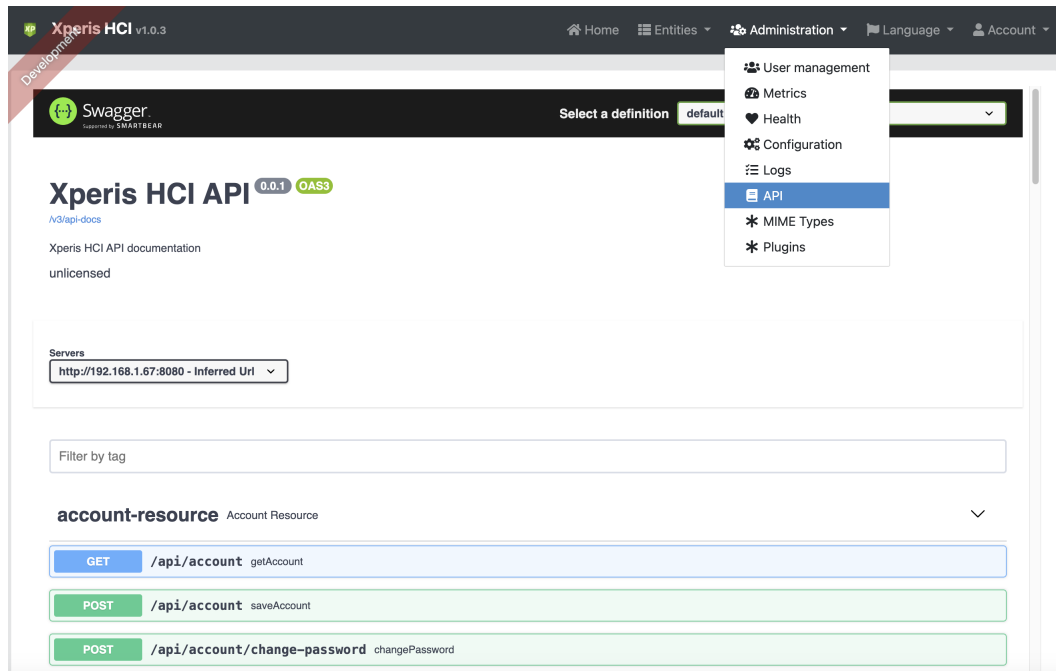


Figure 3.12 – Web Portal’s API Documentation page listing all active endpoints and its parameters.

allows researchers, administrators, and institutions to connect their existing systems and tools, streamlined the process of sending and receiving data.

Through an intuitive RESTful API, Xperis enables the creation, retrieval, modification, and deletion of entities such as users, institutions, experiments, metadata, and data. The API also provides endpoints for uploading and downloading data files. The API is secured using JSON Web Tokens (JWT), which are used to authenticate and authorize users. The API is documented using Swagger (SmartBear Software, 2011), which provides a user-friendly interface for exploring the API’s endpoints and their respective parameters. The API documentation is available at Xperis web site, as shown in Figure 3.12.

The access to the API also respect the system roles. The ADMIN role has full access to the API, while the MASTER and USER roles have limited access. The MASTER role can only access the API endpoints related to experiments, metadata, and data. The USER role can only access the API endpoints related to data upload. The API endpoints related to users and institutions are only accessible to users with the ADMIN role.

The following example illustrates how to use the API. To send the collected data to each participant of an experiment, one can use the `ExperimentData` resource. Here is an example of the call using `curl` (Daniel Stenberg, 1998) command:

```
curl --location --request POST 'http://localhost:8080/api/experiment-data' \
--header 'user-api-key: 61425538d5a03d512659c6fb' \
--header 'Content-Type: application/json' \
--data-raw '{
"participants" : "D04",
```

```
"data" : "MTk7ZmVtYWxl01BvcnR1Z3Vlc2U7NDsxOzE=",  
"dataContentType" : "text/csv",  
"experimentMetaData" : {  
  "id" : "6182046cb28165018a7aeb56"  
}  
'
```

In this example, curl connects to the API using the **user-api-key** header with the experiment owner authentication, sending JSON data containing the participant identifier, the actual experiment data (in Base64 encoding), the data type, and respective metadata ID, as registered in the platform.

Chapter 4

Results

Since it was not possible to conduct tests with real users before the final installation of the platform, only local tests simulating interactions with the Postman (Postman, Inc., 2012) and a remote test with the assistance of a HEI-Lab researcher were performed to validate the integration with an experiment system developed in the Unity game engine.

4.1 Local Test

The local test was executed on a personal computer with a Linux environment and a local MongoDB database. The system was started from the command line:

```
./mvnw
```

making the application available at <http://localhost:8080>.

For configuring the test, data from a questionnaire, entitled *Questionnaire D*, was used to create the experiment in Xperis (see Table 4.1). It contains the metadata of sociodemographics questions like description, type, and expected values applied to the participants.

Abbrev.	Label	Type	Value
Date	Day/Time	String	dd/mm/yyyy
Age	Age	Int	(valid 18 90)
Gender	Gender	String	1: Male; 2: Female; 3: Other
Nat	Nationality	String	1: Portuguese; 2: Other
Educ	Education	Int	1: 1st cycle; 2: 2nd cycle; 3: 3rd cycle; 4: Secondary Education; 5: Bachelor's Degree; 6: Master's Degree; 7: Doctorate
Civ	Marital status	Int	1: Single; 2: Dating/significant relationship; 3: Married or cohabiting; 4: Divorced or separated; 5: Widowed
Prof	Professional status	Int	1: Student; 2: Student-worker; 3: Worker (employed or self-employed); 4: Unemployed; 5: Retired 6: Professional leave

Table 4.1 – Description, Type and Expected Values for the first questions of Sociodemographics Questionnaire D

The figure consists of two screenshots from the Xperis HCI web portal. The top screenshot shows the 'Experiments' page. It features a navigation bar at the top with 'Home', 'Entities', 'Administration', 'Language', and 'Account' menus. Below the navigation bar, there are buttons for 'Refresh list' and '+ Create a new Experiment'. The main content is a table with the following columns: ID, Name, Description, Start Date, Finish Capture Date, Finish Date, Status, Experiment Meta Data, and Created by. A single record is displayed with ID '67ccbad1a8fca11bcc7ee1ae', Name 'D_Questionnaire', Description 'First 6 sociodemographics questions', Start Date '1 Mar 2025 00:00:00', Finish Capture Date '8 Mar 2025 00:00:00', Finish Date '31 Mar 2025 00:00:00', Status 'CAPTURE FINISHED', Experiment Meta Data 'sociodemographic questionnaire (text/csv)', and Created by 'marcelo'. Below the table, it says 'Showing 1 - 1 of 1 items.' and a pagination control shows '1' selected. The bottom screenshot shows the 'Experiment Meta Data' page. It has a similar navigation bar and buttons for 'Refresh list', '+ Create a new Experiment Meta Data', and '+ Create from Template'. The table has columns: ID, Description, Data Structure, Experiment, Data Mime Type, Experiment Data, Plugins, and Created by. A single record is displayed with ID '67ccbbfa8fca11bcc7ee1af', Description 'sociodemographic questionnaire', Data Structure '1.Idade;2.Género;3.Nacionalidade;4.Escolaridade;5.Estado Civil;6.Situação Profissional', Experiment 'D_Questionnaire', Data Mime Type 'text/csv', Experiment Data '5 item(s)', Plugins 'CSVViewer', and Created by 'marcelo'. Below the table, it says 'Showing 1 - 1 of 1 items.' and a pagination control shows '1' selected. Both screenshots have a copyright notice at the bottom: 'Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License'.

Figure 4.1 – Web Portal’s Experiment and Experiment Metadata pages listing the inserted records for Questionnaire D

Thus, the first part of the test involved entering the questionnaire data above into the system. This activity was performed directly on the portal through the management pages mentioned in the previous chapter. In the end, an institution, an experiment, and metadata related to the questionnaire questions were inserted. Figure 4.1 shows the experiments and metadata management screens with the inserted data. In addition, the names of the questions were transcribed as a CSV data structure to be used as a template. This data structure will later be used as the header of the final CSV file containing all the collected data. The data structure was defined in Portuguese as:

1. Idade;
2. Género;
3. Nacionalidade;
4. Escolaridade;
5. Estado Civil;
6. Situação Profissional

In the second part, we enter the responses using the REST API interface that simulates an interaction between the experiment system and Xperis. For this, the Postman software was used. Figure 4.2 shows the request screen.

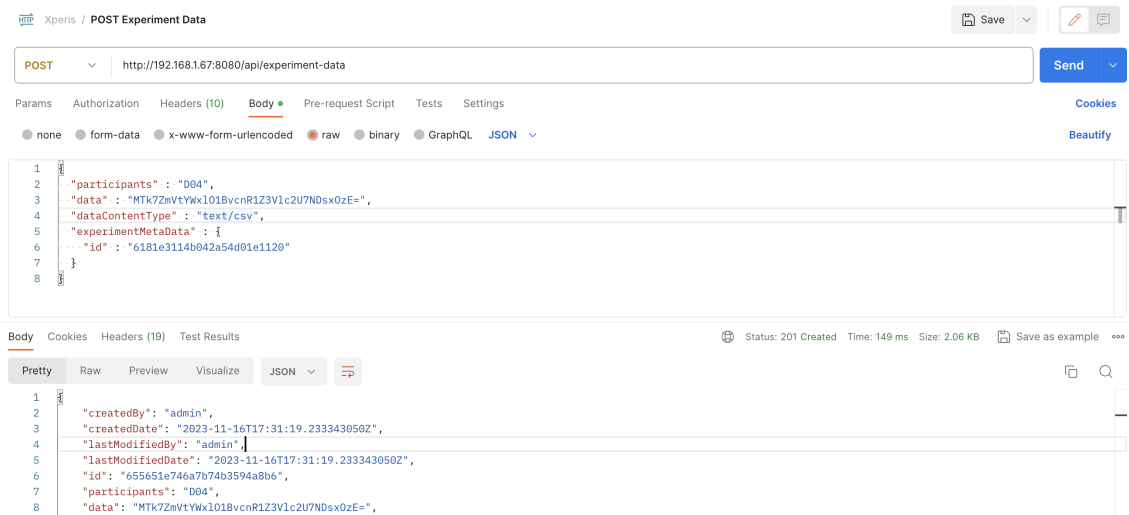


Figure 4.2 – Postman application screen that shows a request to add data to the experiment through POST /metadata REST API endpoint.

The data was sent in the request as a CSV record encoded in Base64. The complete JSON request is shown below:

```

{
  "participants" : "D04",
  "data" : "MTk7ZmVtYWxlO1BvcnR1Z3Vlc2U7NDsxOzE=",
  "dataContentType" : "text/csv",
  "experimentMetadata" : {
    "id" : "67ccbbbf8fca11bcc7ee1af"
  }
}

```

That data value above corresponds to the following answers in CSV format:

```
19;female;Portuguese;4;1;1
```

In this way, after repeating this call for six different participants, it was possible to check that the data had been entered correctly into the system, using the CSV Viewer plugin. After all Postman executions, six new records appear in Xperis Web interface. From the Experiment Metadata Management page, the CSV plugin button was triggered, generating, and downloading a CSV file. The generated CSV file contained the headers and responses of all participants, as can be seen in Figure 4.3.

	A	B	C	D	E	F	G	H	I
1	Date	Participant	1.Idade	2.Género	3.Nacionalidade	4.Escolaridade	5.Estado Civil	6.Situação Profissional	7.
2	2025-03-08 21:53:55	D01	23	female	Portuguese	5	2	3	
3	2025-03-08 21:54:47	D02	18	female	Brazilian	4	1	2	
4	2025-03-08 21:54:51	D03	21	male	Portuguese	4	1	1	
5	2025-03-08 21:54:56	D04	19	female	Portuguese	4	1	1	
6	2025-03-08 21:55:02	D05	20	male	Portuguese	4	1	2	
7									
8									

Figure 4.3 – Google Sheets screen that shows a CSV file generated by Xperis after registering data from 5 participants.

4.2 Remote Test

The initial integrations with the Xperis system began in early 2022, with the collaboration of a HEI-Lab team member. This collaborator developed a Unity-based interface for administering the socio-demographic questionnaire D and implemented the integration required to transmit the collected data to the Xperis system.

To evaluate the integration, a test was conducted simulating participant interaction with the Unity application. The test began with the manual completion of the questionnaire directly within the Unity interface, mimicking the experience of a real participant. Figure 4.4 presents three screenshots of this application:

- The first screen displays the questionnaire title and the initial four questions from Table 4.1.
- The second screen contains behavioral questions from the DS14 scale.
- The third screen shows the final set of questions related to the participant’s experience.

Participants could respond by selecting options or entering text where required. Upon completing the questionnaire, the application automatically transmitted the responses to the Xperis platform using HTTPS requests via the REST API.

Figure 4.5 illustrates the Experiment Data Management page on the Xperis web portal, which lists all records received from the Unity integration. This interface also includes functionality to download individual data files, with the format specified in the associated metadata.

Upon completion of the questionnaire in the Unity application, the data entered by the participant was automatically transmitted to the Xperis platform. This communication occurred via secure HTTPS requests using the Xperis REST API. Each request included the questionnaire responses along with metadata such as the experiment ID and participant code. The API processed the incoming data and confirmed successful receipt with status codes, ensuring data integrity and delivery.

Once received by the system, the data was stored in the Xperis database and made immediately available in the Experiment Data Management section of the Xperis-HCI web portal.

