EVOTION

727521 – EVOTION

DELIVERABLE No: D6.1

Integrated EVOTION platform v1

Authors: Kostas Giannakakis, Ilias Papas, Giorgos Giotis, Nikos Dimakopoulos (ATC), Ioannis Basdekis, Konstantin Pozdniakov (CITY), Michalis Smyrlis (EMP), Panagiotis Katrakazas (ICCS), Niels Henrik Pontoppidan (OTC), Marco Anisetti, Valerio Bellandi, Fulvio Frati (UNIMI)

Dissemination level	
PU	PU - Public

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 727521

Project acronym and GA no	EVOTION- 727521
Project full Title	EVidenced based management of hearing impairments: Public health pOlicy making based on fusing big data analytics and simulaTION.
Project Type	RIA
Start date – end date	01.11.16 – 31.10.19
Website	www.h2020EVOTION.eu
Deliverable type	R + DEM, Report and Demonstrator
Delivery date	M24 (31 OCT 2018) Actual submission date: 5 FEB 2019
Authors:	Kostas Giannakakis, Ilias Papas, Giorgos Giotis, Nikos Dimakopoulos (ATC), Ioannis Basdekis, Konstantin Pozdniakov (CITY), Michalis Smyrlis (EMP), Panagiotis Katrakazas (ICCS), Niels Henrik Pontoppidan (OTC), Marco Anisetti, Valerio Bellandi, Fulvio Frati (UNIMI)
Reviewers:	George Spanoudakis (CITY), Lukas Tietz (OTC)
Contact:	Nikos Dimakopoulos (**n.dimakopoulos@atc.gr**)
To be cited as:	Giannakakis K., Papas I., Giotis G., Dimakopoulos N., Basdekis I., Pozdniakov K., Smyrlis M., Katrakazas P., Pontoppidan N., Anisetti M., Bellandi V., Frati F. (2019), Integrated EVOTION platform v1, Deliverable D6.1 to the EVOTION-727521 Project funded by the European Union, ATC, Greece
Subject and keywords	This report presents the first prototype version of the Integrated EVOTION platform. Keywords: integrated platform, hearing aids, interface, wearable sensor, peripheral devices, API.
Disclaimer:	This document's contents are not intended to replace consultation of any applicable legal sources or the necessary advice of a legal expert, where appropriate. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user, therefore, uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors' view.

Table of Contents

Abstract vi

1 Overview 1

1.1 EVOTION Solution 1

1.2 Purpose and Scope 1

1.3 Approach for this Work Package and Relation to other Work Packages 1

2 System Architecture 2

2.1 Architecture Diagram 2

2.2 EVOTION Components 4

3 EVOTION Hospital System 7

3.1 User interface design and implementation 8

3.2 Integration of Web Interface layer with the EVOTION REST Service layer 9

3.3 EVOTION API 10

4 Components’ Communication 12

4.1 Use Case Scenarios 12

4.2 PHPDM Model Specification, Integrated with Transformation Tool 12

5 Physical Architecture 14

5.1 EVOTION Solution Deployment 14

5.2 System Requirements & Infrastructure 15

6 Demonstrator 18

6.1 Accessibility 18

7 Conclusion and Future Steps 19

References 20

APPENDIX 1 - Web Interface layer of the EVOTION REST Service layer 22

APPENDIX 2 - Additional Use Case Scenarios for the Components’ Communication 38

APPENDIX 3 - EHS Detailed web interface design 44

APPENDIX 4- EVOTION API in details 49

Data Repository API 49

Evotion Hospital System (EHS) API 74

Big Data Analytics Engine API [1] 85

Security API 101

Table of Figures

Figure 1: Representation of the architectural data flow vi
Figure 2: EVOTION Platform Architecture 2
Figure 3: AuditBase and EHS transmit anonymised data in the EVOTION System. 8
Figure 4: Example of interface simplification: Editing ear disease – surgery information: AuditBase (on the left) vs EHS (on the right). 8
Figure 5: Reutilisation of same interface aesthetics introduced in previous developments for consistency: EVOTION Repository Dashboard view of a MOCA answer (on the left) vs EHS view of a patient personal information (on the right) 9
Figure 6: Scenario 1. Retrieval of HA usage data 12
Figure 7: PHPDM integration- Data Flow 13
Figure 8: EHS and EDR architecture 14
Figure 9: EHS sequence diagram for the normal flow of a REST function. 15
Figure 10: End-user's log in 22
Figure 11: All available patients presented in Zebra striping tabular form. Selecting the desired link ID from the list will automatically make it the active selection (patient's specific information to be displayed). Key identification properties are blanked out 23
Figure 12: Creation of a new patient (part A: basic info) 24
Figure 13: Creation of a new patient (part B: details). 25
Figure 14: Portion of current patient's (EX20**********) personal details. Key identification properties are blanked out to preserve the patient's anonymity 26
Figure 15: Portion of current patient's (EX20**********) medical history (Part A: Otoscopy – Audiometry, Ear disease – Surgery, Balance - Hearing). 27
Figure 16: Portion of current patient's (EX20**********) medical history (Part B: Ext. history – Aetiology, Tinnitus - Other history, Lifestyle – Communication, Outcome - Other). 29
Figure 17: View of patient's audiogram (air conductor) data. 29
Figure 18: View of patient's audiogram (bone conductor) data. 30
Figure 19: Editing patient's medical history (Otoscopy). 31
Figure 20: Editing patient's medical history (Ear disease - surgery). 32
Figure 21: Editing patient's medical history (Balance - hearing). 33
Figure 22: Editing patient's medical history (Ext. history - Aetiology). 34
Figure 23: Editing patient's medical history (Tinnitus – Other history). 35
Figure 24: Editing patient's medical history (Lifestyle - Communication). 36
Figure 25: Editing patient's medical history (Outcome - Other). 37
Figure 26: Adding an audiogram: selecting the proper NOAH xml file for upload. 37
Figure 27: Scenario 2. Sudden Deterioration of Hearing 38
Figure 28: Scenario 3. Assessment of initial follow up policies 39
Figure 29: Scenario 4. NIHL Protection of people with hearing impairments from the harmful effects.... 40
Figure 30: Scenario 5. Performing audiological tests through HA (Analytic workflows are pre-loaded and executed) 41
Figure 31: Scenario 6. Prognosis of low HA usage 41
Figure 32: Scenario 7. Mobile hearing coach 42
Figure 33: Scenario 8. Protection of people with hearing impairments from the harmful effects of loud noise 43
Figure 34: End-user's log in. 44
Figure 35: All available patients presented in Zebra striping tabular form. Selecting the desired link ID from the list will automatically make it the active selection (patient's specific information to be displayed)... 44

Figure 36: Creation of a new patient (part A: basic info). 44
Figure 37: Creation of a new patient (part B: details) 45
Figure 38: Portion of current patient’s personal details. Key identification properties will be blanked out to preserve the patient's anonymity. 45
Figure 39: Editing patient’s medical history (Otoscopy). 45
Figure 40: Editing patient’s medical history (Ear disease - surgery). 46
Figure 41: Editing patient’s medical history (Balance - hearing). 46
Figure 42: Editing patient’s medical history (Ext. history - Aetiology). 47
Figure 43: Editing patient’s medical history (Tinnitus – Other history). 47
Figure 44: Editing patient’s medical history (Lifestyle - Communication). 47
Figure 45: Editing patient’s medical history (Outcome - Other). 48
Figure 46: End-user’s log in info. 48

List of Tables

Table 1: List of Abbreviations v
Table 3: EHS Virtual Servers specification 16
Table 4: EDR Virtual Servers specification 17
Table 5: Tables of Evotion LDAP database 104

List of Abbreviations

Table 1: List of Abbreviations

API	APPLICATION PROGRAMMING INTERFACE
ATC	ATHENS TECHNOLOGY CENTER SA
BDA	BIG-DATA ANALYTICS
CITY	THE CITY UNIVERSITY LONDON
DSS	DECISION SUPPORT SYSTEM
EDR	EVOTION Data Repository
EHS	EVOTION Hospital System
EMApp	EVOTION Mobile App
EMP	EMPELOR GMBH
ERESL	EVOTION REST Service Layer
GST	Guys St. and Thomas NHS Trust
HA	HEARING AID
LDAP	Lightweight Directory Access Protocol
OTC	OTICON A/S
PHPDM	Public health policy decision making models
PKI	Public Key Interface
REST	Representational State Transfer
SDK	Software Development Kit
UCL	UNIVERSITY COLLEGE LONDON
UNIMI	UNIVERSITA DEGLI STUDI DI MILANO
WP	WORK PACKAGE

Abstract

Deliverable D6.1 is a description of the initial prototype of the integrated EVOTION platform. The document aims to underline that the module integration ensures the interoperable and extensible integrated systems based on open architecture principles. It focuses on the integration of the different components, tools and applications developed by WP3, WP4 and WP5 into a single EVOTION solution.

Initially, the system architecture and its components are described shortly, based on the related work package (WP2). There is also a description of the implemented specification models and of the use case scenarios. Within chapter 5, the virtual environment of the EHS and EDR architecture is presented, as well as the design and the implementation of the interfaces between the different components and modules. A reference of the technologies used by them is also included.

The integrated EVOTION platform is under continuous testing from the first day. The first integrated version will be followed by a second one to incorporate the early results of the validation phase in the final validations.

In Figure 1 there is an abstract representation of the architectural data flow with the currently available parts of the system. Taking into consideration that the patient side (left) and the medical side (right bottom) are operational and receive updates for more stability, from time to time, second phase development is focused on the policy makers side (right top) to get the full EVOTION platform functionalities.

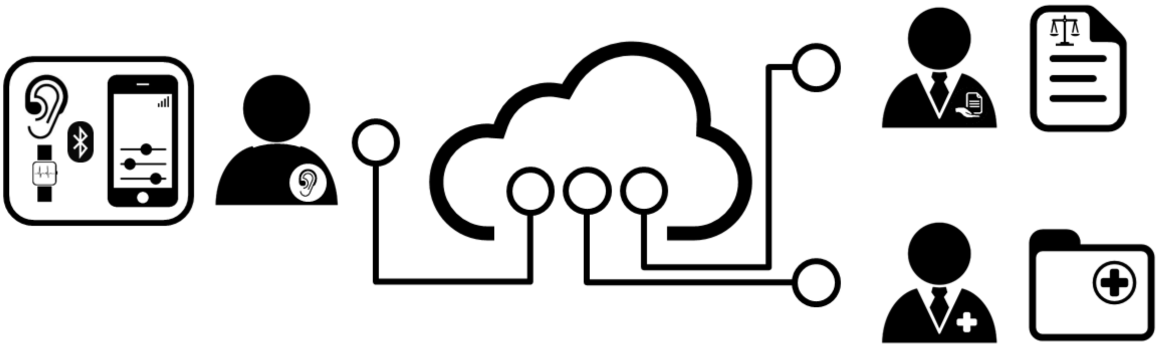


Figure 1: Representation of the architectural data flow

1 Overview

1.1 EVOTION Solution

EVOTION project aims to utilize big datasets (produced by sensors and HAs used by HL patients) in forming holistic HL management policies.

EVOTION develops an integrated platform supporting: (a) the analysis of the above datasets to enable the identification of causal and other effects amongst them, (b) policy decision making focusing on the selection of effective interventions related to the holistic management of HL, based on the outcomes of (a) and the formulation of related public health policies, and (c) the specification and monitoring of such policies in a sustainable manner. To achieve this aim, EVOTION also brings together public health policy organizations, experts and authorities supporting the formation of the targeted policies and the validation of the EVOTION support for it.

EVOTION, far from being a simple container for the individual modules, is a coherent application, where several different components reside and collaborate in harmony. The first prototype is a proof of concept to the EVOTION end users. It encapsulates most of the underlying technologies and gives a clear and easy to use graphical interface, exposing every available feature so far.

1.2 Purpose and Scope

The purpose of this document is to describe the first integrated prototype of the EVOTION project. This version is the first working version of the prototype and will act as the test-bed for the EVOTION stakeholders to experience with the EVOTION provisions and assess the concepts and knowledge conveyed by the project.

1.3 Approach for this Work Package and Relation to other Work Packages

WP6 aims at providing the architectural and implementation aspects for the delivery of the EVOTION components integration in a unified platform. The design of the EVOTION is driven from the requirements definition in WP2 [13] .

Following the implementation of the individual modules in WP3 [10], WP4 [16] and WP5 [1] [4] [6] [7] [8] [12] [15] , this WP delivers an integrated view of the EVOTION to act as the testbed for building the EVOTION pilots in WP7 [3] and evaluating them in real life scenarios. The decisions presented in this deliverable are subject to refinements and modifications, based on the progress of the technical work packages, as well as the validation and evaluation phases. Note that the part related to the EDR integration is more detailed, since this part is more mature and at this stage the integration was focused mainly on this.

2 System Architecture

2.1 Architecture Diagram

The EVOTION platform was developed based on the architecture design presented in D2.2 [13]. The architecture is structured in logical layers, four of them grouping components with homogeneous feature and one vertical layer (Security layer)- meant to be integrated and distributed over most actual components.

The logical layers grouping the components are: Data Acquisition, Data Ingestion and Execution, PHPDM Modelling, Visualization and Security Layer.

