



Computational Models for new Patients Stratification Strategies of Neuromuscular Disorders

D4.2 Atlas Platform Architecture

CoMPaSS Atlas Platform Architecture

D4.1 Requirements Analysis

PROJECT DETAILS AND DELIVERABLE INFORMATION

PROJECT DETAILS

Project Title	Computational Models for new Patients Stratification Strategies of Neuromuscular Disorders
Project Acronym	CoMPaSS-NMD
Call Identifier	HORIZON-HLTH-2022-TOOL-12-two-stage
Grant Agreement No.	101080874
Duration	48 months
Project Start Date	01/05/2023
Project Website	www.compass-nmd.eu

DELIVERABLE INFORMATION

Deliverable No	D4.2
Deliverable Title	D4.2 Atlas Platform Architecture
Deliverable Leader	FINC
Contributing Partners	All partners
Authors	Riccardo Corrias
Contributors	All Partners
Delivery Date to EC	31/06/2025

DISSEMINATION LEVEL

PU	Public — fully open (automatically posted online)
SEN	Sensitive — limited under the conditions of the Grant Agreement
EU classified	EU classified — RESTREINT-UE/EU-RESTRICTED, CONFIDENTIEL-UE/EU-CONFIDENTIAL, SECRET-UE/EU-SECRET under Decision 2015/444

DOCUMENT HISTORY

Version	Date	Author	Description of Change
0.1	19/05/2025	Riccardo Corrias (FINC)	initial draft
0.2	26/05/2025	Riccardo Corrias (FINC)	content organization
1.0	02/06/2025	Riccardo Corrias (FINC)	first complete version
1.1	05/06/2025	Elisabetta Savino (UNIMORE)	content revision
1.2	06/06/2025	Benedikt Schoser (LMUM)	content review
1.3	06/06/2025	Michael Obach (TEC)	content review
1.4	10/06/2025	Ainara Garzo (TEC)	content review
1.5	10/06/2025	Rossella Tupler (UNIMORE)	content review

1.6	16/06/2025	Stefano Bonelli and Hossein Mapar (DBL)	content review
1.7	16/06/2025	Ana Topf (UNEW)	content review
1.8	19/06/2025	Filippo Santorelli	content review
1.9	25/06/2025	Riccardo Corrias and Marcello Scipioni (FINC)	final review
2.0	30/06/2025	Elisabetta Savino and Rossella Tupler	final review and submission

CONTENTS

1. Executive summary.....	7
2. Project abstract.....	7
3. Introduction.....	8
3.1. Purpose of this document.....	8
3.2. Relation with other CoMPaSS-NMD activities.....	8
3.3. Methodology and Co-Design Approach.....	9
4. Overview of the CoMPaSS-NMD Atlas Platform.....	9
4.1. Aim of the CoMPaSS-NMD Atlas.....	9
4.2. Supported Data Types and Workflows.....	10
4.3. Ethical, Legal and Security Considerations.....	12
5. Architecture Overview.....	14
6. Functional Architecture.....	15
6.1. Key Functional Modules.....	15
6.2. User Interface Components.....	17
6.2.1. Patient Creation.....	17
6.2.2. Data Collection Overview - Patients.....	17
6.2.3. Data Collection Overview - Source Data.....	19
6.2.4. Clinical Data Upload and Display.....	20
6.2.5. Histopathological Data Upload, Evaluation and Overview.....	22
6.2.6. MRI Data Upload and Inspection.....	25
6.2.7. Genetic Data Interface.....	27
6.3. Role-Based Access and User Journeys.....	27
7. Technical Implementation.....	29
7.1. Backend Services and APIs.....	29
7.2. Frontend Framework and Tools.....	33
7.3. Data Storage and Databases.....	34
8. Data Model and Metadata Schema.....	35
8.1. Entity-Relationship Overview.....	35
8.2. Data Normalisation and Standards.....	36
8.3. Scalability Considerations.....	37
9. Deployment Architecture.....	38
10. Security, Privacy, and Compliance.....	39

10.1. Data Protection Measures.....	39
10.2. GDPR Compliance Strategy.....	40
10.3. Secure Access and Authentication.....	41
11. Planned Features and Roadmap.....	41
11.1. Requirements Update (M26 Status).....	41
11.2. Feedback Loop from Co-Design Iterations.....	47
12. Conclusions and Next Steps.....	48
13. Annexes.....	49
13.1. DBML for reproducing DBDiagram.io schema.....	49

LIST OF TABLES

Table 01: Requirements Table 2.0	46
----------------------------------	----

LIST OF FIGURES

Figure 01: CoMPaSS Code Structure and Barcode Example	13
Figure 02: Current CoMPaSS-NMD Atlas Platform Architecture Overview	15
Figure 03: CoMPaSS-NMD Patient Creation Screenshot	17
Figure 04: CoMPaSS-NMD Data Collection Overview - Patients	18
Figure 05: CoMPaSS-NMD Barcode View	19
Figure 06: Data Collection Overview - Source Data Screenshot	20
Figure 07: General Information Section of the Clinical Data Upload Form	21
Figure 08: Clinical Form Overview Screenshot	22
Figure 09: Histopathological Data Upload Form	23
Figure 10: Histopathological Evaluation Upload Form	24
Figure 11: Histopathological Data Overview	24
Figure 12: Histopathological Sample Inspect Module	25
Figure 13: MRI Data Upload Screenshot	26
Figure 14: MRI Data Overview Page	27
Figure 15: Possible User Journeys	29
Figure 16: Development and Deployment Environments	38

GLOSSARY OF ACRONYMS

Acronym	Extended Definition
CERBM	Centre Europeen de Recherche en Biologie et Médecine
CoMPaSS-NMD	Computational Models for new Patients Stratification Strategies of Neuromuscular Disorders
DB	Data Base
DICOM	Digital Imaging and Communication in Medicine
DNA	Deoxyribonucleic acid
DPIA	Data Protection Impact Assessment
ECG	Electrocardiogram
FINC	Fincons Group
FSM	Fondazione Stella Maris
GDPR	General Data Protection Regulation
HPO	Human Phenotype Ontology
JCA	Joint Controller Agreement of the Processing of Personal Data
JPEG	Joint Photographic Experts Group
LOPDGDD	Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales (Spanish Law on Personal Data Protection and Guarantee of Digital Rights)

MRC	Medical Research Council
MRI	Magnetic Resonance Imaging
NIfTI	Neuroimaging informatics Technology Initiative
PDF	Portable Document Format
PNG	Portable Network Graphics
SFF	Samfundet Folkhälsan I Svenska Finland RF
SOP	Standard Operating Procedure
SQL	Structured Query Language
LMUM	Ludwig Maximilians Universität München
TIFF	Tagged Image File Format
UNEW	University of Newcastle upon tyne
UNIMORE	Università degli studi di Modena e regione Emilia
SUT	Silesian University of Technology

DISCLAIMER

Views and opinions expressed in this document are those of the author only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

1. EXECUTIVE SUMMARY

This deliverable presents the first full architecture definition of the CoMPaSS-NMD Atlas platform, a web-based system designed to manage, store, and explore multidimensional biomedical data collected prospectively within the CoMPaSS-NMD project. The platform supports data from clinical, genetic, MRI, and histopathological domains, ensuring secure access, traceability, and future interoperability.

Structured across functional and technical dimensions, this document provides detailed insights into the backend services, frontend technologies, data models, and infrastructure underpinning the platform. It describes the co-design methodology that has shaped development so far, the role-based access architecture, and the implementation of metadata schemas built from SOPs and clinical practice.

A significant portion of the platform's features has already been implemented and tested in both the development (DEV) and production (PROD) environments, which are hosted on the infrastructure provided by SUT, one of the project partners. The architecture enables modularity, scalability, and integration with clustering models and phenotype ontologies. Requirements from Deliverable D4.1 have been revised and expanded to reflect real-world feedback from clinical partners and scientific stakeholders, with a clear roadmap for finalisation and exploitation. This document will guide future development, usability testing, and strategic integration activities in the final phase of the project.

2. PROJECT ABSTRACT

The CoMPaSS-NMD project creates novel and universal tools for the diagnostic stratification of patients suffering from Hereditary NeuroMuscular Diseases (HNMDs) aiming at personalised treatments.

HNMDs occur in young people, but can start at any age, causing long-term disability and in some cases, early death; these conditions bring a lack of participation, need for permanent assistance, and may require long-term institutionalisation. Multidimensional HNMD data - clinical, genetic, histopathological and MRI - will be provided by third-level clinical centres in Italy, France, Germany, Finland, and the United Kingdom as part of the European Reference Network for Rare Neurological Diseases. Computational tools for high-dimensional clustering will be applied in an unsupervised learning approach using the internal structure of data to define groups of similar patients. Classification model averaging and integration techniques for federated learning-inspired model building and novel HNMD-specific descriptors of histopathological images will be implemented. The adoption of this multidimensional view has the potential to increase the diagnostic rate of HNMDs by 30% and foster effective actions by European National Health Systems. As the main project outcome, the CoMPaSS-NMD Atlas Platform will be a cost-effective AI-based application providing precise clinical characterisation and diagnosis, with data remaining publicly available for anyone in the research and health community to use.

The project will deliver Recommendations and Guidelines for stratification-based patient management to offer a superior standard-of-care for diagnosis and prognosis and assist in planning clinical trials. It will follow a user-centred, co-design methodology with strong stakeholder engagement and networking with other project consortia.

The project engages partners with clinical, biotechnological, ICT, computational, ethical, legal, communication and exploitation competencies: 6 clinical/academic centres, 1 academic, and 4 industrial partners.

3. INTRODUCTION

3.1. Purpose of this document

This document defines the architectural framework of the CoMPaSS-NMD Atlas platform, expanding on the requirements established in Deliverable D4.1. It provides a comprehensive description of the system architecture, covering both functional and technical components.

The objective is to describe how the Atlas platform is designed to support the collection, integration, and exploration of multidimensional data related to Hereditary Neuromuscular Disorders (HNMDs). This includes clinical, genetic, histopathological, and imaging data collected during the project, with attention to interoperability, security, and long-term scalability.

The document serves as a foundation for the development and deployment of the platform and ensures consistency with the overall strategy defined within Work Package 4 and the broader CoMPaSS-NMD project objectives.

3.2. Relation with other CoMPaSS-NMD activities

This document is developed within the context of Work Package 4 (WP4), which focuses on the design, development, and validation of the CoMPaSS-NMD Atlas platform. The main objective of WP4 is to implement a digital infrastructure capable of supporting the collection, standardisation, and integration of diverse biomedical data produced throughout the project.

The scope of this deliverable specifically concerns the definition of the Atlas platform architecture, based on the requirements collected during Task 4.1 and documented in Deliverable D4.1. It provides the architecture needed to realise the core functionalities outlined in WP4, including secure data ingestion, metadata management, data browsing, filtering, and export features.

While the technical implementation of the platform will continue through subsequent tasks, this document ensures that the architectural foundations are consistent with the methodological and regulatory principles defined across other work packages.

Key inputs have also come from:

- **WP2 and WP3**, which defined the data standards and collection protocols for clinical, histopathological, genetic, and MRI data, forming the basis for the data types and formats the Atlas platform must support;
- **WP5 and WP6**, which focus on the development of unsupervised clustering and multi-omics patient stratification models;
- **WP7**, which provides guidance on ethical, legal, and regulatory requirements, including GDPR compliance, data anonymization, and secure access management. These constraints are embedded in the architecture design;

- **WP8**, in terms of planned dissemination and usability targets, which influence interface design and data accessibility decisions.

3.3. Methodology and Co-Design Approach

The architecture of the CoMPaSS-NMD Atlas platform has been defined through a **collaborative and iterative co-design** process. This approach was chosen to ensure that architectural decisions remain consistent with the operational needs of clinical partners, data providers, and end-users across the consortium.

The co-design methodology adopted in WP4 involves structured engagement between technical and clinical stakeholders, supported by continuous feedback loops. Initial requirements were collected during Task 4.1 through bilateral discussions, joint workshops, and partner feedback sessions. These interactions helped identify both functional expectations and operational constraints related to data ingestion, access control, system usability, and regulatory compliance.

The iterative nature of the process allowed for early validation of critical assumptions, such as the technical feasibility of source data flows and the practical deployment of metadata descriptors. Regular coordination with partners involved in data collection (WP2, WP3), legal and ethical oversight (WP7), and end-user training and dissemination (WP8) supported incremental refinements to the platform design.

This co-design methodology continues to inform development activities in WP4. Feedback from clinical users and domain experts is continuously incorporated into new feature planning and interface refinement. As detailed in Chapter 11.2 - Feedback Loop from Co-Design Iterations, structured co-design meetings held from late 2024 to mid-2025 alongside phased usability validations have provided practical insights into data upload workflows, metadata requirements, and interface clarity. Additional co-design sessions are scheduled to consolidate advanced search and export functionalities and will further shape the platform roadmap toward M36 and M48.

4. OVERVIEW OF THE CoMPaSS-NMD ATLAS PLATFORM

4.1. Aim of the CoMPaSS-NMD Atlas

The CoMPaSS-NMD Atlas platform is designed to serve as a central infrastructure for the secure and structured collection, organisation, and access of prospective data related to Hereditary Neuromuscular Disorders (HNMDs). The platform supports the harmonised management of clinical, genetic, histopathological, and imaging data collected across participating clinical centres.

The main objective of the platform is to enable the exploration and use of multi-dimensional patient data within the boundaries of ethical and regulatory compliance. The system facilitates prospective data acquisition following Standard Operating Procedures (SOPs) defined in WP2 and WP3 and ensures that all data is accompanied by relevant metadata to support interpretation, retrieval, and filtering.

From an architectural perspective, the platform has been designed with the following priorities that also ensures the compliance with the principles defined in the GDPR:

- **Data integrity and traceability:** Each data object is stored with associated metadata that captures context, origin, and collection details.
- **Secure access:** Access to the platform is currently available to all partners in the CoMPaSS-NMD consortium, who operate under the Joint Controller Agreement of the Processing of Personal Data (JCA) signed at the project level. This agreement defines shared responsibility for personal data protection and treatment and allows partners to preview pseudonymised data collected during the project according to the roles specified in the JCA.
- **Future role-based access control:** While not yet active, the architecture has been designed to support differentiated access levels using Keycloak. This will enable fine-grained user group management and credential assignment as the platform matures.
- **Interoperability:** Data structures are defined to be consistent with relevant ontologies and standards, such as the Human Phenotype Ontology¹ (HPO), DICOM² for imaging, FastQ³ for genetic data and NDPI⁴ for Histopathological Data.
- **Scalability and maintainability:** The platform is built with a modular design to accommodate new data types, evolving workflows, or potential integration with external systems.