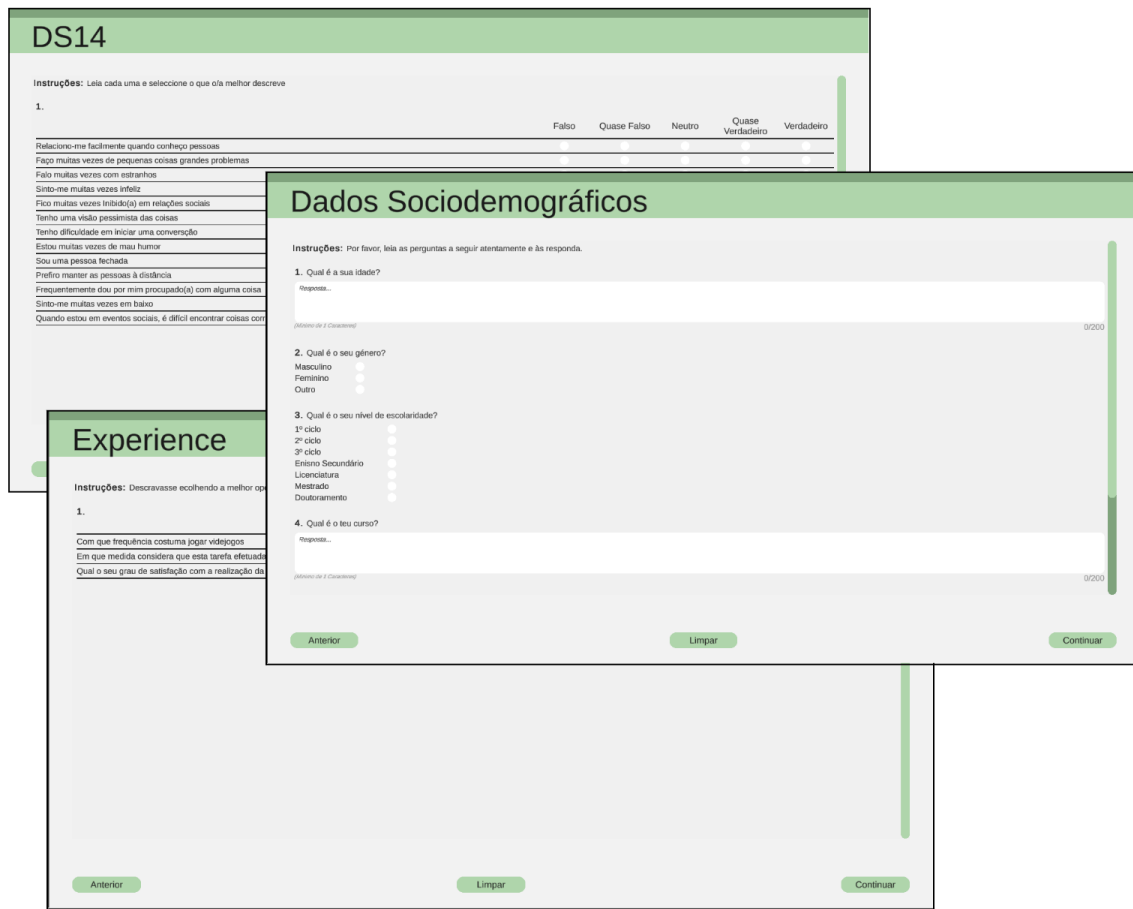


Figure 4.4 – Unity Application screens related to sociodemographics, behavioral and experience questions.

The screenshot shows the 'Experiment Data' management page in the Xperis HCI v1.0.3 web portal. The page includes a navigation bar with 'Home', 'Entities', 'Administration', 'Language', and 'Account' options. Below the navigation bar, there are 'Refresh list' and 'Create a new Experiment Data' buttons. The main content is a table with the following columns: ID, Participant(s), Experiment Meta I, Data, Created date, and Created by. The table lists 11 questionnaire experiments, each with a unique ID, participant type, meta information, data size, creation date, and creator name. At the bottom of the table, there is a pagination control showing 'Showing 41 - 60 of 60 items' and a footer with the Apache License information.

ID	Participant(s)	Experiment Meta I	Data	Created date	Created by
61f7d0692f683f72bda30837	TypeD-1	Questionnaire (application/csv)	application/csv, 49 bytes	31/01/22 12:04:57.437	pedrogtorres
61f7d18e2f683f72bda3083a	TypeD-2	Questionnaire (application/csv)	application/csv, 51 bytes	31/01/22 12:09:50.967	pedrogtorres
61f7d1c32f683f72bda3083b	TypeD-3	Questionnaire (application/csv)	application/csv, 54 bytes	31/01/22 12:10:43.237	pedrogtorres
61f7d1f32f683f72bda3083c	TypeD-4	Questionnaire (application/csv)	application/csv, 61 bytes	31/01/22 12:11:31.575	pedrogtorres
61f7d2292f683f72bda3083d	TypeD-5	Questionnaire (application/csv)	application/csv, 53 bytes	31/01/22 12:12:25.511	pedrogtorres
61f7d2e42f683f72bda3083e	TypeD-6	Questionnaire (application/csv)	application/csv, 53 bytes	31/01/22 12:15:32.193	pedrogtorres
61f7d3112f683f72bda3083f	TypeD-7	Questionnaire (application/csv)	application/csv, 54 bytes	31/01/22 12:16:17.160	pedrogtorres
61f7d3442f683f72bda30840	TypeD-8	Questionnaire (application/csv)	application/csv, 51 bytes	31/01/22 12:17:08.459	pedrogtorres
61f7d3e32f683f72bda30841	TypeD-9	Questionnaire (application/csv)	application/csv, 54 bytes	31/01/22 12:19:47.800	pedrogtorres
61f7d42b2f683f72bda30842	TypeD-10	Questionnaire (application/csv)	application/csv, 52 bytes	31/01/22 12:20:59.713	pedrogtorres
61f7d4642f683f72bda30843	TypeD-11	Questionnaire (application/csv)	application/csv, 53 bytes	31/01/22 12:21:56.103	pedrogtorres

Figure 4.5 – Web Portal’s Experiment Data Management first page listing all received data from Unity integration.

Participant	QuestD_01	QuestD_02	QuestD_03	QuestD_04	QuestD_05	QuestD_06	QuestD_07	QuestD_08	QuestD_09	QuestD_10	QuestD_11	QuestD_12	QuestD_13	QuestD_14	QuestD_15	QuestD_16	QuestD_17	QuestD_18	QuestD_19	QuestD_20	QuestD_21
TypeD-1	27	1	4	Cinema	2	1	0	1	3	4	2	3	1	3	2	0	3	4	3	1	2
TypeD-2	30	0	5	Medicina	3	1	1	1	3	4	1	0	2	3	4	3	4	2	1	1	2
TypeD-3	18	1	0	Veterinaria	1	0	2	3	3	1	2	2	4	3	1	2	3	3	2	1	3
TypeD-4	40	0	6	Telecomunicações	2	1	2	2	3	3	0	0	1	3	0	0	3	4	1	0	2
TypeD-5	21	0	2	Animação	1	1	1	4	1	1	2	3	4	3	4	3	2	0	0	1	2
TypeD-6	33	1	3	Fotografia	0	0	0	4	1	1	2	3	4	3	2	1	0	1	2	3	4
TypeD-7	20	0	0	Veterinaria	1	1	1	3	0	1	0	1	1	1	4	3	3	4	2	2	2
TypeD-8	41	0	4	Medicina	3	1	1	1	2	2	4	0	2	0	4	2	2	0	4	0	2
TypeD-9	29	1	5	Automação	0	0	3	4	0	1	0	0	2	3	4	4	4	3	2	2	2
TypeD-10	30	0	5	Pedagogia	1	1	3	2	0	1	1	2	3	3	4	4	3	4	3	1	1
TypeD-11	24	1	1	Videojogos	1	1	1	3	0	0	2	1	2	3	3	2	2	3	4	3	3

Figure 4.6 – CSV File consolidated with all participants data.

Figure 4.5 shows the interface where each record submitted from the Unity application is listed, including a timestamp and metadata fields. This interface also offers the option to download each individual participant’s data file, in CSV format as defined in the experiment metadata.

To facilitate bulk analysis, the collected data was later aggregated using the CSV Viewer plugin available on the Metadata Management page. This plugin processed the stored entries and generated a single CSV file that included all questionnaire responses. Each question appeared as a separate column header, and each row corresponded to one participant’s set of answers. The plugin also used the metadata to determine the correct structure and format of the output file.

Figure 4.6 presents a visualization of the generated CSV file. The successful use of the plugin not only confirmed the proper formatting of the exported data but also demonstrated the viability of using this approach in future integrations. This functionality is particularly valuable in multi-system environments, as it allows external systems or researchers to easily access, aggregate, and analyze data collected through Xperis using a standardized format.

Chapter 5

Discussion and Future Work

Despite the promising results achieved during the development and validation of Xperis, several limitations hindered the full realization of its potential. These challenges arose due to the constraints faced during the project lifecycle and have implications for the interpretation and utility of the results.

One of the primary limitations was the lack of real-world testing with actual participants. Although local tests using Postman and a limited remote test with a HEI-Lab collaborator validated certain functionalities, the system's performance under real-world conditions remains unexplored. Real-world testing could provide deeper insight into user experience, system robustness, and integration challenges.

Another significant challenge was the limited interaction with other researchers throughout the project. Collaboration with experts in human-computer interaction (HCI) and related fields could have provided valuable feedback, enabling improvements in design, usability, and system capabilities. This limited interaction restricted the project's scope and the opportunity to incorporate diverse perspectives and expertise.

Finally, the inability to deploy the application for large-scale use prevented the system from being tested in a broader, more diverse environment. A large-scale deployment could have offered critical data on scalability, system stability, and its ability to handle heterogeneous user inputs and data formats. This limitation underscores the need for a more comprehensive validation phase to confirm the utility and reliability of the system in real world applications.

To address the aforementioned limitations and enhance the system's applicability and robustness, several directions for future work are proposed:

- **Real-World Testing:** Conduct comprehensive tests with actual participants in various settings. This would involve integrating Xperis into live HCI experiments and collecting feedback to refine its functionality and user interface.
- **Collaboration with Researchers:** Establish partnerships with experts in HCI and related fields to collect information and improve the design and features of the system. Regular interactions with researchers will also help identify emerging needs and ensure that that the system remains relevant.
- **Large-Scale Deployment:** Deploy the system in large-scale, multiinstitutional studies

to validate its scalability and performance under high-demand conditions. This will also provide an opportunity to evaluate its integration with diverse tools and platforms used in HCI research.

- **Enhancing Data Validation:** Expand the dataset used for testing to include varied scenarios and edge cases. This will help validate all functionalities and ensure that the system can handle unexpected inputs or conditions robustly.
- **Extending Functionality:** Explore additional features to enhance the usability of the system, such as advanced data analytics, visualization tools, and support for more data formats. These extensions would broaden its appeal and utility in the research community.
- **Security and Privacy Enhancements:** With increasing data privacy concerns, future iterations should incorporate stronger security measures and compliance with data protection regulations to safeguard participant information.

By addressing these areas, Xperis can evolve into a more robust, user-friendly, and widely applicable system, furthering its contribution to the field of human-computer interaction research.

Chapter 6

Conclusion

This dissertation addressed the challenges in developing a MHCI system for researchers to handle diverse media types captured during HCI experiments. Using Web-based technologies and drawing inspiration from existing solutions in the literature, the developed system, Xperis, successfully demonstrated its capabilities through comprehensive testing scenarios.

Local tests, utilizing simulations of interactions through Postman and a local environment, validated the robustness and efficacy of the system. From configuring experiments to entering participant data, the system showcased its flexibility and seamless integration with external tools. The incorporation of a plugin-driven approach, exemplified by the CSV Viewer plugin, illustrated the adaptability of the system to diverse data export requirements, contributing to its versatility and potential integration with other platforms.

Furthermore, the remote test, which involved collaboration with a HEI-Lab collaborator and integration with Unity, marked a significant milestone. The successful transmission of participant data from the Unity interface to the system via a secure REST API highlighted the system's potential for use in real-world scenarios. The CSV Viewer plugin's role in generating a comprehensive CSV file, showcasing headers and participant responses, emphasized its practical utility in aggregating and exporting experiment data.

In essence, the developed MHCI system not only met its objectives, but also demonstrated its potential to contribute to the evolving landscape of human-computer interaction research. With its modular architecture, web-based interface, and seamless integrations, the system is a valuable tool for researchers looking for efficient and flexible solutions to manage and analyze multimodal data collected during experiments. As technology advances, this system lays the foundation for further innovations in the realm of MHCI, fostering a human-centered approach to computing and interaction.

Bibliography

Apache (2009). Apache ActiveMQ.

Becker, A. S. and Berkemeyer, A. (2002). Rapid application design and testing for usability. *IEEE Multimedia*, pages 38–46.

Chodorow, K. (2013). *MongoDB: The Definitive Guide*. O’Reilly Media.

Cloyd, M. H. (2001). Designing user-centered web applications in web time. *IEEE Software*, pages 62–69.

Crockford, D. (2006). *JavaScript: The Good Parts*. O’Reilly Media.

Daniel Stenberg (1998). curl: Command line tool and library for transferring data with urls. First released in 1998.

Fain, Y. and Moiseev, A. (2018). Angular development with typescript.

Fdez-Riverola, F., Glez-Peña, D., López-Fernández, H., Reboiro-Jato, M., and Méndez, J. (2012). A java application framework for scientific software development. *Software: Practice and Experience*, 42(8):1015–1036.

