Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.5424

Building flexible databases by using web services for computer-aided diagnosis of cardiomyopathies: from conceptual definition to usability evaluation

Larissa Terto Alvim [**University of São Paulo** | ***larissa_terto@usp.br***]
Vagner Mendonça Gonçalves [**University of São Paulo | Federal Institute of Education, Science and Technology of São Paulo** | ***vagner.goncalves@usp.br***]
Fátima L. S. Nunes [**University of São Paulo** | ***fatima.nunes@usp.br***]

Laboratory of Computer Applications for Health Care, School of Arts, Sciences, and Humanities, University of São Paulo, Rua Arlindo Béttio, 1000, Vila Guaraciaba, São Paulo, SP, 03828-000, Brazil.

Received: 14 January 2025 • **Accepted:** 27 October 2025 • **Published:** 20 March 2026

Abstract. Computer-aided diagnosis (CAD) systems based on medical images and records apply computational techniques to process data and extract features from them to provide a second opinion to the health professional. A diverse and organized set of images and records is necessary to develop and validate such systems. However, medical data are generally obtained in a non-standardized way. With each new research and development project in this area, specific data models need to be built to organize and standardize these data and enable their use in the construction of models and computational systems. This article presents a flexible and generic database modeled and implemented to persist Cardiac Magnetic Resonance exams aiming to support the development of CAD schemes of cardiomyopathies. Furthermore, a web application was developed to enable data search and retrieval from the database. An experiment was carried out to evaluate the interface usability of the web application. Results showed that it is possible to develop a generic and flexible DB model, which can be used in several CAD applications. Additionally, the implemented interface received positive evaluations on its functionalities and usability, and users were capable of performing the intended tasks with correct outcomes.

Keywords: Computer-Aided Diagnosis, Generic Database Model, Medical Records, Medical Exams, RESTful Web Services, User Interface Usability, Cardiac MRI

1 Introduction

Computer-aided Diagnosis (CAD) systems provide a second opinion to physicians about a diagnosis based on the multiple analyzed factors, mainly considering medical images as input data [Doi, 2007]. Current CAD systems seek to emulate the decision-making process of medical experts, with the most advanced ones even being able to analyze clinical data and infer new knowledge [Yanase and Triantaphyllou, 2019; Wang *et al*., 2024].

Building computational models applicable in CAD, in general, involves the use of large volumes of medical data and exams [Brzezicki *et al*., 2020; Rank *et al*., 2020; Zhou *et al*., 2021]. Furthermore, the conclusion of a diagnosis automated by CAD requires the system output to be compared to previous medical knowledge, which relies on the analysis and the establishment of relationships among a great volume of medical data [Doi, 2007]. These data and exams must be stored and organized in an efficient and intuitive Database (DB) to support information access and retrieval, and digital technologies can optimize data collection and storage processes in Healthcare through automated routines and output analysis and comparison [Doi, 2007].

In this context, it becomes relevant to build an effective and flexible data storage model and tools for information retrieval, manipulation, and visualization through a friendly and intuitive interface [Silva *et al*., 2020]. Besides that, data on demand at a single interface is the most required fact among physicians, which requires an adaptable interface management process that adjusts itself to the structure of the data to be retrieved [Malik *et al*., 2013].

The literature presents several works that present DB solutions for medical data management, such as the research of Nisanbayev *et al*. [2009], Fernandes *et al*. [2016], and García-Holgado *et al*. [2022]. Reusable interface models for these systems were also proposed, as in the work of Malik *et al*. [2013]. Although these previous studies propose datasets for supporting CAD and medical Information Systems (IS) development, they primarily focus on clinical applications and end-user interfaces for physicians.

In contrast, the target audience of the present proposal is CAD researchers, particularly computer science and information systems professionals who develop these systems. The professionals need to retrieve data with specific conditions for the evaluation of their developments. This audience requires flexible and adaptable data structures to facilitate data collection, integration, and experimentation during the system development.

Besides that, the literature does not present flexible structures that allow the interface to adapt itself according to changes in the DB structure, which is in the scope of the present work. The proposed system follows an Online Transaction Processing (OLTP) model, focusing on efficient DB transactions to support researchers in the development of CAD systems [Conn, 2005]. Unlike Online Analytical Processing (OLAP) systems, designed for complex analysis, our approach prioritizes real-time interaction with the database to facilitate data collection and manipulation [Conn, 2005].

This study presents the development and a usability evaluation of a functional DB of Cardiac Magnetic Resonance (CMR) exams based on a flexible data storage model to support the storage and retrieval of exams in the context of a CAD system for cardiomyopathies. A framework of RESTful web services for DB operations was developed to allow intuitive and accessible DB manipulation for researchers that develop CAD systems, as well as a web application and an adaptable user interface to enable the proposed approach. Finally, to evaluate the interface usability, an experiment was conducted among researchers and students that develop CAD systems, which are the target audience of the proposal presented here.

The research questions raised to evaluate the relevance of the proposed approach are:

1. Is it possible to build a generic data model for CAD that can be used in different applications?
2. Which strategies and technologies are adequate to persist and retrieve data in a flexible DB approach?
3. Is it possible to create an interface adaptable to DB structure changes to minimize the need for code maintenance?

The artifacts developed on this research project, and evaluated through the experiment discussed in this article, aim to provide the following contributions related to the questions previously presented:

1. a functional DB composed by CMR exams for supporting the development of CAD systems;
2. a web application based on RESTful web services for insertion, manipulation, and retrieval of the exams' data persisted in the DB;
3. a dynamically adaptable web application interface to retrieve and visualize the exams' data from the DB.

The remainder of this article is organized into the following sections: Section 2 presents the theoretical framework for the current research project; Section 3 presents previous work and research related to the current project; Section 4 presents the materials and methods used to develop the artifacts and the experiment; Section 5 discusses the current state of the developed artifacts and the experiment results; and, finally, Section 6 presents the conclusions.

2 Background

This section provides the theoretical references and the concepts related to the development of the proposed approach. Section 2.1 presents concepts about web services and databases. Section 2.2 presents the definition of cardiomyopathy, the most common categories of this condition, and highlights concepts about MRI. Section 2.3 summarizes usability and the System Usability Scale.

2.1 Web services

A web service is a standardized interface that allows interaction among applications implemented on different platforms and operating systems [Fensel and Bussler, 2002]. This concept was developed by the World Wide Web Consortium (W3C), which defines a web service as a software system that promotes interaction between two computers over a network [W3C, 1999].

Web services can be implemented using frameworks from different languages, including the RESTful Application Programming Interfaces (APIs) in Python. These APIs follow the constraints of the Representational State Transfer (REST) architecture with additional constraints, such as stateless communication and the use of hyperlinks to find actions after accessing a text or media [Arcuri, 2017].

This architecture was chosen for its scalability, since it allows data to be transferred among systems regardless of their implemented data representation; instead of the information itself, only a representation of its state is transmitted, and the systems can later persist this generic form in their native types of storage [Arcuri, 2017]. Besides that, this technology is aligned with the target users of this application: CAD developers, who are typically computer scientists familiar with data modeling and system integration. As such, this architecture facilitates flexible and intuitive implementation and experimentation, making it easier for users to manipulate and extend the system according to their research needs.

A RESTful service layer provides services for making network requests of create, retrieve, update, and delete (CRUD) operations in the resources persisted in a DB [W3C, 1999]. In the RESTful service layer, a resource is a primary abstraction of existing DB information, such as the structured data persisted in different DB tables [W3C, 1999].

2.2 Cardiomyopathies and Cardiac Magnetic Resonance

Cardiomyopathies are diseases characterized by the morphological abnormality of the myocardium, the muscular tissue that lines the heart [Robbins *et al*., 2010]. They often lead to cardiovascular death or disability related to progressive heart failure [Maron *et al*., 2006].

The present work focuses on the two most common types of cardiomyopathies: Dilated Cardiomyopathy (DCM) and Hypertrophic Cardiomyopathy (HCM) [Maron *et al*., 2006; Pirruccello *et al*., 2020; Ciarambino *et al*., 2021; Izquierdo *et al*., 2021]. DCM is characterized by the globose shape of the heart due to the dilation of one or more cardiac chambers, usually the left ventricle [Maisch *et al*., 2012]. On the other hand, HCM is identified by ventricular myocardial hypertrophy, which reduces the size of these chambers due to the great thickness of their walls [Maisch *et al*., 2012].

The diagnosis of cardiomyopathies makes use of cardiac imaging exams, such as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) [Bergamasco *et al*., 2018]. MRI, in particular, can differentiate tissues and display morphological and functional aspects of any part of the human body, enabling detailed visualization for identifying pathologies [Katti *et al*., 2011]. Figure 1 shows examples of

CMR of the left ventricle. In symptomatic cases, a normal left ventricle (Figure 1a) can be clearly differentiated from a dilated left ventricle due to DCM (Figure 1b) and from a hypertrophied myocardium due to HCM (Figure 1c).

2.3 Usability and System Usability Scale

Usability is the capacity of users to achieve their goals with effectiveness, efficiency, and satisfaction when using a system' interface, minimizing the risk of errors and unwanted results in that specific context of use [ISO, 2018].

The System Usability Scale (SUS) is an evaluation method for the usability of systems and applications [Klug, 2017]. The SUS is an instrument composed of questions regarding different dimensions of the software under evaluation, which helps define a system's usability: complexity, integration of functions, consistency, ease of learning, and intention to use [Klug, 2017]. In this study, the SUS was chosen to evaluate whether the application meets the practical needs of its intended users (CAD researchers) rather than to assess architectural or technical performance. This focus aligns with the goal of ensuring usability and accessibility during their research development phase, where flexible and intuitive interaction with the system is essential.

The questions of SUS usability evaluation are customizable according to the system's context and formulated as affirmations that target the system's usability and complexity [Lewis and Sauro, 2009]. The SUS answers' format is a Likert scale that associates five possible answers to the statements, indicating the user's positive-to-negative strength of agreement with them [McLeod, 2023]. These answers are scored from 1 to 5: a lower value on the scale means strong disagreement, and a higher value means strong agreement [McLeod, 2023].

3 Related work

Health Informatics applications to collect, organize, and analyze medical information are constantly evaluated among physicians and patients in order to validate their contributions to the medical field. With focus on these applications and their evaluation, we divided the studies into two categories for this literature review: Data platforms for Electronic Health Records (EHR) management and analysis, and treatment and rehabilitation-oriented applications (apps) and serious games.

In the first category, the target users are physicians and medical institutions. The current research project fits in this category, as it provides a database application for organizing and retrieving medical data as part of a CAD system to process and analyze medical images and records. However, this study aims to aid CAD researchers during the development phase instead of health professionals. The second category encompasses studies that present approaches commonly used by patients guided by specialists to serve as an auxiliary method of monitoring, since they can collect relevant data on a patient's progress. Similarly to the current project, these applications go through evaluations with the target users to validate the system's suitability and utility for their needs. Our approach is targeted at CAD developers, as previously mentioned; therefore, they were the users to evaluate the usability of the platform.

In this category, the work of Nisanbayev *et al*. [2009] develops an EHR system to assist physicians in managing medical records. The system is modeled using a reusable design pattern, similar to the current research project's proposal of a reusable DB model. The work of Malik *et al*. [2013] also proposes a reusable model for EHR systems, but with a focus on the interface, which should integrate different resources.