4.2. Supported Data Types and Workflows

The CoMPaSS-NMD Atlas platform is designed to manage four main categories of prospective biomedical data: clinical, genetic, histopathological, and MRI data. Each data type is collected according to standardised procedures and accompanied by metadata that enables traceability, searchability, and filtering. Data flows are defined to ensure secure ingestion and storage.

Clinical Data

Clinical data is collected using a structured case report form based on the Standard Operating Procedures (SOP) defined in WP2. Parameters include demographic details, results of clinical examinations, strength scores, comorbidities, and selected laboratory data. The following is a summary of the Data entry workflow for Clinical Data:

1. A new patient is registered via the Atlas platform by clinical personnel.
2. Pseudonymised patient identifiers are generated and stored.
3. Clinical forms are filled through a web interface.

¹ <https://hpo.jax.org/>

² https://dicom.nema.org/medical/dicom/current/output/chtml/part10/chapter_7.html

³ https://en.wikipedia.org/wiki/FASTQ_format

⁴ <https://openslide.org/formats/hamamatsu/>

4. Data is validated and stored.

Genetic Data

Genetic data includes sequencing results, uploaded in standard bioinformatics formats such as VCF⁵ or FASTQ. All prospective sequencing is coordinated with CeGaT to ensure consistency in the sequencing and analysis pipeline. The following is a summary of the Data entry workflow for Genetic Data:

1. DNA samples are extracted at the clinical centres according to established SOPs.
2. Samples are shipped to CeGaT, where sequencing and analysis are performed.
3. FASTQ and CSV resulting files are uploaded directly by CeGaT into the object storage infrastructure of the CoMPaSS-NMD Atlas platform.
4. Each file is linked to the corresponding pseudonymised patient identifier previously assigned during recruitment. The link between the identifier and the personal data of the patients is kept at Hospital level, and only the Principal Investigator can access this information.
5. Associated metadata includes sequencing kit and technology, file format, upload date, and sequencing batch ID.
6. No raw personal identifiable data is handled or stored within the platform.

Histopathological Data

Histopathological data consists of digitised muscle biopsy slides captured in high-resolution formats (e.g., NDP; BigTIFF⁶). These images are prepared and scanned by LMUM. The following is a summary of the Data entry workflow for Histopathological Data:

1. Tissue specimens are stained at local laboratory and shipped on slides to LMUM partner's lab.
2. Slides are scanned using standardised imaging protocols.
3. High-resolution files are uploaded to the platform.
4. Metadata includes staining type, resolution, and acquisition date.
5. Each image is linked to a patient ID and stored for future annotation or analysis.
6. A standardised histopathological report is created and reviewed by LMUM.

MRI Imaging Data

MRI data is captured from participating clinical centres and standardised using defined SOPs. DICOM is the expected file format, compatible with downstream processing workflows. The following is a summary of the Data entry workflow for MRI Data:

1. Muscle MRI scans are acquired at clinical sites according to imaging protocols.
2. Pseudonymised scan files are uploaded through a specific form.
3. Metadata includes anatomical region, scan parameters, and acquisition date.
4. Each scan is indexed and linked to a pseudonymised patient record.

⁵ <https://samtools.github.io/hts-specs/VCFv4.2.pdf>

⁶ <http://bigtiff.org/>

The workflows for each data type have been designed to ensure consistency with ethical constraints, allow retrospective audits, and support future extensions such as data access request management or derived data versioning. Data is stored in a format-agnostic way where possible, preserving original files while enabling structured metadata queries via the platform interface.

4.3. Ethical, Legal and Security Considerations

The CoMPaSS-NMD Atlas platform has been designed in accordance with applicable data protection regulations and technical best practices to ensure secure, ethical, and legally compliant handling of pseudonymised biomedical data.

Access to pseudonymised data stored in the platform is currently available to all project partners under the terms of the JCA signed by all members of the CoMPaSS-NMD consortium. This agreement establishes shared responsibility for compliance with data protection regulations and defines the roles and obligations of each partner with respect to the processing of personal data. Moreover, each partner developed a Data Protection Impact Assessment (DPIA) to evaluate the risks and measures to be taken according to the data collected and the processes to be executed within the project.

Future enhancements of the platform will include the implementation of role-based access control to further refine data access permissions based on user roles and responsibilities.

Data Upload and Transmission

All data transfers to the platform make use of secure, state-of-the-art web protocols. The file upload component implements a multipart upload mechanism over HTTPS⁷, allowing large files to be transmitted reliably and efficiently. Each upload session is authenticated and tracked, ensuring that only authorised users can submit data to the platform.

Storage Infrastructure and Backup

Data is stored in an object storage system hosted within the infrastructure of SUT (Politechnika Śląska), the partner responsible for platform hosting. The storage infrastructure is subject to regular backup procedures and is protected through access control policies consistent with current security standards for biomedical data.

Pseudonymisation

All data collected and stored within the Atlas platform is pseudonymised prior to ingestion. Pseudonymisation is a data protection technique in which identifying information is replaced with artificial identifiers (pseudonyms), making it impossible to directly associate the data with a specific individual without access to separate linkage keys. As it was previously said, the link between the personal data and the codes used for the pseudonymization are kept in the hospitals where the original data collection was carried out and only the principal investigator of that institution has access to the database with the link. This approach ensures compliance with the General Data Protection Regulation (GDPR) while allowing controlled scientific use of the data.

⁷ <https://en.wikipedia.org/wiki/HTTPS>

For each patient and each data object, the CoMPaSS-NMD Atlas platform generates two identifiers:

- A **CoMPaSS ID**, which is a UUIDv4-based⁸ string used as the internal unique identifier across the system. These identifiers are not derived from any personal information and are generated using cryptographically secure randomisation methods. The aim of this identifier is the correlation of the data of each person, but personal identification is not possible using only this ID.
- A **CoMPaSS Code**, which provides a more structured and human-readable identifier, primarily for use by clinical centres managing prospective data. This code is composed using a fixed schema:

DataType-Site-Year-PatNumber-DataNumber, where:

- **DataType** denotes whether the identifier refers to a patient (**PAT**) or a specific data category (e.g. **CLI** for clinical data).
- **Site** is a numeric code assigned to the clinical centre.
- **Year** indicates the year of patient recruitment.
- **PatNumber** is the patient sequence number at that site for the given year.
- **DataNumber** (only for source data codes) distinguishes between multiple datasets linked to the same patient.

Patient Code:

DataType-Site-Year-PatNumber

PAT-001-2025-0001



Source Data Code:

DataType-Site-Year-PatNumber-DataNumber

CLI-001-2025-0001-001



Figure 01: CoMPaSS Code Structure and Barcode Example

All segments in the CoMPaSS Code are expressed in hexadecimal format. This structure enables easy referencing of patient-linked data by clinical centres while ensuring that no direct personal information is embedded in the code. The mapping between these pseudonymised identifiers and patient identifiers is maintained solely within the originating clinical centre and is not accessible through Atlas.

⁸ https://en.wikipedia.org/wiki/Universally_unique_identifier

5. ARCHITECTURE OVERVIEW

The CoMPaSS-NMD Atlas platform is designed as a modular and extensible web-based system for managing prospective biomedical data. Its architecture is built around distinct services responsible for user interaction, data storage, authentication, and backend logic. This modular design supports the current project needs while enabling future integration of additional services, such as patient clustering models or third-party tools. The platform stack consists of the following core components:

- **Frontend (Multi-Page Web Application)**

The frontend is developed using **React**⁹, styled with **Tailwind CSS**¹⁰, and enhanced with **HeroUI**¹¹ components. It consists of multiple structured screens, each corresponding to specific platform functionalities such as patient creation, data upload, and metadata inspection (see Chapter 6 – Functional Architecture). RESTful API¹² calls to the backend are made via the **Axios**¹³ library, enabling secure and asynchronous interaction with data services.
- **Backend (Service Layer and API Gateway)**

The backend is implemented using **Node.js**¹⁴ and serves as an intermediary between the frontend and all core services: PostgreSQL¹⁵, MinIO¹⁶, and Keycloak¹⁷. It handles request validation, data management logic, and interaction orchestration. In future phases, it will also mediate access to AI-based clustering models developed by SUT. All communication is conducted through secure REST APIs.
- **PostgreSQL Database**

Used to store structured metadata, pseudonymised patient records, and submitted clinical forms. Each data entry is linked to its associated patient and metadata tags. Details regarding the database schema and entity relationships are provided in Chapter 8 – Data Model and Metadata Schema.
- **MinIO Object Storage**

A scalable, S3¹⁸-compatible object storage solution hosted on SUT's infrastructure. It is used to store large, non-tabular data files such as MRI scans, histopathological images, and genetic data files. Each file is assigned a CoMPaSS Code together with a UUID-based identifier and is linked to its metadata in the PostgreSQL database.
- **Keycloak Identity and Access Management**

Keycloak is already integrated to manage user authentication across the platform.

⁹ <https://legacy.reactjs.org/>

¹⁰ <https://tailwindcss.com/>

¹¹ <https://www.heroui.com/>

¹² <https://www.redhat.com/en/topics/api/what-is-a-rest-api>

¹³ <https://axios-http.com/>

¹⁴ <https://nodejs.org/en>

¹⁵ <https://www.postgresql.org/>

¹⁶ <https://min.io/>

¹⁷ <https://www.keycloak.org/>

¹⁸ <https://aws.amazon.com/it/s3/>

Although role-based access control is not yet activated, the system is prepared to support future group-based permissions once user roles are formalised.

This architecture allows the platform to support high-volume, multimodal data while preserving flexibility for iterative development and future enhancements. Figure 02 is a representation schema of the abovementioned component stack.

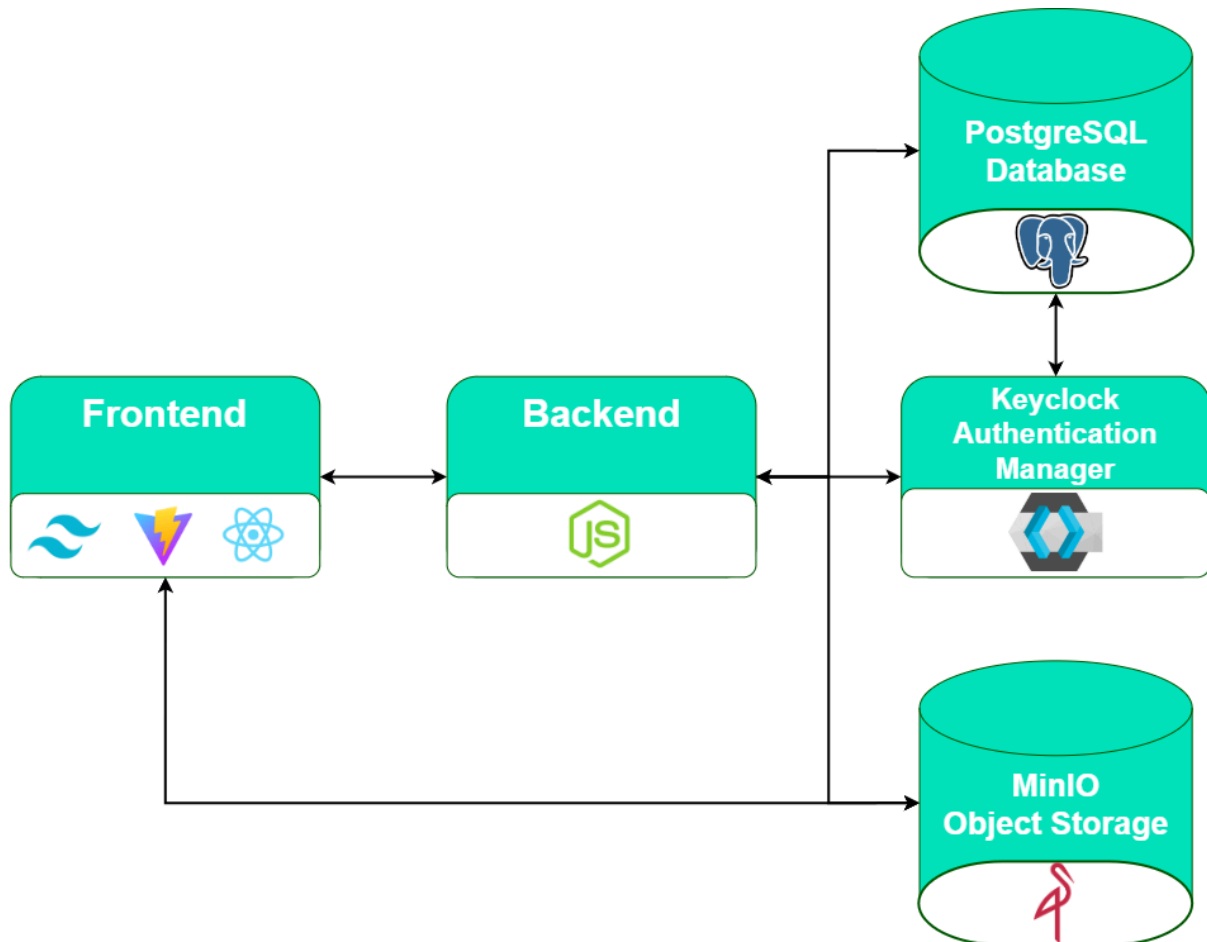


Figure 02: Current CoMPaSS-NMD Atlas Platform Architecture Overview

As illustrated in the diagram, each component interacts with others through clearly defined interfaces. Authentication is managed independently by Keycloak, which stores user credentials and tokens in its own database. The backend handles coordination between services, issuing pre-signed URLs for uploads to MinIO. This architecture ensures that the frontend can upload large files directly to object storage, reducing server-side load and improving scalability — a design choice further detailed in Chapter 7 – Technical Architecture.

6. FUNCTIONAL ARCHITECTURE

6.1. Key Functional Modules

The CoMPaSS-NMD Atlas platform exposes a set of key functional modules through its frontend interface. These modules represent the main user-facing functionalities of the

system and are implemented through coordinated interaction between the frontend, backend services, and underlying data infrastructure. The backend communicates with a MinIO object storage system for file handling and a PostgreSQL database for metadata and platform logic. Details on the underlying architecture and infrastructure are presented in Chapter 7 - Technical Implementation.