Fowler, M. (2002). Patterns of enterprise application architecture. *Addison-Wesley Professional*.

Ivory, M. Y. and Hearst, M. A. (2002). Improving web site design. *IEEE Internet Computing*, pages 56–63.

Jaimes, A. and Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Comput. Vision Image Understanding*, 108(1-2):116–134.

JHipster Team (2013). Jhipster: A development platform to generate, develop and deploy spring boot + angular/react/vue web applications and spring microservices.

Machines, C. (2007). Pscylone.

Morrell, W. C., Birkel, G. W., Forrer, M., Lopez, T., Backman, T. W. H., Dussault, M., Petzold, C. J., Baidoo, E. E. K., Costello, Z., Ando, D., Alonso-Gutierrez, J., George, K. W., Mukhopadhyay, A., Vaino, I., Keasling, J. D., Adams, P. D., Hillson, N. J., and Garcia Martin, H. (2017). The experiment data depot: A web-based software tool for

- biological experimental data storage, sharing, and visualization. *ACS Synthetic Biology*, 6(12):2248–2259. PMID: 28826210.
- Murugesan, S. and Venkatakrishnan, B. A. (2005). Addressing the challenges of web applications on mobile handheld devices. In *Proceedings of International Conference on Mobile Business*, pages 199–205.
- Oracle Corporation (2012). *Java SE Specifications*. Oracle Corporation.
- Pantic, M., Nijholt, A., Pentland, A., and Huang, T. (2008). Human-centered intelligent human-computer interaction (HCI²): How far are we from attaining it. *Int. J. Autonomous Adaptive Commun. Syst.*, 1(2):168–187.
- Pantic, M., Pentland, A., Nijholt, A., and Huang, T. S. (2007). Human computing and machine understanding of human behavior: A survey. In Huang, T. S., Nijholt, A., Pantic, M., and Pentland, A., editors, *Artificial Intelligence for Human Computing (Lecture Notes in Artificial Intelligence)*, volume 4451, pages 47–71. Springer, Berlin, Germany.
- Pantic, M. and Rothkrantz, L. J. M. (2003). Toward an affect-sensitive multimodal human-computer interaction. *Proc. IEEE*, 91(9):1370–1390.
- Postman, Inc. (2012). Postman: Api development environment. First released in 2012.
- Redmond, E. and Wilson, J. R. (2012). *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement*. Pragmatic Bookshelf, Raleigh, North Carolina, USA.
- Schipor, O.-A. and Vatavu, R.-D. (2023). Gearwheels: A software tool to support user experiments on gesture input with wearable devices. *International Journal of Human-Computer Interaction*, 39(18):3527–3545.
- Shen, J. and Pantic, M. (2013). HCI² Framework: A Software Framework for Multimodal Human-Computer Interaction Systems. *IEEE Transactions on Cybernetics*, 43(6):1593–1606.
- Siegel, D. A. (2003). The business case for user-centred design: Increasing your power. *Software*, 18(1):62–69.
- SmartBear Software (2011). Swagger: Open source api tool. First released in 2010.
- Soleymani, M., Lichtenauer, J., Pun, T., and Pantic, M. (2012). A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing*, 3(1):42–55.
- Unity Technologies (2005). Unity game engine. First released in 2005.
- Walls, C. (2016). Spring boot in action.
- Williams, J. (2000). Correctly assessing the “ilities” requires more than marketing hype. *IT Professional*, 2(6):65–67.

Appendix A

User's Guide

Xperis User's Guide

Xperis is the control system for human interaction experiments of the Hei-lab laboratory of the Lusófona University developed by the student Marcelo Alves during the defense of his Master's thesis in Computer Engineering in the year 2023.

Table of Contents

- [Profiles](#)
- [User Registration](#)
 - [Create a new User](#)
 - [New User Validation](#)
 - [Create Master User](#)
- [Institution](#)
 - [Create a new Institution](#)
 - [Edit Institution](#)
 - [Delete Institution](#)
 - [View Institution](#)
- [Experiment](#)
 - [Create Experiment](#)
 - [Edit Experiment](#)
 - [Delete Experiment](#)
 - [View Experiment](#)
- [Experiment Metadata](#)
 - [Create Metadata](#)
 - [Create Metadata with Template](#)
 - [Edit Metadata](#)
 - [Delete Metadata](#)
 - [View Metadata](#)
 - [Download Metadata](#)
 - [Use Plugin](#)
- [Data Type Template](#)
 - [Create Data Type Template](#)
 - [Edit Data Type Template](#)
 - [Delete Data Type Template](#)
 - [View Data Type Template](#)
 - [Create Experiment Metadata from Data Type Template](#)
- [Experiment Data](#)

- [Create Experiment Data](#)
- [Edit Experiment Data](#)
- [Delete Experiment Data](#)
- [View Experiment Data](#)
- [Download Experiment Data](#)
- [Plugin](#)
 - [Add Plugin](#)
 - [Edit Plugin](#)
 - [Deactivate Plugin](#)
 - [Activate Plugin](#)
 - [Delete Plugin](#)
 - [View Plugin](#)
 - [Download Plugin](#)

Profiles

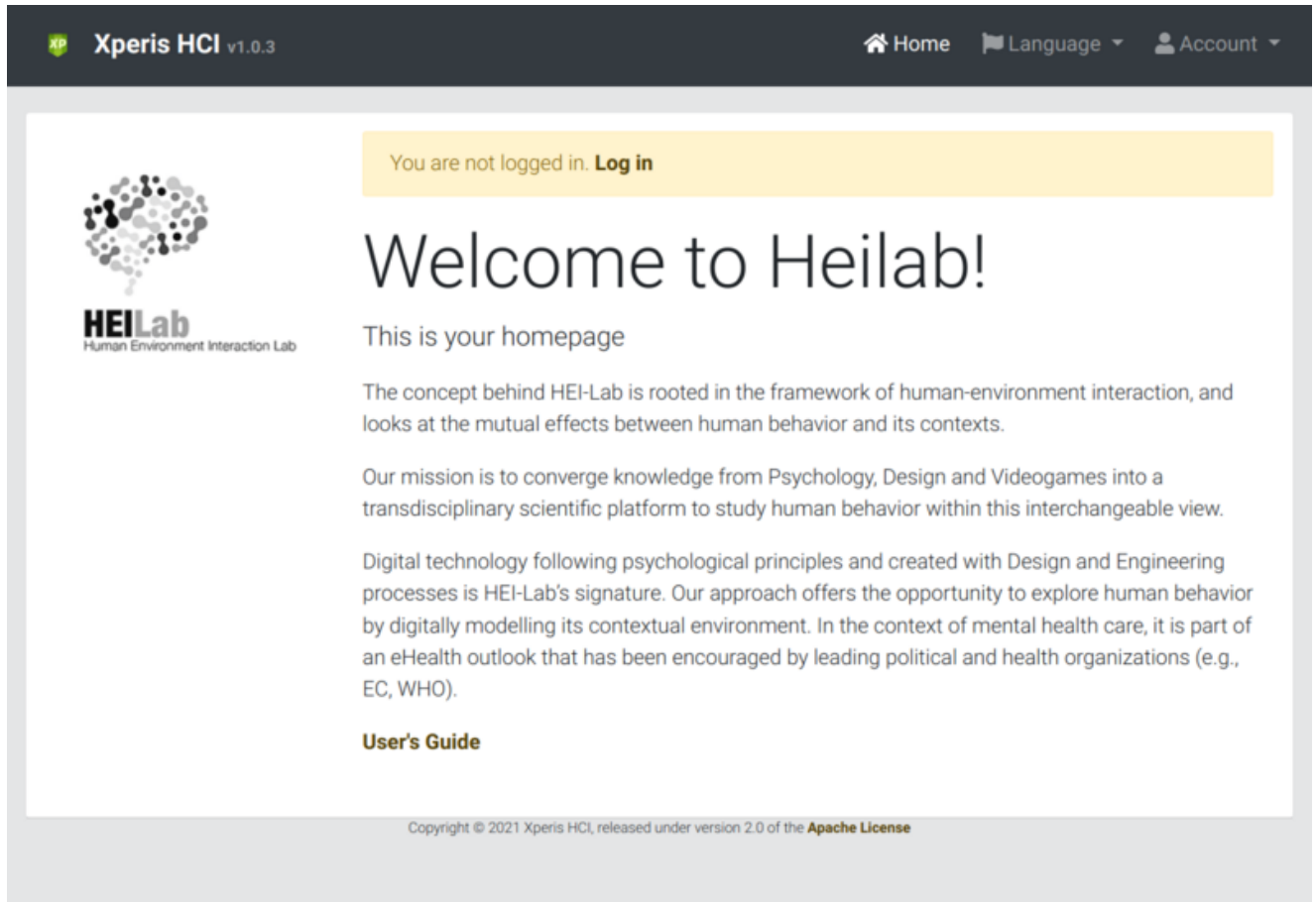
The system has 3 access profiles:

- **ADMIN:** user with access to all features and responsible for administering all other users
- **MASTER:** user with permission to create experiments, initiate data collection and authorize other participants in the system. He can also be a participant in another MASTER user's experiment, and in this case he does not have access to modify the experiment, he only has read access
- **USER:** user with permission only to participate in experiments as a data collector. This user has read-only access to the experiment data and is generally responsible for entering the data into the system.

User Registration

Create a new User

1. Navigate to home



Xperis HCI v1.0.3

Home Language Account

You are not logged in. [Log in](#)

Welcome to Heilab!

This is your homepage

The concept behind HEI-Lab is rooted in the framework of human-environment interaction, and looks at the mutual effects between human behavior and its contexts.

Our mission is to converge knowledge from Psychology, Design and Videogames into a transdisciplinary scientific platform to study human behavior within this interchangeable view.

Digital technology following psychological principles and created with Design and Engineering processes is HEI-Lab's signature. Our approach offers the opportunity to explore human behavior by digitally modelling its contextual environment. In the context of mental health care, it is part of an eHealth outlook that has been encouraged by leading political and health organizations (e.g., EC, WHO).

[User's Guide](#)

Copyright © 2021 Xperis HCI, released under version 2.0 of the [Apache License](#)

- \
2. Choose the option Register in the Account menu on the top right, or choose the "Create a new account" option in the Authentication

3. The registration screen will open

Registration

Username
Your username
Your username is required.

Email
Your email

New password
New password
Password strength: — — — — —

New password confirmation
Confirm the new password

Register

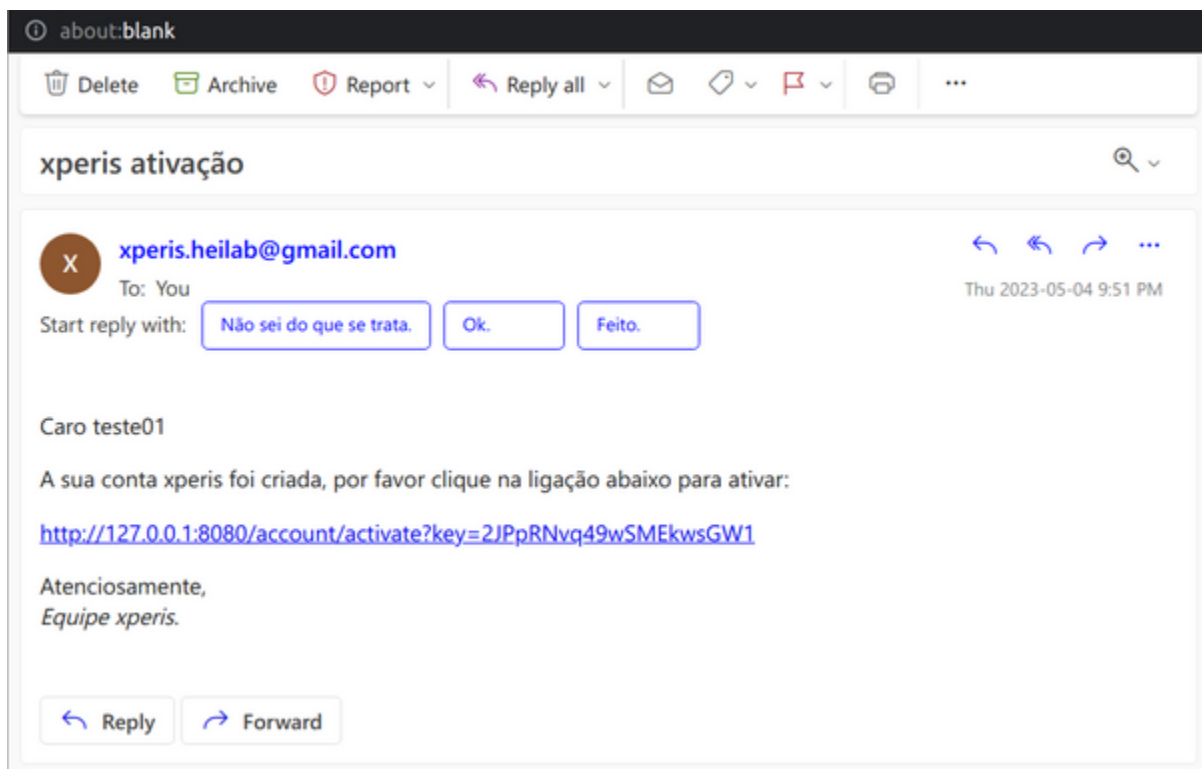
You are not logged in. **Log in**

Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License

4. Fill username, email address and a password.

5. Pressing register it will trigger a verification email to the entered address.

6. The email will look like this



7. After clicking the link in the email, the account will be validated.

New User Validation

However, that user creation still needs the ADMIN's approval.