The studies of Fernandes *et al*. [2016] and García-Holgado *et al*. [2022] also present strategies in line with the current research project. The proposal of Fernandes *et al*. [2016] presents an IS for health data management, aiming to assist in organizing and retrieving social and health care information from Brazilian public health services. This work uses web services to collect stored data and send it to be exhibited in other applications. García-Holgado *et al*. [2022] presents the development and testing of a platform to assist in medical data analysis, including an experiment to validate and test the application on a set of defined use cases.

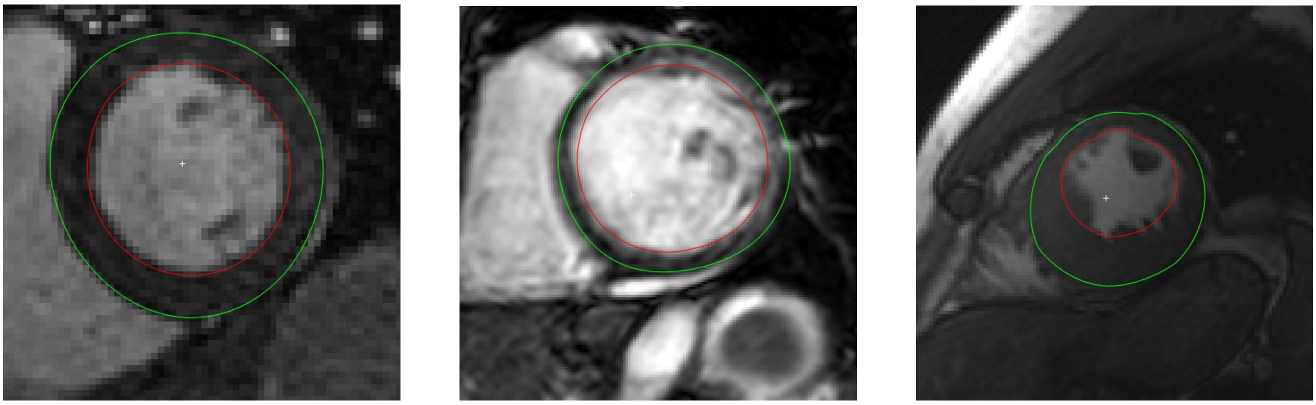
Differently from the presented studies, this article presents flexible structures for both DB and the interface, which allow the dynamic adaptation of the data organization, management, and retrieval tools according to the information available on the dataset.

Regarding the development and evaluation of Health Informatics applications of the second category – and similarly to the current research project –, the work of Cruz *et al*. [2022] assesses an adaptive app for cognitively compromised patients with questionnaires scored with the SUS scale. The studies of Assis *et al*. [2019], Rodrigues *et al*. [2023], and Santos *et al*. [2023] also evaluate their apps with target users using different metrics, such as Pearson's correlation scores of the brain activity, pre- and post-use usability evaluations, heuristics for mHealth apps (HE4EH), and immersion scales, respectively.

Still in this category, the studies of Anjos *et al*. [2024] and Funabashi *et al*. [2018] developed and evaluated serious games for medical monitoring and rehabilitation. Similarly to the current research, Funabashi *et al*. [2018] used a self-efficacy evaluation with specialists to compare the users' perception and ability to perform the system's functions before and after using the system. In contrast, Anjos *et al*. [2024] assessed the severity of the users' difficulties before and after using the application through a five-point Likert scale.

Lastly, the study of Carvalho *et al*. [2024] analyses different evaluation methods used in serious games for autistic children, including the ones used in the current research: questionnaire and experiment. Among the results, the authors identified that the questionnaire method was the only one used in all categories of serious games – either isolated or combined with another method – and the number of participants ranged from 3 to 24. Additionally, the authors reported that the experimental method was used in only 25% of studies to assess the system's impact, effectiveness, and usability; combined with the questionnaire, it helped evaluate the quality of the adopted metrics.

The current research project combines these techniques into one usability evaluation involving self-efficacy and acceptance questionnaires, presenting two different usability



(a) Case with no abnormality. **(b)** Case with DCM. **(c)** Case with HCM.

Figure 1. Slices from CMR exams. In comparison to the case with no abnormality, the DCM image shows a larger internal region of the left ventricle, outlined by the inner red circle, and a thinner ventricular wall, corresponding to the area between the two circles. The opposite occurs in the case of HCM, which has a smaller internal region and a thicker ventricular wall.

metrics that help understand both the efficacy and utility of the developed system.

Overall, the present study contributes to the literature in the topics of data platforms for EHR and the evaluation of system usability through methods that assess how effectively the application meets user requirements. However, this work distinguishes itself from previous studies in terms of its applicability, which targets CAD system developers rather than clinical end-users, and its design approach, which emphasizes flexibility and adaptability. The main differentiating aspects of the proposed system are summarized in Table 1.

4 Materials and methods

The research was conducted according to the phases presented in Figure 2. The initial phases 1 to 3 refer to the requirement analysis, database modeling and implementation, and data acquisition, respectively described in Sections 4.2, 4.3, and 4.4. Phases 4 to 6, in turn, refer to the web application development to store and retrieve data from the flexible database, presenting its design process, implementation, and usability evaluation, respectively detailed in Sections 4.5, 4.6, and 4.7. The evaluation aimed to assess the application's interface usability and the DB's structure efficiency in organizing the CMR exams and patient data, providing the needed information for users to filter a group of patients and retrieve their exams.

4.1 Materials

The data used in this project was obtained from 400 CMR exams, shared by the Heart Institute of the *Hospital das Clínicas* of the *Faculdade de Medicina da Universidade de São Paulo* (InCor-HCFMUSP) as Digital Imaging and Communications in Medicine (DICOM) files, within the context of the research presented by Costa *et al*. [2024] and Gonçalves *et al*. [2024].

The sharing process, as well as the anonymization, security, and privacy of the shared data, were approved and authorized by the Committee for Ethics in Research on Human Beings of the School of Arts, Sciences and Humanities of the University of São Paulo (Aug. 9, 2021, No. 49049021.1.0000.5390), as well as the Committee for Ethics in Research of the HCF-MUSP (Sep. 2, 2021, No. 49049021.1.3001.0068).

From the DICOM files' metadata, it was possible to extract the patient's demographic data, such as age, sex, and weight, which needed to be persisted in the DB. The files also provide two-dimensional (2D) image slices of the patient's cardiac structure at different time points of the cardiac cycle. For each time point, it was possible to reconstruct three-dimensional (3D) objects of the left ventricle. In previous studies [Bergamasco *et al*., 2022], left ventricle 3D objects were reconstructed using the Medviso Segment software [Heiberg *et al*., 2010]. Furthermore, feature descriptors to extract morphological and strain features from 2D and 3D images were developed and validated [Delmondes *et al*., 2015; Bergamasco *et al*., 2022; Costa *et al*., 2024]. With these descriptors, feature values were computed to compose a feature vector to represent each patient. A vector that represents each case was stored in the database, as shown in Section 4.4.

4.2 Requirement analysis

In order to provide a system to meet the needs of the target audience (CAD developers), weekly meetings were held with CAD researchers to collaboratively define both functional and non-functional requirements during the system development. These sessions provided structured discussions, and all identified requirements were documented in a shared file to ensure traceability and alignment. Table 2 summarizes the final requirements that were implemented in this work.

The first functional requirement defines that 2D and 3D objects, as well as the feature descriptor configurations and computed feature values, must be stored in the DB. The second functional requirement defines that this DB must be structured in a way to facilitate search, retrieval, and composition of datasets for building CAD computational models of cardiomyopathies. As for the first non-functional requirement, the DB defined in the present work must be flexible enough to store these and other artifacts that will be developed. This requirement is met by defining a database structure that allows changes without code changes, as presented in Section 4.3.

Additionally, the third functional requirement establishes

Table 1. Comparison between related studies and the current research project.

System	Application	Architecture	Integration	Usability	Flexibility
Related studies	Focused on clinical use by physicians and patients for EHR management or rehabilitation support	Often based on reusable design patterns or static models	Integration with existing medical systems or resources	Evaluated by end-users (physicians, patients) using standard usability metrics	Limited adaptability to changes in data structure or user needs
Current project	Designed for researchers developing CAD systems; supports data collection and experimentation	RESTful web services with a flexible DB model and adaptable GUI	Modular and dynamic interface that adjusts to DB changes	Evaluated by CAD researchers and students using self-efficacy and acceptance questionnaires	High flexibility in both DB and GUI to accommodate evolving datasets and research needs

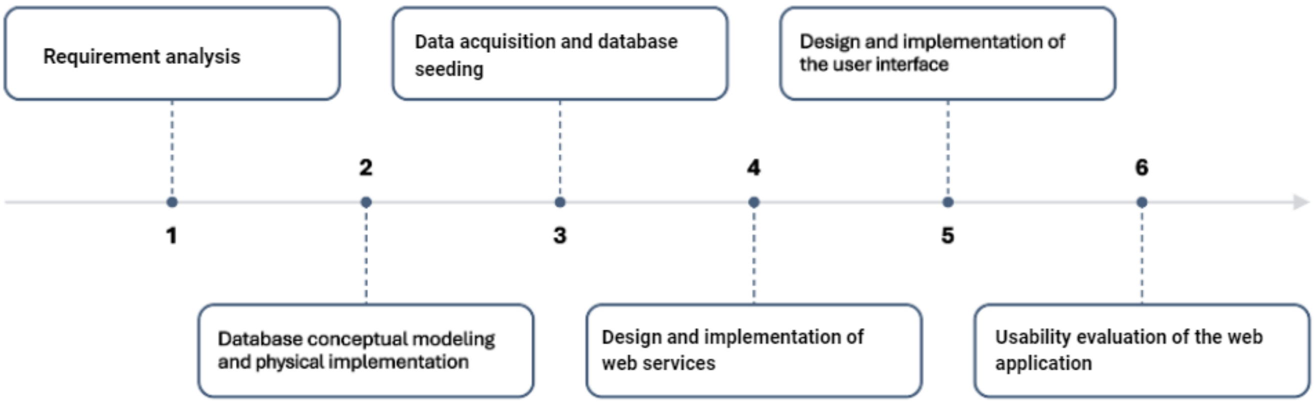


Figure 2. Phases to develop a DB application for CMR exams: from the assessment of DB requirements to the usability evaluation of the DB and its web application.

Table 2. Summary of functional and non-Functional Requirements

Requirement Type	Summary
Functional	1. Store 2D and 3D objects, features, descriptors, and configurations in a structured database. 2. Enable data retrieval and manipulation to support the development of CAD models. 3. Allow access to the database from any location at any time via a web application. 4. Support dynamic adaptation to the database structure for live data manipulation and retrieval by non-technical users and fast-paced research environments.
Non-Functional	1. Ensure flexibility of the database to accommodate current and future artifacts. 2. Minimize system maintenance while maintaining consistency during dynamic operations. 3. Increase productivity by enabling seamless interaction with the database.

the need for a tool that enables the insertion and retrieval of data stored in the database, ensuring that researchers and medical professionals can access it from any location at any time. To fulfill this requirement, a three-layer web application was designed and implemented, as described in Sections 4.3, 4.5, and 4.6.