The main functional modules include:

- **Patient Registration Module**
Responsible for registering new patients in the platform using a pseudonymisation mechanism. Each patient is assigned both a system-generated UUIDv4 (CoMPaSS ID) and a structured CoMPaSS Code for traceability within the clinical workflows. No personal identifiers are stored or processed within the platform.
- **Data Upload Module**
Enables clinical centres and data providers to upload prospective data — including clinical forms, genetic files, histopathology images, and MRI scans — through a secure web interface. The upload workflow supports large files using multipart upload and captures essential metadata (file type, origin, acquisition date) during the process. As of Month 24, this module is fully operational and in use by clinical partners for data collection.
- **Metadata Management Module**
Maintains and indexes metadata associated with all uploaded patient data. This includes descriptive, technical, and contextual metadata necessary for future retrieval and filtering operations. Metadata fields vary by data type and follow the specifications defined together with internal partners.
- **HPO Terms Module** (*under development, Task 4.3*)
This module will provide the ability to associate structured Human Phenotype Ontology (HPO) terms with clinical parameters. The integration will support hierarchical navigation and phenotype-based filtering of patient records.
- **Data Browsing and Filtering Module** (*initial version implemented*)
A user interface is available for browsing patient records and associated data. As of Month 24, filtering is limited to basic attributes such as sex and clinical centre. The module will be progressively extended to support metadata-level filtering based on clinical forms and other prospective data sources, enabling more advanced queries over time.
- **Export Module** (*under development*)
This component will allow authorised users to export data. Logging mechanisms and access tracking will be implemented to ensure compliance with data governance policies. Initial development has focused on establishing secure upload and storage workflows, with export functionality planned for a later phase.
- **Authentication and Access Control Module** (*under development*)
While current access to the platform is granted to all CoMPaSS-NMD partners under

the Joint Controller Agreement, future integration with Keycloak is planned. This will allow the definition of user groups and granular permission control based on partners' roles and project responsibilities.

These modules are exposed through a unified web-based interface and follow a consistent interaction pattern with backend APIs and persistent storage systems. The modularity at the functional level ensures maintainability and clear separation of concerns in the development process.

6.2. User Interface Components

The user interface (UI) of the CoMPaSS-NMD Atlas platform provides web-based access to core functionalities for clinical partners and data contributors. The interface is structured to support a clean and consistent workflow for prospective data registration, upload, and verification. Each view corresponds to a specific operation, and all components interact with backend services connected to a PostgreSQL database and MinIO object storage system.

The currently available UI components include the following key views:

6.2.1. Patient Creation

Enables authorised users to create new pseudonymised patient entries. The interface collects minimal required information: sex, clinical centre, and month/year of birth. Upon submission, the system generates a unique CoMPaSS ID (UUIDv4) and a human-readable CoMPaSS Code, without storing any personally identifying information. Figure 03 shows a screenshot of the Patient Creation Component.

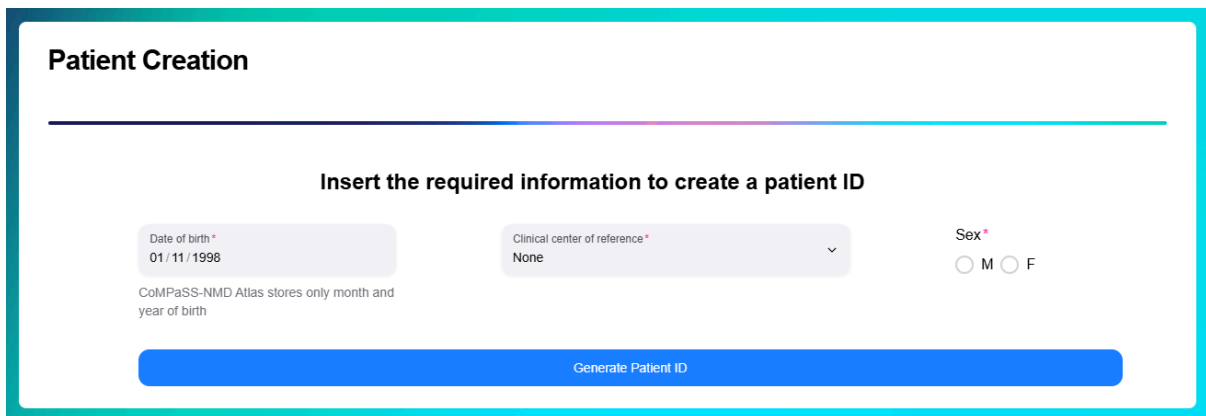


Figure 03: CoMPaSS-NMD Patient Creation Screenshot

The Patient creation is composed of three input components: Date of Birth (only month and year), Clinical Center of reference and Sex. A pseudonymised ID and Code is generated by clicking the “Generate Patient ID” button.

6.2.2. Data Collection Overview - Patients

This interface displays a table listing all pseudonymised patient records stored on the platform. It provides quick access to essential patient metadata and the status of source data

uploads. The current version allows users to filter the view by clinical centre and sex. A search field is also available to locate patients by their pseudonymised Patient ID. Figure 04 shows a screenshot of the aforementioned component.

BAR CODE	CODE	PATIENT ID	CLINICAL CENTER	SEX	DATE OF BIRTH	eSCRIF	TISSUE SCAN	MRI SCAN	DNA
	COMPASS-PAT-001-2025-0000	0f5a8a4e-1d45-48f2-908a-40327ea2151f	Fondazione Stella Maris	Male	01-04-2007	0	2	2	0
	COMPASS-PAT-000-2025-0002	A9dd2bb9-933e-42aa-8d9d-7215ae978da7	Università degli studi di Modena e Reggio Emilia	Male	01-11-1998	0	6	16	0
	COMPASS-PAT-002-2025-0000	A956abb2-9c10-4da4-9161-9ebf3e7ea262	Ludwig Maximilians University	Male	01-11-1970	1	1	1	0

Figure 04: CoMPaSS-NMD Data Collection Overview - Patients

Each patient in the list is presented with the following 10 columns of information:

- **Bar Code** – A button that opens a view showing the barcode representation of the patient's CoMPaSS Code (see Figure 05). The barcode can be downloaded and printed, allowing clinical centres to associate the code with physical samples such as biopsy slides or extracted DNA.
- **Code** – Displays the structured, human-readable pseudonymised CoMPaSS Code (e.g. `COMPASS-PAT-001-2025-0000`).
- **Patient ID** – Displays the internally used UUIDv4-based CoMPaSS ID.
- **Clinical Center** – Indicates the clinical centre of reference for the patient.
- **Sex** – Indicates the sex of the patient.
- **eSCRIF / Tissue Scan / MRI Scan / DNA** – For each prospective data type, a numeric value indicates the number of uploaded files associated with that patient. Clicking on the number opens a context-specific action, allowing users to either upload new data or access existing entries.



Figure 05: CoMPaSS-NMD Barcode View

6.2.3. Data Collection Overview - Source Data

This view provides a detailed listing of all uploaded prospective data entries across patients. Each row corresponds to a single data object and displays its key metadata. The interface allows users to search by Source Data ID or Patient ID and to apply filters by source type, file type, or file extension. This enables project partners to monitor completeness, traceability, and format compliance of all incoming data.

Each data record includes the following 8 metadata fields:

- **Source Data ID** – A unique pseudonymised identifier automatically generated by the platform upon upload.
- **Patient ID** – The UUIDv4-based pseudonymised identifier linking the data to a registered patient.
- **Source** – Indicates the origin of the data, such as “MRI,” “Genetic,” “Histopathological,” or “Clinical.”
- **Upload Date** – The date when the file was uploaded to the Atlas.
- **Collection Date** – The date when the data was originally collected from the patient.

- **Type** – Specifies the format or modality of the data, e.g., DICOM or bigTIFF.
- **Extension** – Indicates the file compression or packaging format used (e.g., `.zip`, `.gz`, `.tiff`).
- **Size** – Displays the file size, which supports integrity checks and aids in resource planning.

Figure 06 shows a screenshot of the Data Collection Overview - Source Data Page.

Data Collection Overview

Patients Source Data

Q Search by Source Data ID Q Search by Patient ID Filter by Source Filter by Type Filter by Extension

Source Data ID	Patient ID	Source	Upload Date	Collection Date	Type	Extension	Size
0af3efd4-f5e7-4da0-a2b8-40dce3750a05	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	03-04-2025	03-04-2025	dicom	zip	38.86 MB
f22dfbcb-d26d-406f-a49d-7d6d876cb3dd	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	03-04-2025	03-04-2025	dicom	zip	38.86 MB
e1c1b842-d486-4993-8e21-6e56990483b4	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	03-04-2025	03-04-2025	dicom	gz	17.00 MB
1042a134-4d83-4c77-9bd7-bb0abf52d27a	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	03-04-2025	03-04-2025	dicom	gz	79.49 MB
4ab963c8-3fbf-4158-8fd9-cd8a63df5e38	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	04-04-2025	04-04-2025	dicom	gz	17.00 MB
4a82b384-bb40-46a4-9536-7018ee31dd68	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	04-04-2025	04-04-2025	dicom	zip	38.86 MB
8f6df472-6e17-48f8-822f-eaadaf1f4034c	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	04-04-2025	04-04-2025	dicom	zip	38.86 MB
590bc317-b373-4568-a177-1da3d35025a4	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	04-04-2025	04-04-2025	dicom	zip	615.97 MB
d3672cf7-262d-44d4-a4a4-8fc62532e3d5	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	04-04-2025	04-04-2025	dicom	zip	38.86 MB
fa3118c-d3b8-413e-b161-644742c46dcf	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	07-04-2025	07-04-2025	dicom	gz	17.00 MB
6bc272fe-9b66-40f1-972e-4c7661f086ae	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	07-04-2025	07-04-2025	dicom	zip	38.86 MB
d2ecc3cb-fc30-4be6-afc4-68a3fd598a75	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	07-04-2025	07-04-2025	dicom	gz	79.49 MB
4268c9f5-bc49-4ea1-b1fe-e8f231ca25b0	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	07-04-2025	07-04-2025	dicom	zip	1713.36 MB
0ad16951-9849-47c8-bd36-cbfc10c41957	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	07-04-2025	07-04-2025	nifti	zip	38.86 MB
c53948ef-fe37-4d8f-b30c-979b8c652e4e	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	MRI	07-04-2025	07-04-2025	dicom	gz	79.49 MB
16bddcb1b-405a-463f-846d-71d560d41ca1	0f5a8a4e-1d45-48f2-908a-40327ea2151f	MRI	07-04-2025	07-04-2025	dicom	zip	121.92 MB
97651668-1b3a-46e7-96e2-06e939b88262	a956abb2-9cf0-4da4-9161-9ebf3e7ea262	MRI	10-04-2025	10-04-2025	dicom	zip	38.17 MB
9d143931-5be5-4201-acf9-dd2ab5f1488f	a9dd2bb9-933e-42aa-8d9d-7215ae978da7	Histopathological	04-04-2025	04-04-2025	bigTiff	.tiff	1833.45 MB

< 1 2 >

Figure 06: Data Collection Overview - Source Data Screenshot

6.2.4. Clinical Data Upload and Display

The clinical data interface supports the structured collection and visualisation of clinical parameters using a digital version of the eCoreCRF described in Deliverable D4.1 (Requirements Analysis). The upload interface is organised into multiple sections and subsections aligned with clinical workflows and phenotyping practices. It includes over 200 input components for recording a wide range of patient-specific variables.

During data entry, users navigate across grouped sections such as general information, neurological examination, and family history. Each subsection is presented as an isolated input block, enabling incremental and modular completion of the form.

Given the extensive scope of the clinical form, it has been designed to save partial data through the same session. Data entered during an incomplete session is automatically stored in the local session storage of the browser. This allows users to pause the process and resume form compilation without data loss. However, only one clinical form can be compiled at a time — initiating a new form for the same patient before submission will overwrite existing unsaved data. Figure 07 shows an initial view of the aforementioned Clinical Data Upload Form.

General Information

Date of compilation *
28 / 05 / 2025

Selected date: Wednesday, May 28, 2025

Handedness *
 Left Right

Weight * 0.0 Kg Height * 0.0 cm BMI

Family Ancestry - geographic origins:

Maternal * Paternal *

Consanguinity *
 Yes No

Figure 07: General Information Section of the Clinical Data Upload Form

After submission, clinical data is displayed in a dedicated visualisation interface. This screen presents a list of all clinical forms submitted for the patient, including the compilation date and the associated CoMPaSS Code. When a specific form is selected, the parameters are displayed in a structured overview, grouped by thematic accordion-style sections for improved readability. Figure 08 shows the Clinical Data Overview Page.

This interface allows clinical partners to verify submitted data, monitor progress, and prepare complete datasets for downstream validation and analysis.