1. Login as administrator
2. Go to the Administration menu and choose "User Management" option
3. Find the new user and press the "Inactive" button in front of the user's email address to activate it.

ID	Login	Email	Language	Profiles	Created date	Modified by	Modified date		
user-1	admin	admin@localhost	Activated	en	ROLE_MASTER ROLE_ADMIN	25/04/21 00:57	admin	08/08/21 19:35	View Edit Delete
60908c7f83c8ca54d6457906	marcelo	marcelo.balves@outlook.com	Activated	en	ROLE_MASTER	04/05/21 00:51	admin	28/06/21 12:58	View Edit Delete
60942a22304a986d48503f3d	nuno	nuno.fachada@ulusofona.pt	Activated	en	ROLE_MASTER	06/05/21 18:40	admin	28/06/21 12:59	View Edit Delete
60db36489d163040e930f02b	tryout	fabio.dias@ulusofona.pt	Activated	en	ROLE_MASTER	29/06/21 16:03	tryout	30/06/21 14:44	View Edit Delete
61687327761f5026b58d051a	pedrojtorres	pedrojuantorres.info@gmail.com	Activated	en	ROLE_MASTER	14/10/21 19:12	anonymousUser	20/04/23 14:06	View Edit Delete
64270f20b2d196291f5c599f	ajpinha	ana.pinha@ulusofona.pt	Activated	en	ROLE_MASTER ROLE_ADMIN	31/03/23 17:49	admin	31/03/23 23:06	View Edit Delete
64540c854d99b424284c8040	teste01	marceloborgesmp@gmail.com	Activated	pt-pt	ROLE_USER	04/05/23 20:50	admin	04/05/23 23:40	View Edit Delete

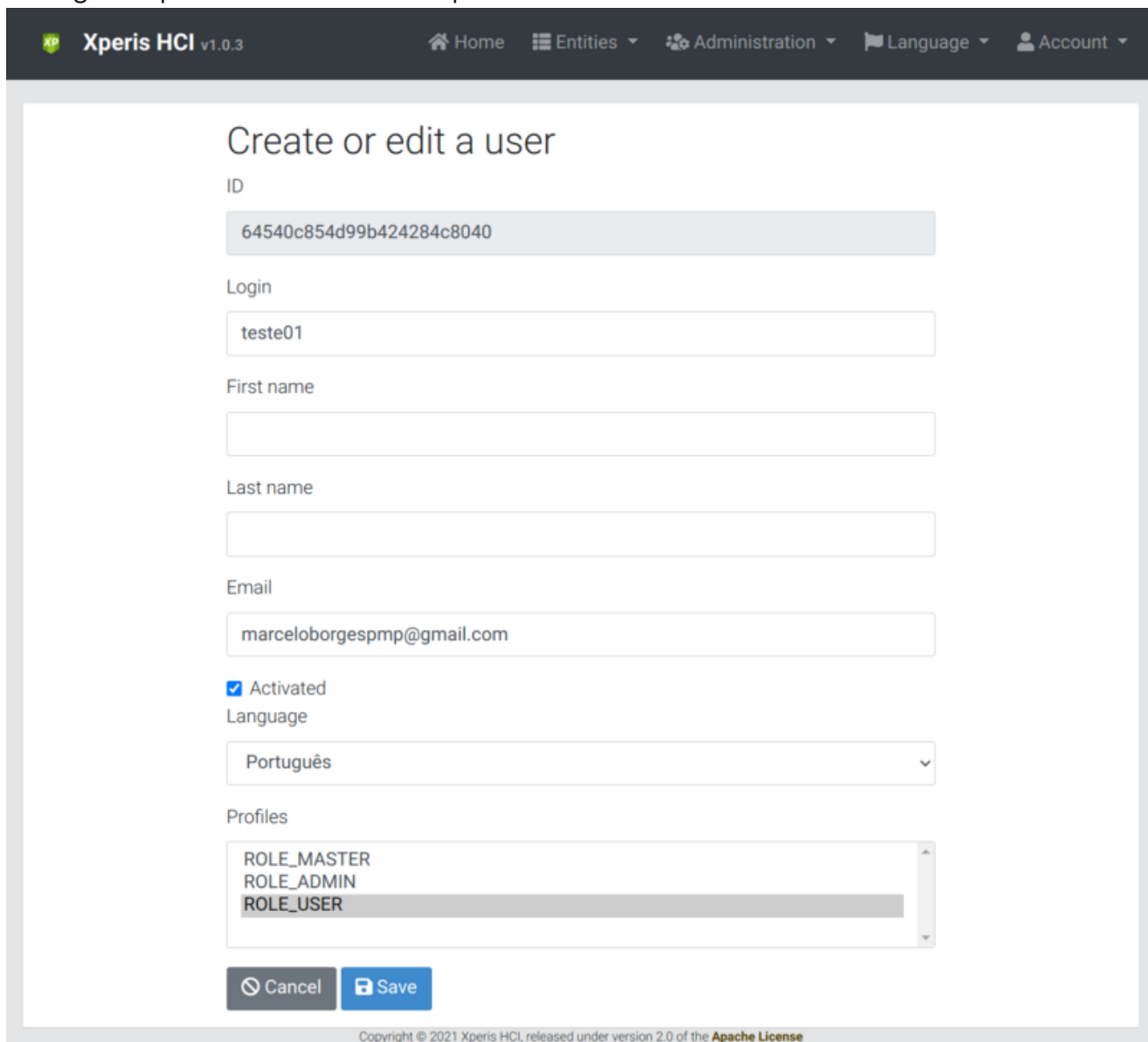
4. Only after Activated, the user in question will be able to log in to the application. Every newly created user gets the USER profile.

Create Master User

To add new MASTER users, the process should be done the same way. After the ADMIN approves it, still need to change the profile manually:

1. in the "User Management" choose the "Edit" button.

2. Change the profile to MASTER and press "Save"



The screenshot shows the Xperis HCI v1.0.3 interface. The top navigation bar includes Home, Entities, Administration, Language, and Account. The main content area is titled "Create or edit a user" and contains the following fields and options:

- ID:** 64540c854d99b424284c8040
- Login:** teste01
- First name:** (empty field)
- Last name:** (empty field)
- Email:** marceloborgesmp@gmail.com
- Activated:**
- Language:** Português
- Profiles:** ROLE_MASTER, ROLE_ADMIN, ROLE_USER (selected)

At the bottom, there are "Cancel" and "Save" buttons. A copyright notice at the very bottom reads: "Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License".

3. On this screen you can also change the user's name and nickname.

Institution

Create a new Institution

1. Login as MASTER user
2. Go to the Entities menu and choose "Institution" option
3. Choose the "Create Institution" option

4. Will open the screen "Create or edit Institution"

The screenshot shows the 'Create or edit a Institution' form. The 'Name' field is filled with 'Teste'. The 'Employees' list contains the following names: marcelo, nuno, tryout, pedrojtorges, ajpinha, teste01. The 'marcelo' entry is selected. At the bottom, there are 'Cancel' and 'Save' buttons. The footer indicates 'Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License'.

5. Fill the name and choose the employees of the institution in the list of already registered users.
6. After you have entered the data and pressed the "Save" button, the system returns to the list of Institutions and displays the basic data as well as the options to "View", "Edit" and "Delete".

The screenshot shows the 'Institutions' list. A green notification bar at the top says 'A new Institution is created with identifier 6474c82523ce5d6a244f1d77'. Below is a table with 3 columns: Id, Name, and actions (View, Edit, Delete). The table shows three institutions: CHPL, JF Benfica, and Teste. A pagination bar at the bottom shows 'Showing 1 - 3 of 3 items'.

Id	Name	Actions
60d9ba055d49c243e4eef349	CHPL	View Edit Delete
60d9ba105d49c243e4eef34a	JF Benfica	View Edit Delete
6474c82523ce5d6a244f1d77	Teste	View Edit Delete

Showing 1 - 3 of 3 items.

Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License

Edit Institution

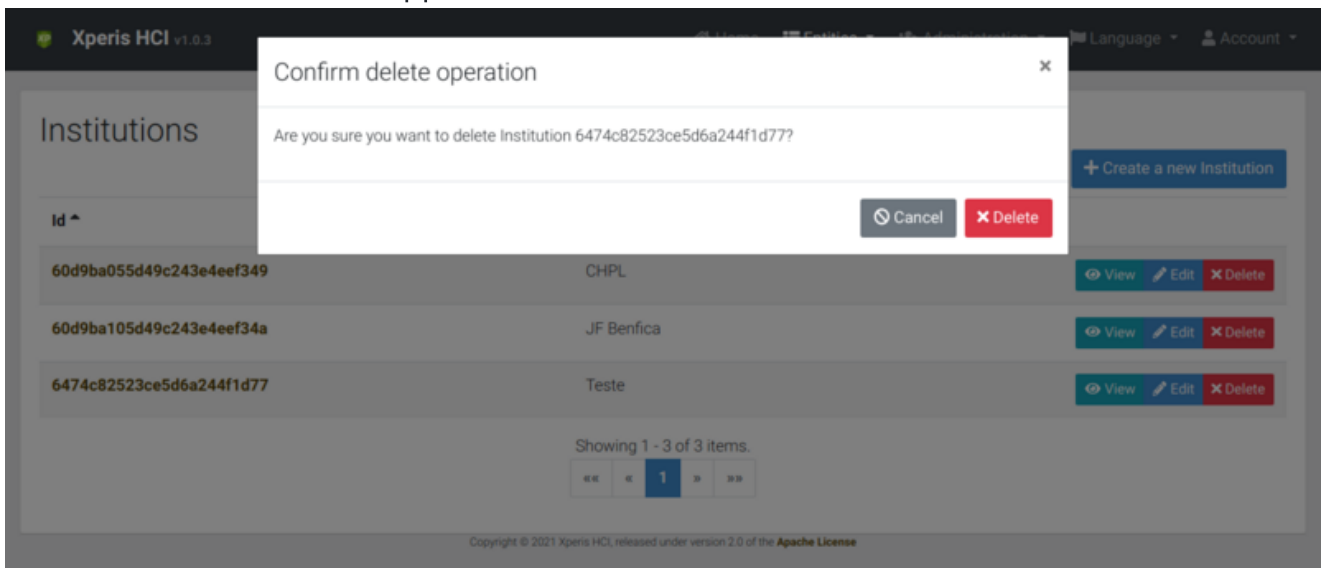
To change any of the data entered when creating the Institution:

1. Choose the "Edit" option directly from the "Institution" list
2. The screen "Create or edit Institution" will open again, with the data already entered.
3. To remove a user from the Employees list, just click with the mouse and hold the CTRL key.
4. By choosing "Save" the changes will take effect
5. It is possible to cancel the process just by pressing the "Cancel" button

Delete Institution

To delete an Institution

1. Choose the "Delete" option directly from the "Institution" list
2. A confirmation screen will appear.

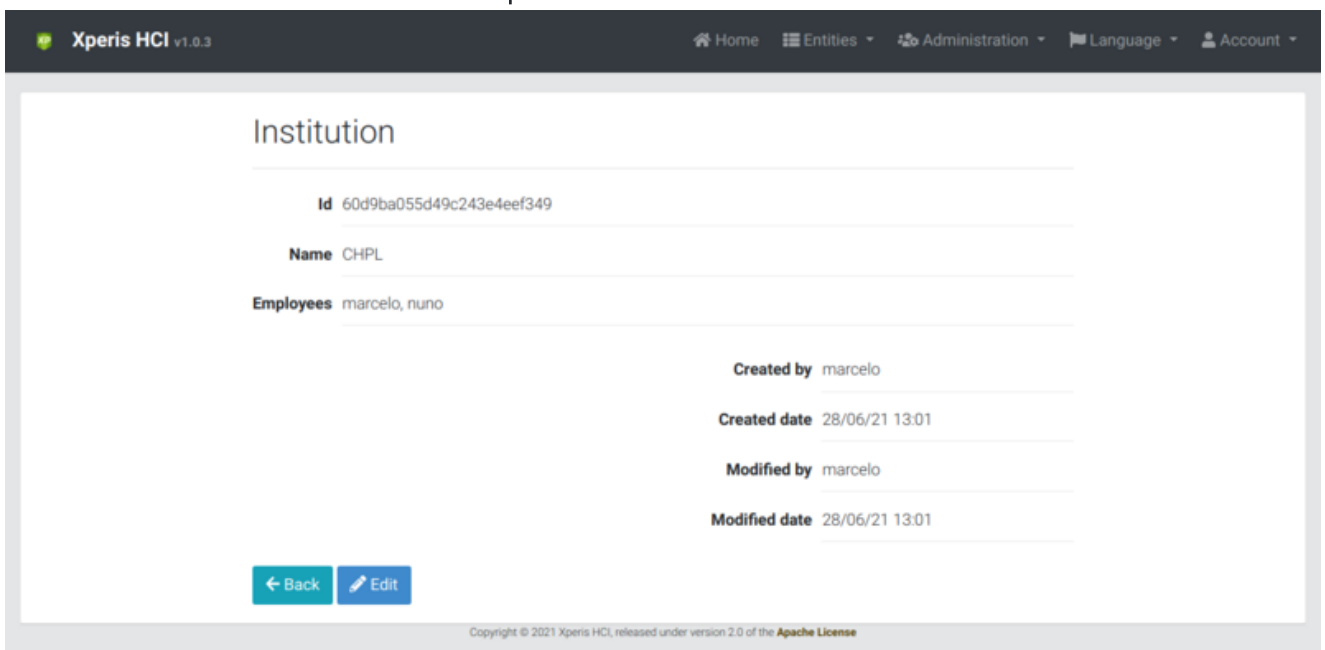


3. If you choose "Delete" the Institution will be permanently deleted. If you choose "Cancel" the Institution will not be deleted.