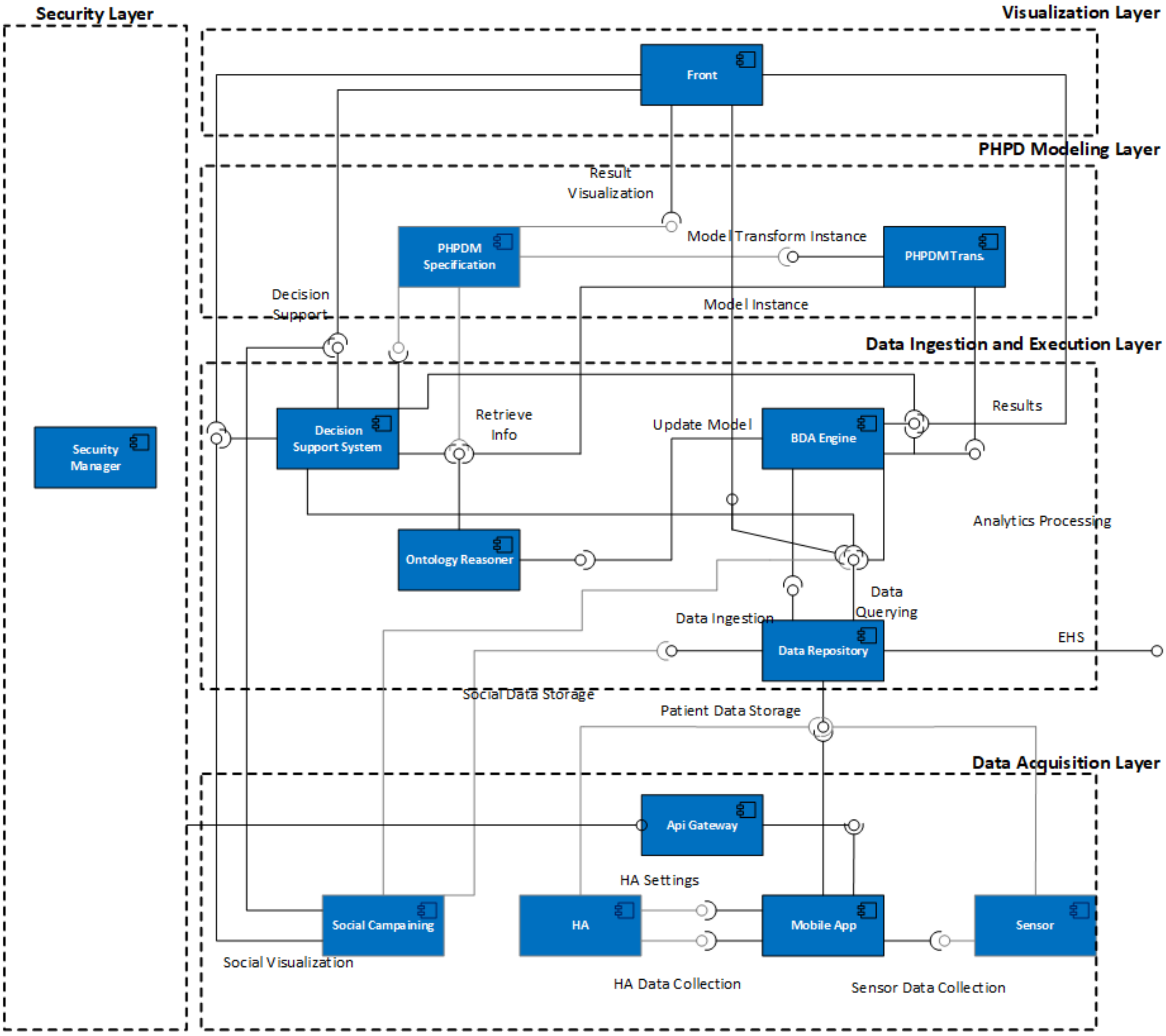


Figure 2: EVOTION Platform Architecture

Data Acquisition layer: All components in charge of gathering runtime data for the EVOTION platform are grouped. These ones represent data collected from the HA devices, the wearable sensors, the mobile app, and social campaigns. More specific this layer includes the components:

a. Hearing Aids (HA)

b. Sensor
c. MobileApp
d. Social Campaigning

Data ingestion and Execution layer: Here the data collected by Data Acquisition layer, are stored and used for logical operations. Operations on data could be either based on analytics (e.g. queries, statistical analyses, machine learning, etc.) or driven by documents supporting decisions, in both cases they are informed by an ontology-oriented representation of policy models.

This layer includes the following components:

a. Data Repository
b. BDA Engine
c. Ontology Reasoner
d. Decision Support System

PHPDM Modelling layer: It is responsible to handle the process of PHPDM model definition and the relative analytic definition starting from the specification of *declarative analytics* produced by PHPDM Specification tool using EVOTION models and language, to the *procedural analytics* produced by the PHPDM Transformation tool. It interacts with the Data Ingestion and Execution Layer for i) providing procedural analytics to be executed by the BDA Engine, ii) retrieving suggestions from the Decision Support System and iii) retrieving models form the Ontology Reasoner.

It includes the following components:

a. PHPDM Specification tool
b. PHPDM Transformation tool

Visualization layer: provides the frontend to the EVOTION platform, offer the results of operations, and drives social campaigns. The only component included here is **EVOTION Front**. This component aims at helping EVOTION users in interaction with the EVOTION Platform and providing visualization facilities mainly for showing the results of a given analytic task. It interacts with Data Acquisition Layer for showing the results of the Social Campaigning, with the Data Ingestion and Execution Layer for visualization of analytics and for suggestion obtained by the Decision Support System, and with Modelling layer for providing interface to the model specification.

Security layer: It is showed as a vertical logical layer with no specific interfaces and a unique generic Security manager. This representation means that security properties are spread on several functional components, possibly coordinated through global policies. Specific security-oriented components will be adopted at network or physical levels, as well as component configurations, activity monitoring, and technology solutions will be adopted and described in following deliverables.

2.2 EVOTION Components

The components comprising the EVOTION platform are briefly described in the following paragraphs. Since some components have a detailed description in terms of technology and functionality in separate deliverables, only a short description is presented in this section in order to help the reader understand the architecture and physical diagrams.

- **HA**[15] **:** It is the prototype EVOTION hearing aids which the provides the HA users with compensation for their hearing loss after having being adjusted by the hearing care professionals. When EVOTION HAs are used and connected to the Mobile App, the hearing aids continuously transmit sound environment data (e.g. how loud, how much noise, etc) and HA controls (volume adjustment and restricted settings adjustment) through the HADataCollection Interface and HASetting interface - also through the HASetting interface the default settings and default volume can be set from the EVOTION platform.
- **Mobile App** [7] **:** It is used in the data gathering process of the EVOTION platform as it will be the main tool to collect information from the EVOTION HAs, the environment and the HA users. It is the main way of interaction with the HA users by getting input by them as well as serving them with relevant information according to their needs. Mobile App component is also used to feed the EVOTION Platform with its collected data.
- **Api Gateway:** EVOTION platform includes many components, communicating each other. To avoid service competition and malfunction of our system, an additional layer is added ensuring the smooth operation of the system, as well as to ensure the integrity of the EVOTION platform. As such, EVOTION Api- Gateway is implemented to provide a unique gateway to the security and the mobile engine. It can be further used for the integration of the data repository services and BDA Engine.
  These services are implemented following a microservice approach. The EVOTION Api-Gateway serves as a proxy service among the security, the mobile engine and the data repository services. Every call between these services is accomplished via this gateway. This module serves as the front door of the system and providing authentication and security for each component, via the Secure Layer, and dynamic routing requests to different backend services as needed.
- **Sensor** [6] **:** It is the wearable/embedded sensors module component, which offers the physiological data measurements collection to Mobile App
- **Social Campaigning**: This component runs campaigns on social media, with the purpose of informing the public about policies generated by PHPDM models and subsequently collect and analyse feedback from different stakeholder groups for these policies.
- **BDA Engine** [1]: It realizes executions of queries and analytics tasks (AnalyticsProcessing interface). It is responsible for the data processing, offering feedback towards Ontology Reasoner component and Visualization Layer.
  The BDA Engine is based on a set of sub-components:
  - **BDA Infrastructure**: Big Data Processing infrastructure based on Apache Hadoop framework enhanced with additional components, like Apache Oozie, to support analytic workflow execution and libraries to supports statistical and data mining tasks, such as Mlib
  - **Task Catalogue**: list of Tasks (i.e., Processing, Statistical and Data Mining) for which an executable implementation is available in the BDA. For instance, the Task Catalogue will include Spark ANOVA as an implementation of a task ANOVA defined in the PHPDM model instance
  - **Workflow Catalogue**: list of EDAWs related to a corresponding PHPDM policy model instance. These workflows are composed of tasks appearing in the Task Catalogue and a coded logic driving the task execution. An EDAW can be scheduled according to PHPDM

preference. It contains also a set of pre-defined EDAWs that are not derived from a PHPDM model instance, but are needed to cope with clinical specific scenarios.

  - **Management/Catalogue Backend**: management backend for the BDA Infrastructure based on Ambari and for the Task and Workflow catalogues. It is used mainly for administrative management of the BDA Engine and Catalogues management.
  - **API Module**: RESTFUL APIs for the EVOTION components interacting with the BDA

- **Data Repository** [4]: It consolidates data from various sources (mobile application, AuditBase, EVOTION Hospital System), provides the storing facilities to all the EVOTION components requiring it and directly interacts with BDA Engine to provide data for the execution of the analytic tasks. More specifically, it receives hearing aid usage, biosensor and mobile application data from EMApp, and anonymised patients' medical information (medical questionnaires, and clinical data). Data repository is intended to reliably store and process huge volume of information (EVOTION Bjg Data). For those reasons it's based on Apache Hadoop ecosystem designed for Big Data operations. In particular, data is stored in HBase - non-relational distributed database. It automatically replicates information and distributes it among cluster components for reliability purpose. In order to exploit advantages of traditional relational databases, the Phoenix Interface was incorporated to support SQL-like queries. Additionally, all accumulated data is backed up on a daily basis. Currently data repository can be accessed via developed REST interface
- **Ontology Reasoner**: This component is responsible for creating a new ontology model instance according to the user's input via the specification tool and checking whether the ontology model instance is consistent. The ontology model instance is saved for auditing purposes. This component also reads the current version of the ontology model and provides the available algorithms and their configuration options to the specification tool. If there are changes to the available algorithms of the ontology model, the updated ontology model is uploaded and read by the ontology reasoner and the specification tool is updated accordingly.
- **Decision Support System** [10]: The purpose of the EVOTION Decision Support System (DSS) is to provide data retrieval and summarization functionalities for text-mining related tasks, aimed mainly at PHPD makers and clinicians, in order to define and produce decision (in terms of suggesting sets of interventions) scenarios, based on the information produced by the EVOTION platform.
- **PHPDM Specification tool** [10]: It allows EVOTION end user to specify analytics tasks using PHPDM Specification Language and Models in a declarative way to be – at a later step – transformed into procedural analytics to be executed by the BDA Engine, , As such, provides necessary interaction elements (Public Health Policy Decision Models e-service) to support this process (i.e., interfaces to create a decision support model instances – policies and workflows - and manage the created model instances).
- **PHPDM Transformation tool** [10]: It transforms a declarative analytics pipeline written with the EVOTION specification tool, into a procedural set of commands, which is used as an input for the BDA Engine.
- **Front-End Visualization Dashboard:** EVOTION Data Repository and Public Health Policy Decision Models e-service both use a common way of interacting with RESTful services. The EVOTION Dashboard. EVOTION Dashboard provides appropriate mechanisms to support the administration of data been acquired via different medical questionnaires, the mobile registration, and the administration of Policy Models. To serve this purpose, the design of interaction and presentation elements took into account specific requirements and comments received mainly from the EVOTION clinical partners. A user centered approach for the design and implementation of the web GUI elements has been followed, with the aim to develop highly usable services, which can

be easily operated by clinicians, who might not be highly experienced in using computers and similar applications. Following this approach, the introduction of new design features (new functionality) reutilises a consistent look and feel, to simplify the both end-user interaction and inter-component communication. Design principles and key elements of the continuous evolving Dashboard were presented in D5.2.

3 EVOTION Hospital System

As described in DOW, EVOTION system collects “existing and periodically collected patient data (i.e., audiological, medical, clinical and medication, personal and occupational data)”. For this purpose, some of EVOTION clinical partners used pre-existing (to their premises) medical management systems. Thus, they administer healthcare data such as patient’s electronic medical record, with emphasis on issues related to clinical HA fitting, generic ear procedures and diagnostic results, audiograms, vestibular testing and other medical information. UK based clinical partners (University College London, Guys and St Thomas) have been utilising AuditBase^{TM} (AuditData A/S, 2015) as a central client case record database for this reason. AuditBase is an advanced clinic management system for hospital audiology clinics. This legacy system administers among others:

- Basic client's information, name, client number, birthday, ID numbers (such as National Health Security number in the UK).
- Address, phone numbers, and e-mail addresses.
- Medical information such as referring physician, and medical background.
- Audiograms and other hearing test results transferred directly from Noah-compatible measurement modules.
- Hearing aid fitting data stored by Noah-compatible fitting modules.

As for the majority of the clinical partners of the EVOTION consortium (Athens Medical Centre, University of Athens, University College London, Oticon), the EVOTION Hospital System (EHS) developed in the context of EVOTION. It serves as an alternative to the functionality of the AuditBase system (Figure 1). EHS is a web-based system (Figure 2) responsible for medical data collection, to meet the goals of EVOTION project. Subset of both functionality and data tables was transformed into web e-services responsible for the gathering medical data, in a similar format. To this extent, the EHS is a web-based tool that reutilises web templates and other basic elements derived from the EVOTION Dashboard, and supports:

- Secure registration of end-users (clinicians)
- Secure login of end-users (clinicians) giving them access to supported functionality via Role–based access control (RBAC)
- The administration and collection of medical patient data during the clinical pathway of an EVOTION patient. By the clinical pathway we mean the administration of real patients and their basic personal information (administration per patient of otoscopy and audiometry record, ear disease – surgery record, balance – hearing record, hearing history and aetiology record, tinnitus records, lifestyle and communication record, outcome information record)

Section “REST API” presents the REST web services (Figure 3) that have been implemented to connect the platform with the data repository of EHS (i.e., to enable the insertion, deletion, modification and retrieval of data from it).

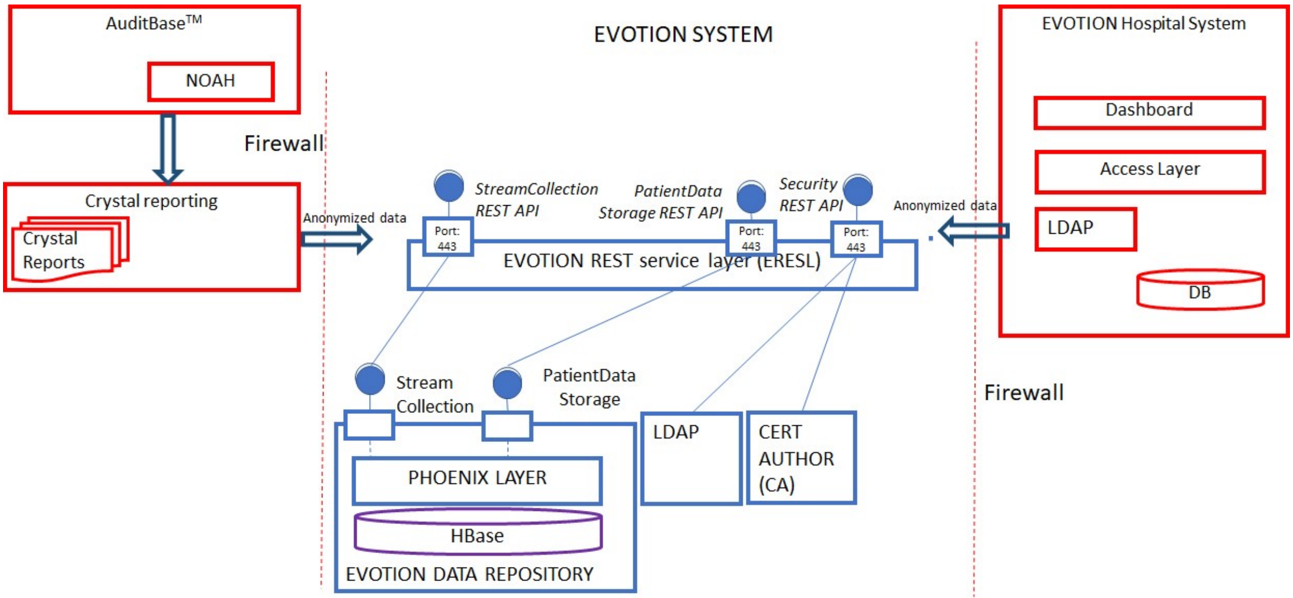
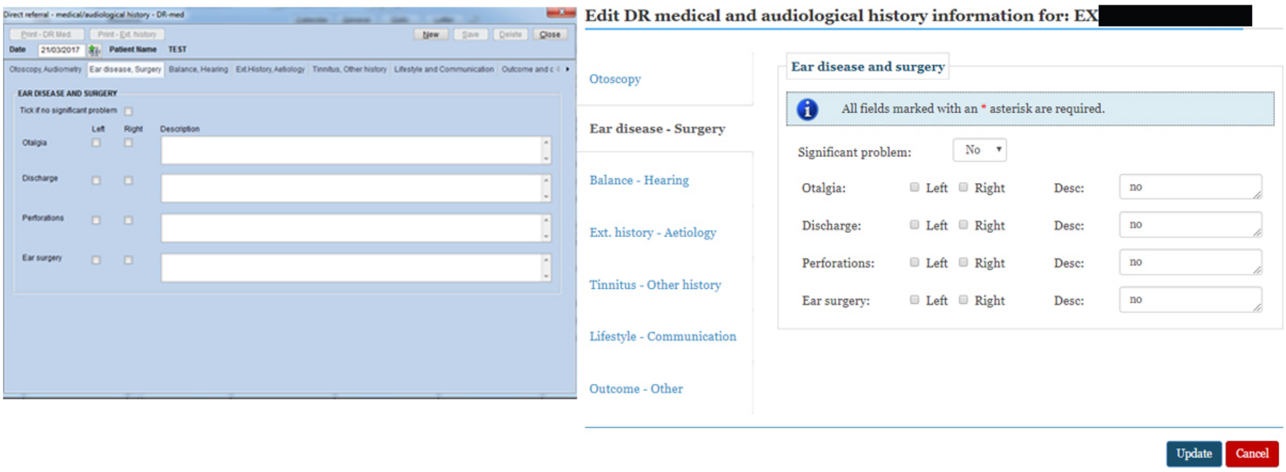


Figure 3: AuditBase and EHS transmit anonymised data in the EVOTION System.