Because the system must support online data operations by non-technical users, it demands a dynamic adaptation to the database structure. The chosen implementation allows these interactions to occur with minimal maintenance and without compromising consistency. This implementation also satisfies the fourth functional requirement, as well as the second and third non-functional requirements, as described in Section 4.6.

Figure 3 presents the web application architecture. The first layer corresponds to the DB itself. The second layer consists of a RESTful web services module, named `CADDatabaseAPI`, which contains services for performing CRUD operations on the DB data, which will be further detailed in Section 4.5.

The third layer is a graphical user interface (GUI) created to facilitate data retrieval from the DB, including filters and functionalities for downloading DICOM images and data structured in CSV files. The latter can be easily used as datasets to build computational models. The interface implementation is detailed in Section 4.6

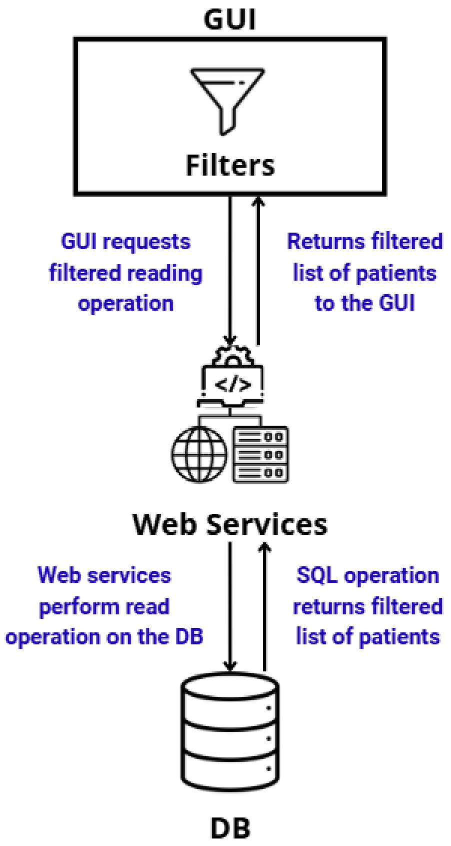


Figure 3. Web application architecture. The reading operation is triggered by the GUI's filtering and fetching functions, and the web services perform the filtered SQL reading operation on the DB. The result is retrieved by the web services and passed on to the GUI, where it is visualized by the user.

4.3 Database modeling and implementation

A relational DB was modeled with the DbSchema software, a tool for designing database diagrams that contains integration features for several Database Management Systems (DBMS) [Wise Coders GmbH, 2023]. The DB model was validated by means of a physical implementation in PostgreSQL, an open-source object-relational DBMS that uses and extends the SQL language, provides many features for storing and scaling data, and is known for its reliability, feature robustness, and performance. [The PostgreSQL Global Development Group, 2023].

As mentioned in Section 1, the proposed application follows OLTP model. The defined system requirements emphasize real-time data operations to support researchers in the development of CAD systems, including selecting and organizing relevant data for a patient's diagnosis [Conn, 2005].

The DB model involves registering the demographic (table `Demographic Data`) and exam attributes (tables `Exam` and `Exam Category`) in individual tables, and each attribute can be associated with a patient and a value in a relationship table (DB model is presented in Section 5.1). These tables also store the data type of these attributes, similar to how the DBMS would store a column's metadata.

Differently to adding columns for each attribute, this approach allows the patients to have all of their available data registered without keeping null values in the columns, minimizing inconsistencies. Additionally, if a new attribute is necessary for patients, no change in the table structure is required. In this case, it does not require the creation of a new column, but only its insertion in the table of attributes and its association with the patient in the relationship table.

This strategy significantly increases productivity by reducing the need for maintenance in the DB schema, since every new attribute can be registered for a patient as a tuple. Keeping separate tables to register the attributes also allows for the dynamic update of the interface, further described in Section 4.6. The interface can simply fetch all of the currently available attributes and data types, and generate the filter elements based on the current state of the DB.

The developed strategy was applied in building the entire DB. The example above (considering patients and their attributes) was also applied to store the several descriptors developed in the CAD system and their respective values. In this way, we defined an approach that requires a greater complexity for conception and implementation, but is able to decrease the necessity of maintenance and, thus, increase productivity during the use of the CAD system.

4.4 Data acquisition

The DICOM images of the 400 CMR exams were processed to extract metadata for insertion into the database. The metadata includes patients' demographic data of age, sex, and weight. A server path for each exam file was provided in the server's file system and stored in the DB to allow users to download the exams through web application requests.

CSV files produced in previous work were also provided, containing the values of attributes computed through morphological feature descriptors of 2D and 3D objects. Scripts were created to properly insert the CSV data into the tables, respecting the relationships with the data already persisted in the DB.

The extraction of metadata from the DICOM images and the data acquisition with the retrieved data were made with the Python language and the libraries `SQLAlchemy` [Myers and Copeland, 2015], `csv`, `pydicom` [Mason, 2011], and `psycopg2`.

4.5 CRUD web services

As mentioned in Section 2.1, RESTful web services allow for scalable data transferring in a computational system, matching the current DB system's flexibility requirements: a single web service can be used to perform a CRUD operation on any data persisted in a DB table, without requiring changes in the code to adapt to different data types and representations.

The RESTful API for retrieving data persisted in the DB was created using Python and the `Flask` framework [Pallets, 2023]. Overall, ten services were implemented and tested for the six resources available on the DB. Table 3 presents the name and description of each resource in terms of which DB entities they represent.

Table 3. Resources and descriptions. For each resource, a web service was implemented, including CRUD methods for data retrieval and manipulation in the DB.

Resource	**Description**
Patient	Represents a patient and their demographic data that can be registered in the DB associated with a type of demographic data.
Exam	Represents a type of exam that can be registered in the DB associated with an exam category.
Exam Execution	Represents an execution of exam that can be registered in the DB associated with a Patient and an Exam.
Object from a Patient's Exam Execution	Represents an object that can be registered in the DB associated with a Patient, Exam, Exam Execution, and object type.
Feature Descriptor	Represents a feature descriptor that can be registered in the DB associated with a configuration, category, and object type.
Feature	Represents a feature that can be registered in the DB associated with a Feature Descriptor.

For each resource, the five methods described in Table 4 were implemented. A Service Documentation (see Supplementary Material I) was written to list the services, their resources, and their methods' parameters, operation, and return state or value.

Table 4. Methods implemented for each resource.

Method	**Description**
`get` (by id)	Retrieves the current state of a resource instance from the DB by the id of its main component entity
`get` (all)	Retrieves the current state of all resource instances of a type from the DB.
`post`	Inserts a new resource instance
`put`	Updates a resource instance already persisted on the DB
`delete`	Deletes a resource instance

4.6 User interface

The user interface was designed by using the concept of a single-page application [Kowalczyk and Szandala, 2024] that turns the user navigation simpler.

The interface's filter elements are dynamically generated to reflect the current DB structure and persisted data: the initialization of the interface calls a method that performs a query to retrieve the current attributes registered on the table `Demographic Data`, as well as the distinct values of patients' demographic data and exams' types and categories. Based on this information, the interface dynamically creates filter elements for patients' and exams' attributes with the currently available values, without requiring maintenance in its implementation. Figure 4 shows an example of how the interface elements reflect the current state of the DB.

The GUI was developed using Python, HTML, JavaScript, CSS, Bootstrap framework, and Bootstrap's DataTables library; its final state is presented in Section 5.2.

4.7 Usability evaluation of the web application

The experiment to evaluate the web application and the usability of its interface was carried out among graduate and undergraduate computer science students, professors, and researchers at the University of São Paulo, who could validate the requirements of the application as developers of CAD systems. The methodology was defined after a careful literature review of the related works presented in Section 3, and it is documented in a protocol containing:

1. the context of the research project;
2. the environment: a Personal Computer (PC) or a computer available at LApIS;
3. the participants' requirements: basic English to read the interface, familiarity with computers, and a computer or mobile device connected to the internet;
4. the procedure to be followed by the users;
5. the description of the tasks to be executed by the users.

A control group was not included in the experiment due to the absence of a comparable application that could be used to evaluate usability for benchmarking purposes. The medical exams utilized in this study are stored in a restricted-access database hosted on Webpax [Simmons, 2013], and only non-sensitive information was made available in the form of unstructured DICOM files. These limitations meant that the specific data and tasks involved in the scope of this research could not be replicated or accessed externally, making a usability evaluation by a control group unfeasible.

The experiment involved the use of the web application guided by instructions on executing specific tasks that cover all of the elements and functionalities of the interface. Table 5 presents the overview of the tasks performed by the users (see Supplementary Material II for detailed descriptions of the tasks). To help the participants, a User Guide was created to explain the elements and actions available in the interface.

A form was created on Google Forms to unify the procedure, instructions, link to the application, User Guide, and questions to evaluate the interface's usability. It was shared with researchers and students in an invitation to participate in the experiment. The form was organized into five sections and guided the user through all of the sections of questions and tasks. The entire procedure followed by the participants is presented in Figure 5.

The answers from the Participant Profile (second section of the form) were used to identify whether the participants' personal, educational, and professional experiences with the

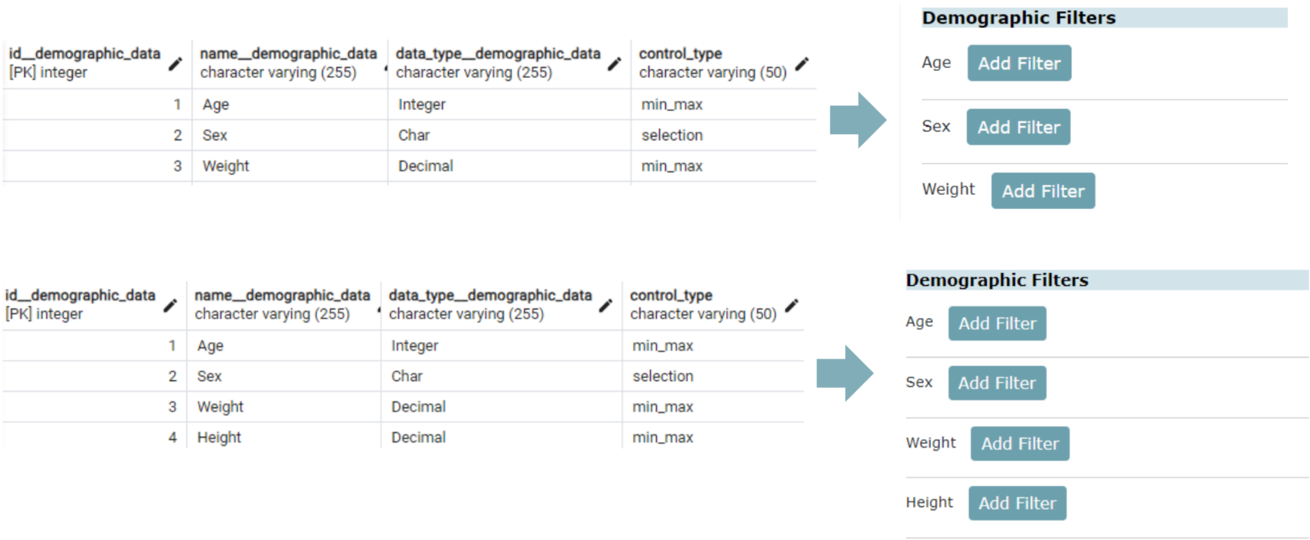


Figure 4. Two different states of the `Demographic Data` table and the user interface filter elements generated based on the attributes' name and data type. The first state does not include the attribute 'Height'; when this attribute is inserted, the interface will show its filter element automatically after reloading, without requiring changes in the code.