View Institution

To view the details of an Institution

1. Choose the "View" option directly from the "Institution" list
2. The screen "View Institution" will open



3. In addition to the basic data, , this screen will also show the date/time stamp of the creation and of the last change.

4. It is possible to return to the list of Institutions by pressing the "Back" button

Experiment

Create Experiment

To create a new experiment:

1. Login as MASTER user
2. Go to the Entities menu and choose "Experiment" option
3. Choose the "Create Experiment" option (For a user with the profile USER this option will not be visible)
4. Will open the screen "Create or edit Experiment"
5. Fill the name, the description, the start date of the experiment, as well as a forecast of the start and end of the data collection.

The screenshot shows the 'Create or edit a Experiment' form in the Xperis HCI v1.0.3 interface. The form is titled 'Create or edit a Experiment' and contains the following fields:

- Name:** Test New 01
- Description:** Test for User Guide
- Start Date:** 05/27/2023, 12:00 AM
- Finish Capture Date:** 05/27/2023, 12:00 AM
- Finish Date:** 05/27/2023, 12:00 AM
- Members:** A list of users including pedrojtortes, tryout, nuno, and ajpinha.
- Assistants:** A list of users including tryout, pedrojtortes, ajpinha, and teste01.

At the bottom of the form, there are two buttons: 'Cancel' and 'Save'.

Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License

6. On this screen you can choose other MASTER users to have access to the experiment, called members. And you can also choose the assistants that will have access to collect data (those with the USER profile)
7. After you have entered the data and pressed the "Save" button, the system returns to the list of Experiments and displays the basic data as well as the options to "View", "Edit" and "Delete".

ID	Name	Description	Start Date	Finish Capture Date	Finish Date	Status	Experiment Meta Data	Created by
6182044eb28165018a7aeb55	D_Questionaire	Prmeiro Exemplo Experiencia 2	3 Nov 2021 00:00:00	30 Nov 2021 00:00:00	4 Dec 2021 00:00:00	FINISHED	D_Sociodemographics (text/csv) Teste Envio 7z (application/x-zip-compressed) Teste Envio Ficheiro LSL (application/zip)	marcelo
6472738c23ce5d6a244f1d73	Test New 01	Test for User Guide	27 May 2023 00:00:00	27 May 2023 00:00:00	27 May 2023 00:00:00	FINISHED		marcelo

Showing 1 - 2 of 2 items.

The status of the experiment is automatically obtained according to the dates entered. For those experiments that already have metadata created will be displayed also a direct link to each metadata.

Edit Experiment

To change any of the data entered when creating the experiment:

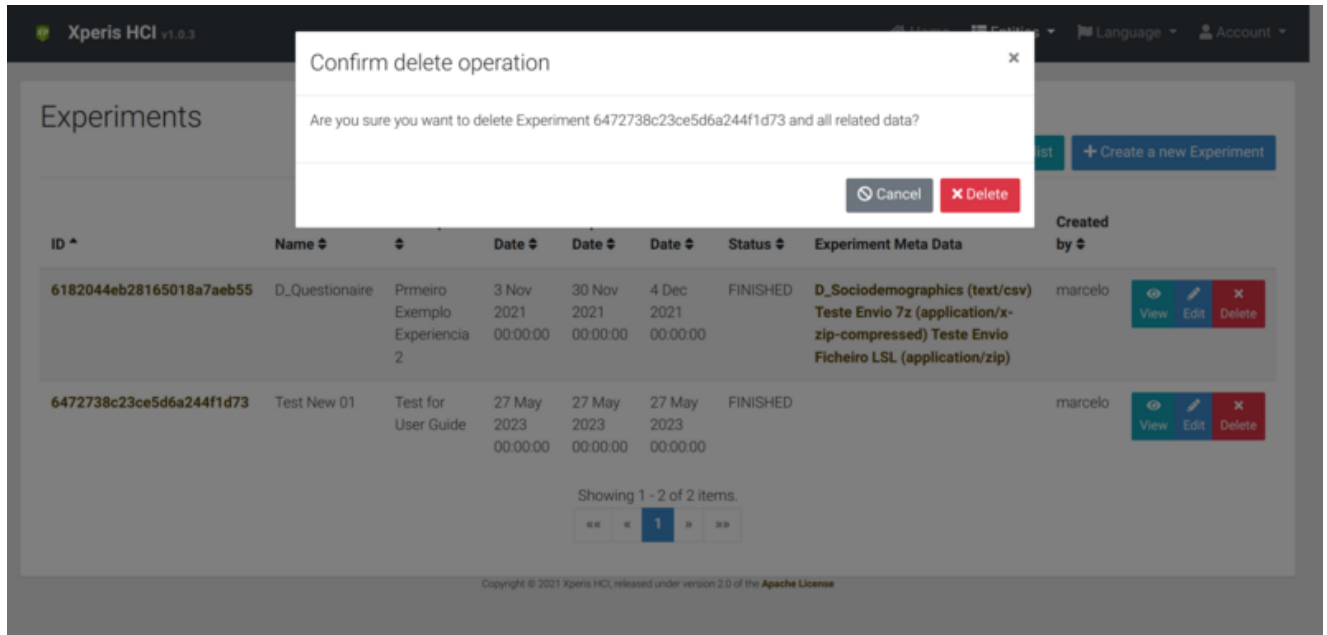
1. Choose the "Edit" option directly from the "Experiments" list
2. The screen "Create or edit Experiment" will open again, with the data already entered.
3. To remove a user from the "members" or "assistants" list, just click with the mouse and hold the CTRL key.
4. By choosing "Save" the changes will take effect
5. It is possible to cancel the process just by pressing the "Cancel" button

Delete Experiment

To delete an experiment

1. Choose the "Delete" option directly from the "Experiments" list

2. A confirmation screen will appear



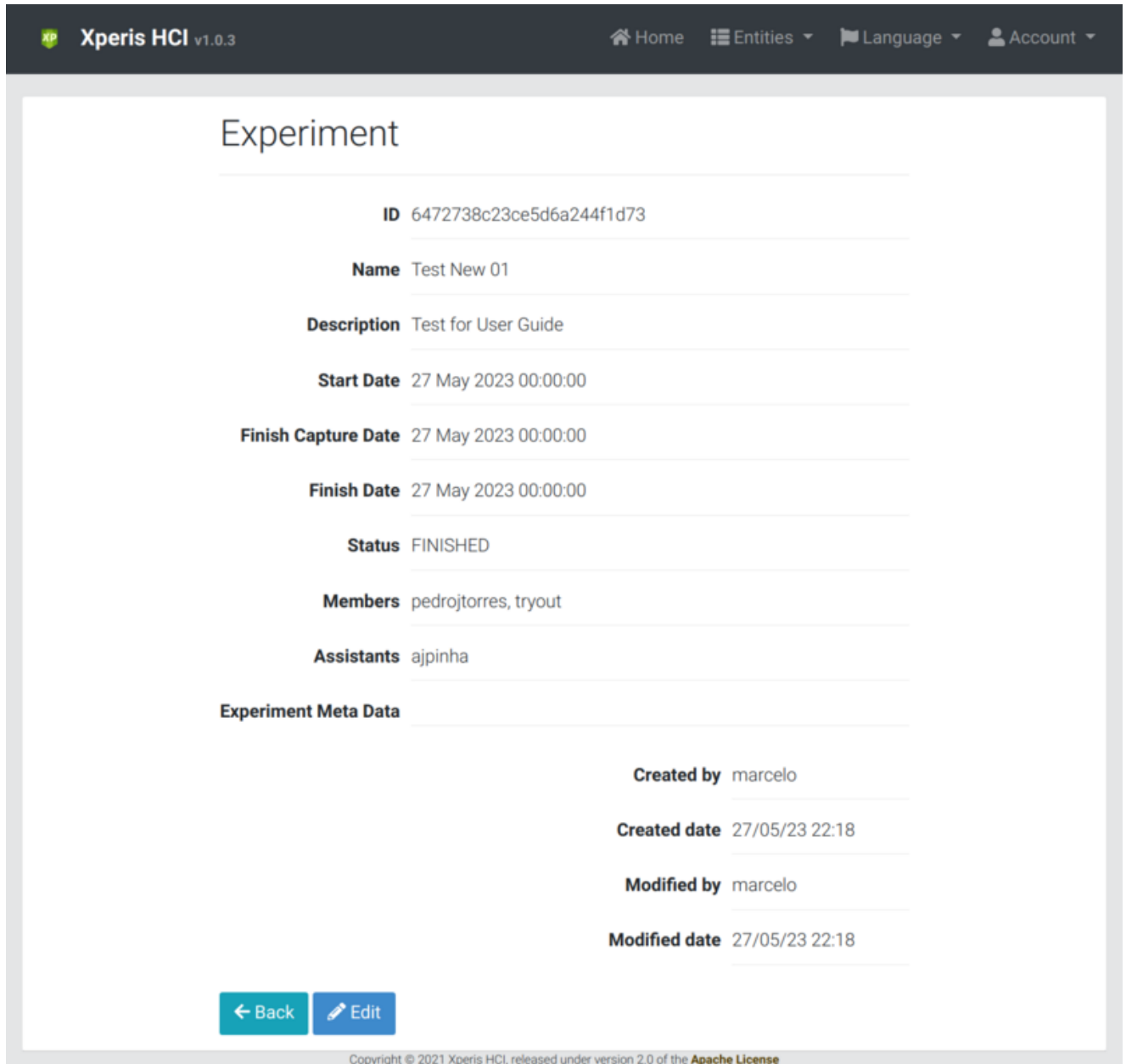
3. If you choose "Delete" the experiment will be permanently deleted. If you choose "Cancel" the process will be canceled and the experiment will not be deleted.

View experiment

To view all the data entered when creating the experiment

1. Choose the "View" option directly from the "Experiments" list

2. The screen "View Experiment" will open



3. In addition to the initial data entered, this screen will also show the date/time stamp of the creation and of the last change. Also, in the "Metadata" section, a link to the metadata associated with the experiment will appear.
4. It is possible to return to the list of Experiments by pressing the "Back" button

Experiment Metadata

Create Metadata

An experiment can collect various different types of data like questionnaires, electronic data feedbacks, impulses, and more. In order to create a new metadata type associated with an experiment:

1. Login as MASTER user
2. Go to the Entities menu and choose "Experiment Metadata" option

- Choose the "Create new Experiment MetaData" option (For a user with the profile USER this option will not be visible)
- Will open the screen "Create or edit Experiment MetaData"

Xperis HCI v1.0.3 Home Entities Language Account

Create or edit a Experiment Meta Data

Experiment

Description

Data Structure

Data Mime Type

Copyright © 2021 Xperis HCI, released under version 2.0 of the **Apache License**

- Choose Experiment, fill the description, the data structure and choose the data mime type.\
- After you have entered the data and pressed the "Save" button, the system returns to the list of Experiments and displays the basic data as well as the options to "View", "Edit" and "Delete"

Xperis HCI v1.0.3 Home Entities Language Account

Experiment Meta Data

ID	Description	Data Structure	Experiment	Data Mime Type	Experiment Data	Plugins	Created by	
6182046cb28165018a7aeb56	D_Sociodemographics	Idade;Gênero;Nacionalidade;Escolaridade;Estatuto Civil;Situação Profissional	D_Questionaire	text/csv	7 item(s)	CSVView	marcelo	Download Zip View Edit Delete
6188f36b7f508359e089bab	Teste Envio Ficheiro LSL	Ficheiro	D_Questionaire	application/zip	6 item(s)		pedrojtorres	Download Zip View Edit Delete
61950458bec8de1cd8b67c57	Teste Envio 72	file	D_Questionaire	application/vnd-zip-compressed	1 item(s)		pedrojtorres	Download Zip View Edit Delete
6472771623ce5d6a244f1d74	Teste for User Guide	column01, column02, total	Test New 01	text/csv	0 item(s)	CSVView	marcelo	View Edit Delete

Showing 1 - 4 of 4 items.

Copyright © 2021 Xperis HCI, released under version 2.0 of the **Apache License**

For those metadata that already have collected data will be displayed also "Download Zip" button to download all data in a compressed file. And for those metadata with a data mime type that already have a plugin associated, will be displayed also a button to each plugin in order to be able to use it.

Create Metadata with Template

In order to create a new metadata using a template:

1. Go to the Entities menu and choose "Experiment Metadata" option
2. Choose the "Create from Template" option (For a user with the profile USER this option will not be visible)
3. Will open the screen "Create or edit Experiment MetaData"
4. Choose Experiment and fill the description. The data structure and the data mime type will be automatically filled with the data from the template.
5. After you have entered the data and pressed the "Save" button, the system returns to the list of Experiments and displays the basic data as well as the options to "View", "Edit" and "Delete".