3.1 User interface design and implementation

Since the EHS serves as an alternative to the AuditBase system, the implementation reutilised same database schema, in order to recreate (almost) identical usage scenarios as well as properties of patients' medical history to ensure the acceptability of data been stored by both of those systems. However, to achieve the necessary efficiency and usability levels, AuditBase interaction elements had to be amended. The interface (in comparison to the AuditBase) was (re-)designed providing access to the EHS services. Hence there is a significant reduction of the information load (Figure 4). Moreover, an end-user interface has been implemented, with certain web UI consistency constraints to be introduced in the development of the EDR Dashboard (presented in D5.2 "Data repository and collection components" section 2.3 "EVOTION Dashboard Design" [4]) (Figure 5).



Date 21/03/2017 Patient Name TEST

EAR DISEASE AND SURGERY

Tick if no significant problem □

	Left	Right	Description
Otalgia	□	□	
Discharge	□	□	
Perforations	□	□	
Ear surgery	□	□	

Edit DR medical and audiological history information for: EX[redacted]

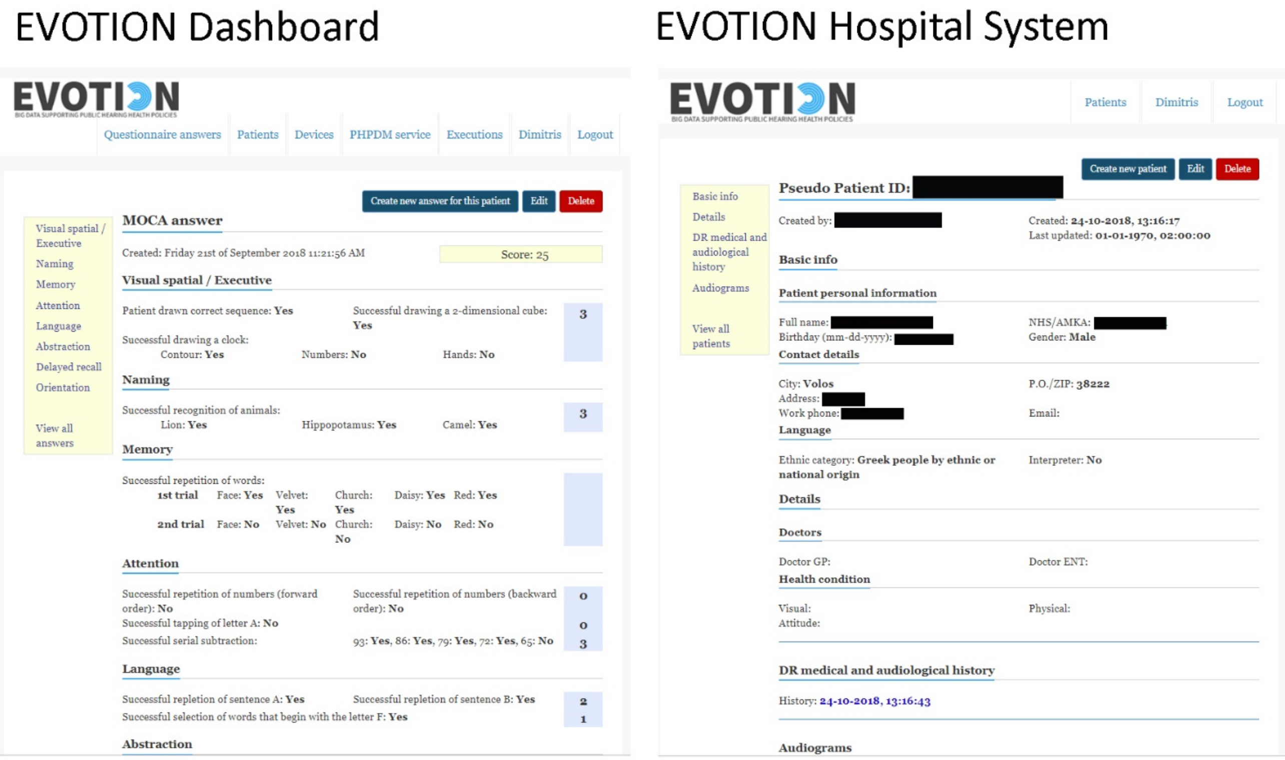
Ear disease and surgery

All fields marked with an * asterisk are required.

Significant problem: No

Otalgia:	□ Left	□ Right	Desc:	no
Discharge:	□ Left	□ Right	Desc:	no
Perforations:	□ Left	□ Right	Desc:	no
Ear surgery:	□ Left	□ Right	Desc:	no

Figure 4: Example of interface simplification: Editing ear disease – surgery information: AuditBase (on the left) vs EHS (on the right).



EVOTION Dashboard

MOCA answer

Created: Friday 21st of September 2018 11:21:56 AM — Score: 25

Visual spatial / Executive

Patient drawn correct sequence: **Yes** — Successful drawing a 2-dimensional cube: **Yes** — 3

Successful drawing a clock:
Contour: **Yes** — Numbers: **No** — Hands: **No**

Naming

Successful recognition of animals: — 3
Lion: **Yes** — Hippopotamus: **Yes** — Camel: **Yes**

Memory

Successful repetition of words:

1st trial	Face: **Yes**	Velvet: **Yes**	Church: **Yes**	Daisy: **Yes**	Red: **Yes**
2nd trial	Face: **No**	Velvet: **No**	Church: **No**	Daisy: **No**	Red: **No**

Attention

Successful repetition of numbers (forward order): **No** — Successful repetition of numbers (backward order): **No** — 0
Successful tapping of letter A: **No** — 0
Successful serial subtraction: 93: **Yes**, 86: **Yes**, 79: **Yes**, 72: **Yes**, 65: **No** — 3

Language

Successful repletion of sentence A: **Yes** — Successful repletion of sentence B: **Yes** — 2
Successful selection of words that begin with the letter F: **Yes** — 1

Abstraction

EVOTION Hospital System

Pseudo Patient ID: [redacted]

Created by: [redacted] — Created: **24-10-2018, 13:16:17**
Last updated: **01-01-1970, 02:00:00**

Basic info

Patient personal information

Full name: [redacted] — NHS/AMKA: [redacted]
Birthday (mm-dd-yyyy): [redacted] — Gender: **Male**

Contact details

City: **Volos** — P.O./ZIP: **38222**
Address: [redacted]
Work phone: [redacted] — Email:

Language

Ethnic category: **Greek people by ethnic or national origin** — Interpreter: **No**

Details

Doctors

Doctor GP: — Doctor ENT:

Health condition

Visual: — Physical:
Attitude:

DR medical and audiological history

History: **24-10-2018, 13:16:43**

Audiograms

Figure 5: Reutilisation of same interface aesthetics introduced in previous developments for consistency: EVOTION Repository Dashboard view of a MOCA answer (on the left) vs EHS view of a patient personal information (on the right)

The overall design of the usage scenarios of the EHS ensure the exact and correct sequencing of actions necessary for performing the same tasks as with AuditBase system. Modifications were dictated by the need to ensure that contextual relevance of the presented information and functionality to the different user tasks. On the basis of clinical experts' comments, design mock ups for all functions organized into usage scenarios were prepared. Mock-ups included design alternatives for a secondary navigation bar, and the way information was presented, while most of the comments made referred to the terminology that was used for labelling. As a result, alternative terms were introduced, which according to the clinical experts' opinion were more intuitive and easier for them to understand. On the basis of the above, the design of the EHS was finalized, and the implementation proceeded using the options preferred by the clinical experts. The implemented design targeted towards capturing and addressing in the best possible way user requirements, and the functionality required for the purpose of the study with a user-friendly experience (APPENDIX 3 - EHS Detailed web interface design).

3.2 Integration of Web Interface layer with the EVOTION REST Service layer

For the development of the EHS e-services, as with EVOTION Dashboard (ED), a RESTful architecture was chosen. As previously reported, this approach allows the different layers of functionality (presentation – security policies – e-services - repository) to be separated, and interactions between components to be quite simple (and at the same time easy to reutilise). The RESTful services paradigm allows for the hierarchical organization of all the resources through URI paths that make the discovery of the services rather intuitive. Corresponding to such approach, the development of the web interface layer (e.g., view pages, insertion forms, etc.) reuses components to achieve consistent look-and-feel, and provide easy access to:

- Login (Figure 6)
- All available patients: Viewing – Filtering results (Figure 7)
- Patient’s personal information: Adding-Editing (Figure 8, Figure 9), Viewing (Figure 10),
- DR medical and audiological history: Adding-Editing (Figure 15, Figure 16, Figure 17, Figure 18, Figure 19, Figure 20, Figure 21), Viewing (Figure 11, Figure 12)
- Audiograms data: Uploading (Figure 22), Viewing (Figure 14)

For the development of web user interface elements of the platform, the server-side language PHP (ver. 7) was used. PHP was utilised to produce HTML5 mark-up (i.e., the latest revision of the HTML standard). These were complemented by custom-made CSS 2.1, CSS Bootstrap ver. 3.3.6 style sheets, custom open source libraries (bootstrap-tabs-x v1.3.3 by Kartik Visweswaran) and JQuery.

In D2.2 [13], security aspects have been presented covering the whole range of components form the EVOTION solution. In the context of EHS, and due to the sensitive nature of the data that the EVOTION platform manages, implementation key aspect ensured patients’ anonymity and associated information. The security measures implemented are:

- EHS data security: The connection to any of the EHS REST services and the access to the underlying database is password protected. For this, a REST service enables the authorisation of valid end-users. Hence, for an end-user to get hold of a connection on the ERESL he/she must be authorised through a username and a password that he/she must receive externally to the system.
- User authentication: For a registered end-user to use the EHS, he/she must type in a username and a password. The credentials that are typed in are compared against a list of users that are stored in the user’s database. This process happens dynamically every time the end-user logs into the system.
- Preventing Cross-Site Scripting Attacks:
  - Data validation for all input elements: Ensuring that user input has the correct form.
  - In most cases HTML5 input type="number", multiple selects, and radio options are used (free text inputs are seldomly used). But even when free text inputs occur, sanitization functions are utilised to convert special characters into their respective symbols (e.g., input elements cannot contain and pass SQL statements).
- URL vulnerabilities: POST method utilized in most cases as parameters are not stored in browser history or URL or in web server logs.

For a detailed demonstration of the Web Interface layer with the EVOTION REST Service layer, please refer to APPENDIX 1.

3.3 EVOTION API

For the intended purpose of the implementation of the web services, a token protected REST API has been built around the data repository to allow for the interaction of only registered end-users through the web user interface.

REST is a software architectural style that defines a set of constraints to be used for creating web services. RESTful services lent really well themselves for building systems that treat its data like resources. In the case of EHS, all the tables in the data repository are regarded as resources and for that reason a REST service has been produced for every operation against the EDR tables. The RESTful services paradigm allows for the hierarchical organization of all the resources through URI paths that make the discovery of the services rather intuitive. APPENDIX 4- EVOTION API in details provides a breakdown of all the RESTful services that have been built for the purpose.

4 Components' Communication

4.1 Use Case Scenarios

In this section, the communication among the previous components is presented. The following sequence diagrams are on implementation level and they are based on the user scenarios elaborated in section 6 of Deliverable D2.2 [13]. Within this section a basic scenario (Retrieval of HA usage data) is presented.

Some additional scenarios, demonstrating the interaction among the components of the platform are presented in

Scenario 1. Retrieval of HA usage data

This scenario describes how objective data will be retrieved by the EVOTION platform, allowing evidence based evaluation of HA usage

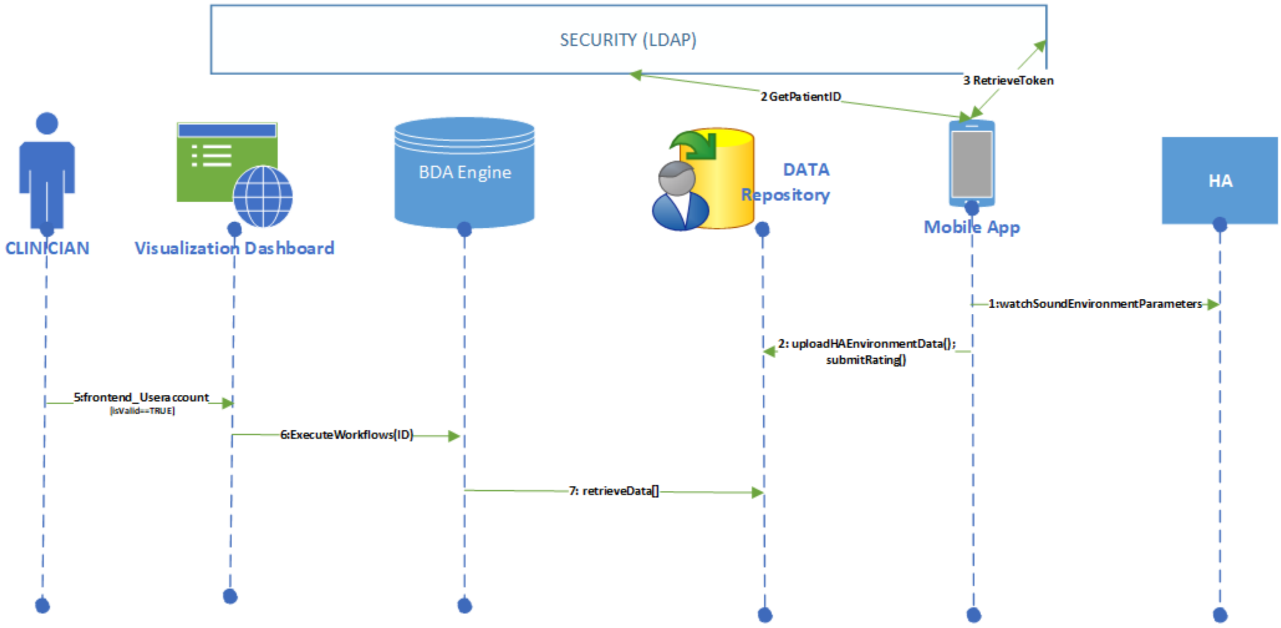


Figure 6: Scenario 1. Retrieval of HA usage data

4.2 PHPDM Model Specification, Integrated with Transformation Tool

This section describes the communication of all the components to achieve the execution of the associated workflows of a Policy on the BDA engine.