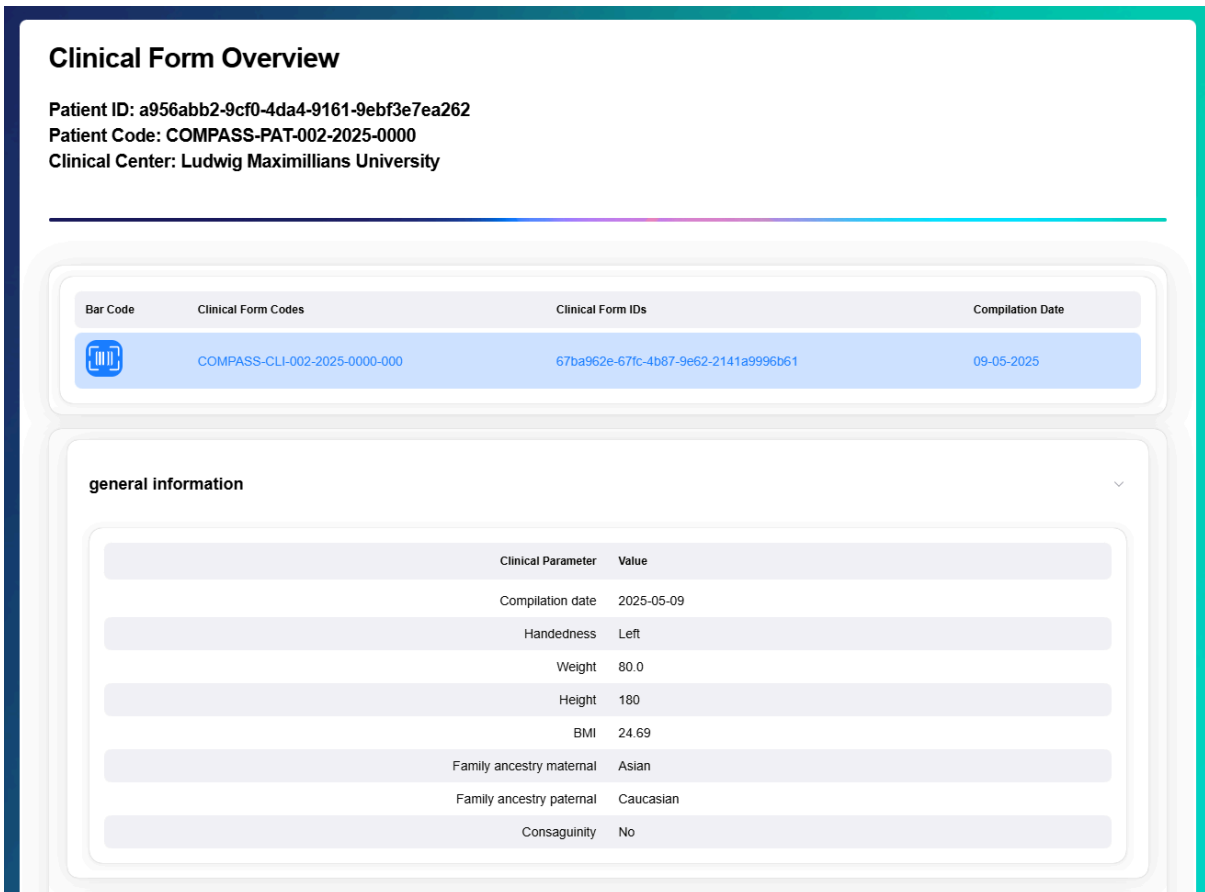


Figure 08: Clinical Form Overview Screenshot

6.2.5. Histopathological Data Upload, Evaluation and Overview

The CoMPaSS-NMD Atlas platform provides a dedicated interface for uploading and managing histopathological data. This functionality allows clinical centres to submit high-resolution digital biopsy images together with both objective metadata and subjective evaluations. The upload process is divided into two distinct steps to separate technical file attributes from expert assessments.

The **Histopathological Data Uploader** view enables the submission of raw data files (e.g., NDP, BigTIFF) along with essential metadata such as muscle name, biopsy method, and histochemical stain used. In agreement with LMUM and other clinical partners, the uploader also supports the inclusion of a smaller preview image of the sample. This preview facilitates quick sample identification in the overview interface without loading the full-resolution file. Figure 09 shows the Histopathological Upload Screen.

Once the sample and metadata are submitted, users can proceed to the **Histopathological Evaluation** screen. This interface allows the entry of subjective observations using a standardised form with multiple binary, categorical, and percentage fields. Parameters include features such as fiber shape, presence of vacuoles, and specific pathological structures. Figure 10 shows the Histopathological Evaluation Upload Screen.

Histopathological Data Uploader

Patient ID: 0f5a8a4e-1d45-48f2-908a-40327ea2151f
Clinical Center: Fondazione Stella Maris

Required Metadata

Name of Muscle*
None

Biopsy method*
 Open Needle

Frozen*
 Yes No

Formalin fixed*
 Yes No

Paraffin embedded*
 Yes No

Handling artifacts*
 Yes No

Histochemical stains*
None

File Type*
None

Collection Date*
28 / 05 / 2025

Upload Date*
28 / 05 / 2025

Source Data

Preview File

Upload Histopathological Data

Figure 09: Histopathological Data Upload Form

All uploaded samples and evaluations are listed in the **Histopathological Data Overview** table. Each row represents a unique histopathological sample, with columns displaying the CoMPaSS Code, clinical center, muscle name, stain type, and acquisition dates. A magnifying glass icon in the final column opens a detailed view. Figure 11 shows the Histopathological Data Overview Screen.

The **Inspection View** allows the user to examine all metadata fields, download preview images, and review linked evaluations. This consolidated view aids pathologists and data managers in verifying sample quality, checking completeness, and ensuring traceability. Figure 12 shows the Inspection Sample View.

Histopathological Evaluation

Histopathological Data ID: de73dbbc-2096-4a73-8bed-39281bb79d16

Patient ID: a9dd2bb9-933e-42aa-8d9d-7215ae978da7

Clinical Center: Università degli studi di Modena e Reggio Emilia

Histopathological Evaluation Form

Ice Crystals*

Yes No

Fatty infiltration*

None

Endomysial fibrosis*

None

Myofiber degeneration*

None

Necrosis*

None

Myophagocytosis*

Absent Present

Basophilic fibers, Large nuclei*

Absent Present

Hypertrophic fibers*

Absent Present

Atrophy / Hypotrophy*

Absent Present

All fibers within the specimen*

Yes No

Subsets of fibers, leading to excessive variation in fiber size*

Yes No

Single fibers*

Yes No

Groups of fibers*

Yes No

Perifascicular distribution*

Yes No

Nuclear bags / clumps*

Yes No

Atrophic / Hypotrophic fiber*

Yes No

Fiber shape

Angulated Round

Figure 10: Histopathological Evaluation Upload Form

Histopathological Data Overview

Patient ID: 0f5a8a4e-1d45-48f2-908a-40327ea2151f

Patient Code: COMPASS-PAT-001-2025-0000

Clinical Center: Fondazione Stella Maris

Bar code	Code	Histopathological data id	Clinical center	Name of muscle	Histochemical stains	Collection date	Upload date	Inspect
	COMPASS-HIS-001-2025-0000-000	11714d2a-f947-4bcc-9d4c-57584c5b2c2d	fsm	Vastus lateralis	H&E	07-04-2025	07-04-2025	
	COMPASS-HIS-001-2025-0000-001	8681e045-4dbe-4aef-8a4a-e5ec8d11263c	fsm	Quadriceps femoris	H&E	07-04-2025	07-04-2025	

Figure 11: Histopathological Data Overview

Histopathological Data ID: 9d143931-5be5-4201-acf9-dd2ab5f1488f

Upload date 04-04-2025	Collection date 04-04-2025
Type bigTiff	Extension .tiff
Size 1833.45 MB	Name of muscle Biceps brachii
Biopsy method Open	Frozen Yes
Formalin fixed No	Paraffin embedded No
Handling artifacts No	Histochemical stains H&E

Code
COMPASS-HIS-000-2025-0002-001

Evaluations List
a0f66d0f-f7b4-4828-9008-cab0b94efd4b

Parameter	Value
Ice crystals	No
Fatty infiltration	Absent
Endomysial fibrosis	Absent
Myofiber degeneration	Mild
Necrosis	Mild

Add Evaluation

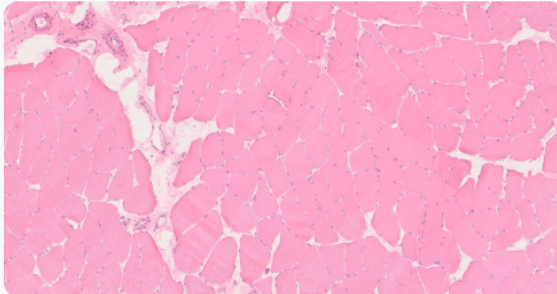


Figure 12: Histopathological Sample Inspect Module

6.2.6. MRI Data Upload and Inspection

The MRI data interface allows authorised users to upload pseudonymised imaging files, typically in DICOM format, along with descriptive metadata. The goal is to collect standardised muscle imaging data from participating centres in accordance with the SOPs developed in WP3.

The **MRI Data Upload** form includes fields for specifying anatomical region, acquisition date, file type, and additional context useful for downstream interpretation. Users are required to fill in essential metadata at the time of upload.

To streamline metadata entry, the platform includes a tool to assist with DICOM files. By clicking the **Parse Header DICOM File** button, the platform automatically reads the DICOM header and extracts relevant metadata such as acquisition modality, study date, and imaging orientation. This pre-filled information can then be reviewed and adjusted before submission. The actual upload is triggered via the **Upload MRI Data** button. Figure 13 shows the MRI Data Upload View.


MRI Uploader

Patient ID: 0f5a8a4e-1d45-48f2-908a-40327ea2151f
Clinical Center: Fondazione Stella Maris

Metadata

Modality	Scanning sequence	Sequence variant	Repetition time (TR)
Echo time (TE)	Flip angle	Slice thickness	Spacing between slices
Body part examined*	Patient position	File Type*	

Collection Date* 28/05/2025 Upload Date* 28/05/2025



Parse Header Dicom File

Upload MRI Data

Figure 13: MRI Data Upload Screenshot

Once uploaded, MRI files can be accessed through the **MRI Data Overview** page (figure 14). This screen displays a scrollable table of MRI scans uploaded for the selected patient. Each row includes the CoMPaSS Code, file type, format, upload and collection dates, anatomical region, and associated metadata.

The overview page is designed for easy verification and monitoring of MRI submissions, ensuring that each patient’s imaging data is complete and properly documented.

MRI Overview

Patient ID: c8c61fd0-d048-4803-b36f-543681feb5d6
 Patient Code: COMPASS-PAT-000-2025-0003
 Clinical Center: Università degli studi di Modena e Reggio Emilia

Upload date	Collection date	Type	Extension	Size	Modality	Scanning sequence	Sequence variant	Repetition time	Echo time	Flip angle	Slice thickness	Body par
28-05-2025	28-05-2025	dicom	.dcm	0.55 MB	XA	RM	OSP	4844	123	90	1.5	Lower B

Figure 14: MRI Data Overview Page

6.2.7. Genetic Data Interface

The platform includes a dedicated interface for uploading genetic data, designed with future scalability and exploitation in mind. While this interface has been fully implemented on the frontend, it is not currently used as part of the active data collection workflow within the CoMPaSS-NMD project.

As defined in WP3, prospective genetic data is generated through whole genome or exome sequencing performed externally by CeGaT. The resulting files — including formats such as VCF and FASTQ — are uploaded directly by CeGaT technicians into the MinIO object storage associated with the platform. This upload is performed outside the UI, following a secured internal process that ensures data integrity and traceability.

The Genetic Data Upload interface includes fields for metadata such as sequencing type, collection and upload dates, file format, extension, and file size. It also includes inputs for tracking sequencing batch and source centre, ensuring compatibility with both retrospective quality assurance and future multi-centre usage.

Although currently unused within CoMPaSS-NMD operations, this module provides a foundation for future access by other projects and clinical partners beyond the consortium. Furthermore, controlled user uploads subject to future governance agreements.

6.3. Role-Based Access and User Journeys

Access to the CoMPaSS-NMD Atlas platform is currently governed by the JCA signed by all members of the project consortium. This agreement establishes a shared responsibility model for data protection and enables full access to pseudonymised data across all partners. All authenticated users from the consortium currently operate with equivalent access rights.

However, the platform architecture has been designed to support more granular, role-based access control in future iterations, as established during the co-design meetings. The

authentication and user management system will be extended, an open-source identity and access management solution. Once integrated, Keycloak will enable fine-grained control of user permissions based on organisational roles and responsibilities.

Planned user roles include:

- **Clinical Partner**
Can register patients, upload and edit prospective source data (clinical, imaging, histopathological, and genetic), and access metadata and overview pages relevant to their institution.
- **Data Manager**
Has visibility across the entire dataset. Responsible for verifying metadata integrity, managing corrections, and monitoring completeness and compliance across submissions.
- **Research Partner**
Can search, filter, and export pseudonymised datasets once the export module is active and approval workflows are in place. Cannot perform uploads or modifications.
- **Administrator** (*restricted to platform host*)
Maintains platform operations, user provisioning, system backups, and technical configuration. Does not access any clinical or pseudonymised data.

Each user's interaction with the platform follows a specific navigation path depending on their role and objectives. To visually represent this, a diagram (Figure 15) has been prepared outlining the available pages and the directional flow between functionalities. This platform access map illustrates how a user, such as a clinical partner, may log in, register a new patient, upload clinical and imaging data, view metadata, or access submitted forms.

It is possible to observe from the diagram below that a user without logging in can only access the pseudonymised Data Collection Overview page; in order to access any other functionality of the Atlas platform the user needs to login to the platform.

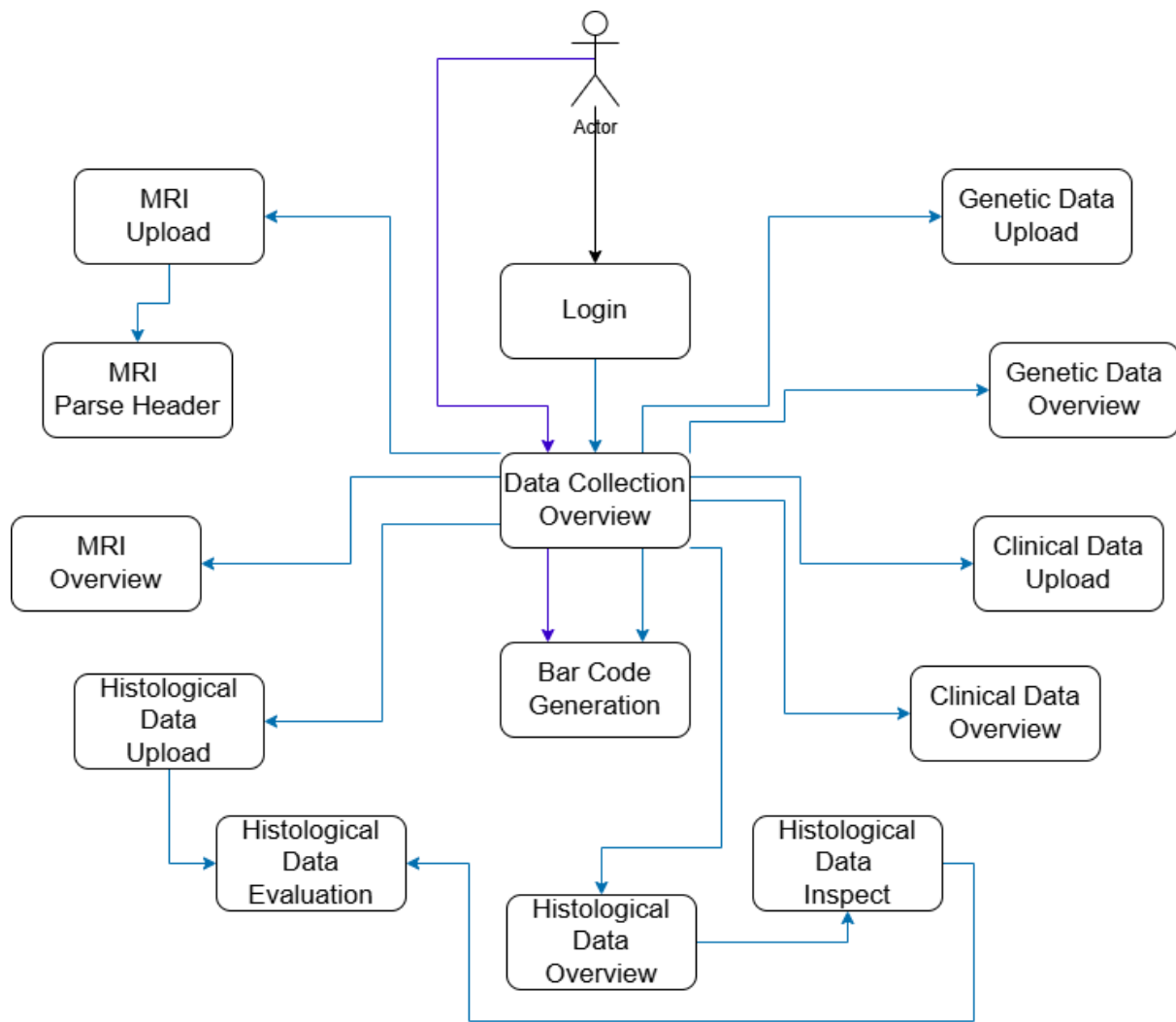


Figure 15: Possible User Journeys

7. TECHNICAL IMPLEMENTATION

7.1. Backend Services and APIs

The backend of the CoMPaSS-NMD Atlas platform is developed using **Node.js** and serves as the core processing layer that coordinates logic between the user interface, storage systems, database, and authentication service. It exposes a structured set of RESTful APIs that allow frontend components to perform operations such as patient creation, data upload, metadata retrieval, and clinical form submission.