Edit Metadata

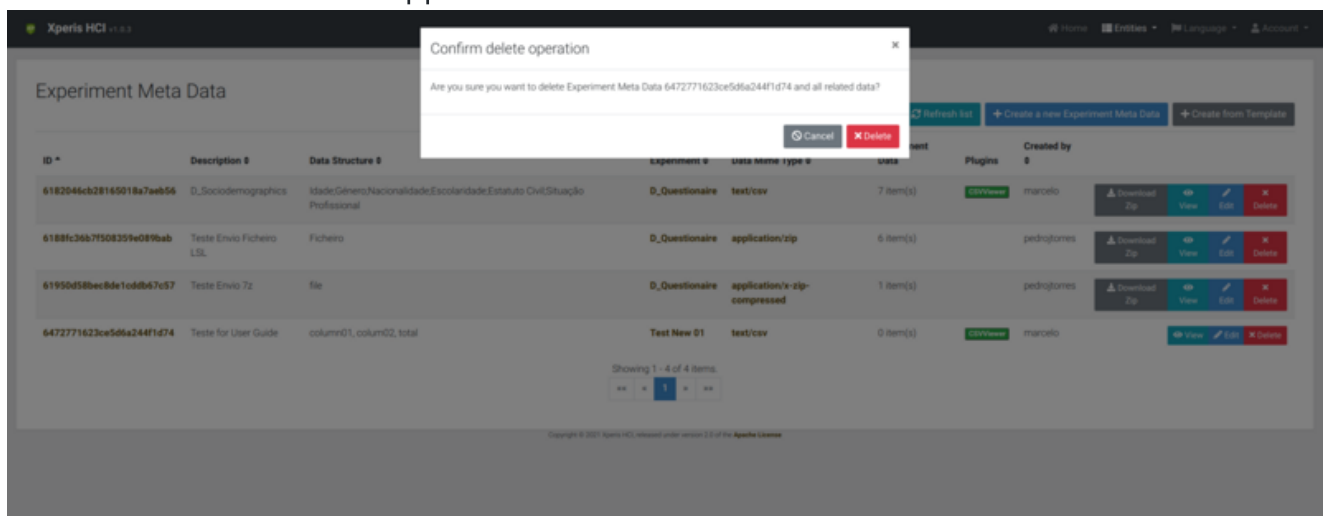
To change any of the data entered when creating the experiment metadata:

1. Choose the "Edit" option directly from the "Experiment Metadata" list
2. The screen "Create or edit Experiment Metadata" will open again, with the data already entered.
3. By choosing "Save" the changes will take effect
4. It is possible to cancel the process just by pressing the "Cancel" button

Delete Metadata

To delete an experiment metadata

1. Choose the "Delete" option directly from the "Experiment Metadata" list
2. A confirmation screen will appear



3. If you choose "Delete" the experiment metadata will be permanently deleted. If you choose "Cancel" the process will be canceled and the experiment metadata will not be deleted.

View Metadata

To view all the data entered when creating the experiment metadata

1. Choose the "View" option directly from the "Experiment Metadata" list
2. The screen "View Experiment" will open

The screenshot displays the 'View Experiment' interface in Xperis HCI v1.0.3. The page title is 'Experiment Meta Data'. The metadata fields are as follows:

- ID:** 6182046cb28165018a7aeb56
- Description:** D_Sociodemographics
- Data Structure:** Idade;Género;Nacionalidade;Escolaridade;Estatuto Civil;Situação Profissional
- Experiment:** D_Questionaire
- Data Mime Type:** text/csv (with a CSVViewer button)
- Experiment Data:** D03 (03/11/21 03:43:22), D04 (03/11/21 03:45:20), D01 (03/11/21 03:42:45), PedroUnity4 (09/11/21 14:43:09), D02 (03/11/21 03:43:07), D05 (03/11/21 03:43:57), D06 (03/11/21 03:44:14)
- Change Logs:** A table with columns 'Modified date', 'Modified by', and 'Action'.
- Created by:** marcelo
- Created date:** 03/11/21 03:39
- Modified by:** admin
- Modified date:** 11/04/23 22:46

At the bottom, there are three buttons: 'Back', 'Download Zip', and 'Edit'. The footer contains the copyright notice: 'Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License'.

Modified date	Modified by	Action
2023-04-14T20:26:41.814Z	admin	DOWNLOAD_DATA
2021-11-04T02:29:41.315Z	marcelo	DOWNLOAD_DATA
2023-04-11T21:46:00.351Z	admin	UPDATE
2023-04-11T21:46:25.889Z	admin	DOWNLOAD_DATA
2021-11-03T09:40:04.801Z	marcelo	DOWNLOAD_DATA
2021-11-04T19:07:09.422Z	admin	DOWNLOAD_DATA
2023-04-12T19:34:35.597Z	admin	DOWNLOAD_DATA
2023-05-27T22:02:52.778Z	marcelo	DOWNLOAD_DATA

In addition to the initial data entered, this screen will also show the date/time stamp of the creation and of the last change. In the "Experiment Data" section, a link to the data associated with the experiment metadata will appear. Also, it will appear a change log section, where will be displayed all changes made to the metadata, with the date/time stamp and the user who made the change.

3. It is possible to return to the list of Experiments Metadata by pressing the "Back" button

Download Metadata

To download all data associated with the experiment metadata

1. Choose the "Download Zip" option directly from the "Experiment Metadata" list or from the "View Experiment Metadata" screen
2. A compressed file will be downloaded with all data associated with the experiment metadata.

Use Plugin

To use a plugin associated with the experiment metadata

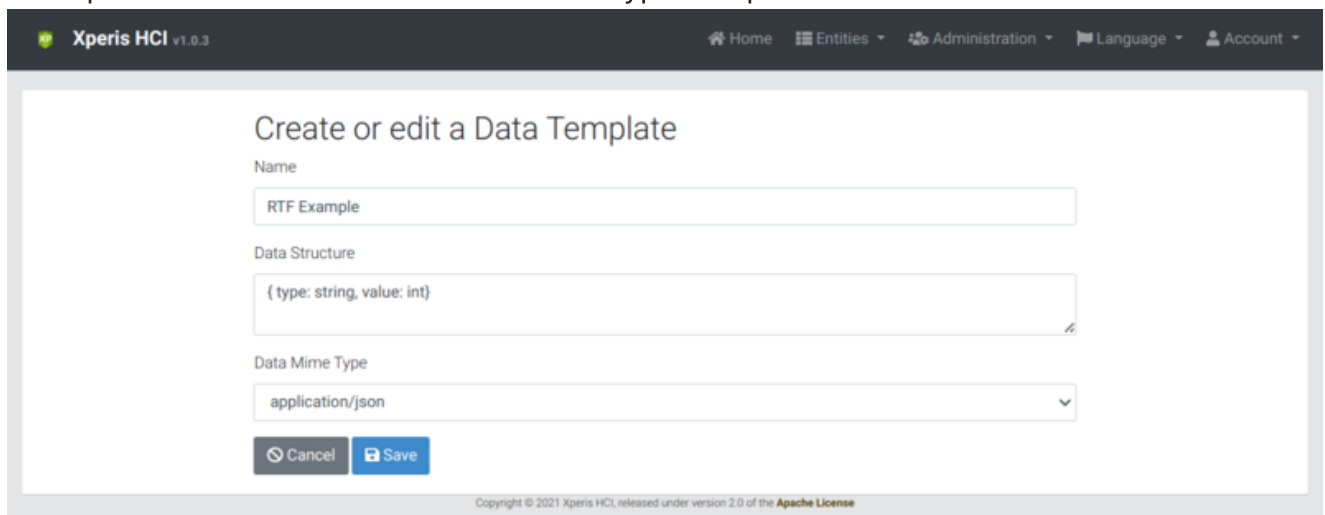
1. Choose the "Plugin" directly from the "Experiment Metadata" list
2. A file generated by the plugin will be downloaded with all collected data associated with the experiment metadata.

Data Type Template

Create Data Type Template

To create a new data type template to utilize in the creation of metadata:

1. Login as MASTER user
2. Go to the Entities menu and choose "Data Type Template" option
3. Choose the "Create new Data Type Template" option (For a user with the profile USER this option will not be visible)
4. Will open the screen "Create or edit Data Type Template"

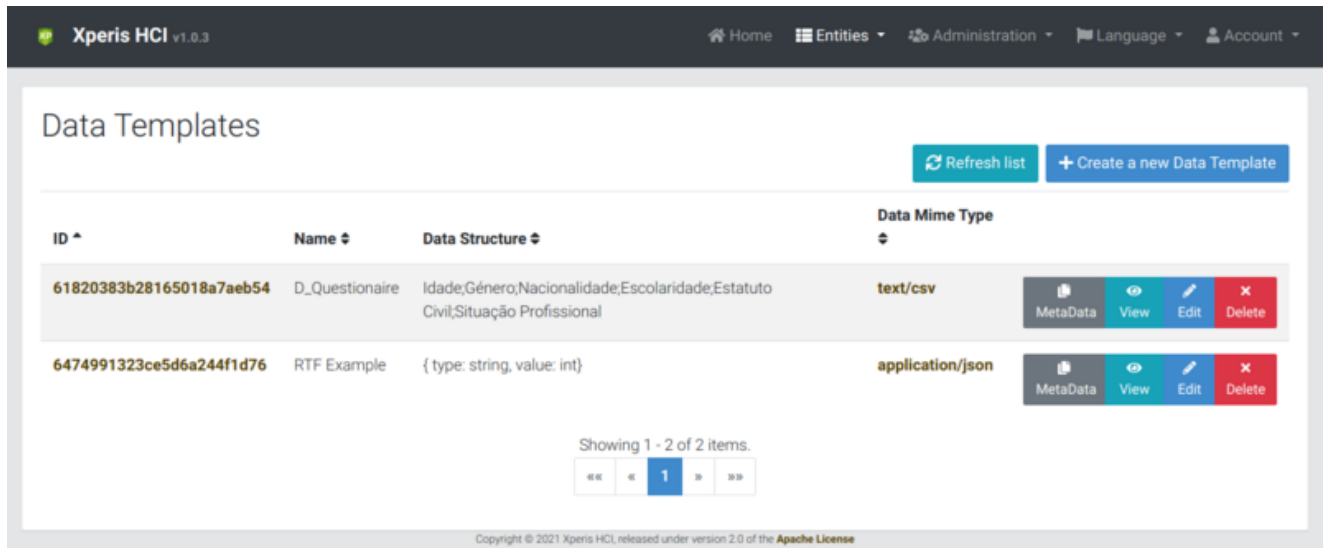


The screenshot shows the 'Create or edit a Data Template' interface. The form contains the following fields and values:

- Name:** RTF Example
- Data Structure:** { type: string, value: int}
- Data Mime Type:** application/json

At the bottom of the form, there are two buttons: 'Cancel' and 'Save'. The footer of the page reads: 'Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License'.

5. Fill the name, the description, the data structure and choose the data mime type.\
6. After you have entered the data and pressed the "Save" button, the system returns to the list of Data Type Templates and displays the basic data as well as the options to "View", "Edit" and "Delete"



Edit Data Type Template

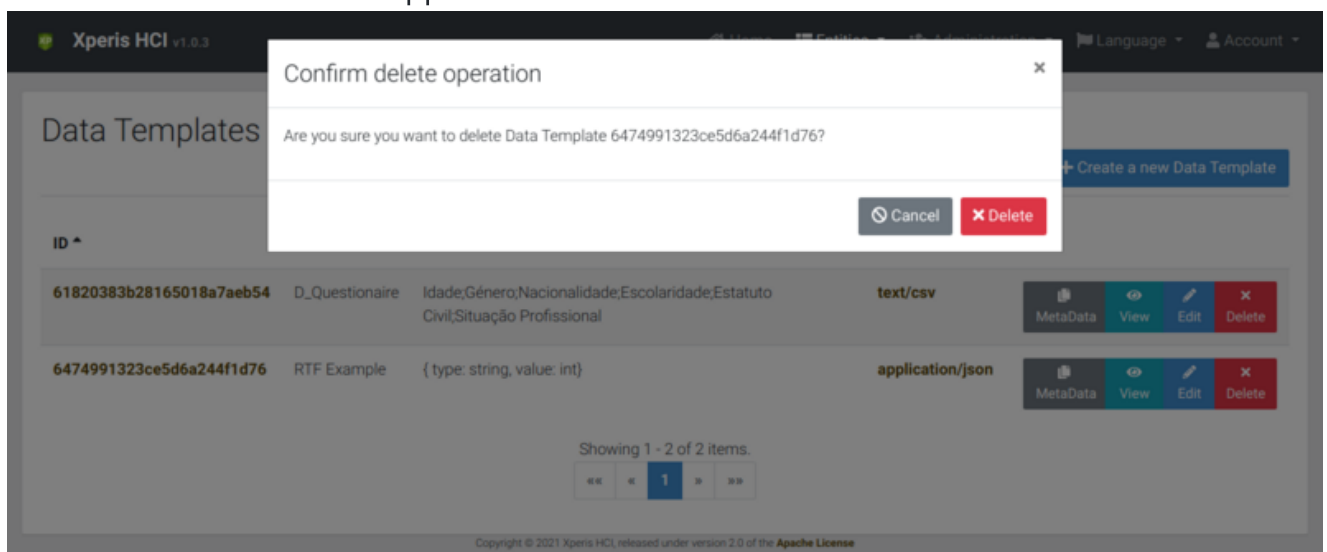
To change any of the data entered when creating the data type template:

1. Choose the "Edit" option directly from the "Data Type Template" list
2. The screen "Create or edit Data Type Template" will open again, with the data already entered.
3. By choosing "Save" the changes will take effect
4. It is possible to cancel the process just by pressing the "Cancel" button