In Figure 30, there is a description after the execution of a previous validated policy. This action triggers the PHPDM Transformation Tool [16]. This process will transform the model specification into an executable command suitable for further execution by the BDA Engine, while results of this execution will be return back to the PHPDM e service.

During the whole procedure the following components need to cooperate each other:

Security Layer: This is responsible for the end user authentication
EVOTION Dashboard (ED): This is the end-user interface by which all user-input information is gathered.

PHPDM Specification Tool (PHPDM e-service, EVOTION REST Service Layer): manipulates input data, creates data table clones and retrofits instances of the ontology for later execution purposes (by the BDA engine). **Ontology Reasoner:** IT receives input data from PHPDM e-service, compares them to match the ontology
in use and responds back. The Ontology Reasoner renders this information package to the PHPDM Transformation tool.

PHPDM Transformation tool: It compiles workflow specs into commands suitable for the BDA Engine

BDA engine: It interacts to Ontology Reasoner to utilise the updated ontological model instances, to PHPDM Transformation tool to receive an Data Analytic Workflow in an executable form, to EVOTION Data
Repository to grab cloned store data and execute the analytics on stored data and to update PHPDM e-service tables, from which results will be presented to the end-user via EVOTION Dashboard (ED).

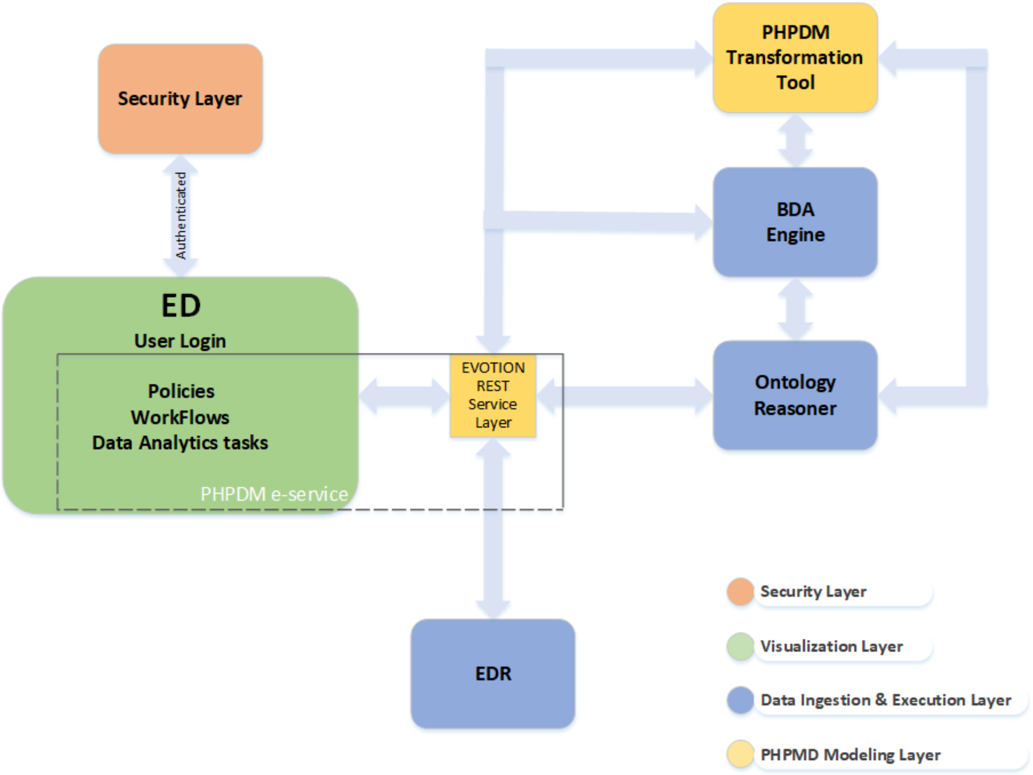


Figure 7: PHPDM integration- Data Flow

5 Physical Architecture

5.1 EVOTION Solution Deployment

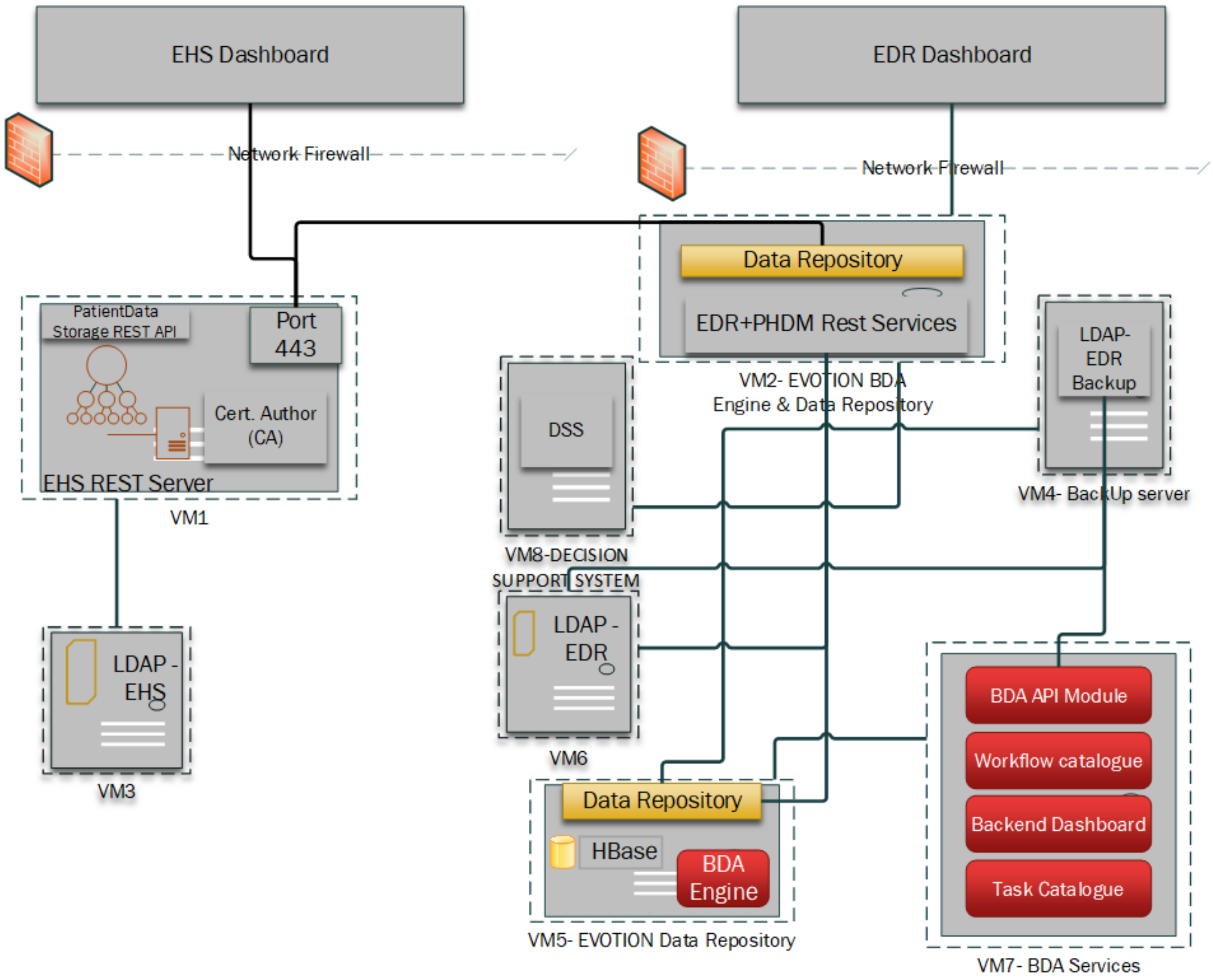


Figure 8: EHS and EDR architecture

EHS and EDR architecture is spread across the virtual environment consisting of eight virtual machines. Dashboard components, in their turn, are located on the client side and provide an interface to access all services needed within the data repository ecosystem. EHS Rest server is located on VM1. It processes the external requests from Dashboard for patients' data storage. This information is anonymised and stored securely in the data repository server. User authentication for both Dashboards is processed by EHS and EDR LDAP servers located on VM3 and VM6 accordingly. In order to support a fast-distributed data retrieval and analysis, BDA engine components were deployed in VM5 and VM7. BDA services receive input data from PHPDM tool residing on VM2 for following information transformation and reasoning.

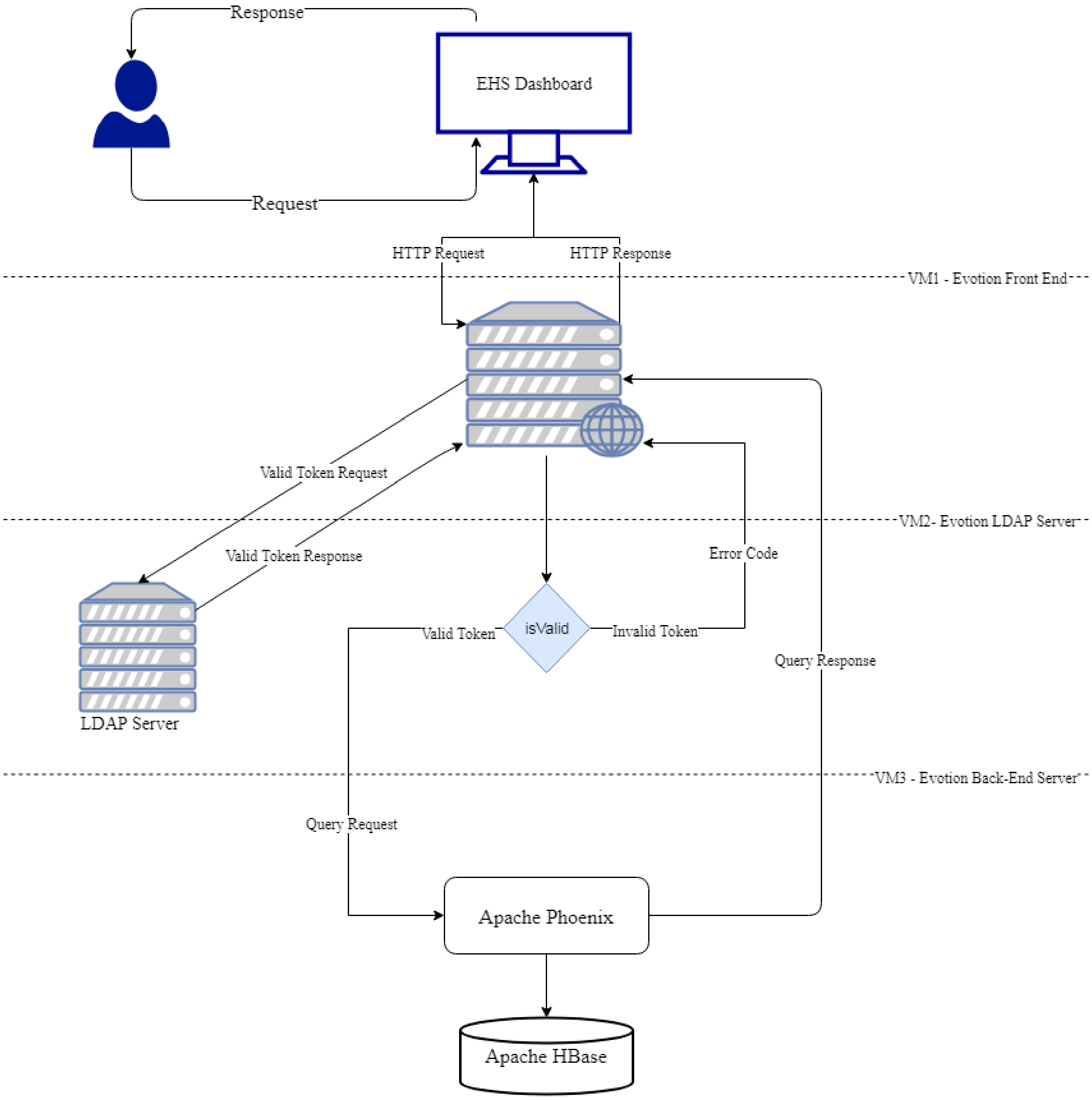


Figure 9: EHS sequence diagram for the normal flow of a REST function.

As with AuditBase, the EHS, prior to the secure transmission of any data to the EDR, anonymises core patient data and encrypts portions or all of them. In this context, encrypts real patients' IDs and their associations with the pseudo ones generated by the ED. The following top-level architecture of the EHS defines the components of the system, and Virtual servers utilised for hosting each component (Figure 2).

5.2 System Requirements & Infrastructure

The minimal hardware and software requirements of the infrastructure to deploy EHS (Figure 2) are estimated and reported in Table 1. This information used to set-up the EHS infrastructure in each location (hospital). The requirements for different Virtual Machines (VMs) are estimated on the basis of possible internet traffic and the amount of data to manage. The computational effort to produce early diagnoses, previsions and treatment plans in the scope of the clinical trials to execute in the project, affects also the requirements. In addition, a small overestimation to guarantee the successful deployment of the system.

The table reports:

- Access level: if a VM is open to public access or not
- Usage: fundamental software components installed
- Operating System: the required operating system to execute all software components
- CPU properties: the suggested type of the CPU that will guarantee the execution of the hosted components
- Disk space: the amount of space required to host the components
- Memory properties: the amount of RAM needed for ensuring the system functionalities will be provided with acceptable response times
- Ports: specific port numbers reserved for hosted services

Table 2: EHS Virtual Servers specification

	VM1	**VM3**
Access	open	protected
Usage	Web server (Tomcat, Apache, PHP) REST server	LDAP Encrypted (Base64) end-users and patients information
Operating System	Ubuntu Server 16.04.3 LTS	Ubuntu Server 16.04.3 LTS
CPU	1.00GHz	1.00GHz
Disk Memory	10Gb	10Gb
RAM	4Gb	4Gb
Open Ports	443, 80, 8080 (open to public)	443 None open to public

Table 3: EDR Virtual Servers specification

	VM2	VM4	VM7
Access	protected	protected	protected
Usage	AMBARI HDP-2.6.2.0 (Hadoop, Hbase, Phoenix, Ambari Metrics)	Backup	BDA services (API, Workflow, Task Catalogue, Backend Dashboard)
Operating System	Ubuntu Server 16.04.3 LTS	Ubuntu Server 16.04.3 LTS	Centos7
CPU	2.30GHz	2.30GHz	2.30GHz
Disk Memory	50Gb	50Gb	40Gb
RAM	12-16Gb	12-16Gb	4Gb
Open Ports	443, 8080 None open to public	443, 8080 None open to public	443, 8080 None open to public

6 Demonstrator

For the purpose of verifying the demonstrator Deliverable D6.1 Integrated EVOTION platform this report contains a video demonstrating the EVOTION platform in use. The video includes a specific demonstration scenario with the following steps:

Step 1: A new patient visits the clinician.