All API calls are secured using **token-based authentication** managed by **Keycloak**. Before executing any operation, requests are validated for structure, required fields, and token authenticity to ensure robustness and security. Communication with external systems — including **MinIO**, **PostgreSQL**, and **Keycloak** — is handled directly within the backend through signed or credential-based requests. For example, when a user uploads a file, the backend generates a presigned URL, which allows the frontend to upload directly to MinIO without routing the file through the server.

Similarly, during patient creation, the backend handles the generation of a UUIDv4 and a CoMPaSS Code and links these identifiers to metadata stored in the PostgreSQL database.

What follows is a complete listing of the available API routes. Each entry includes a short description of the route's purpose, how it operates, and how it supports broader platform functionality:

Authentication & Session Management

- `POST /api/login:`

This route handles user login by forwarding credentials to the Keycloak authentication server. It sends a username and password to the `/token` endpoint of the configured Keycloak realm. If the credentials are valid, the response includes an access token, refresh token, and other session metadata. This token is then used to authorise further API calls.

- `POST /api/logout:`

This route revokes the current session by sending a logout request to the Keycloak `/logout` endpoint. It requires both the access token (for authorisation) and the refresh token (to identify the session to terminate). This ensures that tokens cannot be reused after logout and invalidates the user session on the Keycloak server.

- `GET /api/isTokenValid:`

This route validates whether an incoming bearer token is still active. It does so by sending the token to Keycloak's `/token/introspect` endpoint. If the token is active and valid, the API responds with a 200 OK and a simple confirmation. If not, it returns 403 Forbidden or 401 Unauthorized, depending on the case. This route is primarily used by the frontend to check session status before performing user-level actions.

Patient Management

- `GET /api/getAllPatients:`

This route retrieves all registered patient records from the PostgreSQL `patients` table. Upon connection, it initiates a transaction and performs a `SELECT` query on all rows of table `patients`. The route finalises the transaction and returns the full list of pseudonymised patient metadata.

- `POST /api/postPatient:`

This route creates a new patient record. Upon validation, the route generates a pseudonymised CoMPaSS Code derived from the clinical centre, current year, and a counter that ensures unique indexing. Simultaneously, a UUIDv4 is created to serve as the internal system identifier. The patient counter is retrieved and incremented within the counter table to preserve uniqueness across submissions. With these identifiers in place, a new patient entry is inserted into the `patients` table alongside demographic fields.

Clinical Data

- **POST** /api/postClinicalData:

This route handles the submission of a new clinical form for a specific patient. It verifies the patient's existence in the database and calculates a unique CoMPaSS Code for the clinical form based on the clinical centre, year, patient code, and current clinical form count. A UUIDv4 is generated for the clinical_form_id, and the clinical parameters, structured in nested sections, are flattened and inserted into the clinical_form table. The patient's record is then updated to include the new clinical form ID.

- **GET** /api/getClinicalData:

This route retrieves the full content of a clinical form given its clinical_form_id. It queries the clinical_form table and matches the result against the expected structure defined in the platform's clinical schema (ClinicalDataModel.json). The output JSON reflects all standard parameters, even if some values are not present in the database, in which case they are returned as null.

Genetic Data

- **GET** /api/getGeneticPresignedUrl:

This route generates a presigned URL that allows direct upload of genetic data files to the MinIO object storage, bypassing the backend for large file transfer efficiency. It retrieves patient details, including clinical centre and form count, from the database to construct a unique CoMPaSS Code for the upload. A new UUIDv4 is generated for the file, and the appropriate file extension is maintained. The presigned URL is valid for a limited time and is returned along with the UUID and code to the client for immediate use in uploading the file.

- **POST** /api/postGeneticMetadata:

This route records metadata about a genetic data file that was uploaded to MinIO. It dynamically parses the data object provided in the request, extracting key-value pairs and inserting them into the Genetic_data table. It then updates the patients table by appending the new Genetic_data_id to the array of genetic records linked to that patient. This maintains a complete reference trail between patient records and uploaded genetic data.

- **GET** /api/getGeneticData:

This route retrieves all genetic data records associated with a specific patient. It performs a query on the Genetic_data table using the patient's UUID and returns the results as an array. If no records are found, a 404 response is returned, indicating that no data has yet been uploaded for that patient.

MRI Data

- **GET** /api/getMRIPresignedUrl:

This route prepares a multipart upload process for large MRI files using MinIO. It begins by retrieving patient details from the database and uses that information to generate a unique CoMPaSS Code and file identifier. A multipart upload session is initiated in MinIO, and the route calculates the number of upload chunks required based on the file size. It then generates a list of presigned URLs, one for each chunk, allowing the frontend to upload the file in parallel segments. The session metadata (upload ID, file ID, object key, and presigned chunk URLs) is returned to the client for use during the upload process.

- **POST** /api/postMRIMetadata:

This route records the metadata associated with an MRI file that has been uploaded to MinIO. It dynamically parses the fields provided in the request body and inserts them into the MRI table. The associated patient's record in the patients table is then updated by appending the new MRI entry's UUID to the MRI_id array. This maintains a referential link between patient profiles and their corresponding imaging data.

- **GET** /api/getMRI:

This route retrieves all MRI metadata entries associated with a specific patient UUID. It queries the MRI table for all records where Patient_id matches the given identifier and returns the resulting list as a JSON response. If no records are found, a 404 message is returned, indicating that the patient does not yet have MRI data registered in the system.

Histopathological Data & Evaluation

- **GET** /api/getHistopathologicalPresignedUrl:

This route initiates a multipart upload session for histopathological data in MinIO, generating presigned URLs for both the main file and a preview image. It retrieves patient metadata, calculates a unique CoMPaSS Code and UUID for the upload, and prepares the preview path within a designated Preview/ subdirectory. Based on the file size, it determines how many upload chunks are needed and issues a presigned URL for each. The route returns all necessary identifiers and upload instructions to the frontend.

- **POST** /api/postHistopathologicalMetadata:

This route saves the metadata associated with a histopathological data file uploaded to object storage. It parses key-value pairs from the request and inserts them into the Histopathological_data table. It then updates the associated patient's record to include the new Histopathological_data_id.

- **GET** /api/getHistopathologicalData:

This route retrieves all metadata entries from the Histopathological_data table that are linked to the specified patient. If found, the records are returned as an array. If not, a 404 response is sent to indicate no histopathological data has been uploaded for that individual.

- `GET /api/getHistopathologicalPreviewImage:`

This route fetches the preview image of a histopathological data sample stored in MinIO. It constructs the object path in the Preview/ directory and attempts to retrieve the file. If successful, it streams the binary image data directly to the response with appropriate MIME headers. If the object is missing, it responds with a 404 error.

- `POST /api/postHistopathologicalEvaluation:`

This route handles the submission of evaluation data for a specific histopathological sample. It starts by retrieving the sample's metadata from the Histopathological_data table and determines the next available evaluation index. A unique CoMPaSS Code and UUID are generated for the evaluation, and all form data is inserted into the Histopathological_evaluation table. The associated data entry is also updated to include a reference to the new evaluation record.

- `GET /api/getHistopathologicalEvaluation:`

This route retrieves a specific histopathological evaluation by ID. It queries the Histopathological_evaluation table and returns the matching row if found. If no match exists, it sends a 404 response indicating that the evaluation could not be located.

Generic Upload Handling

- `GET /api/getAllSourceData:`

This route consolidates metadata from all prospective data types MRI, Genetic, Histopathological, and Clinical into a unified response. It queries each corresponding database table. The final combined array is returned to support the platform's source data overview interface.

- `POST /api/completeMultipartUpload:`

This route finalizes a multipart upload session in MinIO. It receives the object type (bucket), upload ID, object key, and a list of part identifiers and ETags as input. These are passed to MinIO's completeMultipartUpload function to assemble the full file from its individual uploaded chunks. Upon successful completion, the backend confirms upload success to the client. If any part fails, an appropriate error is returned to allow retry or fallback handling.

All routes in the CoMPaSS-NMD backend apply consistent security and data integrity practices. Before processing, each route validates the requester's token using Keycloak's /token/introspect endpoint to ensure authentication and authorisation. Additionally, operations involving data persistence or modification are wrapped in PostgreSQL transactions. This guarantees **atomicity** and enables safe **rollback** on failure, preserving database consistency across all actions.

7.2. Frontend Framework and Tools

The frontend of the CoMPaSS-NMD Atlas platform is developed as a multi-page web application using **React.js**, structured to support modular growth and easy integration of new

views and functionalities. Each screen corresponds to a specific task (e.g., Patient Creation, Upload, Overview) and is built as a separate React component, encapsulating both UI and logic.

Styling is managed using **Tailwind CSS**, a utility-first framework that allows fine-grained control of layout and responsiveness while keeping the styling consistent and performant. UI widgets and interactive components (modals, dropdowns, input groups) are based on the **Hero UI** React library, which provides accessible, pre-built components aligned with Tailwind.

Axios is used as the HTTP client for making RESTful API requests to the backend. Requests include bearer tokens managed via browser session and are routed securely to backend endpoints for data retrieval, upload orchestration, and metadata posting.

Routing between views is handled via React Router, allowing smooth page-to-page transitions, parameter passing (e.g., `patient_id`), and preservation of navigation history. Internal application state (e.g., active session info or temporary form data) is managed through React state hooks, with some large-form session data (such as clinical form drafts) preserved in the **browser's session storage**, enabling form completion across multiple user sessions.

Input validation and error feedback are handled client-side for user responsiveness but mirrored on the backend for full data integrity. Views are modular, and state changes in one part of the application (e.g., patient registration) are propagated to dependent views (e.g., upload dashboard) through reactive data fetching.

All assets and logic are bundled via **Vite**, enabling fast hot-reload during development and optimized production builds.

This architecture supports **clear separation of concerns**, making the platform extensible (e.g., adding HPO-based filtering or patient stratification views) while maintaining a consistent interaction model and UI behaviour.

7.3. Data Storage and Databases

The CoMPaSS-NMD Atlas platform separates structured and unstructured data storage using two dedicated systems: PostgreSQL for metadata and tabular clinical data, and MinIO for large binary files and imaging datasets. This separation ensures optimal performance, scalability, and ease of data management across different modalities.

Structured data including patient identifiers, clinical forms, metadata descriptors, and links to uploaded files are stored in a PostgreSQL relational database. All prospective data entries (MRI, genetic, histopathological, clinical) are referenced through a pseudonymised CoMPaSS ID and linked to metadata fields in their respective tables. UUIDv4 identifiers are used throughout to enforce uniqueness and interoperability with external systems. Transactional integrity and referential consistency are ensured through foreign key constraints and committed SQL transactions.

Unstructured data such as DICOM files, compressed genomic results, histopathological images, and preview snapshots are stored in MinIO, an S3-compatible object storage service hosted on SUT infrastructure. Each file is associated with a unique objectCode and stored within a domain-specific bucket (mri, genetic, histopathological). Metadata describing the file (e.g., upload date, type, size, associated patient) is maintained separately in PostgreSQL and linked via identifiers.

Data ingestion is orchestrated through presigned URL flows. For performance and scalability, large files are uploaded directly from the frontend to MinIO using URLs generated by the backend. Multipart upload support is enabled to handle files exceeding standard HTTP limits, such as large imaging series or genomic data sets. Once upload is complete, metadata is submitted via a separate API call to register the object in the database and associate it with the correct patient.

While this section focuses on the technical stack used for storage and access, further discussion on data modeling, normalization, and metadata schema design is presented in Chapter 8 – Data Model and Metadata Schema.

Chapter 10.1 - Data Protection Measures will expand on backup measures in place for the Data storage hosted on SUT's premises.

8. DATA MODEL AND METADATA SCHEMA

The CoMPaSS-NMD Atlas platform relies on a relational PostgreSQL schema designed for scalability, modularity, and data consistency. The schema integrates structured metadata from clinical forms, imaging, histopathological, and genetic datasets, all traceable to pseudonymised patient records.

Each patient is identified by a UUIDv4 and linked to one or more data entries in domain-specific tables. Relationships are explicitly enforced using foreign key references to ensure data integrity across modalities. All collected information remains pseudonymised, and no personally identifiable information is stored in the database.

The full DBML (Database Markup Language) source code used to define and generate the entity-relationship diagram is included in Annex 13.1 of this document. This code can be imported directly into dbdiagram.io for interactive exploration, editing, or further development.

The schema supports both structured constraints (via enums and typed fields) and future extensibility (via open-ended fields), providing a robust yet flexible foundation for handling prospective biomedical data.