Delete Data Type Template

To delete a data type template

1. Choose the "Delete" option directly from the "Data Type Template" list
2. A confirmation screen will appear

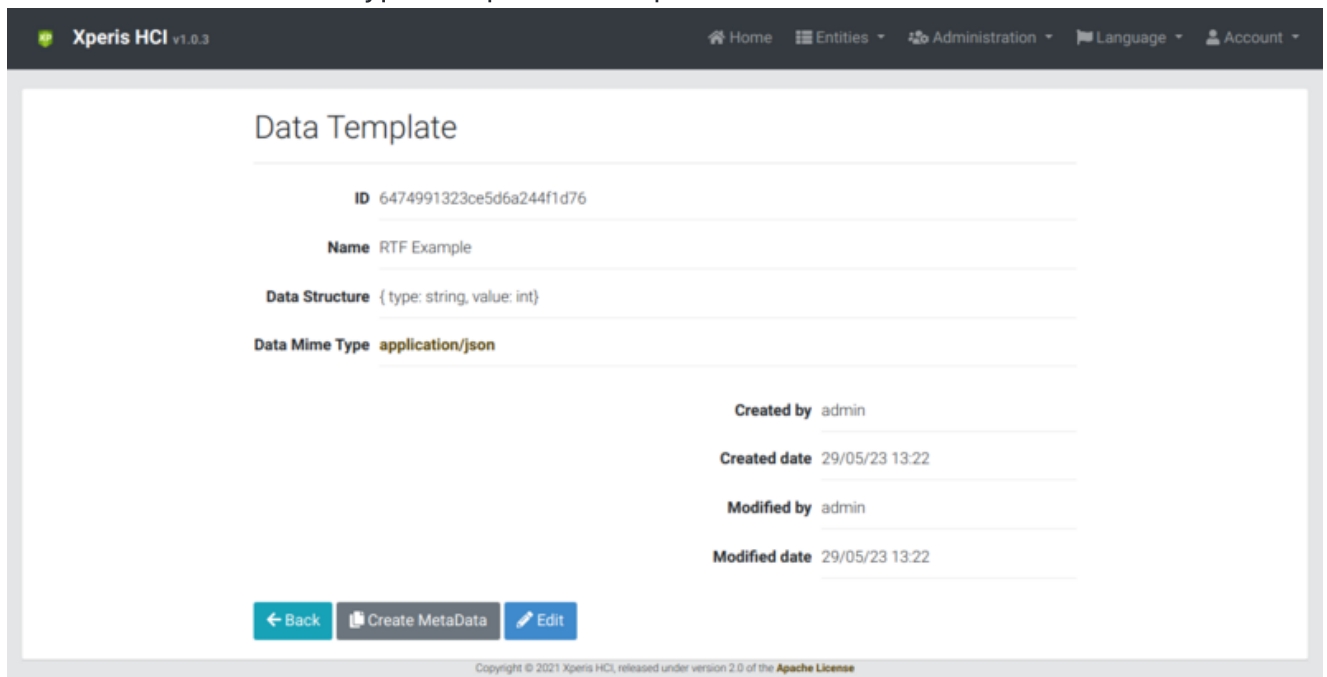


3. If you choose "Delete" the data type template will be permanently deleted. If you choose "Cancel" the process will be canceled and the data type template will not be deleted.

View Data Type Template

To view all the data entered when creating the data type template

1. Choose the "View" option directly from the "Data Type Template" list
2. The screen "View Data Type Template" will open



3. In addition to the initial data entered, this screen will also show the date/time stamp of the creation and of the last change. As well as the "Create Metadata" button.

Create Experiment Metadata from Data Type Template

To create a new experiment metadata using a data type template:

1. Choose the "Metadata" option directly from the "Data Type Template" list or in the "View Data Type Template" screen.
2. Will open the screen "Create or edit Experiment MetaData"
3. Choose Experiment and fill the description. The data structure and the data mime type will be automatically filled with the data from the template.
4. After you have entered the data and pressed the "Save" button, the system returns to the list of Experiments Metadata.

Experiment Data

Create Experiment Data

To create a new experiment data:

1. Login as MASTER user

2. Go to the Entities menu and choose "Experiment Data" option
3. Choose the "Create new Experiment Data" option (For a user with the profile USER this option will not be visible)
4. Will open the screen "Create or edit Experiment Data"

The screenshot shows the 'Create or edit a Experiment Data' form. At the top, there is a navigation bar with 'Home', 'Entities', 'Administration', 'Language', and 'Account'. The form title is 'Create or edit a Experiment Data'. Below the title, there are three main sections: 'Participant(s)' with a text input field containing '10001a'; 'Experiment Meta Data' with a dropdown menu showing 'Alert Development (text/csv)'; and 'Data' with a file upload area showing 'text/csv, 2 359 bytes' and a download icon. At the bottom of the form, there are 'Cancel' and 'Save' buttons. A copyright notice at the bottom reads 'Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License'.

5. Fill the participants identifier (in order to hide the participant identity), the experiment metadata and the upload the data file. The file must be in the same format as informed in the data mime type field of the experiment metadata.
6. After you have entered the data and pressed the "Save" button, the system returns to the list of Experiment Data and displays the basic data as well as the options to "View", "Edit" and "Delete"

The screenshot shows the 'Experiment Data' list. At the top, there is a navigation bar with 'Home', 'Entities', 'Administration', 'Language', and 'Account'. Below the navigation bar, there are two buttons: 'Refresh list' and '+ Create a new Experiment Data'. The table has the following columns: ID, Participant(s), Experiment Meta Data, Data, Created date, and Created by. The table contains 10 rows of data. The last row is highlighted and shows the data entered in the previous step: ID: 6474ce8b23ce5d6a244f1d78, Participant(s): 10001a, Experiment Meta Data: Alert Development (text/csv), Data: text/csv, 2 359 bytes, Created date: 29/05/23 17:10:51.705, Created by: admin. At the bottom of the table, there is a pagination bar showing 'Showing 201 - 210 of 210 items' and a set of navigation buttons. A copyright notice at the bottom reads 'Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License'.

ID	Participant(s)	Experiment Meta Data	Data	Created date	Created by
62d6c1d2a673ef56745016f7	Com@Rehab-58	Data (application/csv)	application/csv, 30 bytes	19/07/22 15:38:10.210	pedrojtortes
62d6c613a673ef56745016f8	Com@Rehab-59	Data (application/csv)	application/csv, 31 bytes	19/07/22 15:56:19.911	pedrojtortes
62d6c968a673ef56745016f9	Com@Rehab-60	Data (application/csv)	application/csv, 26 bytes	19/07/22 16:10:32.838	pedrojtortes
62d6cd08a673ef56745016fa	Com@Rehab-61	Data (application/csv)	application/csv, 32 bytes	19/07/22 16:26:00.270	pedrojtortes
62d6cfc6a673ef56745016fb	Com@Rehab-62	Data (application/csv)	application/csv, 37 bytes	19/07/22 16:37:42.837	pedrojtortes
62d9307c24087b43e84d1cfd	Com@Rehab-63	Data (application/csv)	application/csv, 31 bytes	21/07/22 11:54:52.641	pedrojtortes
62d93e8193a251498479bab1	Com@Rehab-64	Data (application/csv)	application/csv, 26 bytes	21/07/22 12:54:41.620	pedrojtortes
6332da8a58015d3a02eafa8c	Com@Rehab-65	Data (application/csv)	application/csv, 29 bytes	27/09/22 12:12:10.172	pedrojtortes
6332e29c58015d3a02eafa8d	Com@Rehab-66	Data (application/csv)	application/csv, 34 bytes	27/09/22 12:46:36.802	pedrojtortes
6474ce8b23ce5d6a244f1d78	10001a	Alert Development (text/csv)	text/csv, 2 359 bytes	29/05/23 17:10:51.705	admin

Edit Experiment Data

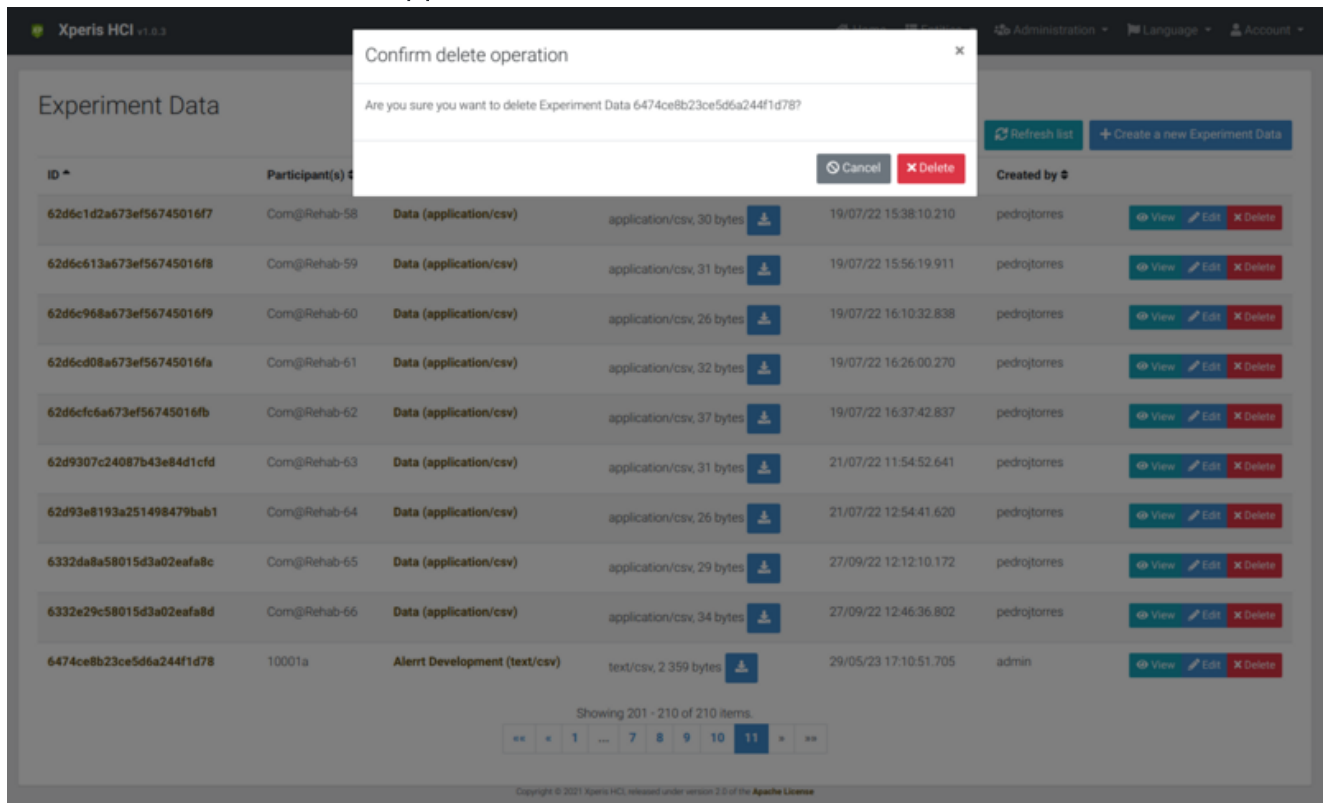
To change any of the data entered when creating the experiment data:

1. Choose the "Edit" option directly from the "Experiment Data" list
2. The screen "Create or edit Experiment Data" will open again
3. By choosing "Save" the changes will take effect
4. It is possible to cancel the process just by pressing the "Cancel" button

Delete Experiment Data

To delete an experiment data

1. Choose the "Delete" option directly from the "Experiment Data" list
2. A confirmation screen will appear



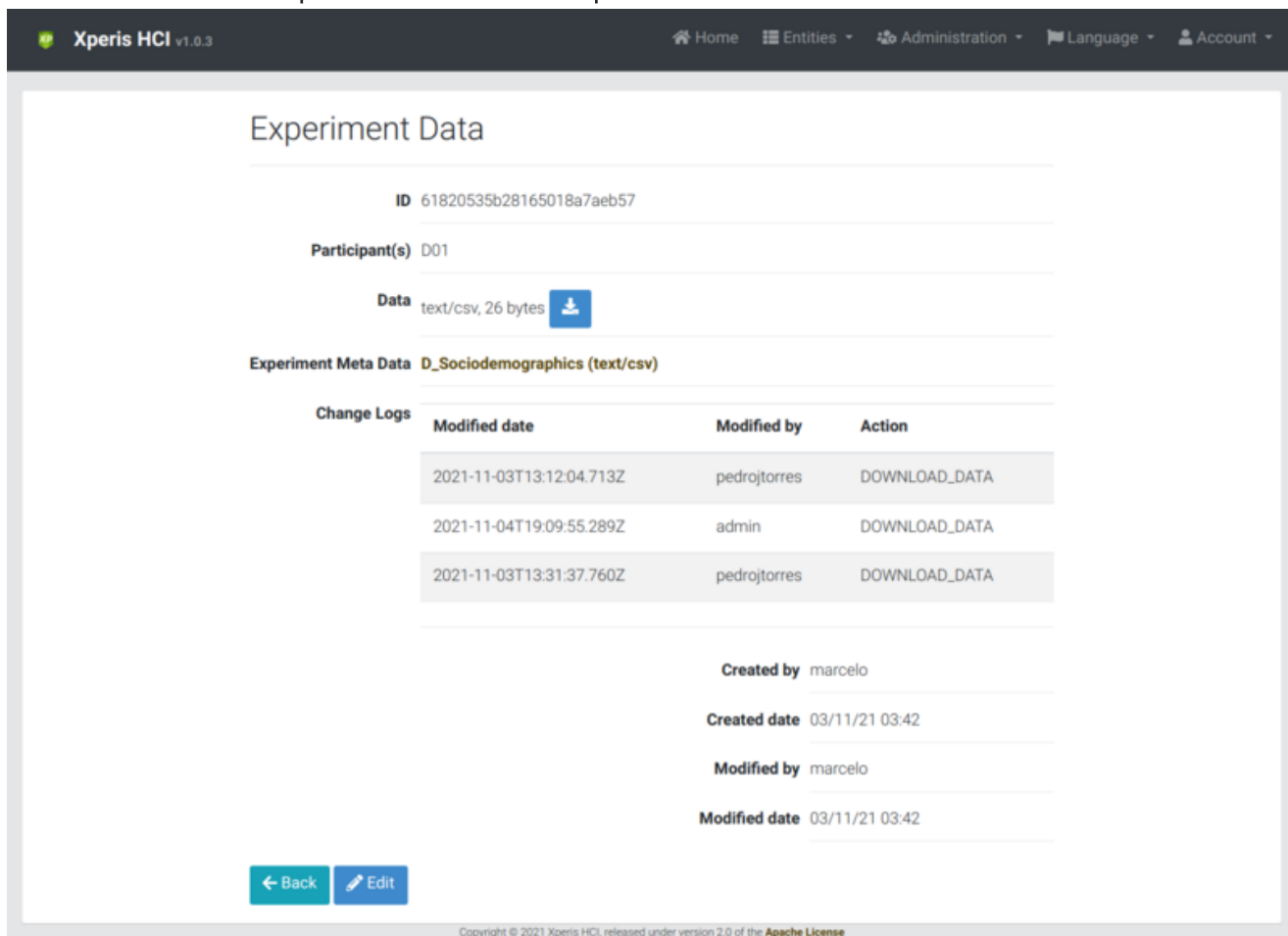
3. If you choose "Delete" the experiment data will be permanently deleted. If you choose "Cancel" the process will be canceled and the experiment data will not be deleted.