Step 2: The doctor provides to patient the EVOTION devices (mobile phone+ Hearing Aids) and correlates mobile device IMEI and this patient pseudo ID.

Step 3: The patient starts using the EVOTION devices

Step 4: Patient’s data collection

6.1 Accessibility

The related demonstration video is available at the following webpage:

http://h2020evotion.eu/?ddownload=956

7 Conclusion and Future Steps

The first integrated prototype of the EVOTION platform has been successfully deployed and will be evaluated by a group of users during the following period. The implementation and integration is not limited to the current level. Most modules, as well as the platform itself are designed to be flexible and easily customisable. The overall architecture of the system facilitates modifications and makes the platform improvement to be easy, at any point. This will be the aim of the following period: to continue the enhancement of this platform capabilities, make it more stable, more sophisticated and more efficient for the target end users, by plugging additional functionalities and calibrating existing ones.

The future steps include the assessment of the feedback that will be received and also the integration of the next versions of the components as planned according with the description of work.

References

[1] Anisetti, M., Bellandi, V., Cremonini, M., Damiani, E., Dimakopoulos, N., Kokkinakis, P., Papagrigoriou, P., Smyrlis, M., Pontoppidan, N.H., 2018. Big Data Analytics Engine, Deliverable D5.5 to the EVOTION-727521 Project funded by the European Union. University of Milan, Milan, Italy.

[2] Bamiou, D.-E., Dritsakis, G., Damiani, E., Anisetti, M., Cremonini, M., Spanoudakis, G., Meedom, N.H., 2018. GEN- Requirement No. 1, Confidential Deliverable D9.1 to the EVOTION-727521 Project funded by the European Union. University College London, London, United Kingdom.

[3] Bamiou, D.-E., Dritsakis, G., Murdin, L., Koloutsou, N., Bibas, A., Kikidis, D., Cremonini, M., Spanoudakis, G., Meedom, N.H., Pontoppidan, N.H., Dimakopoulos, N., Śliwińska-Kowalska, M., Milas, J., Katrakazas, P., 2017. Study protocol and Ethics Approval Application Report, Confidential Deliverable D7.1 to the EVOTION-727521 Project funded by the European Union. University College London, London, United Kingdom.

[4] Basdekis, I., Ye, B., Spanoudakis, G., Dimakopoulos, N., Kokkinakis, Panagiotis, Smyrlis, M., Katrakazas, P., Pontoppidan, N.H., 2017. Data Repository and Collection Components, Confidential Deliverable D5.2 to the EVOTION-727521 Project funded by the European Union. CITY University, London, United Kingdom.

[5] Cremonini, M., Anisetti, M., Spanoudakis, G., Meedom, N.H., 2017. POPD – Requirement No. 3, Confidential Deliverable D9.3 to the EVOTION-727521 Project funded by the European Union. University of Milan, Milan, Italy.

[6] Dimakopoulos, N., 2017. EVOTION Sensoring interface component, Confidential Deliverable D5.3 to the EVOTION-727521 Project funded by the European Union. Athens Technology Center, Athens, Greece.

[7] Dimakopoulos, N., Kokkinakis, P., Papas, I., Smyrlis, M., Stefanakis, M., 2017b. Mobile application (Public Report), Deliverable D5.4 to the EVOTION-727521 Project funded by the European Union. Athens Technology Center, Athens, Greece.

[8] Dritsakis, G., Bamiou, D.-E., Dimakopoulos, N., Pontoppidan, N.H., Kikidis, D., Murdin, L., Sladen, M., Ye, B., Spanoudakis, G., 2018. Auditory Training component and auditory tests, Deliverable D5.7 to the EVOTION-727521 Project funded by the European Union. University College London, London, United Kingdom.

[9] EVOTION Project Consortium, 2018. First periodic report, Confidential Deliverable D1.3 to the EVOTION-727521 Project funded by the European Union. Oticon A/S, Snekkersten, Denmark.

[10] Katrakazas, P., Androutsou, T., Milas, J., Brdarić, D., Dudarewicz, A., Pawlaczyk-Łuszczyńska, M., Śliwińska-Kowalska, M., Smith, A., Murdin, L., Koloutsou, N., Pontoppidan, N.H., Trenkova, L., Kaloyanova, G., Tsokova, N., Bamiou, D.-E., Dritsakis, G., Bibas, A., Kikidis, D., 2017. Public Health Policy Decision Models (PHPDM) v1, Deliverable D3.1 to the EVOTION-727521 Project funded by the European Union. Institute of Communications and Computer Systems, Athens, Greece.

[11] Katrakazas, P., Meedom, N.H., Koutsouris, D., 2018a. EVOTION Quality Assurance Plan, Confidential Deliverable D1.2 to the EVOTION-727521 Project funded by the European Union. BEL, ICCS, NTUA, Athens, Greece.

[12] Katrakazas, P., Mpostanis, A., Christensen, J.H., 2018. Decision Support System and Simulation Component, Deliverable D5.6 to the EVOTION-727521 Project funded by the European Union. BEL, ICCS, NTUA, Athens, Greece.

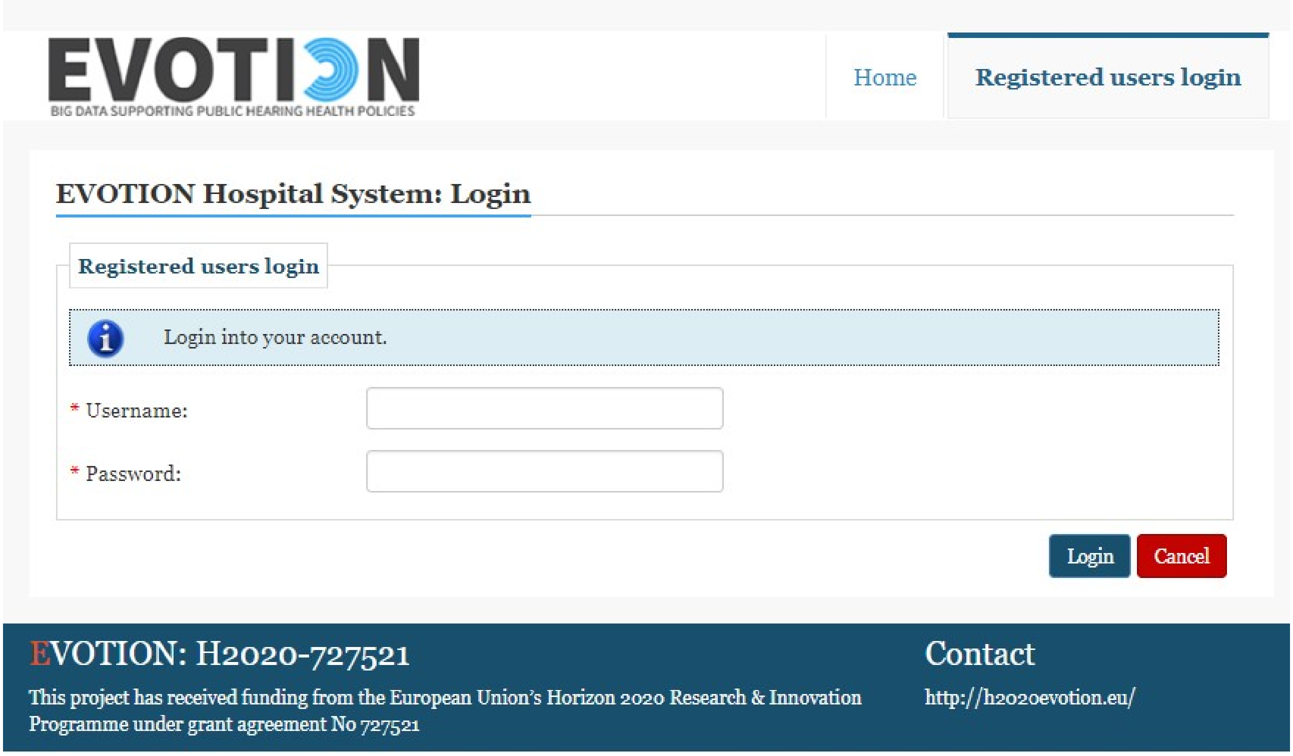
[13] Ye, B., Spanoudakis, G., Prasinos, M., Dimakopoulos, N., Kokkinakis, P., Giotis, G., Papas, I., Papagrigoriou, P., Smyrlis, M., Stefanakis, M., Katrakazas, P., Koutsouris, D., Brdaric, D., Pandzic, Z., Salavarda, D., Urban, M., Milas, J., Pontoppidan, N.H., Trenkova, L., Kaloyanova, G., Tsokova, N., Anisetti, M., Bellandi, V., Cremonini, M., Damiani, E., 2017. EVOTION Architecture and Detailed Design, Deliverable D2.2 to the EVOTION-727521 Project funded by the European Union. CITY University, London, United Kingdom.

[14] Laplante-Lévesque, A., Dimakopoulos, N., Papagrigoriou, P., Gutenberg, J., Brdarić, D., Sifniadis, P., 2017. First Market Analysis and Exploitation Report, Deliverable D8.4 to the EVOTION-727521 Project funded by the European Union. Oticon, Eriksholm Research Centre, Oticon A/S.

[15] Li, X., Pontoppidan, N.H., Hafez, A., 2017. EVOTION hearing aids, Confidential Deliverable D5.1 to the EVOTION-727521 Project funded by the European Union. Eriksholm Research Centre, Oticon A/S, Snekkersten, Denmark.

[16] Prasinos, M., Spanoudakis, G., Basdekis, I., Smith, A., Anisetti, M., Cremonini, M., Katrakazas, P., Androutsou, T., Ioannidou, P., 2017. PHPDM model specification language, Confidential Deliverable D4.1 to the EVOTION-727521 Project funded by the European Union. CITY University, London, United Kingdom.

APPENDIX 1 - Web Interface layer of the EVOTION REST Service layer



EVOTION Hospital System: Login

Registered users login

Login into your account.

* Username:

* Password:

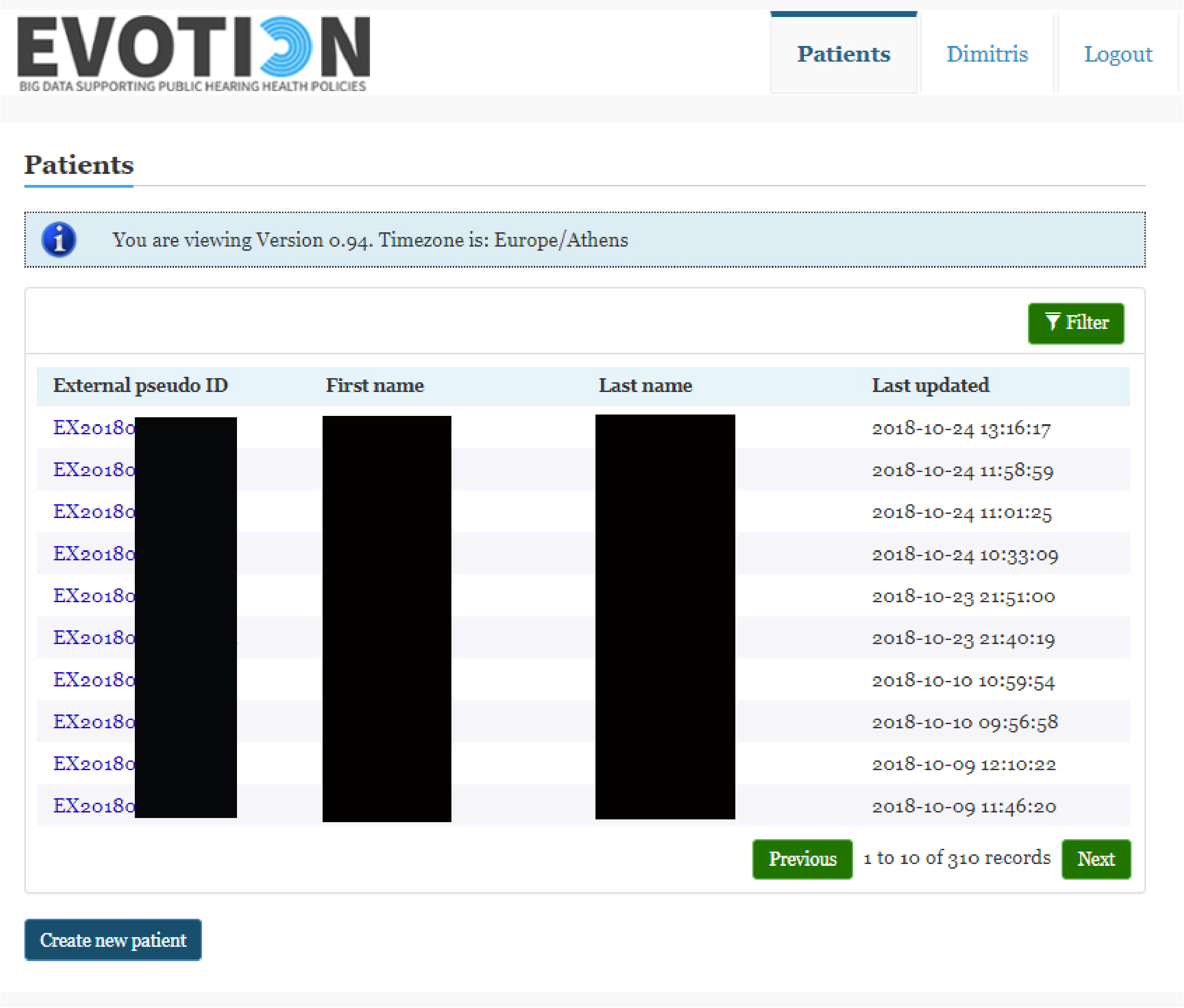
EVOTION: H2020-727521

This project has received funding from the European Union's Horizon 2020 Research & Innovation Programme under grant agreement No 727521