8.1. Entity-Relationship Overview

The core of the schema is the patients table, which houses the UUID, CoMPaSS pseudonymised code, clinical center, sex, and other demographic metadata. This table links to four main modality-specific tables:

- `clinical_form` for structured clinical parameters

- MRI for imaging metadata
- Genetic_data for NGS file descriptors
- Histopathological_data and Histopathological_evaluation for tissue sample metadata and qualitative analysis

Each of these tables references patient_id as a foreign key, enabling full traceability across data modalities. Additional utility tables such as counters are used to support CoMPaSS code generation logic by tracking yearly increments per clinical site.

One of the standout features of this data model is its extensive use of PostgreSQL enum types across the schema. These enumerations define fixed allowed values for many clinical fields, ranging from gait type to biopsy method, disease subtypes, hormone disorders, imaging modalities, and more. This approach ensures consistent terminology and controlled vocabularies across all submitted forms.

For example:

- The sex field is constrained to Male or Female.
- Clinical parameters like Diagnosis, Electromyographic pattern, and Use of wheelchair follow structured enum types aligned with domain-relevant medical taxonomies.
- Imaging and file upload types (e.g., .nii, .dcm, .vcf, .fastq.gz) are validated using predefined enum constraints in their respective metadata tables.

However, to support long-term platform adaptability and unforeseen clinical scenarios, the model also includes flexible fields where appropriate:

- Free-text arrays (e.g., Other_diseases_type, Diagnosis_of_cancer_type) allow nuanced user inputs.
- Numerical ranges (integer, doubleprecision) are used where values vary between patients.

Together, this hybrid design of strict typing and soft expansion enables the platform to enforce data consistency without compromising on extensibility or project growth potential.

8.2. Data Normalisation and Standards

The data schema used in the CoMPaSS-NMD Atlas platform follows a normalised relational structure. Different types of data such as patient profiles, clinical forms, MRI scans, histopathological evaluations, and genetic information are separated into dedicated tables and linked through UUIDv4 identifiers. This allows for data consistency and traceability across modules.

Field names, structures, and accepted values were developed through a series of bilateral and co-design meetings with clinical and technical partners. These discussions ensured alignment with real clinical workflows and data entry practices. The final model also reflects the output of standard operating procedures (SOPs) defined in earlier project deliverables.

To enforce consistency, the schema uses enumerated types (enums) for many fields, especially within the clinical data table. Examples include controlled values like "Yes/No",

"Right/Left", "Normal/Abnormal", and other project-specific classifications. This approach helps ensure standardised input, supports filtering, and simplifies analysis.

At the same time, other fields particularly those intended to capture open-ended clinical notes or evolving medical observations are kept as free text, numeric values, or string arrays. This flexibility supports future updates and platform evolution, without requiring structural changes.

Date fields follow the ISO¹⁹ format (YYYY-MM-DD), and all references across tables use UUIDs to ensure unique and consistent linking of records.

The schema is designed to accommodate later integration with ontology-based tools (e.g., HPO) and external analysis modules.

8.3. Scalability Considerations

The CoMPaSS-NMD Atlas platform has been developed with scalability in mind, ensuring it can support the growing volume, variety, and complexity of data that will emerge over the course of the project and beyond its formal duration.

From a data model perspective, the relational schema is designed to be extensible. UUID-based primary keys ensure unique identification across distributed systems and simplify cross-referencing when adding new data types. New tables or columns can be introduced with minimal disruption due to the modular nature of the schema and the use of foreign-key references.

For clinical data in particular, the platform uses enumerated fields to enforce structure and consistency. However, the model also incorporates free-text and flexible string/array fields, allowing clinical centres to capture additional information that might not yet be standardised. This hybrid approach supports both structured data collection and exploratory research needs.

On the data ingestion side, the use of MinIO as a scalable, S3-compatible object storage system enables the platform to manage large volumes of non-tabular files such as MRI scans, genetic data, and high-resolution histopathological images. The presigned URL mechanism used during uploads shifts the load from the backend server to direct client-to-storage communication, reducing bottlenecks and enabling concurrent uploads.

The PostgreSQL database is hosted on SUT-managed infrastructure with the option for horizontal scaling and replication should demand increase. Tables are indexed appropriately for the most common queries (e.g. filtering patients by clinical centre, sex, or source data availability).

The separation of responsibilities across the frontend, backend, storage, and authentication services also contributes to scalability. Each component can be independently scaled or redeployed based on load, user activity, or evolving system requirements.

¹⁹ https://en.wikipedia.org/wiki/International_Organization_for_Standardization

Future scalability will also consider integration with other platforms or services (e.g., HPO indexing, AI clustering models, or external cohorts) by exposing well-defined APIs and maintaining a clean separation between internal schemas and external interfaces.

9. DEPLOYMENT ARCHITECTURE

The deployment strategy for the CoMPaSS-NMD Atlas platform ensures a clear separation between development, testing, and production environments. This modular and staged deployment approach allows for robust validation, early partner feedback, and risk mitigation before data reaches the production system. Figure 16 highlights differences between the three environments used in the project.

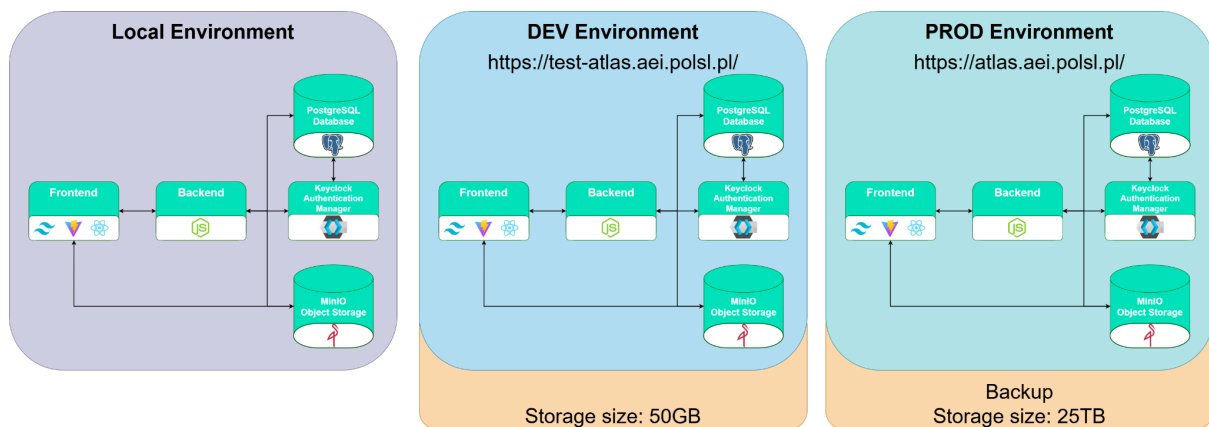


Figure 16: Development and Deployment Environments

Initial development and feature implementation take place in local environments operated by the WP4 technical team. Upon reaching a functional milestone, the fully dockerised platform components are pushed to a shared GitHub²⁰ repository. Dockerisation²¹ guarantees environment consistency across all stages of deployment and simplifies orchestration of services.

From GitHub, the updated containers are deployed to the Development Virtual Machine (DEV VM) hosted on SUT premises. This DEV VM mirrors the production environment and is used for:

- Debugging integration issues across services (MinIO, PostgreSQL, Keycloak)
- Validating upload workflows, authentication, and metadata handling
- Partner feedback collection during active testing phases

This intermediate step ensures any bugs or interface improvements are resolved before public deployment.

Once validated on the DEV VM, the platform is deployed to the Production Virtual Machine (PROD VM) — also hosted on SUT premises. This instance is used by CoMPaSS-NMD clinical partners for uploading and reviewing pseudonymised patient data in live conditions.

²⁰ <https://github.com/>

²¹ <https://www.docker.com/>

The PROD environment runs the same containerised architecture as the DEV VM, including:

- React-based multi-screen frontend built with Tailwind and HeroUI
- Node.js backend serving as the core logic and API gateway
- PostgreSQL database for structured metadata
- MinIO object storage system for large file uploads (MRI, genetic, histopathology)
- Keycloak-based authentication service

Backup procedures are regularly executed, and system monitoring ensures uptime and performance. By isolating development from production, the project ensures stable operation while enabling a responsive feedback-driven development cycle.

As shown in Figure 16, the DEV environment is publicly deployed at <https://test-atlas.aei.polsl.pl/>. It includes a 50 GB MinIO storage space and is accessible using test credentials provided to selected evaluators. The PROD environment, deployed at <https://atlas.aei.polsl.pl/>, is used for real data collection within the consortium. It includes a 25 TB storage allocation and is accessible only to authorised partners via secure credentials. Backup measures and other protections in place will be described in detail in Chapter 10.1 – Data Protection Measures.

10. SECURITY, PRIVACY, AND COMPLIANCE

The CoMPaSS-NMD Atlas platform is engineered to handle sensitive biomedical data in accordance with current security and privacy standards, including the General Data Protection Regulation (GDPR). The system architecture, data management workflows, and user access procedures are all designed to support secure, auditable, and privacy-respecting operations across development, testing, and production environments.

10.1. Data Protection Measures

The CoMPaSS-NMD Atlas platform applies a multi-layered approach to data protection, ensuring the secure handling of sensitive biomedical data in both development and production environments:

Encrypted Data Transfer: All communications between client, backend, PostgreSQL, and MinIO services occur over encrypted HTTPS connections secured with SSL/TLS²² protocols.

Authenticated Access: Access to the platform is gated by token-based authentication managed via Keycloak. API endpoints and UI components are only accessible to validated users. Roles have been defined and roles management is also done.

Presigned URL Uploads: To reduce backend load and improve scalability, file uploads (e.g., MRI, histopathological, genetic) are performed directly from the frontend to MinIO via presigned URLs, minimising intermediate handling.

²² https://en.wikipedia.org/wiki/Transport_Layer_Security

Transactional Database Writes: All data insertions and updates to the PostgreSQL database are conducted through transactional queries, with rollback capabilities in case of error to preserve consistency.

Environment Isolation: The platform's DEV and PROD environments are entirely separated to avoid cross-contamination. DEV is used for testing and validation, while PROD is the live instance used by consortium partners to upload prospective patient data.

Automated Backup Strategy: The PROD environment (<https://atlas.aei.polsl.pl/>) includes an incremental, versioned backup system based on the **restic** tool. Snapshots of both the PostgreSQL database and MinIO object storage are automatically generated **daily** via cron jobs. Backup retention follows the policy to keep the last 7 daily snapshots, to keep 4 weekly snapshots (1 per week), to keep 6 monthly snapshots (1 per month) and automatically prune older snapshots and remove unused data. Backup status is auditable through restic script or cron job logging.

These measures ensure data availability and traceability while supporting future disaster recovery and analysis if needed.

10.2. GDPR Compliance Strategy

The CoMPaSS-NMD Atlas platform is designed and implemented in alignment with the core principles of the GDPR, key compliance considerations include:

Pseudonymisation: All patient data is pseudonymised at the point of entry. Each patient is assigned a UUIDv4-based CoMPaSS ID, along with a human-readable CoMPaSS Code. No identifiable personal data (e.g., names, addresses, national IDs, DNA information) is stored or processed by the platform. The codification of the data allows the application of the rights of the subjects (access, rectification, erasure, restriction of processing, data portability, object to individual decision-making, ...) in case that any request is received.

Data Residency and Control: Original identities and re-identification mappings are retained solely at the originating clinical centres, in accordance with their local data protection protocols.

Consent and Data Ownership: Data entry into the Atlas platform is governed by the clinical centres, which are responsible for the lawfulness of processing, including obtaining proper patient consent and ensuring ethical use of data under their jurisdiction.

Access Control: Platform access is restricted to authorised users via token-based authentication managed by Keycloak. Role-based access control will be introduced to further refine data access based on professional responsibilities.

Data Processing Transparency: All user actions involving patient data (e.g., uploads, edits) are logged internally for traceability and audit purposes.

Backups: Periodic backups will be carried out with the aim of ensuring the integrity and availability of the information.

Physical safety: The data will be stored in local servers which are in a video surveillance room and with access control, to avoid unauthorized access to the data and the integrity of the information.

10.3. Secure Access and Authentication

Access to the CoMPaSS-NMD Atlas platform is protected through an integrated authentication service using Keycloak, which enforces token-based login across all components. All API routes and frontend operations require valid bearer tokens for interaction, ensuring that only authorised users can perform operations.

The system currently supports secure authentication for all consortium users, and its design includes support for future role-based access control, allowing more granular permission management as needed.

Credentials are issued individually to project partners, and session validity is verified with each request. All authentication operations are conducted over secure HTTPS channels.

As previously described, authentication logs and access attempts are monitored for security compliance and auditing. In case that a security breach is identified, it will be notified to the corresponding authorities.

11. PLANNED FEATURES AND ROADMAP

This chapter provides an overview of the current development status of the CoMPaSS-NMD Atlas Platform as of Month 26 (M26) and outlines the upcoming features planned for the final phase of the project. The development roadmap has been guided by clinical and technical requirements gathered through iterative co-design workshops, feedback from pilot testing on the DEV environment, and alignment with project objectives. Emphasis has been placed on ensuring that implemented features are stable, secure, and usable across the consortium, while future developments aim to extend platform capabilities, improve data accessibility, and enhance patient stratification functionalities.

11.1. Requirements Update (M26 Status)

This section presents an updated and extended version of the functional requirements originally outlined in Deliverable D4.1 – Requirement Analysis. The table presented below (Table 01) originates from the structure of the requirements matrix in D4.1 but has been refined, expanded, and adapted to reflect the evolving needs and feedback collected throughout the project.

Importantly, several new functional priorities have emerged as a result of the continuous co-design process involving bilateral meetings with clinical partners, iterative development cycles, and practical usage of the DEV environment. As a result, some requirements were clarified, others expanded, and additional functionalities were introduced that were not initially foreseen but proved essential in supporting real-world workflows.

Table 01 therefore serves both as a retrospective validation of completed requirements and a reflection of how the platform's feature set has matured up to Month 26 of the

CoMPaSS-NMD project. The content of the table aligns with the current implementation status of the production platform used to collect prospective data.