View Experiment Data

To view all the data entered when creating the experiment data

1. Choose the "View" option directly from the "Experiment Data" list

2. The screen "View Experiment Data" will open



3. In addition to the initial data entered, this screen will also show the date/time stamp of the creation and of the last change. Also, it will appear a change log section, where will be displayed all changes made to the metadata, with the date/time stamp and the user who made the change.

4. It is possible to return to the list of Experiments Data by pressing the "Back" button

Download Experiment Data

To download the file associated with the experiment data

1. Choose the "Download" button directly from the "Experiment Data" list or from the "View Experiment Data" screen
2. The file associated with the experiment data will be downloaded.
3. It is possible to return to the list of Experiments Data by pressing the "Back" button

Plugin

Add Plugin

To add a new plugin:

1. Login as ADMIN user
2. Go to the Administration menu and choose the "Plugins" option
3. Choose the "Add new Plugin" button
4. Will open the screen "Add or edit a Plugin"

Add or edit a Plugin

Name

Class Name

Activated

JAR File
test-plugin-example-txt.jar (1 559 bytes)

Data Mime Type

Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License

5. Fill the name, the class name, choose the data mime type and upload the plugin file. The file must be a .jar file. This plugin needs to be developed using the plugin core library implementing the interface "PluginInterface" and the method "executePlugin".
6. After you have entered the data and pressed the "Save" button, the system returns to the list of Plugins and displays the basic data as well as the options to "View", "Edit" and "Delete"

Plugins

ID	Name	Status	Class Name	JAR File Name	Data Mime Type	
645002f0aa5b34337589b137	CSVViewer	Activated	pt.ulusofona.heilab.xperis.plugin.CSVViewer ✓	xperis-plugin-csv-1.1.0.jar	text/csv	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
646be30c6335916a1aef568e	CSVViewer2	Invalid	pt.ulusofona.heilab.xperis.plugin.CSVViewer2 ✗	xperis-plugin-csv-1.2.0.jar	text/csv	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
6474d75b23ce5d6a244f1d79	TXTPluginExample	Invalid	com.company.test.TXTPluginExample ✗	test-plugin-example-txt.jar	text/plain	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Showing 1 - 3 of 3 items.

Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License

At this moment the plugin will be validated and loaded to the application in order to be available to be used in the experiment metadata screen. If the plugin is not a valid implementation, a red cross will be displayed after the class name, the status will be changed to "Invalid" and the plugin will not be loaded. If the plugin is valid, a green check mark will be displayed after the class name, the status will be changed to "Activated" and the plugin will be loaded.

Edit Plugin

To change any of the data entered when adding the plugin:

1. Choose the "Edit" option directly from the "Plugins" list
2. The screen "Add or edit a Plugin" will open again, with the data already entered. It is possible to change the name, the class name, the data mime type and the plugin file. For a valid plugin is able to change the status of the plugin from activated to inactivated and vice versa.
3. By choosing "Save" the changes will take effect
4. It is possible to cancel the process just by pressing the "Cancel" button

Deactivate Plugin

To deactivate a plugin:

1. Click at "Activated" button after plugin name directly from the "Plugins" list. This option will not be available for plugins with status "Invalid".
2. The plugin will be deactivated and the status will be changed to "Deactivated"

The screenshot displays the Xperis HCI v1.0.3 interface for managing plugins. At the top, there are navigation links for Home, Entities, Administration, Language, and Account. The main content area is titled "Plugins" and includes buttons for "Refresh list", "Add new Plugin", and "Reload Plugins Folder". A green notification bar states: "A Plugin is updated with identifier 645002f0aa5b34337589b137". Below this is a table with the following data:

ID	Name	Status	Class Name	JAR File Name	Data Mime Type	Actions
645002f0aa5b34337589b137	CSVViewer	Deactivated	pt.ulusofona.heilab.xperis.plugin.CSVViewer ✓	xperis-plugin-csv-1.1.0.jar	text/csv	View Edit Delete
646be30c6335916a1aef568e	CSVViewer2	Invalid	pt.ulusofona.heilab.xperis.plugin.CSVViewer2 ✗	xperis-plugin-csv-1.2.0.jar	text/csv	View Edit Delete
6474d75b23ce5d6a244f1d79	TXTPluginExample	Invalid	com.company.test.TXTPluginExample ✗	test-plugin-example-txt.jar	text/plain	View Edit Delete

At the bottom of the table, it says "Showing 1 - 3 of 3 items." with pagination controls. The footer contains the copyright notice: "Copyright © 2021 Xperis HCI, released under version 2.0 of the Apache License".

Activate Plugin

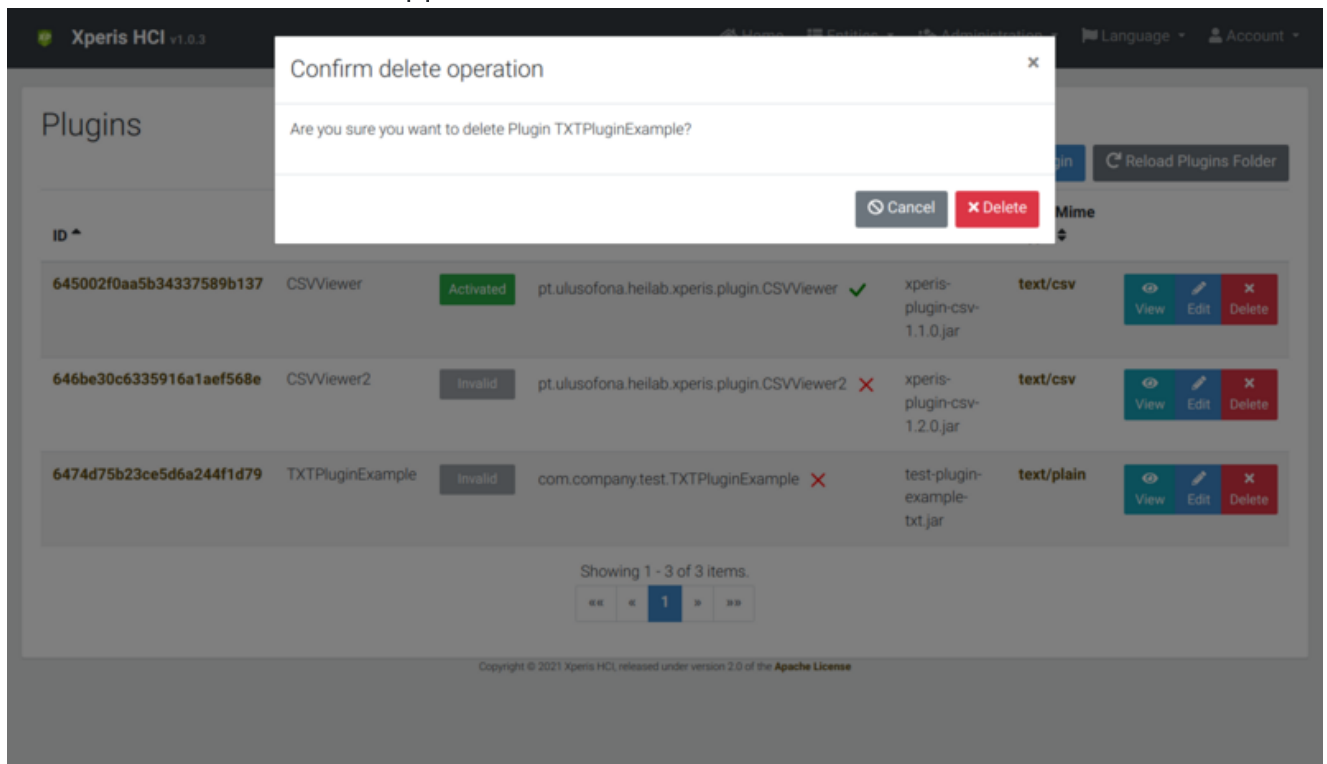
To re-activate a plugin:

1. Click at "Deactivated" button after plugin name directly from the "Plugins" list. This option will not be available for plugins with status "Invalid".
2. The plugin will be re-activated and the status will be changed to "Activated"

Delete Plugin

To delete a plugin

1. Choose the "Delete" option directly from the "Plugins" list
2. A confirmation screen will appear



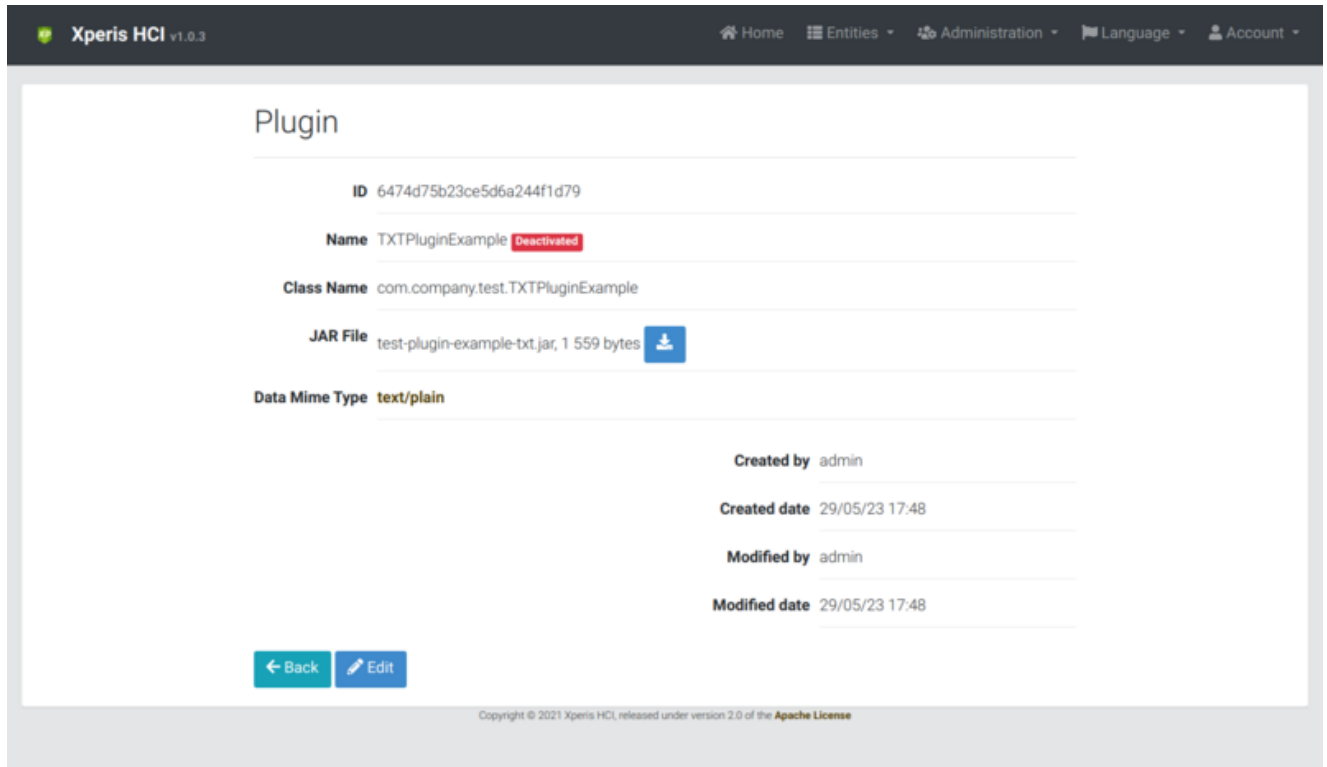
3. If you choose "Delete" the plugin will be permanently deleted. If you choose "Cancel" the process will be canceled and the plugin will not be deleted.

View Plugin

To view all the data entered when adding the plugin

1. Choose the "View" option directly from the "Plugins" list

2. The screen "View Plugin" will open



3. In addition to the initial data entered, this screen will also show the date/time stamp of the creation and of the last change.
4. It is possible to return to the list of Plugins by pressing the "Back" button

Download Plugin

To download the jar file associated with the plugin

1. Choose the "Download" button from the "View Plugin" screen
2. The jar file associated with the plugin will be downloaded.