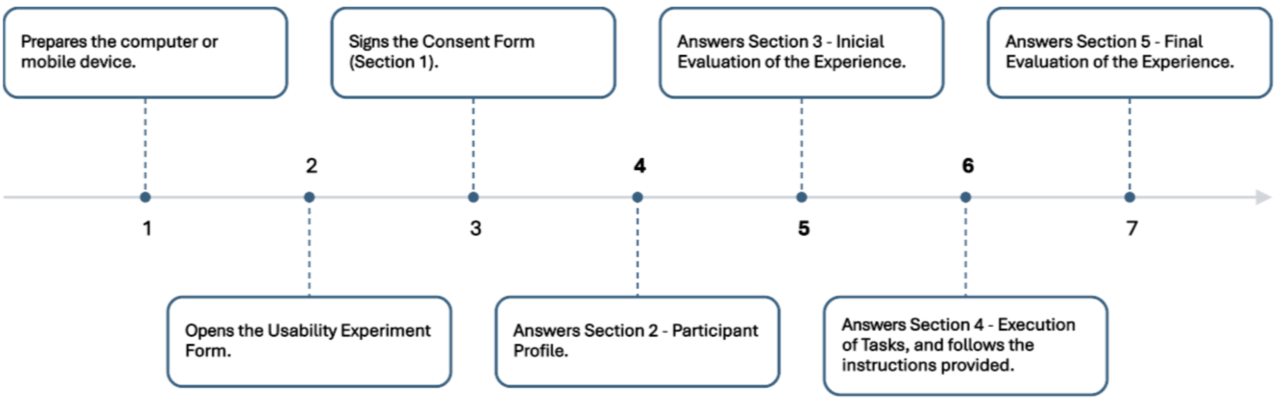


Figure 5. Steps of the experiment procedure.

Table 5. Tasks performed by the participants in the experiment to evaluate a DB system interface.

Task	Description
1	Obtain the data of all patients.
2	Obtain the data of patients diagnosed with dilated cardiomyopathy with ages between 20 and 80 years.
3	Obtain the data of patients diagnosed with hypertrophic cardiomyopathy.
4	Obtain the data of patients between the ages of 20 and 80 years.
5	Obtain the data of female patients who weigh more than 95 kilograms and have any registered exam types.
6	Freely navigate through the interface to test the available features (filters and buttons).

areas within the scope of this research affected their evaluation of the interface. The goal was to assess whether more or less familiarity with similar artifacts may result in different perceptions of the functionalities, errors, and self-efficacy.

The third and the fifth sections of the form consist of a self-efficacy evaluation with Likert scale assessment questions, which identify the user's belief in their capacity to execute the actions provided by the interface before and after using it[Bandura, 1978]. The goal was to verify if, after using the interface, the users believe they are more capable of executing those actions than before its use.

In the first self-efficacy evaluation, the functionalities are introduced as a hypothetical application, and the user must answer whether they believe they could perform the described tasks by using it. In the second evaluation, the user is already familiar with the application and must evaluate if they can perform those tasks through the interface. The Self-Efficacy Statements (SES) in this form section are listed in Table 6.

The fourth section of the form provides the tasks to be executed in the interface, as well as the links to the application and the User Guide. This section aims to identify any errors that may occur during the usage of the interface, collect feedback on the visualization and functionality of elements, and verify if the generated results are compatible with those obtained from performing direct queries on the DB, without using the interface. The overall flow of the tasks is represented in Figure 6, and their detailed descriptions are in Supplementary

Table 6. Self-efficacy statements.

SES ID	Statement
SES1	I am capable of filtering a group of patients based on their diagnosis and demographic data.
SES2	I am capable of obtaining the demographic data of a group of patients.
SES3	I am capable of obtaining the CMR exam files of a group of patients.
SES4	I am capable of obtaining the files of different exam types and categories.
SES5	I am capable of obtaining a table with the patients' demographic data and values of their CMR exams' images features.

Material II.

After completing the tasks, the user answers a general evaluation of their acceptance of the interface by agreeing or disagreeing with the SUS-based Acceptance Statements (AS) listed in Table 7.

Table 7. Interface acceptance statements.

AS ID	Description
AS1	I was able to complete all the desired tasks.
AS2	I was able to perform all the desired tasks without encountering any errors in the functionalities and interface elements.
AS3	I was able to perform all tasks with ease and had no difficulty understanding the elements and functionalities.
AS4	I was able to identify the interface elements easily and had no issues with the layout or visual presentation of the elements.
AS5	I was able to execute all tasks without using the User Guide provided.
AS6	Overall, I am satisfied with the use of the interface.

The server where the application runs has an Asus ROG MAXIMUS Z790 HERO motherboard, version Rev 1.xx; 192.0 GB of Random Access Memory (RAM); an Intel© Core™ i9 processor 14900KF processor with 24 cores; and 5.0 TB of storage. The operating system is Ubuntu 64-bit version 22.04.5 LTS.

4.7.1 Statistical analysis

To assess the impact of system usage on users' self-efficacy, we tested the null hypothesis (H_0) that the mean scores obtained from responses to each self-efficacy statement before using the system (SES 1B to 5B) are statistically equivalent to the mean scores obtained after system use (SES 1A to 5A). For this purpose, the Mann-Whitney U test was employed: this non-parametric test is suitable for comparing two independent groups when the data do not follow a normal distribution, allowing us to determine whether the observed differences in self-efficacy scores are statistically significant [Mann and Whitney, 1947].

Before applying the Mann-Whitney U test to evaluate the H_0 for each self-efficacy statement, we first verified whether the responses, collected before and after system use, follow a normal distribution. To do this, we applied the Shapiro-Wilk [Shapiro and Wilk, 1965] and Anderson-Darling [Stephens, 1974] tests, which evaluate the null hypothesis that the data were drawn from a normally distributed population. This step is crucial to justify the use of a non-parametric test like Mann-Whitney.

The Shapiro-Wilk test returns a test statistic and a p-value: if the p-value is below the conventional threshold of 0.05, we reject the null hypothesis that the sample follows a normal distribution, justifying the use of a non-parametric test. The Anderson-Darling test is more general and can evaluate the compatibility of a sample with various types of distributions and across different significance levels (p-values). To specifically test for normality, you simply set the parameter 'dist='norm'' when calling the function. The test returns a test statistic, five standard significance levels, and corresponding critical values for each level. These outputs are all essential for evaluating the null hypothesis. To evaluate H_0, it is necessary to compare the test statistic to the critical value at a chosen significance level. If the statistic exceeds the critical value, H_0 is rejected. In this case, we assume a significance level of 0.05.

5 Results and discussions

This section presents and discusses the achieved results about the current DB implementation (Section 5.1), the web application (Section 5.2), and the usability evaluation (Section 5.3).

5.1 Database

Figure 7 presents an overview of the implemented DB. The blocks represent database tables (relations), and the arrows indicate the direction of the dependency: from the table where an attribute serves as a primary key to the table where that same attribute is referenced as a foreign key. Table 8 presents the description of each table illustrated in Figure 7. Supplementary Material III presents the detailed relational DB model and its data dictionary.

As previously mentioned, the proposal aims to provide a model that allows flexibility, i.e., changing the content with no or minimal change in the DB structure. This flexible model allows for the dynamic inclusion of new attributes without requiring changes or maintenance in the DB structure. For example, table `Patient Demographic Data` currently stores the patient's age, sex, and weight. If these attributes were created as columns on the table, every new attribute that one wished to register would require changing the table structure and, therefore, the DB schema. With this model, however, it is possible to register every new attribute in the `Demographic Data` table, and `Patient Demographic Data` materializes the relationship between a registered patient, a type of demographic data, and its value for that patient.

This flexibility minimizes the need for maintenance in the DB structure, and it does not depend on the persisted information, making it applicable to various DB and CAD systems

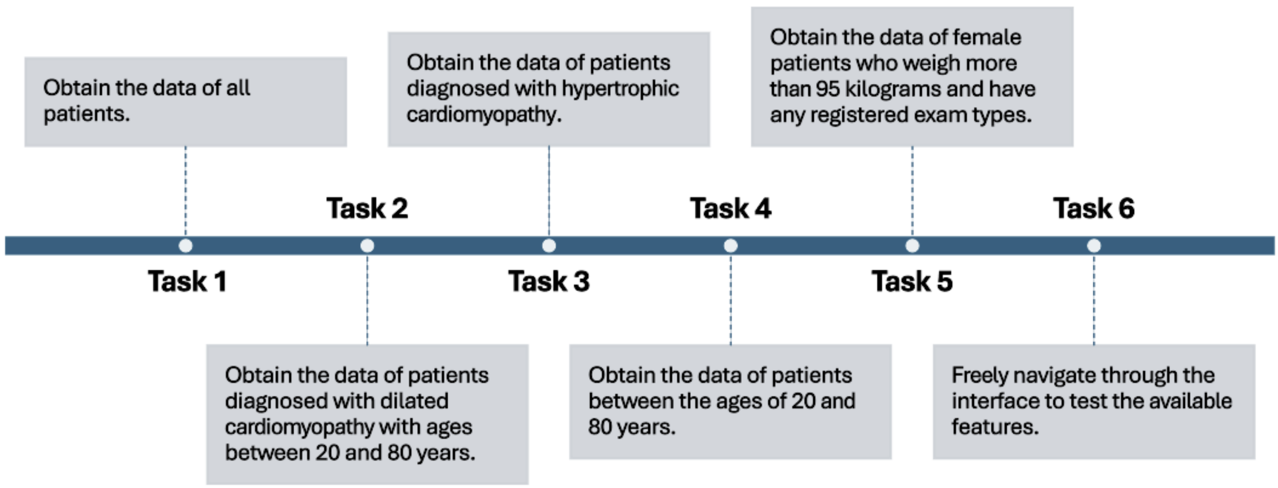


Figure 6. Tasks executed by the participants in the experiment to evaluate the usability of the web application.