The fields assigned to each requirement are the following:

- **Requirement Code**
Identifier for each feature, composed by an incrementing integer.
- **Requirement**
Indicates the name of the specific requirement of the CoMPaSS-NMD Atlas platform.
- **Description**
Description of the requirement to give the reader an overview of its specifics.
- **Priority**
Indicates the prioritisation of the requirement (MUST, SHOULD, COULD)
- **Implementation Status**
Indicates the status of the requirement (✔ Completed, ✘ Not Implemented, ● Partially Available)

Requirement Code	Requirement	Description	Priority	Implementation Status
R01	Login Authentication	Token-based login with Keycloak	MUST	✓ Completed
R02	Patient Creation	UI for creating patients and generating Compass Code	MUST	✓ Completed
R03	Bar Code Generation	Possibility to generate, visualise and download barcode to identify patients, or source data	SHOULD	✓ Completed
R04	Clinical Data Upload (UI)	Clinical Form with more than 200 structured fields	MUST	✓ Completed
R05	Partial session storage for Clinical Form	Clinical Form Data are saved in Session storage during compilation so that it's possible to continue after a delay or in a second moment	SHOULD	✓ Completed
R06	Clinical Data Upload (File)	Upload of structured form via file	COULD	✗ Not Implemented
R07	MRI Data Upload	Multipart upload to MinIO with metadata registration	MUST	✓ Completed
R08	MRI Metadata Parsing	Parsing of DICOM headers for metadata fields	SHOULD	✓ Completed
R09	Histopathological Data Upload	Upload of primary and preview file	MUST	✓ Completed
R10	Histopathological Evaluation Upload	Upload of subjective form for sample assessment	SHOULD	✓ Completed

Requirement Code	Requirement	Description	Priority	Implementation Status
R11	Genetic Data Upload (to MinIO)	Upload directly to MinIO Object Storage through user authentication for CeGaT	MUST	✓ Completed
R12	Genetic Data Upload (from UI)	Upload with pre-signed URL from UI	SHOULD	● Partially Available
R13	Family Tree Chart Upload	Patient's family tree chart upload	SHOULD	✗ Not Implemented
R14	Family Tree Chart View	Patient's family tree chart visualisation	SHOULD	✗ Not Implemented
R15	Clinical Data View	Visualisation of submitted clinical forms	MUST	✓ Completed
R16	MRI Data View	Visualisation of metadata related to submitted MRI prospective data	MUST	✓ Completed
R17	Histopathological Data View	Visualisation of metadata and preview image for collected histopathological samples	MUST	✓ Completed
R18	Genetic Data View	Visualisation of metadata associated to submitted Genetic prospective data	MUST	● Partially Available
R19	Patient Overview UI	Data collection screen showing patient metadata and file counters	MUST	✓ Completed
R20	Source Overview UI Data	Table of uploaded data entries per type	MUST	✓ Completed

Requirement Code	Requirement	Description	Priority	Implementation Status
R21	Search Filtering and	Basic Search by CoMPaSS Code and filter by Sex or Clinical center of reference	MUST	✓ Completed
R22	Advanced Search and Filtering	Search and Filter based on all metadata or HPO terms	MUST	✗ Not Implemented
R23	Export Features	Possibility of download or export source data	MUST	✗ Not Implemented
R24	Clustering Model Results	Integration of Clustering model developed by SUT	SHOULD	✗ Not Implemented
R25	Error & Feedback Handling	The platform inform the user when an upload fail, a form is incomplete or a session expires	SHOULD	✓ Completed
R26	DEV Deployment for testing	DEV instance at test-atlas.aei.polsl.pl	MUST	✓ Completed
R27	PROD Deployment	PROD instance with 25TB for collecting prospective data at https://atlas.aei.polsl.pl/	MUST	✓ Completed
R28	Backup and Snapshotting	Daily restic backups with versioning	MUST	✓ Completed
R29	Integration of HPO Terms	The platform support semantic tagging of clinical data using HPO terms	MUST	✗ Not Implemented

Requirement Code	Requirement	Description	Priority	Implementation Status
R30	Audit Logging	The platform maintains logs of user actions on sensitive operations for compliance	MUST	✓ Completed
R31	Role-based Access Control	Group-specific permissions in Keycloak	MUST	✗ Not Implemented
R32	Data Correction	Privileged technical users are able to correct/modify data incorrectly inserted in the CoMPaSS-NMD Atlas	COULD	✗ Not Implemented
R33	Security and Standard Upload	Prospective Data are uploaded following current WEB standards	MUST	✓ Completed

Table 01: Requirements Table 2.0

The features marked as **Completed** in Table 01 are fully implemented, tested, and currently accessible via the DEV environment and actively used in the PROD environment for real data collection by consortium partners. Features marked as **Partially Available** or **Not Implemented** are scheduled for development and refinement in the upcoming months. The goal is to ensure that all partially implemented features reach a functional state between **Month 30 (M30)** and **Month 36 (M36)**, enabling iterative testing and feedback through additional co-design rounds with clinical partners. Progress will be documented in future Deliverable D4.4 and Consortium reviews.

This iterative process will support the enhancement and validation of existing functionalities, ensuring the platform aligns with real-world workflows. The final project phase, leading up to **Month 48 (M48)**, will focus on the **exploitation** of the platform — including feature finalisation, interface stabilisation, and preparing for **integration with other biomedical platforms** and data ecosystems already active in the neuromuscular domain.

11.2. Feedback Loop from Co-Design Iterations

The development of the CoMPaSS-NMD Atlas platform has been guided by a **collaborative, iterative co-design approach** that actively involves clinical, technical, and research partners from across the consortium. This process ensured that the platform not only met functional requirements, but also reflected real-world clinical workflows, usability needs, and data handling practices.

Throughout the reporting period, feedback was continuously collected via bilateral meetings, co-design workshops, and asynchronous reviews of deployed features in the DEV environment. Partners were encouraged to test functionalities such as clinical data input, patient management, and data browsing interfaces, and their observations led to key adjustments for instance, the modularisation of clinical forms into subsections, the use of preview images for histopathological data, and the inclusion of DICOM header parsing for MRI uploads.

These feedback loops were not isolated to early development stages. They remain an **ongoing, embedded part of the development cycle**, with several requirements evolving directly as a result of partner needs. This is especially evident in how the requirements table (Table 01) has diverged from the version defined in Deliverable D4.1, incorporating newly emerged features and clarifying priorities based on practical partner feedback.

In the upcoming development phases (M26–M48), these feedback mechanisms will be reinforced through:

- Regularly scheduled usability testing sessions,
- Partner-driven feature validation exercises,
- Integration discussions with external systems and registries.

In particular the following is a list of officially done, scheduled and planned CO-Design meetings:

First Co-Design Meeting (Late 2024): Organised with the support of DBL, this initial session gathered structured feedback from internal clinical partners and members of the Scientific Advisory Board (SAB). Key usability insights and requirements emerged, especially around patient creation, data upload clarity, and visual presentation of records, leading to several foundational modifications in the user interface and clinical workflows.

Second Co-Design Meeting (Phased, 2025):

- *Phase I (Q1 2025, Strasbourg Plenary Meeting):* This in-person session featured broad participation from consortium partners and SAB members. It focused on validating implemented features and gathering feedback on the data visualisation layers and navigation between patient views.
- *Phase II (April 2025):* A targeted session with internal clinical partners aimed at consolidating all data upload workflows (clinical, histopathological, MRI, and genetic),

with live feedback sessions on upload usability, session persistence, and metadata input.

- *Phase III (Upcoming)*: Scheduled in the coming months, this session will again include SAB and clinical representatives and will focus on advanced search and filtering features especially the integration of HPO terms and the export pipeline for clinical and imaging data.

Third Co-Design Meeting (Early 2026): This future meeting will provide a comprehensive review of the platform's functionality in its near-final state and begin shaping discussions around post-project exploitation, system interoperability, and long-term sustainability. Additional co-design events may be organised as needed to support final refinement and partner alignment toward the project's closing milestones.

More information regarding the co-design workshops are available in D8.1 Communication plan and D8.2 Exploitation plan.

The feedback loop is thus central to maintaining a platform that is both **technically robust** and **clinically meaningful**, ensuring sustainability beyond the end of the project.

12. CONCLUSIONS AND NEXT STEPS

The current version of the CoMPaSS-NMD Atlas platform architecture reflects a solid foundation that integrates technical robustness with the functional demands emerging from real clinical workflows and co-design collaboration. The implemented features, now available for evaluation and use in both DEV and PROD environments, confirm the viability of the chosen architecture and its ability to support the multidimensional data collection required for patient stratification in neuromuscular diseases.

In the coming months, development will focus on completing partially implemented and pending requirements, particularly those enabling advanced search, semantic annotation (e.g., HPO integration), and data export functionalities. These features are targeted for completion by Month 36 (M36), ensuring that clinical partners can engage with the full spectrum of capabilities during the remaining co-design iterations.

Between M30 and M36, emphasis will be placed on collecting feedback from platform usage in real scenarios, refining existing workflows, and validating new functionalities. The final stretch of the project, from M36 to M48, will be dedicated to stabilisation, refinement, and preparation for exploitation. This includes strengthening links with external data platforms, improving interoperability, and ensuring the platform can support sustainable use beyond the project's lifespan.

The architectural strategy defined here will continue to guide platform evolution, ensuring that all future enhancements are grounded in clinical relevance, ethical compliance, and technological reliability.

13. ANNEXES

13.1. DBML for reproducing DBDiagram.io schema

```
// Use DBML to define your database structure
// Docs: https://dbml.dbdiagram.io/docs

Enum "ability_to_walk" {
  "Normal"
  "On tiptoes only"
  "On heels only"
  "Unable"
}
Enum "altered_normal_notperformed" {
  "Altered"
  "Normal"
  "Not performed"
}
Enum "clinical_center" {
  "Università degli studi di Modena e Reggio Emilia"
  "Fondazione Stella Maris"
  "Ludwig Maximillians University"
}
Enum "creatine_kinase" {
  "Normal range"
  "< 4x normal value (<1000 U/L)"
  "> 4x normal value (>1000 U/L)"
}
Enum "degree_range" {
  ">45° - 90°"
  "≤45°"
}
Enum "diagnosis" {
  "Suspected"
  "Definitive"
  "None"
}
Enum "electromyographic_pattern" {
  "Myopathic pattern"
  "Neurogenic pattern"
  "Normal"
  "Abnormal"
```

```

    "Not performed"
}
Enum "ethnicity" {
    "Caucasian"
    "Hispanic"
    "African-American"
    "Asian"
    "Others"
}
Enum "exom_genome_singlegene" {
    "Exom"
    "Genome"
    "Single gene analysis"
}
Enum "general_local_epidural" {
    "General"
    "Local"
    "Epidural"
}
Enum "genetic_data_extension" {
    ".vcf"
    ".vcf.gz"
    ".gz"
    ".zip"
    ".fastq"
    ".fq"
    ".fastq.gz"
    ".fq.gz"
}
Enum "genetic_data_type" {
    "vcf"
    "fastq"
}
Enum "histopathological_data_extension" {
    ".tiff"
    ".tif"
    ".zip"
    ".gz"
    ".md2"
    ".MD2"
    ".ndpi"
    ".jpeg"
}

```

```

    ".jpg"
    ".png"
}
Enum "histopathological_data_type" {
    "tiff"
    "BigTiff"
    "bigTiff"
    "md2"
    "MD2"
    "ndpi"
    "jpeg"
    "png"
    "zip"
}
Enum "hypothyroidism_hyperthyroidism" {
    "Hypothyroidism"
    "Hyperthyroidism"
}
Enum "inthepast_ongoing" {
    "In the past"
    "Ongoing"
}
Enum "left_right" {
    "Left"
    "Right"
}
Enum "mri_data_extension" {
    ".nii"
    ".zip"
    ".gz"
    ".dcm"
}
Enum "mri_data_type" {
    "dicom"
    "nifti"
}
Enum "neurogen_myositis_dystrophy" {
    "Neurogen"
    "Myositis"
    "Muscular dystrophy"
    "Myopathy"
}

```

```

Enum "none_worsening_improvement" {
  "None"
  "Worsening"
  "Improvement"
}
Enum "normal_abnormal" {
  "Normal"
  "Abnormal"
}
Enum "normal_abnormal_notperformed" {
  "Normal"
  "Abnormal"
  "Not performed"
}
Enum "normal_delayed" {
  "Normal"
  "Delayed"
}
Enum "normal_unable" {
  "Normal"
  "Unable"
}
Enum "normal_waddling_hyperlordotic_steppage" {
  "Normal"
  "Waddling"
  "Hyperlordotic"
  "Steppage"
}
Enum "notherapy_medicationotherthaninsulin_insulin" {
  "No Therapy"
  "Medication other than insulin"
  "Insulin"
}
Enum "onceperweek_2ormoreperweek" {
  "1 x per week"
  "2 or more per week"
}
Enum "pelvic_girdle_options" {
  "Without support"
  "Without support but abnormally"
  "With support"
  "Unable"
}

```

```

}
Enum "performed_notperformed" {
    "Performed"
    "not performed"
    "Not performed"
}
Enum "positive_negative" {
    "Positive"
    "Negative"
}
Enum "professional_amateur" {
    "Professional"
    "Amateur"
}
Enum "proximal_distal" {
    "Proximal"
    "Distal"
}
Enum "right_left_both" {
    "Right"
    "Left"
    "Both"
}
Enum "sex" {
    "Male"
    "Female"
}
Enum "substitute_notsubstitute" {
    "Substitute"
    "Not substitute"
}
Enum "symmetric_asymmetric" {
    "Symmetric"
    "Asymmetric"
}
Enum "thoracic_lumbar" {
    "Thoracic"
    "Lumbar"
}
Enum "typei_typeii" {
    "Type I"
    "Type II"
}

```

```

}
Enum "wheelchair" {
  "Not necessary"
  "With manual control"
  "With electric control"
  "Bed bound"
}
Enum "yes_no" {
  "Yes"
  "No"
}
Enum "yes_no_notevaluated" {
  "Yes"
  "No"
  "Not Evaluated"
  "Not evaluated"
}
Table "Genetic_data" {
  "Genetic_data_id" uuid [pk, not null]
  "Upload_date" date
  "Collection_date" date
  "Type" genetic_data_type
  "Extension" genetic_data_extension
  "Size" text
  "Source" text
  "Tissue" text
  "Library_strategy" text
  "NGS_kit" text
  "Sequencing_strategy" text
  "Genome_build" text
  "Phenotype" text
  "Clinical_center" text
  "Patient_id" uuid
  "Code" text
}
Table "Histopathological_data" {
  "Histopathological_data_id" uuid [pk, not null]
  "Evaluation_id" "uuid[]"
  "Upload_date" date
  "Collection_date" date
  "Type" histopathological_data_type
  "Extension" histopathological_data_extension
}