Contact

http://h2020evotion.eu/

Figure 10: End-user's log in



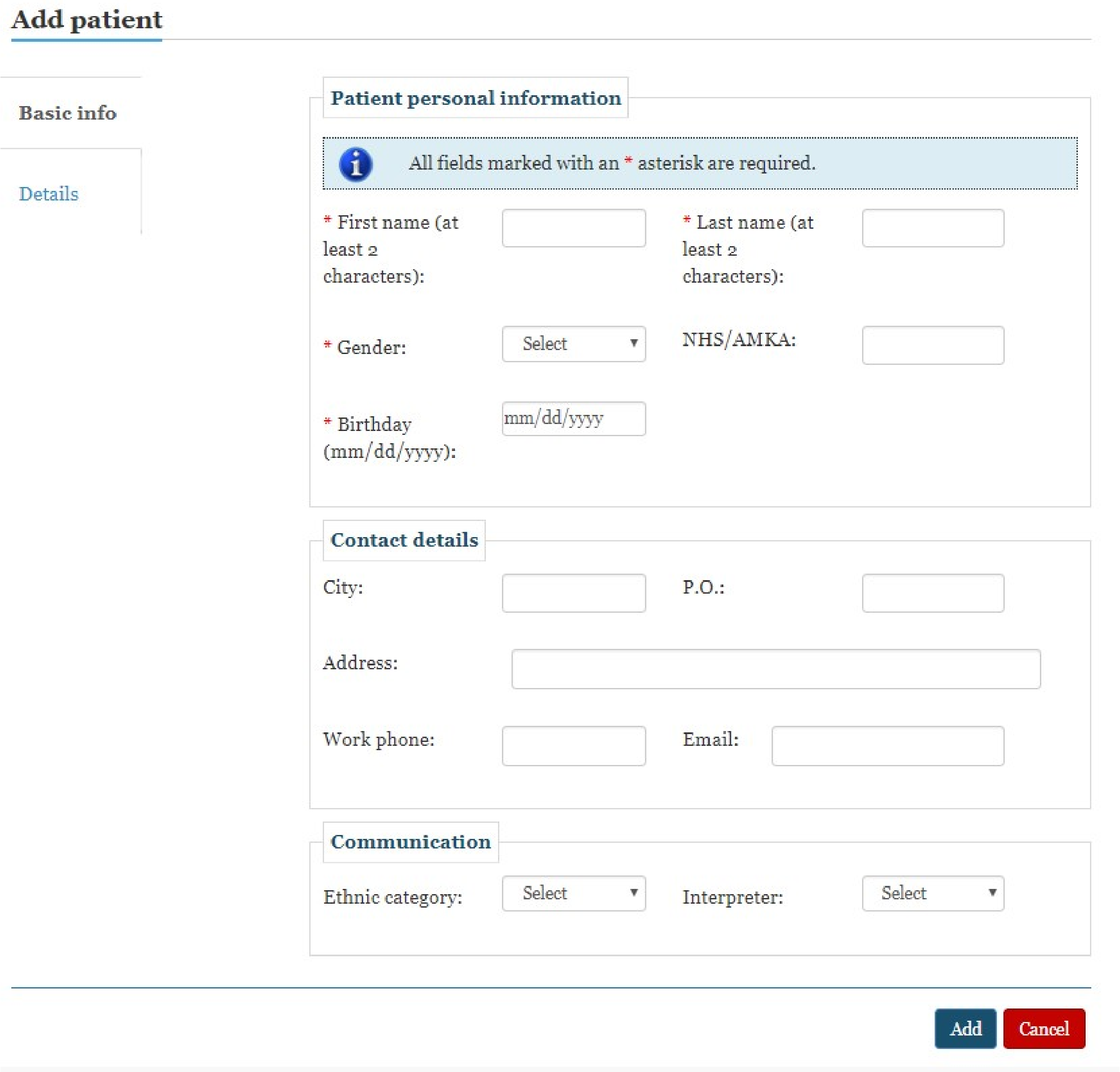
Patients

You are viewing Version 0.94. Timezone is: Europe/Athens

External pseudo ID	First name	Last name	Last updated
EX2018o [redacted]	[redacted]	[redacted]	2018-10-24 13:16:17
EX2018o [redacted]	[redacted]	[redacted]	2018-10-24 11:58:59
EX2018o [redacted]	[redacted]	[redacted]	2018-10-24 11:01:25
EX2018o [redacted]	[redacted]	[redacted]	2018-10-24 10:33:09
EX2018o [redacted]	[redacted]	[redacted]	2018-10-23 21:51:00
EX2018o [redacted]	[redacted]	[redacted]	2018-10-23 21:40:19
EX2018o [redacted]	[redacted]	[redacted]	2018-10-10 10:59:54
EX2018o [redacted]	[redacted]	[redacted]	2018-10-10 09:56:58
EX2018o [redacted]	[redacted]	[redacted]	2018-10-09 12:10:22
EX2018o [redacted]	[redacted]	[redacted]	2018-10-09 11:46:20

1 to 10 of 310 records

Figure 11: All available patients presented in Zebra striping tabular form. Selecting the desired link ID from the list will automatically make it the active selection (patient's specific information to be displayed). Key identification properties are blanked out



Add patient

Basic info

Details

Patient personal information

All fields marked with an * asterisk are required.

* First name (at least 2 characters):		* Last name (at least 2 characters):	
* Gender:	Select ▾	NHS/AMKA:	
* Birthday (mm/dd/yyyy):	mm/dd/yyyy		

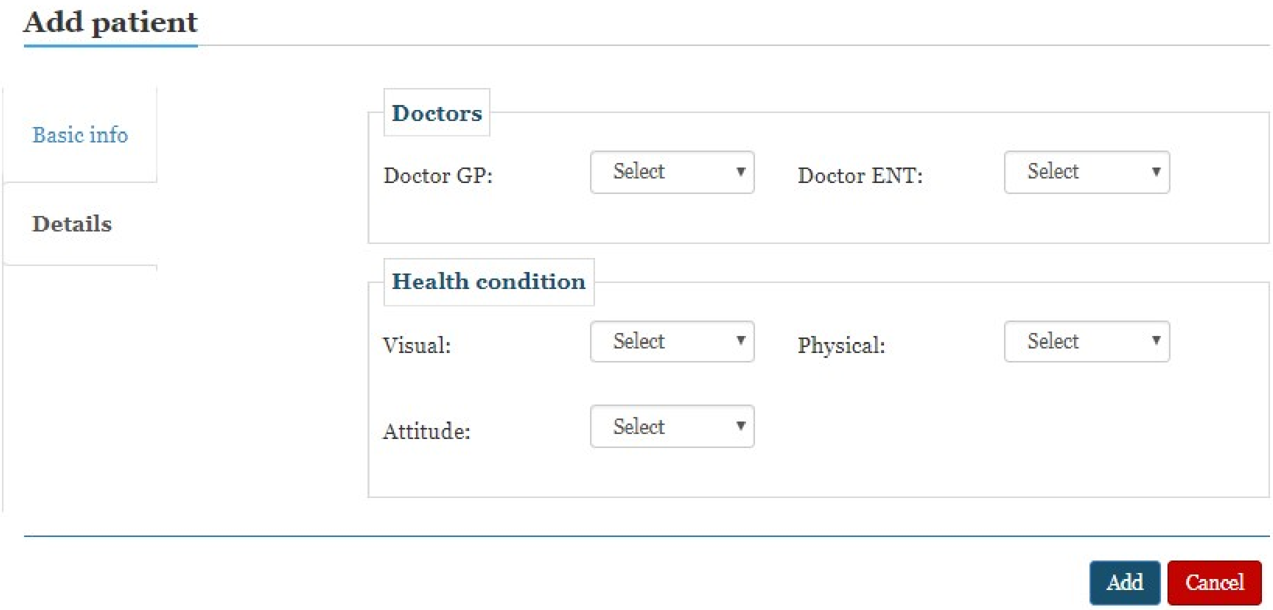
Contact details

City:		P.O.:	
Address:			
Work phone:		Email:	

Communication

Ethnic category:	Select ▾	Interpreter:	Select ▾

Figure 12: Creation of a new patient (part A: basic info)



Add patient

Basic info

Details

Doctors

Doctor GP: Select ▼

Doctor ENT: Select ▼

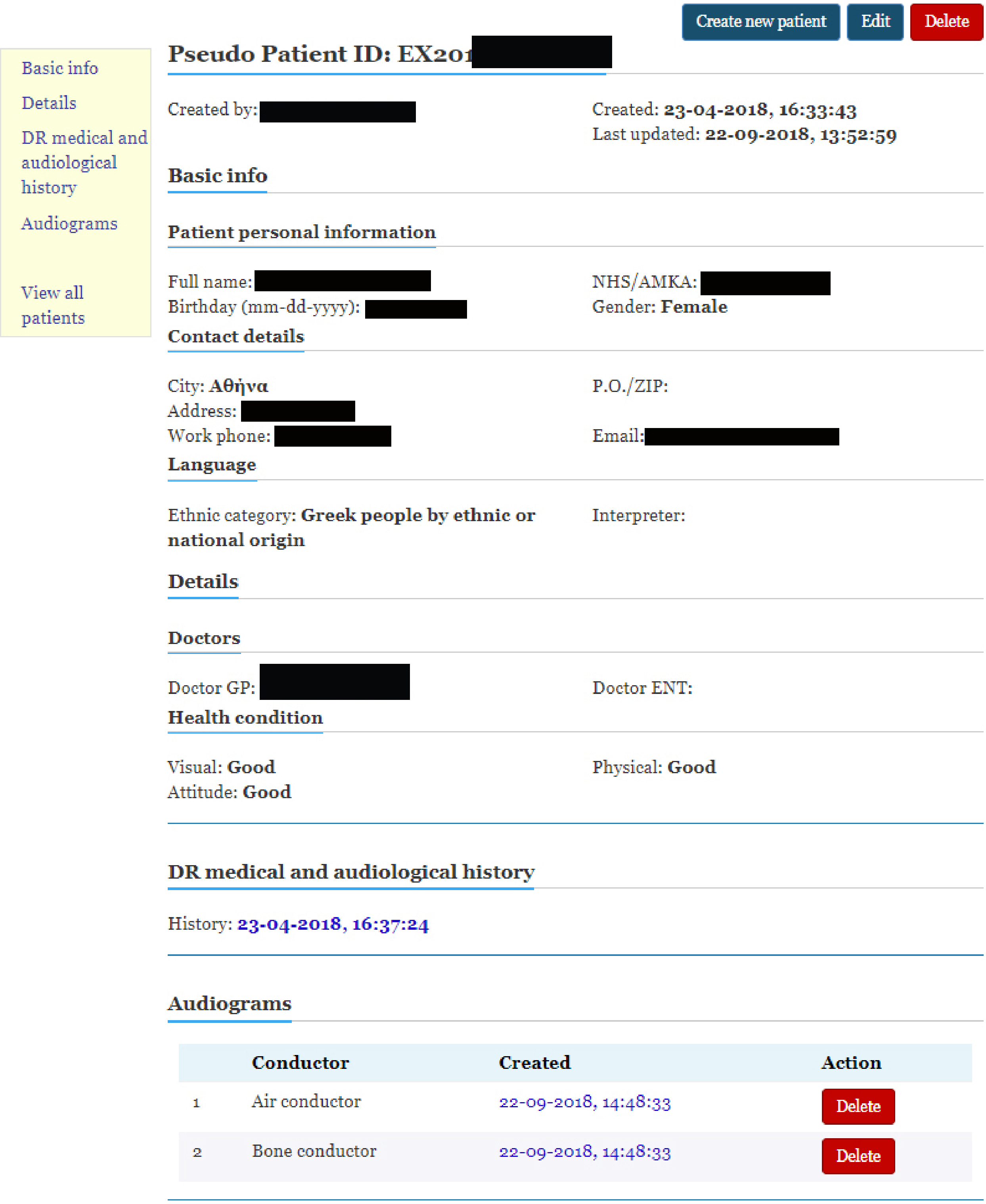
Health condition

Visual: Select ▼

Physical: Select ▼

Attitude: Select ▼

Figure 13: Creation of a new patient (part B: details).



Pseudo Patient ID: EX20[redacted]

Created by: [redacted]

Created: **23-04-2018, 16:33:43**
Last updated: **22-09-2018, 13:52:59**

Basic info

Patient personal information

Full name: [redacted]
Birthday (mm-dd-yyyy): [redacted]

NHS/AMKA: [redacted]
Gender: **Female**

Contact details

City: **Αθήνα**
Address: [redacted]
Work phone: [redacted]

P.O./ZIP:

Email: [redacted]

Language

Ethnic category: **Greek people by ethnic or national origin**

Interpreter:

Details

Doctors

Doctor GP: [redacted]

Doctor ENT:

Health condition

Visual: **Good**
Attitude: **Good**

Physical: **Good**

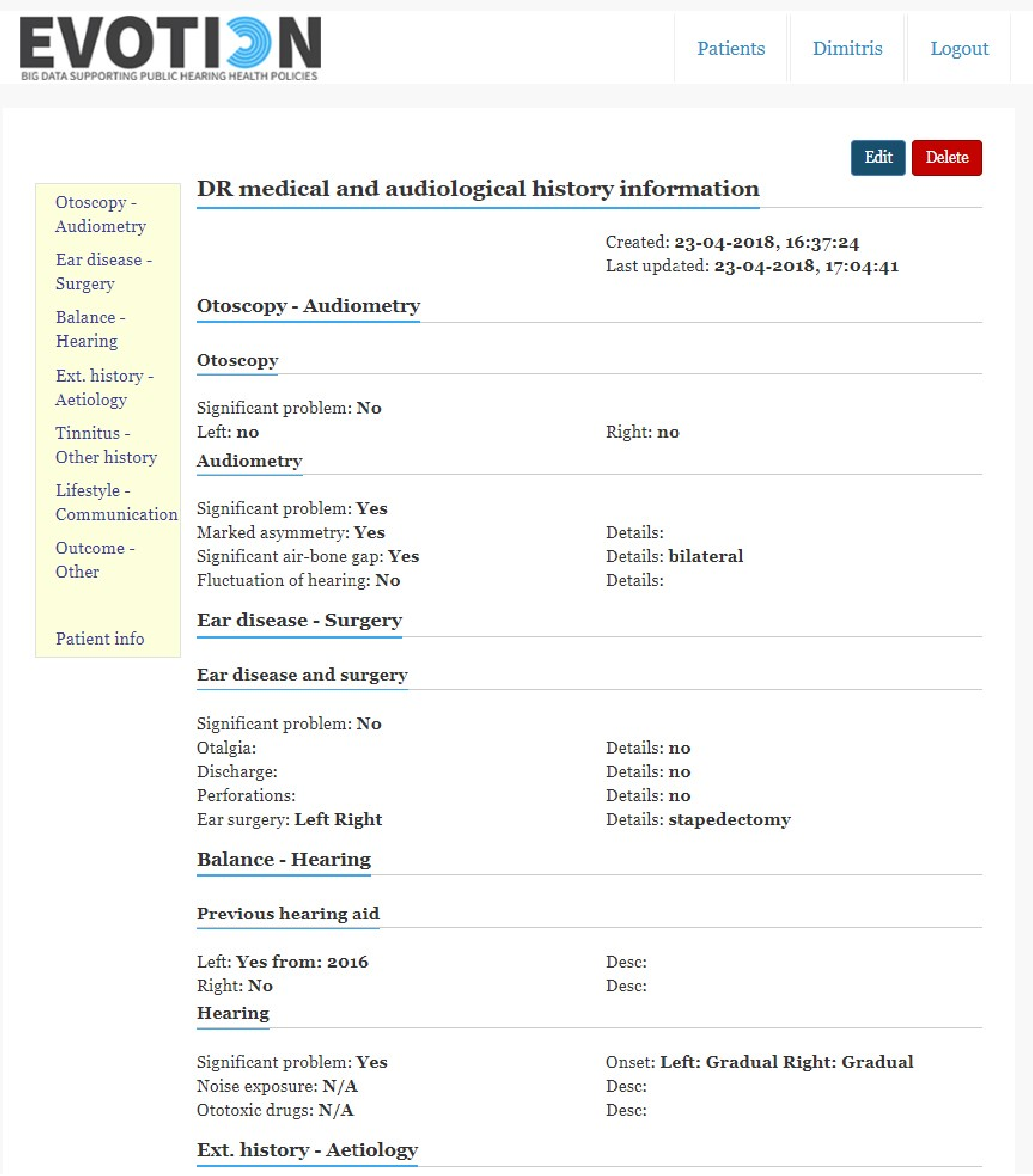
DR medical and audiological history

History: **23-04-2018, 16:37:24**

Audiograms

	Conductor	Created	Action
1	Air conductor	22-09-2018, 14:48:33	Delete
2	Bone conductor	22-09-2018, 14:48:33	Delete