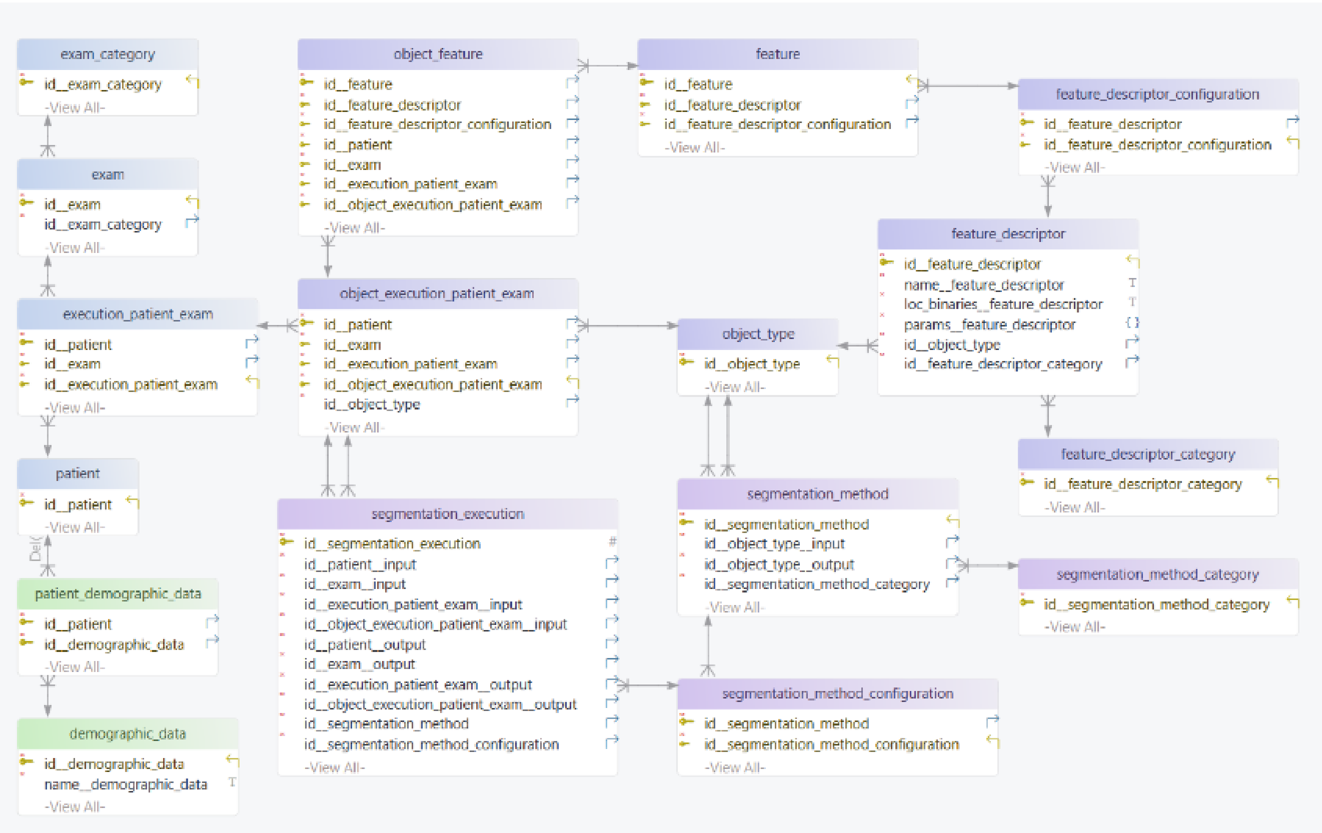


Figure 7. Overview of tables, keys, and constraints of the DB.

beyond the medical concepts explored in this research.

5.2 Application with web services

The DB records can be accessed through RESTful web services that, in turn, provide structured data (resources) to render the user interface. As mentioned in Section4.5, RESTful web services help meet the current DB system's flexibility requirements since they minimize the need for code maintenance to adapt to different data formats. These services also allow the dynamic generation of filter elements on the user interface, making it automatically adaptable to changes in the DB structure and minimizing the need for maintenance on its implementation.

Additionally, as mentioned in Sections 4.3 and 4.6, the DB model incorporates the registering of attributes similarly to a DB column's metadata, which allows the interface to fetch all currently available attributes and data types to generate the appropriate filter elements. Since the fetching involves a simple query on a DB table, without requiring an analysis of the DB schema, the interface can execute this query every time it is loaded, dynamically adapting to the current state of the DB with no need for code maintenance.

Figure 8 presents the user interface after initialization. In

Table 8. Descriptions of Tables in the Database.

Table Name	**Description**
`demographic_data`	Stores the demographic data that can be associated with a patient
`exam`	Stores the exam types
`exam_category`	Stores the exam categories that can be associated with an exam type
`execution_patient_exam`	Stores an execution of an exam type associated with a patient
`patient`	Stores the patients
`patient_demographic_data`	Stores the values of the demographic data associated with the patients
`feature`	Stores the features that can be associated with an object extracted from a patient's exam
`feature_descriptor`	Stores the feature descriptors that can be associated with the features, each descriptor is associated with a category and a configuration
`feature_descriptor_category`	Stores the categories of feature descriptors
`feature_descriptor_configuration`	Stores the configurations of feature descriptors
`object_type`	Stores the types of object that can be extracted from an exam
`segmentation_method`	Stores the segmentation methods that can be performed in an exam, each method is associated with a category and a configuration
`segmentation_method_category`	Stores the categories of segmentation methods
`object_execution_patient_exam`	Stores the object type extracted from a patient's exam
`segmentation_method_configuration`	Stores the configurations of segmentation methods
`object_feature`	Stores the value of a feature calculated from the object type extracted from a patient's exam
`segmentation_execution`	Stores the execution of a segmentation method that was performed in a patient's exam

this initial state, it is possible to add a filter for the patient's class of diagnosis, age, sex, weight, and height, as well as the exam's type and category. For categorical attributes, the filter element presents a list of all distinct values available in the DB; for numerical attributes, the filter is a range slider element from zero to a maximum, currently set to 120. The filter element is within a pop-up, which appears after selecting the 'Add Filter' button of each attribute.

After selecting the desired filters, the chosen values are available below the attribute's name, and the "Add Filter" buttons are changed to "Alter Filter". Then, the user can select the "Fetch Data" button, which retrieves the patients whose information is within the filtered data. The interface, then, shows a table with the retrieved patients' IDs and demographic data, which is shown in Figure 9.

After fetching the data, the buttons to download the exams and the CSV appear on the screen. The "Download Exams" button triggers the download of a zip file with the retrieved patients' exams DICOM files, and the "Download CSV" button downloads a CSV file with each row containing a retrieved patient's ID, sex, age, weight, and the values of each of the registered feature values of the exams' images. To perform a new search, the user can restart the page using the browser or the "Reset" button.

Therefore, the system supports stratified data exploration through dynamic filtering mechanisms integrated into the user interface. As illustrated in Figure 9, the interface displays the filtered patient list along with their demographic attributes, facilitating exploratory selection and organization. Once filtered, the data can be exported in CSV and ZIP formats, enabling external stratified analyses without requiring modifications to the database schema or interface code. This flexible approach empowers CAD researchers to tailor their investigations to specific population groups while preserving the system's adaptability. As future work, we may integrate dashboards to provide further analysis value to the tool.

5.3 Usability evaluation

In the Usability experiment, 31 volunteers executed the six tasks and answered the experiment form presented in Section 4.7. Most participants were male Computer Science undergraduate students and PhDs, with basic to no knowledge of cardiomyopathies, CMR, medical image processing,

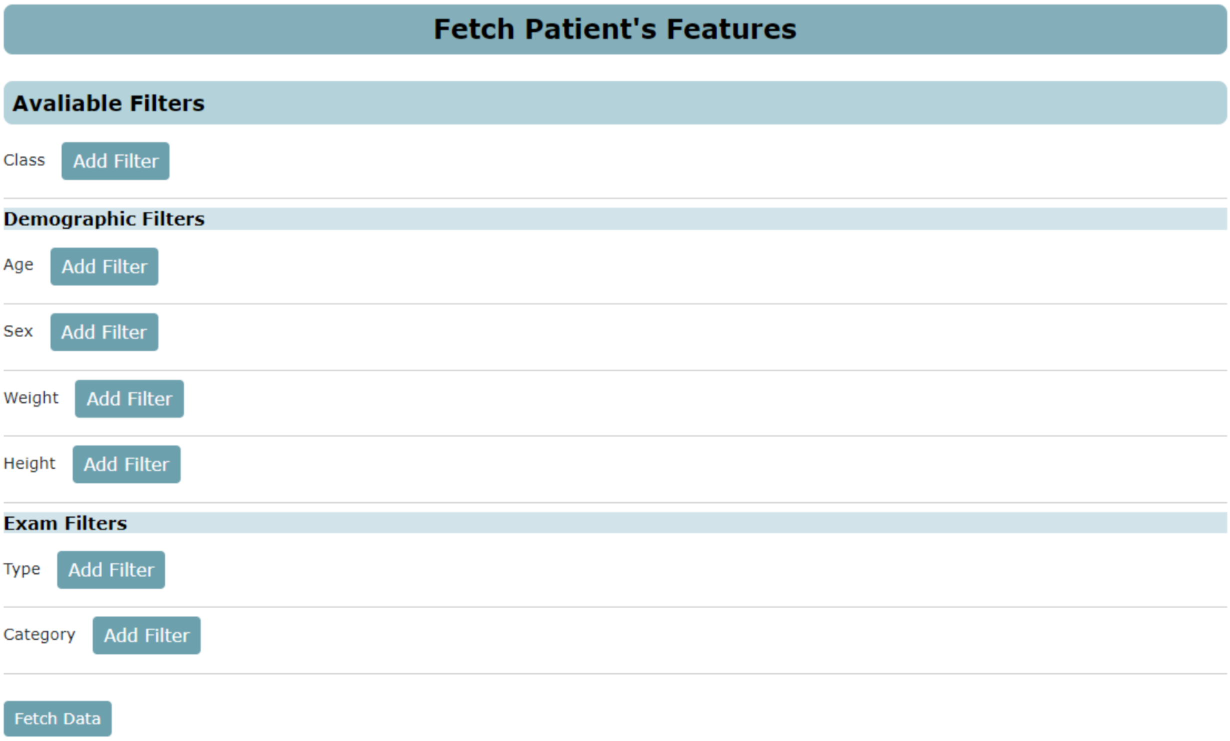


Figure 8. DB application user interface after initialization. It presents filter options for the class of diagnosis, demographic data, and exam type and category. The demographic filters' elements are automatically generated based on the current state of the DB without requiring code maintenance.