```

```

"Size" text
"Name_of_muscle" text
"Biopsy_method" text
"Frozen" text
"Formalin_fixed" text
"Paraffin_embedded" text
"Handling_artifacts" text
"Histochemical_stains" text
"Clinical_center" text
"PreviewImageExtension" text
"Patient_id" uuid
"Histopathological_preview_data_id" uuid
"Code" text
}
Table "Histopathological_evaluation" {
  "Histopathological_evaluation_id" uuid [pk, not null]
  "Ice_crystals" text
  "Fatty_infiltration" text
  "Endomysial_fibrosis" text
  "Myofiber_degeneration" text
  "Necrosis" text
  "Myophagocytosis" text
  "Basophilic_fibers_Large_nuclei" text
  "Hypertrophic_fibers" text
  "Atrophy_or_Hypotrophy" text
  "All_fibers_within_the_specimen" text
  "Subsets_of_fibers" text
  "Single_fibers" text
  "Groups_of_fibers" text
  "Perifascicular_distribution" text
  "Nuclear_bags_or_clumps" text
  "Atrophic_or_Hypotrophic_fiber_shape" text
  "Central_nuclei" text
  "Central_nuclei_estimated_percentage_of_fibers" text
  "Internal_nuclei" text
  "Internal_nuclei_estimated_percentage_of_fibers" text
  "Nemaline_rods_or_bodies" text
  "Red_incursions_trichrome" text
  "Rimmed_vacuoles" text
  "Round_vacuoles" text
  "Ragged_red_fibers" text
  "Central_cores" text

```

```

"Minicores" text
"Moth_eaten_fibers" text
"Tubular_aggregates" text
"Ring_fibers" text
"Split_fibers" text
"Lobulated_fibers" text
"Inflammation" text
"Intramuscular_nerve_branches" text
"Muscle_spindles" text
"Myotentinous_insertion_sites" text
"Atrophic_or_Hypotrophic_fiber" text
"Evaluation_date" date
"Clinical_center" text
"Histopathological_data_id" uuid
"Patient_id" uuid
"Code" text
}
Table "MRI" {
  "MRI_id" uuid [pk, not null]
  "Upload_date" date
  "Collection_date" date
  "Type" mri_data_type
  "Extension" mri_data_extension
  "Size" text
  "Modality" text
  "Scanning_sequence" text
  "Sequence_variant" text
  "Repetition_time" text
  "Echo_time" text
  "Flip_angle" text
  "Slice_thickness" text
  "Body_part_examined" text
  "Patient_position" text
  "Clinical_center" text
  "Patient_id" uuid
  "Code" text
  "Spacing_between_slices" text
}
Table "clinical_form" {
  "clinical_form_id" uuid [pk, not null]
  "Compilation_date" date
  "Handedness" left_right
}

```

"Weight" text
"Height" text
"BMI" text
"Family_ancestry_maternal" ethnicity
"Family_ancestry_paternal" ethnicity
"Consanguinity" yes_no
"Diabetes_mellitus" yes_no_notevaluated
"Diabetes_mellitus_type" typei_typeii
"Diabetes_mellitus_therapy" notherapy_medicationotherthaninsulin_insulin
"Thyroid_hormones_alterations" yes_no_notevaluated
"Thyroid_hormones_alterations_type" hypothyroidism_hyperthyroidism
"Thyroid_hormones_alterations_therapy" substitute_notsubstitute
"Diagnosis_of_cancer" yes_no
"Diagnosis_of_cancer_type" "text[]"
"Diagnosis_of_cancer_therapy" "text[]"
"Retinal_vasculopathy" yes_no_notevaluated
"Sensorineural_deafness" yes_no
"Audiometry" altered_normal_notperformed
"Epilepsy" yes_no
"Cognitive_impairment" yes_no
"Dementia" yes_no
"Other_diseases" yes_no
"Other_diseases_type" "text[]"
"Falls" yes_no
"number_of_falls_over_the_past_3_months" integer
"Previous_trauma" yes_no
"Joint_or_bone_fractures" yes_no
"Statins" yes_no
"Statins_duration_period" inthepast_ongoing
"Other_chronic_treatments" yes_no
"Other_chronic_treatments_duration_period" inthepast_ongoing
"Other_chronic_treatments_type" "text[]"
"Pregnancy" yes_no
"Spontaneous_abortion" yes_no
"Modification_of_the_disease_after_pregnancy" none_worsening_improvement
"Menopause" yes_no
"Hormonal_therapy" yes_no
"Regularly_played_sport" yes_no
"Physical_activity_type" professional_amateur
"Physical_age_from_age" integer
"Physical_age_to_age" integer
"Physiotherapy" yes_no

"Physiotherapy_frequency" onceperweek_2ormoreperweek
"Surgery" yes_no
"Anesthesia" general_local_epidural
"Modification_of_the_disease_after_the_surgery" none_worsening_improvement
"Prolonged_ICU_staying" yes_no
"Prolonged_weaning" yes_no
"Episode_of_malignant_hyperthermia" yes_no
"Relatives_wheelchair_bound" yes_no
"Relatives_swallowing_difficulties" yes_no
"Relatives_cardiac_disease" yes_no
"Relatives_cramps_or_myalgia" yes_no
"Relatives_epileptic_seizures" yes_no
"Relatives_mental_retardation_or_dementia" yes_no
"Relatives_autism" yes_no
"Relatives_eye_diseases_including_cataract" yes_no
"Age_of_onset_of_motor_impairment" integer
"Motor_mile_stones" normal_delayed
"Facial_muscles_weakness" yes_no
"Facial_muscles_weakness_side" right_left_both
"Shoulder_girdle_muscles_weakness" yes_no
"Shoulder_girdle_muscles_weakness_side" right_left_both
"Abdominal_muscle_weakness" yes_no
"Abdominal_muscle_weakness_side" right_left_both
"Pelvic_girdle_muscles_weakness" yes_no
"Pelvic_girdle_muscles_weakness_side" right_left_both
"Distal_lower_limb_muscles_weakness" yes_no
"Distal_lower_limb_muscles_weakness_side" right_left_both
"Distal_upper_limb_muscles_weakness" yes_no
"Distal_upper_limb_muscles_weakness_side" right_left_both
"Asymmetry_at_onset" yes_no
"Asymmetry_at_onset_side" right_left_both
"Triggering_events" yes_no
"Triggering_events_side" right_left_both
"Muscle_pain" yes_no
"Muscles_pain_side" "text[]"
"Precocious_muscle_fatigue_before_onset_of_muscle_impairment" yes_no
"widened_palpebral_fissures" yes_no
"Horizontal_smile" yes_no
"Dysarthria" yes_no
"Orbicularis_oculi_evaluation" normal_unable
"Ability_to_protrude_lips" normal_unable
"Ability_to_puff_cheeks" normal_unable

"Asymmetric_involvement_of_facial_muscles" yes_no
"Asymmetric_involvement_of_facial_muscles_side" left_right
"Asymmetric_ability_to_abduct_arms" yes_no
"Scapular_winging_at_rest" yes_no
"Scapular_winging_at_rest_type" symmetric_asymmetric
"Scapular_winging_at_rest_side" left_right
"Ability_to_climb_4_stairs" pelvic_girdle_options
"Inability_to_climb_4_stairs_since_age" integer
"Ability_to_walk" pelvic_girdle_options
"Inability_to_walk_since_age" integer
"Ability_to_stand_up_from_a_chair" pelvic_girdle_options
"Inability_to_stand_up_from_a_chair_since_age" integer
"Ability_to_rise_from_the_floor" pelvic_girdle_options
"Inability_to_rise_from_the_floor_since_age" integer
"Use_of_wheelchair" wheelchair
"Ability_to_walk_on_tiptoes_and_or_heels" ability_to_walk
"Beavor_sign" positive_negative
"Scoliosis" yes_no
"Scoliosis_type" thoracic_lumbar
"Hyperlordosis" yes_no
"Ridgid_spine_syndrome" yes_no
"Extrarotator_muscles_of_upper_limb_right_MRC_score" doubleprecision
"Extrarotator_muscles_of_upper_limb_left_MRC_score" doubleprecision
"Extrarotator_muscles_of_upper_limb_atrophy" text
"Triceps_brachii_right_MRC_score" doubleprecision
"Triceps_brachii_left_MRC_score" doubleprecision
"Triceps_brachii_atrophy" text
"Biceps_brachii_right_MRC_score" doubleprecision
"Biceps_brachii_left_MRC_score" doubleprecision
"Biceps_brachii_atrophy" text
"Common_finger_extensors_right_MRC_score" doubleprecision
"Common_finger_extensors_left_MRC_score" doubleprecision
"Common_finger_extensors_atrophy" text
"Wrist_extensors_right_MRC_score" doubleprecision
"Wrist_extensors_left_MRC_score" doubleprecision
"Wrist_extensors_atrophy" text
"Long_finger_flexors_right_MRC_score" doubleprecision
"Long_finger_flexors_left_MRC_score" doubleprecision
"Long_finger_flexors_atrophy" text
"Wrist_flexors_right_MRC_score" doubleprecision
"Wrist_flexors_left_MRC_score" doubleprecision
"Wrist_flexors_atrophy" text

"Gluteus_medius_right_MRC_score" doubleprecision
"Gluteus_medius_left_MRC_score" doubleprecision
"Gluteus_medius_atrophy" text
"Iliopsoas_right_MRC_score" doubleprecision
"Iliopsoas_left_MRC_score" doubleprecision
"Iliopsoas_atrophy" text
"Biceps_femoris_right_MRC_score" doubleprecision
"Biceps_femoris_left_MRC_score" doubleprecision
"Biceps_femoris_atrophy" text
"Quadriceps_femoris_right_MRC_score" doubleprecision
"Quadriceps_femoris_left_MRC_score" doubleprecision
"Quadriceps_femoris_atrophy" text
"Triceps_surae_right_MRC_score" doubleprecision
"Triceps_surae_left_MRC_score" doubleprecision
"Triceps_surae_atrophy" text
"Tibialis_anterior_right_MRC_score" doubleprecision
"Tibialis_anterior_left_MRC_score" doubleprecision
"Tibialis_anterior_atrophy" text
"Horizontal_clavicles" yes_no
"Axillary_creases" yes_no
"Sunken_or_flattened_appearance_of_the_chest" yes_no
"Poly_hill_sign" yes_no
"Myotonia" yes_no
"Rippling_muscles" yes_no
"Eyelid_ptosis" yes_no
"Eyelid_ptosis_type" symmetric_asymmetric
"Eyelid_ptosis_side" left_right
"Extra_ocular_weakness" yes_no
"Pharyngeal_and_lingual_muscle_weakness" yes_no
"Early_contractures" yes_no
"Early_contractures_type" "text[]"
"Dropped_head" yes_no
"Pes_cavus" yes_no
"Myoglobinuria" yes_no
"Ogival_palatus" yes_no
"Teeth_deformities" yes_no
"Value_of_two_blood_assays_separated_by_at_least_one_month" creatine_kinase
"Cardiac_involvement" yes_no
"Cardiac_involvement_type" "text[]"
"ECG" normal_abnormal_notperformed
"Echocardiogram" normal_abnormal_notperformed
"Electromyographic_pattern_of_four_limbs" electromyographic_pattern

"Electroneurography_of_four_limbs" normal_abnormal
"Report_of_last_pulmonary_function_tests" performed_notperformed
"FVC" normal_abnormal_notperformed
"MIP" normal_abnormal_notperformed
"MEP" normal_abnormal_notperformed
"P0_1" normal_abnormal_notperformed
"Muscle_biopsy" yes_no
"Muscle_biopsy_result" normal_abnormal
"Muscle_biopsy_type" neurogen_myositis_dystrophy
"Genetic_test_previously_performed" yes_no
"Genetic_test_previously_performed_type" exom_genome_singlegene
"Diagnosis" diagnosis
"Orphanet_code" integer
"clinical_center" clinical_center
"patient_id" uuid
"Code" text
"Dysphagia" yes_no
"Ability_to_abduct_arms" text
"Ability_to_abduct_arms_degree" text
"Gait" "text[]"
"Gluteus_maximus_right_MRC_score" doubleprecision
"Gluteus_maximus_left_MRC_score" doubleprecision
"Gluteus_maximus_atrophy" text
"Hip_adductors_right_MRC_score" doubleprecision
"Hip_adductors_left_MRC_score" doubleprecision
"Hip_adductors_atrophy" text
"Neck_extensors_right_MRC_score" doubleprecision
"Neck_extensors_left_MRC_score" doubleprecision
"Neck_extensors_atrophy" text
"Neck_flexors_right_MRC_score" doubleprecision
"Neck_flexors_left_MRC_score" doubleprecision
"Neck_flexors_atrophy" text
"Deltoid_right_MRC_score" doubleprecision
"Deltoid_left_MRC_score" doubleprecision
"Deltoid_atrophy" text
"Pectoralis_right_MRC_score" doubleprecision
"Pectoralis_left_MRC_score" doubleprecision
"Pectoralis_atrophy" text
"Deep_tendon_reflexes" text
"Electroneurography_of_four_limbs_type" text
"Electromyographic_pattern_of_four_limbs_side" "text[]"
"Electromyographic_pattern_of_four_limbs_type" "text[]"

```

}
Table "counter" {
  "clinical_center" clinical_center [not null]
  "year" smallint [not null]
  "patients" integer [not null, default: 0]
}
Table "patients" {
  "patient_id" uuid [pk, not null]
  "Genetic_data_id" "uuid[]"
  "Histopathological_data_id" "uuid[]"
  "MRI_id" "uuid[]"
  "clinical_form_id" "uuid[]"
  "Date_of_birth" date
  "Sex" sex
  "Clinical_center" clinical_center
  "Medic" text
  "Code" text
}
Ref: "Histopathological_data"."Histopathological_data_id" <
"patients"."Histopathological_data_id"
Ref: "clinical_form"."clinical_form_id" < "patients"."clinical_form_id"
Ref: "Histopathological_evaluation"."Histopathological_evaluation_id" <
"Histopathological_data"."Evaluation_id"
Ref: "Genetic_data"."Genetic_data_id" < "patients"."Genetic_data_id"
Ref: "MRI"."MRI_id" < "patients"."MRI_id"

```