Figure 14: Portion of current patient's (EX20********) personal details. Key identification properties are blanked out to preserve the patient's anonymity**



DR medical and audiological history information

Created: **23-04-2018, 16:37:24**
Last updated: **23-04-2018, 17:04:41**

Otoscopy - Audiometry

Otoscopy

Significant problem: **No**	
Left: **no**	Right: **no**

Audiometry

Significant problem: **Yes**	
Marked asymmetry: **Yes**	Details:
Significant air-bone gap: **Yes**	Details: **bilateral**
Fluctuation of hearing: **No**	Details:

Ear disease - Surgery

Ear disease and surgery

Significant problem: **No**	
Otalgia:	Details: **no**
Discharge:	Details: **no**
Perforations:	Details: **no**
Ear surgery: **Left Right**	Details: **stapedectomy**

Balance - Hearing

Previous hearing aid

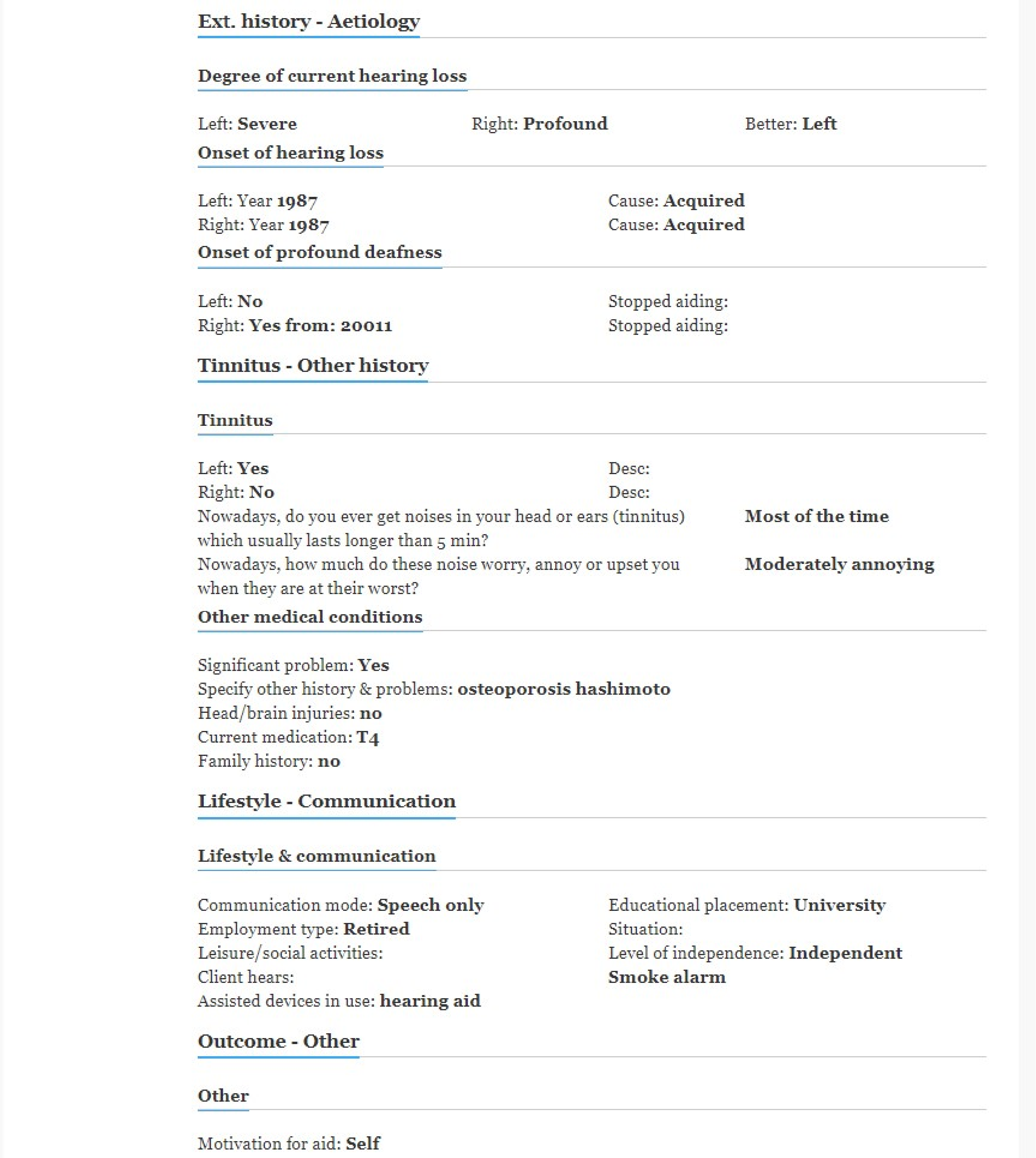
Left: **Yes from: 2016**	Desc:
Right: **No**	Desc:

Hearing

Significant problem: **Yes**	Onset: **Left: Gradual Right: Gradual**
Noise exposure: **N/A**	Desc:
Ototoxic drugs: **N/A**	Desc:

Ext. history - Aetiology

Figure 15: Portion of current patient's (EX20********) medical history (Part A: Otoscopy – Audiometry, Ear disease – Surgery, Balance - Hearing).**



Ext. history - Aetiology

Degree of current hearing loss

Left: **Severe** Right: **Profound** Better: **Left**

Onset of hearing loss

Left: Year **1987** Cause: **Acquired**
Right: Year **1987** Cause: **Acquired**

Onset of profound deafness

Left: **No** Stopped aiding:
Right: **Yes from: 20011** Stopped aiding:

Tinnitus - Other history

Tinnitus

Left: **Yes** Desc:
Right: **No** Desc:

Nowadays, do you ever get noises in your head or ears (tinnitus) which usually lasts longer than 5 min?	**Most of the time**
Nowadays, how much do these noise worry, annoy or upset you when they are at their worst?	**Moderately annoying**

Other medical conditions

Significant problem: **Yes**
Specify other history & problems: **osteoporosis hashimoto**
Head/brain injuries: **no**
Current medication: **T4**
Family history: **no**

Lifestyle - Communication

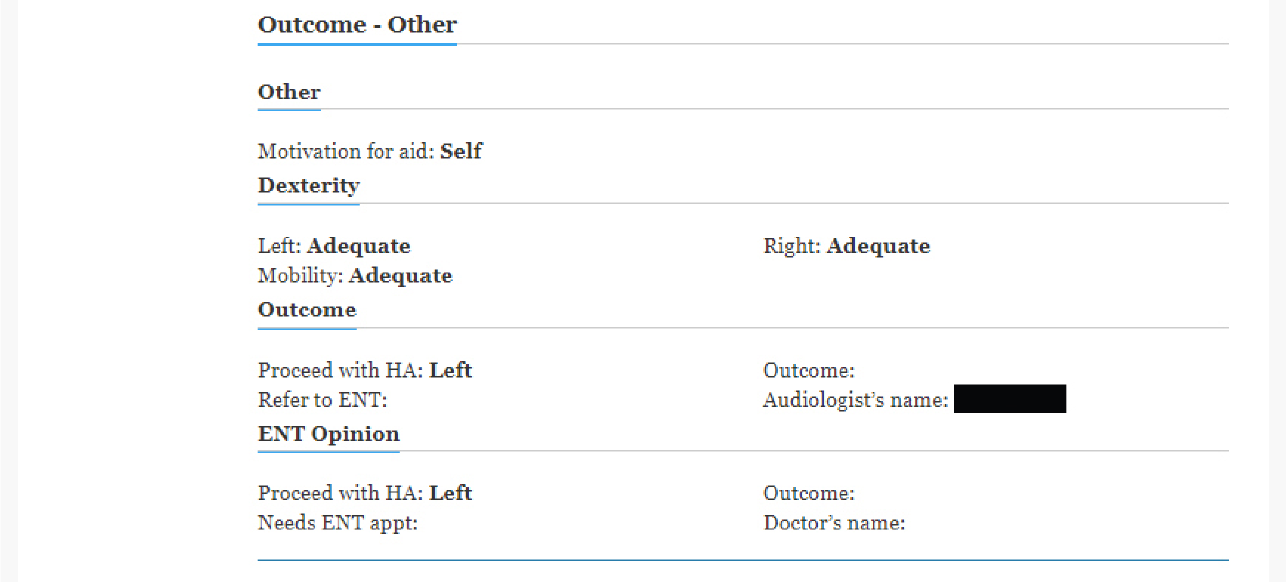
Lifestyle & communication

Communication mode: **Speech only** Educational placement: **University**
Employment type: **Retired** Situation:
Leisure/social activities: Level of independence: **Independent**
Client hears: **Smoke alarm**
Assisted devices in use: **hearing aid**

Outcome - Other

Other

Motivation for aid: **Self**



Outcome - Other

Other

Motivation for aid: **Self**

Dexterity

Left: **Adequate**

Right: **Adequate**

Mobility: **Adequate**

Outcome

Proceed with HA: **Left**

Outcome:

Refer to ENT:

Audiologist's name: [redacted]

ENT Opinion

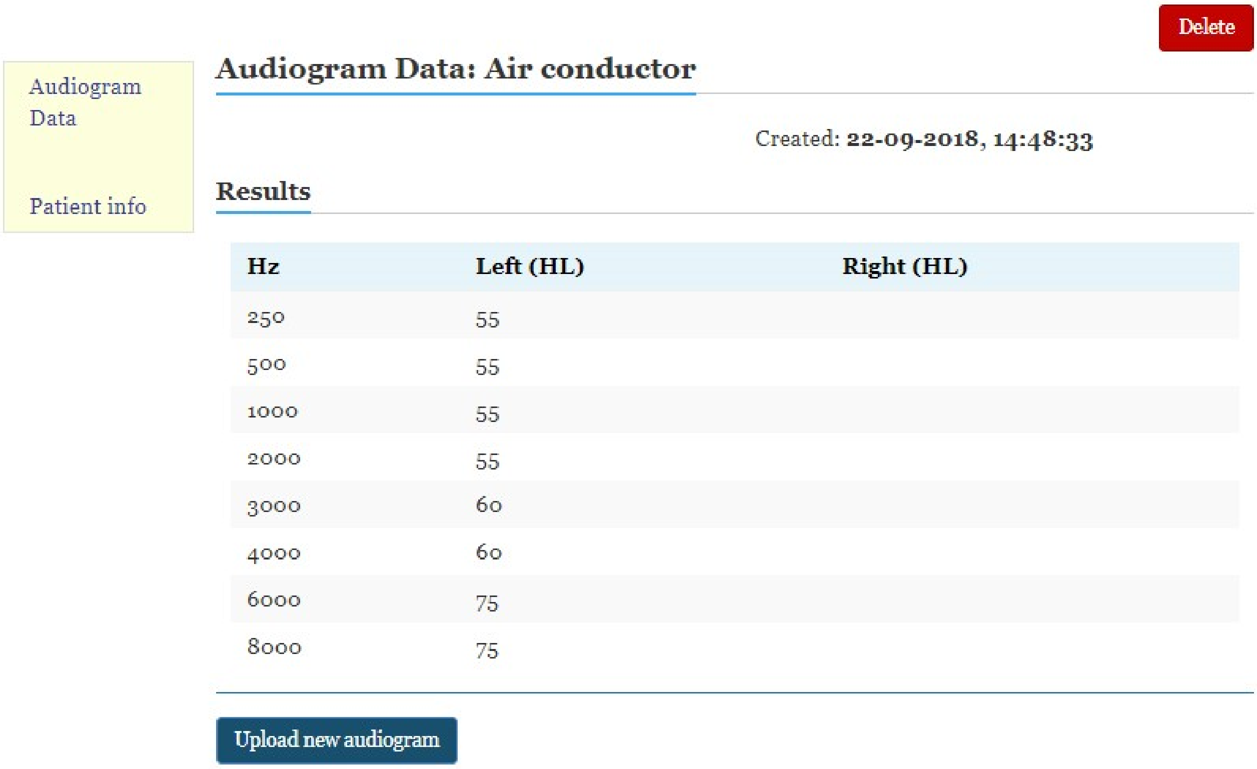
Proceed with HA: **Left**

Outcome:

Needs ENT appt:

Doctor's name:

Figure 16: Portion of current patient's (EX20********) medical history (Part B: Ext. history – Aetiology, Tinnitus - Other history, Lifestyle – Communication, Outcome - Other).**



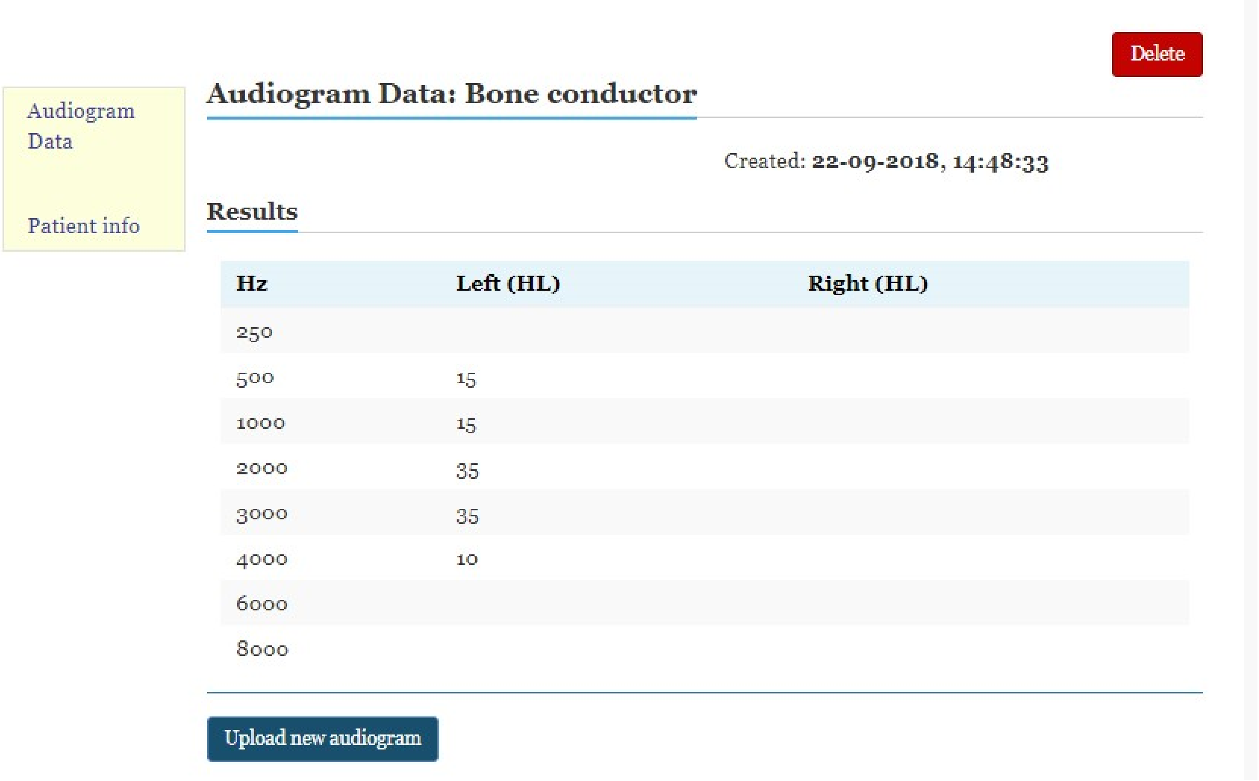
Audiogram Data: Air conductor

Created: **22-09-2018, 14:48:33**

Results

Hz	Left (HL)	Right (HL)
250	55	
500	55	
1000	55	
2000	55	
3000	60	
4000	60	
6000	75	
8000	75	

Figure 17: View of patient's audiogram (air conductor) data.