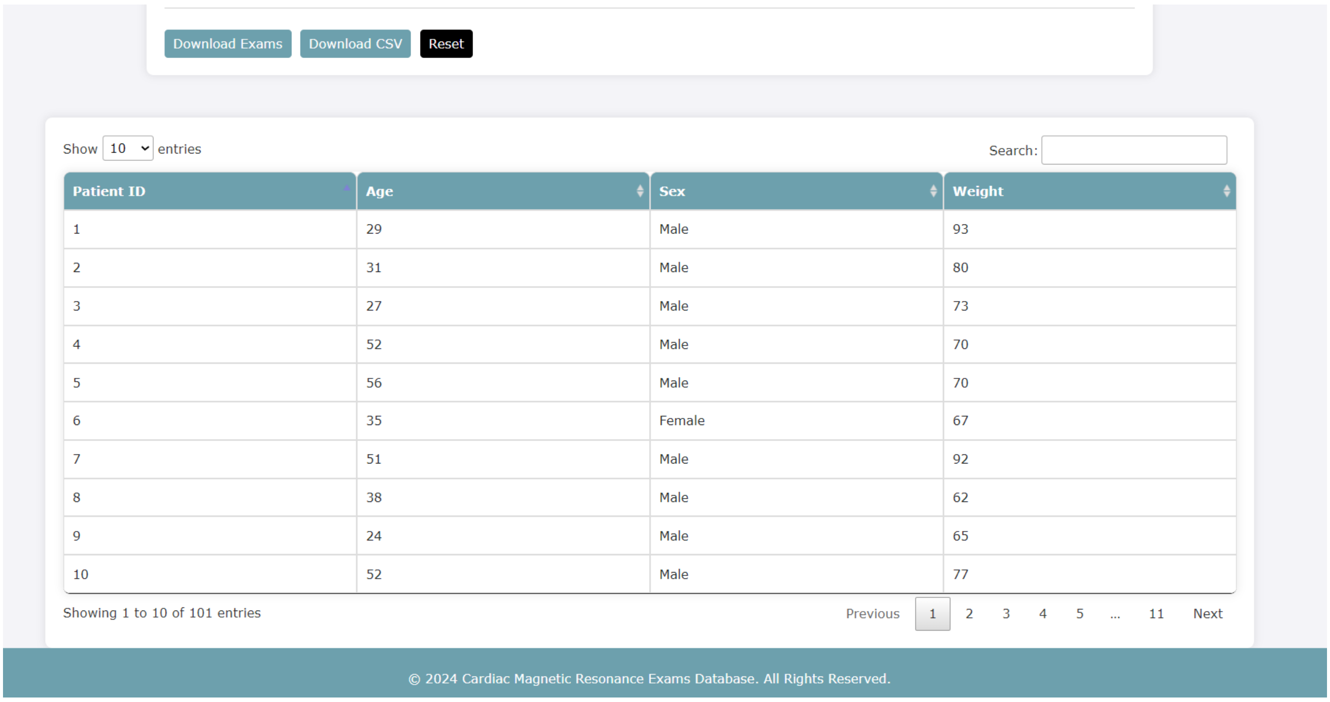


Figure 9. Web application user interface highlighting the rendered table of the patient's demographic data after the execution of a search.

and CAD systems. The profile of the participants, as well as the percentages of completed tasks and accurate results, are presented in Table 9.

Except for Task 5, nearly every participant could complete the tasks, and those who did not complete them specified that the incompleteness was due to connectivity issues. On average, only 4% of answers were incorrect when comparing the obtained results with the expected values.

Associating these results with the participants' profiles, it was possible to identify that the wrong answers were all from male participants. However, this was most likely due to males being over 70% of the participants instead of there being a correlation between gender and accuracy when performing the tasks.

The accuracy of the tasks' results was also associated with the participants' education level and familiarity with the concepts in this research project. However, the percentage of wrong answers was consistent among all the groups. There-

Table 9. Percentages of completed and correct tasks per demographic group.

Group	Task	Number (%)	Completed	Correct
Gender				
Male	1-4	23 (74%)	96%	90%
	5		70%	57%
Female	1-5	8 (26%)	100%	100%
Education Level				
Doctorate	1-4	14 (45%)	100%	99%
	5		63%	50%
Masters	1-4	6 (19%)	100%	81%
	5		100%	75%
Undergraduate	1-4	11 (36%)	100%	99%
	5		85%	80%
Familiarity with Concepts				
Computer-Aided Diagnosis				
None or Basic	1-4	19 (61%)	100%	93%
	5		80%	75%
Intermediate	1-4	5 (16%)	100%	100%
	5		40%	40%
Advanced or Specialist	1-4	7 (23%)	90%	90%
	5		55%	30%
Cardiac Magnetic Resonance				
None or Basic	1-4	24 (77%)	100%	94%
	5		76%	63%
Intermediate	1-4	4 (13%)	100%	100%
	5		50%	50%
Advanced or Specialist	1-4	3 (10%)	67%	67%
	5		33%	33%
Databases				
None or Basic	1-4	2 (7%)	100%	100%
	5		50%	50%
Intermediate	1-4	9 (29%)	100%	94%
	5		89%	78%
Advanced or Specialist	1-4	20 (65%)	96%	92%
	5		59%	48%
Image Processing				
None or Basic	1-4	19 (61%)	100%	94%
	5		77%	69%
Intermediate	1-4	3 (10%)	100%	100%
	5		33%	33%
Advanced or Specialist	1-4	9 (29%)	92%	92%
	5		58%	42%
Cardiomyopathies				
None or Basic	1-4	25 (81%)	100%	95%
	5		69%	57%
Intermediate	1-4	3 (10%)	100%	100%
	5		33%	33%
Advanced or Specialist	1-4	3 (10%)	67%	67%
	5		67%	67%

fore, these parameters are not necessarily correlated to the execution of the tasks.

Task 5 was the only exception in terms of completeness and accuracy: one-third of the participants did not complete the task, and out of the ones who did, only four did not report an error. The issue encountered by these participants was the same: when selecting the "Download Exams" button, the zip file with the patients' exams was not downloaded, or took over ten minutes to download.

The reason for this issue was a network connection limitation of the server computer. To access the public Internet, the server connects to a foreign proxy, which was probably the reason for the decrease in the connectivity speed, particularly when downloading large files. To solve this issue, a local internet connection could be created for the server to increase the connectivity speed, or the DB and its application can be migrated to a hardware and software infrastructure designed to meet the computational and connectivity demands of the DB application.

Figure 10 summarizes the self-efficacy evaluation (Supplementary Material IV presents percentages of all answer labels for each question). The answers show an average increase of 30.4% of "strongly agree" answers after using the interface, suggesting a general increase in the participants' self-evaluated capacity of performing the desired tasks by using the interface, which validates the system's efficacy and utility for the target users.

As mentioned in Section 4.7, we applied Shapiro-Wilk and Anderson-Darling statistical tests to evaluate the distribution of the answers. Both tests were applied to each set of responses, before (B) and after (A) using the system, for all five self-efficacy statements. The results in Table 10 show that the null hypothesis, which assumes that the data follows a normal distribution, was rejected for all statements in both conditions. Particularly, all p-values of the Shapiro-Wilk test were significantly below the conventional threshold of 0.05, indicating the non-normality of the distribution in every case. For the Anderson-Darling test, the test statistic exceeded the corresponding critical value in all cases, which means that the null hypothesis, which assumes a normal distribution, can be rejected with a 95% level of certainty.

This outcome reinforces the conclusion that the data does not exhibit normality under the tested conditions, which justifies the use of non-parametric statistical methods, such as the Mann-Whitney U test. It also reinforces the robustness of the experimental design, as the assumption of normality was explicitly tested and ruled out.

Therefore, the Mann-Whitney U test was applied to compare the distributions of self-efficacy scores before and after system usage. In this test, the key factor for determining whether to accept or reject the null hypothesis is the computed p-value. As with previous tests, we adopt a threshold of 0.05: if the p-value is less than 0.05, we reject H_0 with 95% confidence in our decision. For each statement, H_0 was be tested by comparing the sample of responses collected before using the system (B) with the sample collected after using the system (A), allowing us to assess whether there is a statistically significant difference in user perceptions.

The analysis of Table 11 shows that for statements SES 1, 2, and 5, the null hypothesis was rejected. This provides statistical evidence that, for these specific tasks, there is a significant difference in users' self-efficacy evaluations before and after using the system. On the other hand, for SES 3 and 4, the null hypothesis was accepted. This suggests that, for these two tasks, there is no statistically significant difference in users' self-efficacy evaluations before and after using the system.

Table 10. Normality test results for each Self-Efficacy Statement using Shapiro-Wilk and Anderson-Darling methods.

SES	**Shapiro-Wilk Statistic**	**SW p-value**	**Anderson-Darling Statistic**	**H_0 Status (0.05)**
1B	0.6463	1.06e-06	3.72	Rejected
2B	0.6331	7.29e-07	4.43	Rejected
3B	0.6685	2.02e-06	3.96	Rejected
4B	0.6530	1.28e-06	3.54	Rejected
5B	0.6751	2.46e-06	3.22	Rejected
1A	0.1980	6.83e-11	9.53	Rejected
2A	0.1980	6.83e-11	9.53	Rejected
3A	0.4936	2.16e-08	6.37	Rejected
4A	0.4295	5.27e-09	7.25	Rejected
5A	0.1980	6.83e-11	9.53	Rejected

In fact, the only statements where more respondents answered lower values after using the interface were SES 3 and 4, with an average 5% increase on the "disagree" and "strongly disagree" answers. These statements are associated with the retrieval of the CMR exams and the files of different exam types and categories. Due to the limitation with the zip file download aforementioned, some participants did not feel more capable of achieving these tasks by using the interface. However, it was still possible to observe an increase in the user's self-evaluated capacity to perform these activities after using the interface, reinforcing the system's overall positive evaluation.

Lastly, the agreement levels to the system's acceptance statements are presented in the graph in Figure 11 (see Supplementary Material IV for more details). The percentages of "strongly agree" and "agree" answers made up over 60 % of the total for every statement, which indicates a good overall acceptance of the interface.

The only statement that had less than 85% of "agree" and "strongly agree" answers was AS2, which evaluates if the user was able to perform every task without encountering errors. The participants who did not complete Task 5 considered the incompleteness of the zip file download an interface error, and disagreed with this statement. However, as previously discussed, this was most likely due to the network

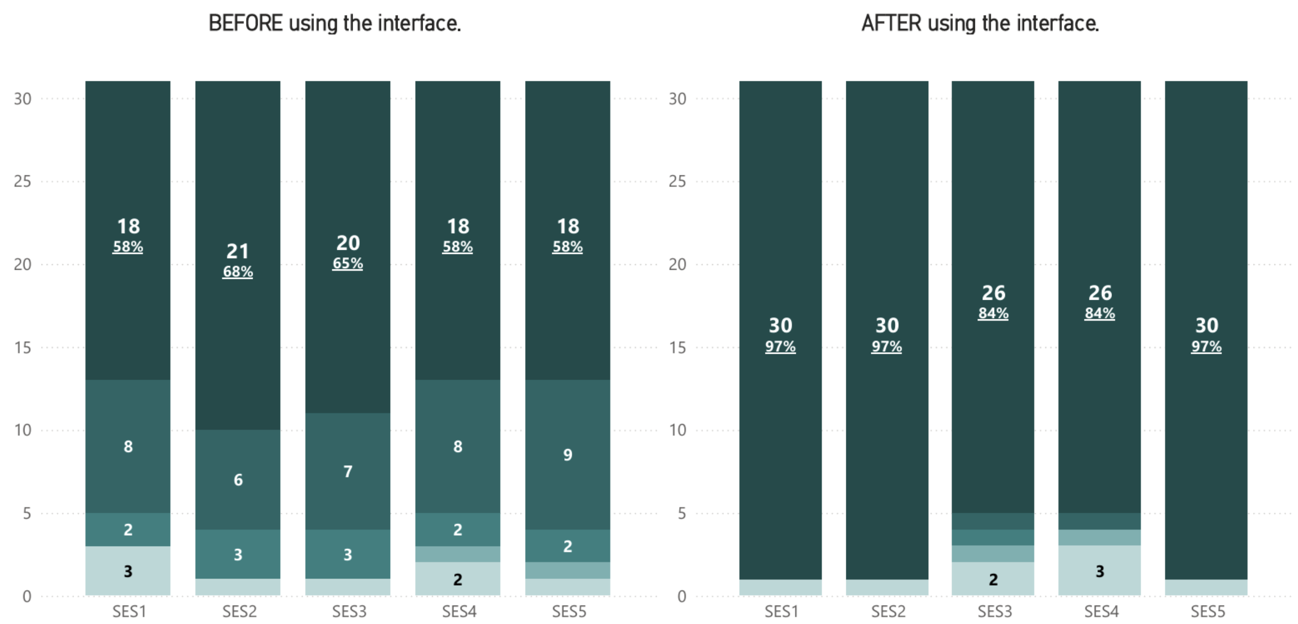


Figure 10. Answer of self-efficacy statements before and after using the web application.

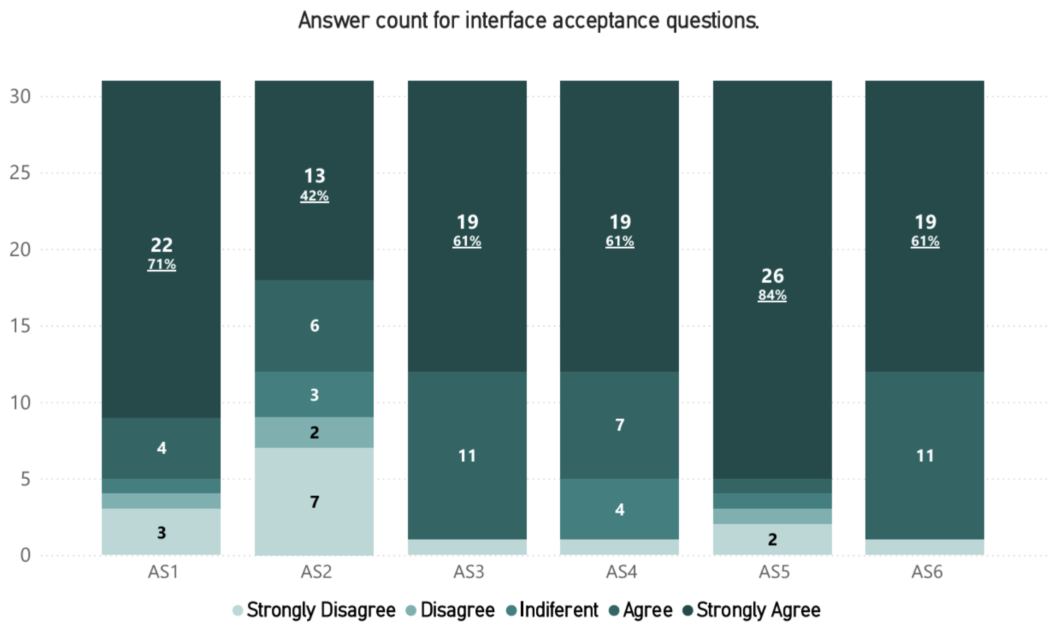


Figure 11. Answer counts of participants' acceptance of the web application.

connection limitations instead of an issue with the interface implementation.

To further comprehend the acceptance of the interface regarding AS2, the graph in Figure 12 associates the participant's agreement level to AS2 and the percentage of tasks they completed successfully and with accurate results, and the bubbles' size is proportional to the number of respondents. It was possible to visualize that participants who were unsuccessful in any of the tasks were more likely to answer lower levels of agreement to AS2, thus, participants who correctly completed the tasks did not find any major errors with the functionalities and elements of the interface.

The results obtained in the experiment suggest that the web application built using web services concepts was capable of automatically adapting to the need for searching and retrieving data in a friendly and flexible way, addressing the needs of developers of CAD systems. Although our case study was developed in the context of CAD for cardiomyopathies, other subjects can be addressed with the same concepts previously presented.

6 Conclusion

The conceptual modeling and the implementation of the database by using direct access to the attributes as metadata allowed building a flexible database application that can be automatically updated, i.e., with no change in the source code when the database scheme is changed, minimizing the need for code maintenance and increasing productivity. The RESTful web services in the application also provide flexibility by allowing the system to perform CRUD operations on the DB

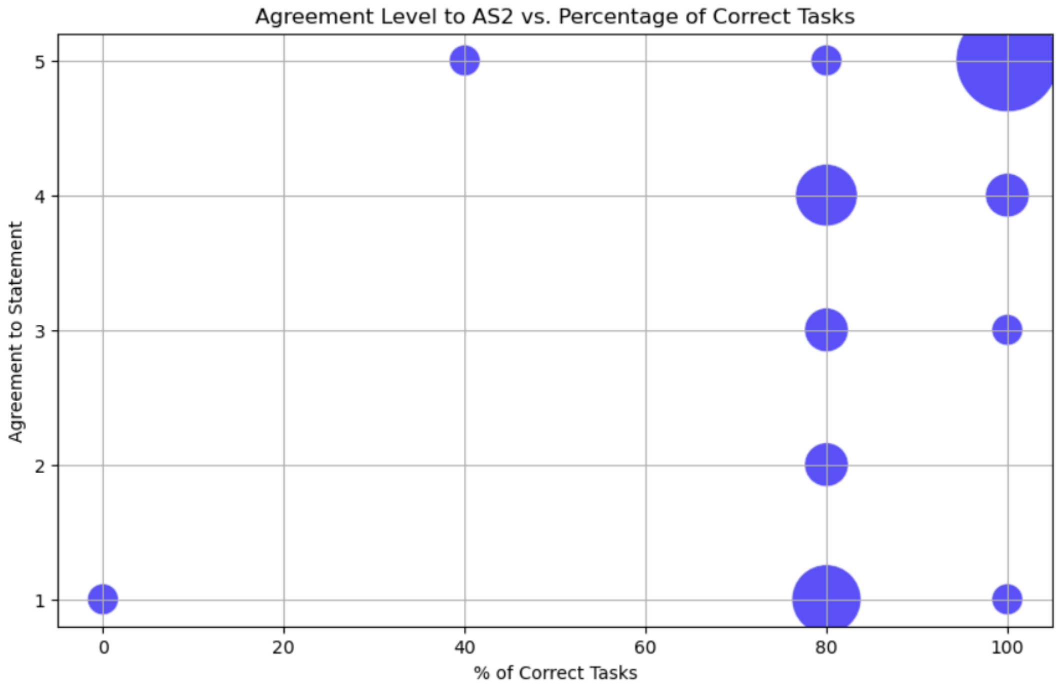


Figure 12. Bubble chart relating each agreement level with the percentage of correct tasks executed by participants who chose the level as answer for the AS2 ("I was able to perform all the desired tasks without encountering any errors in the functionalities and interface elements").

Table 11. Mann-Whitney U test results for each Self-Efficacy Statement.

SES	Mann-Whitney U Test Summary
1	U statistic: 212.0000; p-value: 0.0018; H_0 status: Rejected (non-normal distribution)
2	U statistic: 251.0000; p-value: 0.0162; H_0 status: Rejected (non-normal distribution)
3	U statistic: 302.0000; p-value: 0.4040; H_0 status: Accepted (normal distribution)
4	U statistic: 271.0000; p-value: 0.1180; H_0 status: Accepted (normal distribution)
5	U statistic: 213.0000; p-value: 0.0020; H_0 status: Rejected (non-normal distribution)

with no need to adapt the code to different data formats.

Overall, the experiment showed that the implemented interface received positive evaluations on its functionalities and usability, and users were capable of performing the intended tasks with correct outcomes. There was an average 30.4% increase in the positive self-efficacy perceptions after using the interface, and all acceptance statements had over 60% of agreements, suggesting that the system is efficient and useful to its target users.

The next steps of the research include the implementation and testing of Information Visualization functionalities on the interface, integrating dashboards to improve the analysis value, readability, and interactivity with the DB data. In addition, other types of evaluation can be conducted, such as architectural and technical performance evaluation, among others.

Declarations

Acknowledgements

The authors would like to thank the InCor-HCFMUSP by sharing CMR exams, as well as the Brazilian National Council for Scientific and Technological Development (CNPq) and São Paulo Research Foundation (FAPESP) for financial support.

Funding

This work was supported in part by the CNPq [grant numbers 307710/2022-0 and 120150/2023-9], and FAPESP through the National Institute of Science and Technology – Medicine Assisted by Scientific Computing (INCT-MACC) [grant number 2014/50889-7] and a scientific initiation scholarship [grant number 2023/12566-0].

Authors' Contributions

LTA: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Visualization, and Writing – original draft. VMG: Conceptualization, Data curation, Investigation, Methodology, Project administration, Validation, and Writing – review & editing. FLSN: Conceptualization, Funding acquisition, Methodology, Resources, Supervision, Validation, and Writing – review & editing. All authors read and approved the final manuscript.

Competing interests

The authors declare they do not have any competing interests.

Availability of data and materials

The following materials supplement this article:

- Supplementary Material I presents the documentation of the web services implemented for data retrieval and manipulation in the DB application.
- Supplementary Material II presents the detailed descriptions of the tasks performed by the participants in the experiment to evaluate the DB application and its interface.
- Supplementary Material III presents the relational DB model and its data dictionary.
- Supplementary Material IV presents the percentages of all answer labels for each self-efficacy and acceptance statement in the Experiment Usability Form evaluation questions.

References

Anjos, F. M. S. d., Oliveira, H. M., Estevam, F. E., Vaz, N. A. P., Mata, L. R. F. d., and Carvalho, S. T. (2024). Analyzing the impact of gamification on a mhealth application for treating urinary incontinence in prostate cancer patients. *Journal on Interactive Systems*, 15(1):728–740. DOI: 10.5753/jis.2024.4190.

Arcuri, A. (2017). Restful api automated test case generation. In *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, pages 9–20. DOI: 10.1109/QRS.2017.11.

Assis, G. A. d., Brandão, A. F., Côrrea, A. G. D., and Castellano, G. (2019). Evaluation of a protocol for fmri assessment associated with augmented reality rehabilitation of stroke subjects. *Journal on Interactive Systems*, 10(1). DOI: 10.5753/jis.2019.717.

Bandura, A. (1978). Self-efficacy: Toward a unifying theory of behavioral change. *Advances in Behaviour Research and Therapy*, 1(4):139–161. DOI: 10.1016/0146-6402(78)90002-4.

Bergamasco, L. C. C., Lima, K. R., Rochitte, C. E., and Nunes, F. L. (2022). A bipartite graph approach to retrieve similar 3d models with different resolution and types of cardiomyopathies. *Expert Systems with Applications*, 193:116422. DOI: 10.1016/j.eswa.2021.116422.

Bergamasco, L. C. C., Lima, K. R. P. S., Rochitte, C. E., and Nunes, F. L. S. (2018). 3D medical objects retrieval approach using spharms descriptor and network flow as similarity measure. In *Proceedings of the 31st Conference on Graphics, Patterns and Images (SIBGRAPI 2018)*, pages 329–336. DOI: 10.1109/SIBGRAPI.2018.00049.

Brzezicki, M. A., Bridger, N. E., Kobetić, M. D., Ostrowski, M., Grabowski, W., Gill, S. S., and Neumann, S. (2020). Artificial intelligence outperforms human students in conducting neurosurgical audits. *Clinical Neurology and Neurosurgery*, 192:105732:1–105732:6. DOI: 10.1016/j.clineuro.2020.105732.

Carvalho, A. P. d., Braz, C. S., and Prates, R. O. (2024). An analysis of the evaluation methods being applied to serious games for autistic children. *Journal on Interactive Systems*, 15(1):55–78. DOI: 10.5753/jis.2024.3288.

Ciarambino, T., Menna, G., Sansone, G., and Giordano, M. (2021). Cardiomyopathies: an overview. *International Journal of Molecular Sciences*, 22(14):7722:1–7722:25. DOI: 10.3390/ijms22147722.

Conn, S. (2005). Oltp and olap data integration: a review of feasible implementation methods and architectures for real time data analysis. In *Proceedings. IEEE SoutheastCon, 2005.*, pages 515–520. DOI: 10.1109/SECON.2005.1423297.

Costa, S. S. H., Gonçalves, V. M., and Nunes, F. L. S. (2024). Applying ventricular wall shape and motion features from CMRI for aiding diagnosis of cardiomyopathies. In *Anais do XXIV Simpósio Brasileiro de Computação Aplicada à Saúde*, pages 142–153. DOI: 10.5753/sbcas.2024.2066.

Cruz, D., Junior, C. C., Brucki, S., Takada, L., and Nunes, F. (2022). Automatic adaptation in self-monitoring systems for mci patients: conceptual approach and a software prototype. In *Proceedings of the 2nd Life Improvement in Quality by Ubiquitous Experiences Workshop*, pages 1–4. DOI: 10.5753/lique.2022.19994.

Delmondes, P. H. M., Bergamasco, L. C. C., and Nunes, F. L. S. (2015). Recuperação de modelos médicos por conteúdo usando extrator local de Área e distância. *Revista de Informática Teórica e Aplicada*, 22(2):10–30. DOI: 10.22456/2175-2745.53751.

Doi, K. (2007). Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, 31(4):198–211. DOI: 10.1016/j.compmedimag.2007.02.002.

Fensel, D. and Bussler, C. (2002). The web service modeling framework wsmf. *Electronic Commerce Research and Applications*, 1(2):113–137. DOI: 10.1016/S1567-4223(02)00015-7.

Fernandes, E., Silva, M. A., and Cagnin, M. I. (2016). Sigs-s: A set of applications for social and health care data management. *iSys – Revista Brasileira de Sistemas de Informação*, 9(1):81–100. Available at:`https://journals-sol.sbc.org.br/index.php/isys/article/download/301/301`.

Funabashi, A. M. M., Aranha, R. V., Silva, T. D., Monteiro, C., Silva, W. S., and Nunes, F. (2018). A serious game for virtual rehabilitation: evaluation with patients and physiotherapists. *Journal on Interactive Systems*, 9(2). DOI: 10.5753/jis.2018.698.

García-Holgado, A., Vázquez-Ingelmo, A., Alonso-Sánchez, J., García-Peñalvo, F. J., Therón, R., Sampedro-Gómez, J., Sánchez-Puente, A., Vicente-Palacios, V., Dorado-Díaz, P. I., and Sánchez, P. L. (2022). Koopaml, a machine learning platform for medical data analysis. *Journal on Interactive Systems*, 13(1):154–165. DOI: 10.5753/jis.2022.2574.

Gonçalves, V. M., Bergamasco, L. C. C., and Nunes, F. L. S. (2024). 3D Hough Transform-based left ventricle 3D object classification for cardiomyopathy diagnosis. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. DOI: 10.1109/MLSP58920.2024.10734784.

Heiberg, E., Sjögren, J., Ugander, M., Carlsson, M., Engblom, H., and Arheden, H. (2010). Design and validation of Segment - freely available software for cardiovascular image analysis. *BMC Med. Imaging*, 10(1):1:1–13. DOI:

10.1186/1471-2342-10-1.

ISO, C. S. (2018). Ergonomics of human-system interaction — part 11: Usability: Definitions and concepts. Standard, International Organization for Standardization. Available at:`https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en`.

Izquierdo, C., Casas, G., Martin-Isla, C., Campello, V. M., Guala, A., Gkontra, P., Rodríguez-Palomares, J. F., and Lekadir, K. (2021). Radiomics-based classification of left ventricular non-compaction, hypertrophic cardiomyopathy, and dilated cardiomyopathy in cardiovascular magnetic resonance. *Frontiers in Cardiovascular Medicine*, 8:764312:1–764312:10. DOI: 10.3389/fcvm.2021.764312.

Katti, G., Ara, S. A., and Shireen, A. (2011). Magnetic resonance imaging (mri)–a review. *International journal of dental clinics*, 3(1):65–70. Available at:`https://www.researchgate.net/publication/279471369_Magnetic_resonance_imaging_MRI_-_A_review`.

Klug, B. (2017). An overview of the system usability scale in library website and system usability testing. *Weave: Journal of Library User Experience*, 1(6). DOI: 10.3998/weave.12535642.0001.602.

Kowalczyk, K. and Szandala, T. (2024). Enhancing SEO in single-page web applications in contrast with multipage applications. *IEEE Access*, 12:11597–11614. DOI: 10.1109/ACCESS.2024.3355740.

Lewis, J. and Sauro, J. (2009). The factor structure of the system usability scale. In *Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International*, volume 5619, pages 94–103. DOI: 10.1007/978-3-642-02806-$9_1$2.

Maisch, B., Noutsias, M., Ruppert, V., Richter, A., and Pankuweit, S. (2012). Cardiomyopathies: classification, diagnosis, and treatment. *Heart Failure Clinics*, 8(1):53–78. DOI: 10.1016/j.hfc.2011.08.014.

Malik, A., Sheraz, M., and Sulaiman, S. (2013). Towards the development of an interface model for information visualization in multiple electronic health records. In *2013 International Conference on Computer Medical Applications (ICCMA)*, pages 1–5. DOI: 10.1109/ICCMA.2013.6506164.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18:50–60. DOI: 10.1214/aoms/1177730491.

Maron, B. J., Towbin, J. A., Thiene, G., Antzelevitch, C., Corrado, D., Arnett, D., Moss, A. J., Seidman, C. E., and Young, J. B. (2006). Contemporary definitions and classification of the cardiomyopathies: an American Heart Association scientific statement from the Council on Clinical Cardiology, Heart Failure and Transplantation Committee; Quality of Care and Outcomes Research and Functional Genomics and Translational Biology interdisciplinary working groups; and Council on Epidemiology and Prevention. *Circulation*, 113(14):1807–1816. DOI: 10.1161/CIRCULATIONAHA.106.174287.

Mason, D. (2011). Su-e-t-33: pydicom: an open source dicom library. *Medical Physics*, 38(6Part10):3493–3493. DOI: 10.1118/1.3611983.

McLeod, S. (2023). Likert Scale Questionnaire: Examples & Analysis. Available at:`https://www.simplypsychology.org/likert-scale.html#`.

Myers, J. and Copeland, R. (2015). *Essential SQLAlchemy: Mapping Python to Databases*. " O'Reilly Media, Inc.", 2 edition. Book.

Nisanbayev, Y., Na, H., Lim, D., and Ko, F. (2009). Designing an electronic medical records system using design patterns. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, pages 1410–1415. DOI: 10.1145/1655925.1656185.

Pallets (2023). Flask. Available at:`https://flask.palletsprojects.com/en/stable/`.

Pirruccello, J. P., Bick, A., Wang, M., Chaffin, M., Friedman, S., Yao, J., Guo, X., Venkatesh, B. A., Taylor, K. D., Post, W. S., Rich, S., Lima, J. A. C., Rotter, J. I., Philippakis, A., Lubitz, S. A., Ellinor, P. T., Khera, A. V., Kathiresan, S., and Aragam, K. G. (2020). Analysis of cardiac magnetic resonance imaging in 36,000 individuals yields genetic insights into dilated cardiomyopathy. *Nature Communications*, 11(1):2254:1–2254:10. DOI: 10.1038/s41467-020-15823-7.

Rank, N., Pfahringer, B., Kempfert, J., Stamm, C., Kühne, T., Schoenrath, F., Falk, V., Eickhoff, C., and Meyer, A. (2020). Deep-learning-based real-time prediction of acute kidney injury outperforms human predictive performance. *npj Digital Medicine*, 3(1):139:1–139:12. DOI: 10.1038/s41746-020-00346-8.

Robbins, S. L., Cotran, R. S., Kumar, V., Abbas, A. K., Fausto, N., and Aster, J. C. (2010). *Robbins and Cotran, Pathological Bases of Diseases*. Elsevier Editora Ltda., 8 edition. Book.

Rodrigues, M. E. M., Moura, K. H. S., Branco, K. C., Lelli, V., Viana, W., Andrade, R. M. C., and Santos, I. S. (2023). Exploring user experience and usability of mhealth applications for people with diabetes: An evaluation study using ueq and he4eh checklist. *Journal on Interactive Systems*, 14(1):562–575. DOI: 10.5753/jis.2023.3226.

Santos, C. d. S., Garcia, B. C. C., Souza, L. A. M. d., Gomes, M. F., Fernandes, B. S., Esteves, E. A., and Rocha-Vieira, E. (2023). Development, usability, formative assessment, and story immersion of nutrigame, a mhealth nutrition education app. *Journal on Interactive Systems*, 14(1):470–480. DOI: 10.5753/jis.2023.3274.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611. DOI: 10.2307/2333709.

Silva, L. S., Aranha, R. V., Ribeiro, M. A. d. O., Nakamura, R., and Nunes, F. L. S. (2020). Exploring visual attention and machine learning in 3d visualization of medical temporal data. In *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 146–151. DOI: 10.1109/CBMS49503.2020.00035.

Simmons, C. (2013). WebPAX PACS and Universal Viewer v8 Medical Imaging System announced by Heart IT. Available at:`https://enewschannels.com/webpax-pacs-and-universal-viewer-v8-medical-imaging-system-announced-by-heart-it/`.

Stephens, M. A. (1974). Edf statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69:730–737. DOI: 10.1080/01621459.1974.10480196.

The PostgreSQL Global Development Group (2023). Postgresql. Available at:`https://www.postgresql.org/`.

W3C, W. W. W. C. (1999). Web characterization terminology and definitions sheet. Available at:`https://www.w3.org/1999/05/WCA-terms/`.

Wang, S., Zhao, Z., Ouyang, X., Wang, Q., and Shen, D. (2024). Interactive computer-aided diagnosis on medical image using large language models. *Communications Engineering*, 3(133). DOI: 10.1038/s44172-024-00271-8.

Wise Coders GmbH (2023). Dbschema. Available at:`https://dbschema.com/`.

Yanase, J. and Triantaphyllou, E. (2019). A systematic survey of computer-aided diagnosis in medicine: past and present developments. *Expert Systems with Applications*, 138:112821:1–112821:25. DOI: 10.1016/j.eswa.2019.112821.

Zhou, W., Yang, Y., Yu, C., Liu, J., Duan, X., Weng, Z., Chen, D., Liang, Q., Fang, Q., Zhou, J., Ju, H., Luo, Z., Guo, W., Ma, X., Xie, X., Wang, R., and Zhou, L. (2021). Ensembled deep learning model outperforms human experts in diagnosing biliary atresia from sonographic gallbladder images. *Nature Communications*, 12(1):1259:1–1259:14. DOI: 10.1038/s41467-021-21466-z.