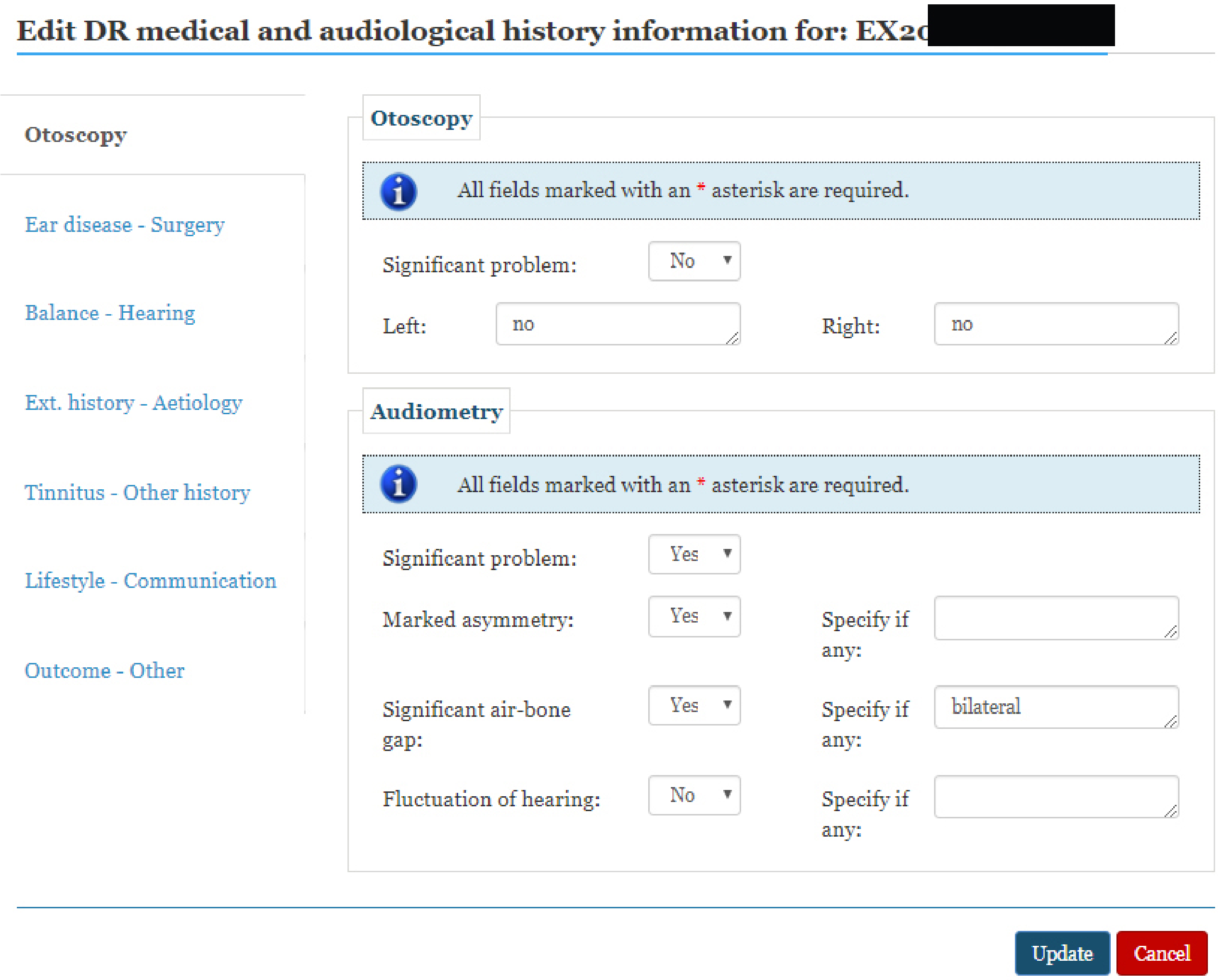
Audiogram Data: Bone conductor

Created: **22-09-2018, 14:48:33**

Results

Hz	Left (HL)	Right (HL)
250		
500	15	
1000	15	
2000	35	
3000	35	
4000	10	
6000		
8000		

Figure 18: View of patient's audiogram (bone conductor) data.



Edit DR medical and audiological history information for: EX2c[redacted]

Otoscopy

All fields marked with an * asterisk are required.

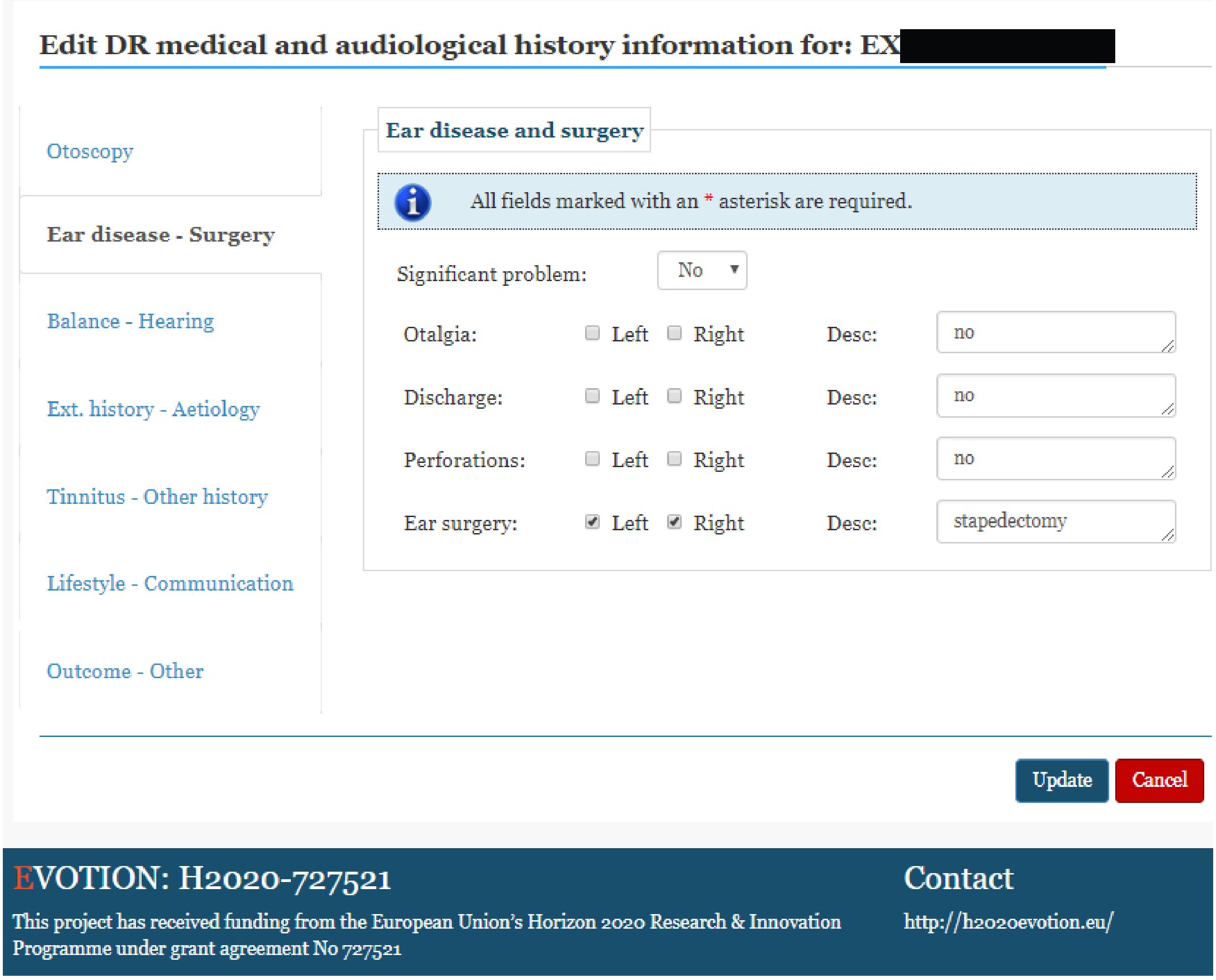
Significant problem:	No		
Left:	no	Right:	no

Audiometry

All fields marked with an * asterisk are required.

Significant problem:	Yes		
Marked asymmetry:	Yes	Specify if any:	
Significant air-bone gap:	Yes	Specify if any:	bilateral
Fluctuation of hearing:	No	Specify if any:	

Figure 19: Editing patient's medical history (Otoscopy).



Edit DR medical and audiological history information for: EX[redacted]

Ear disease and surgery

All fields marked with an * asterisk are required.

Significant problem: No ▾

Otalgia:	□ Left	□ Right	Desc:	no
Discharge:	□ Left	□ Right	Desc:	no
Perforations:	□ Left	□ Right	Desc:	no
Ear surgery:	☑ Left	☑ Right	Desc:	stapedectomy

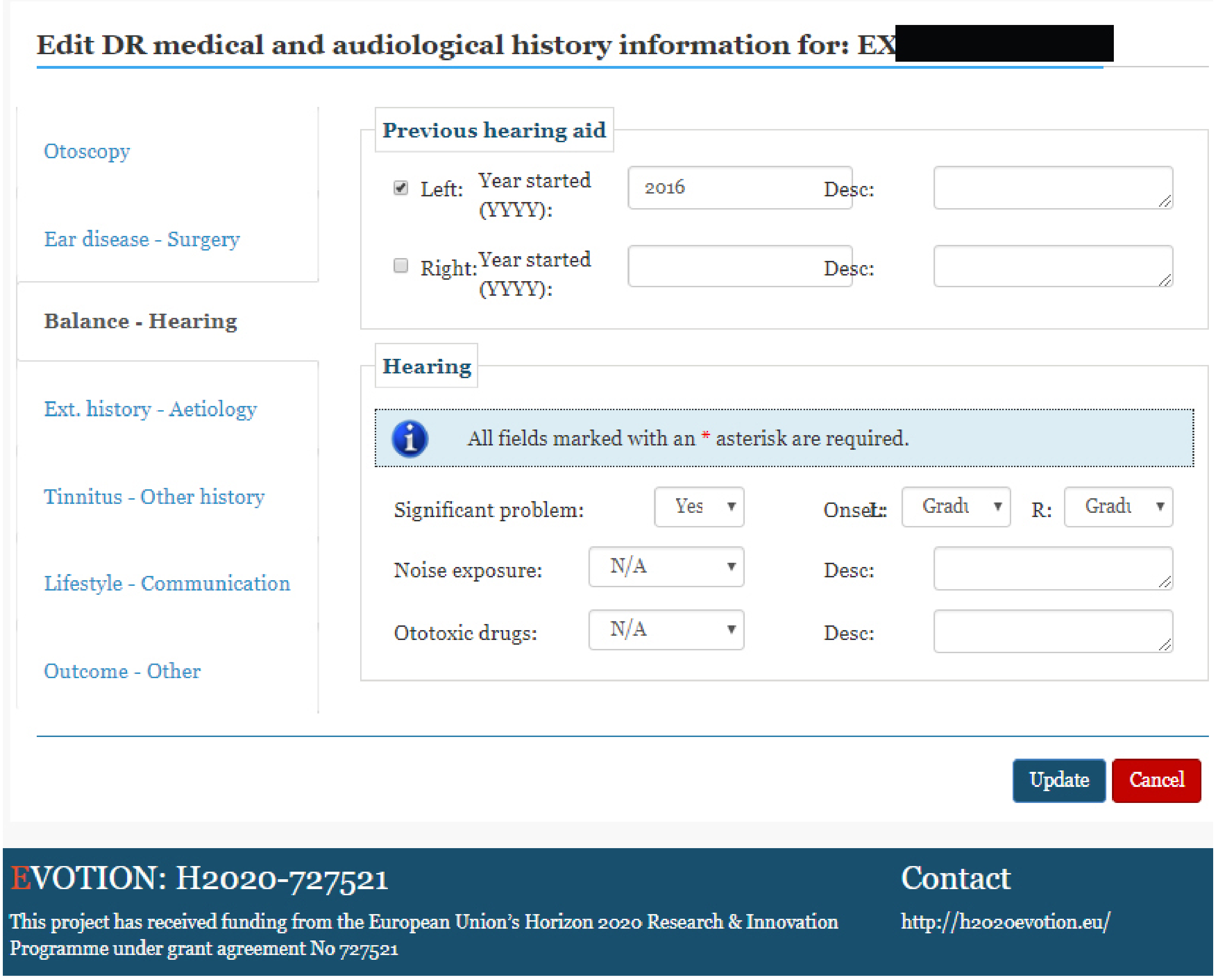
EVOTION: H2020-727521

This project has received funding from the European Union's Horizon 2020 Research & Innovation Programme under grant agreement No 727521

Contact

http://h2020evotion.eu/

Figure 20: Editing patient's medical history (Ear disease - surgery).



Edit DR medical and audiological history information for: EX[redacted]

Previous hearing aid

☑ Left: Year started (YYYY): 2016 Desc:

☐ Right: Year started (YYYY): Desc:

Hearing

All fields marked with an * asterisk are required.

Significant problem: Yes ▾ Onset L: Gradu ▾ R: Gradu ▾

Noise exposure: N/A ▾ Desc:

Ototoxic drugs: N/A ▾ Desc:

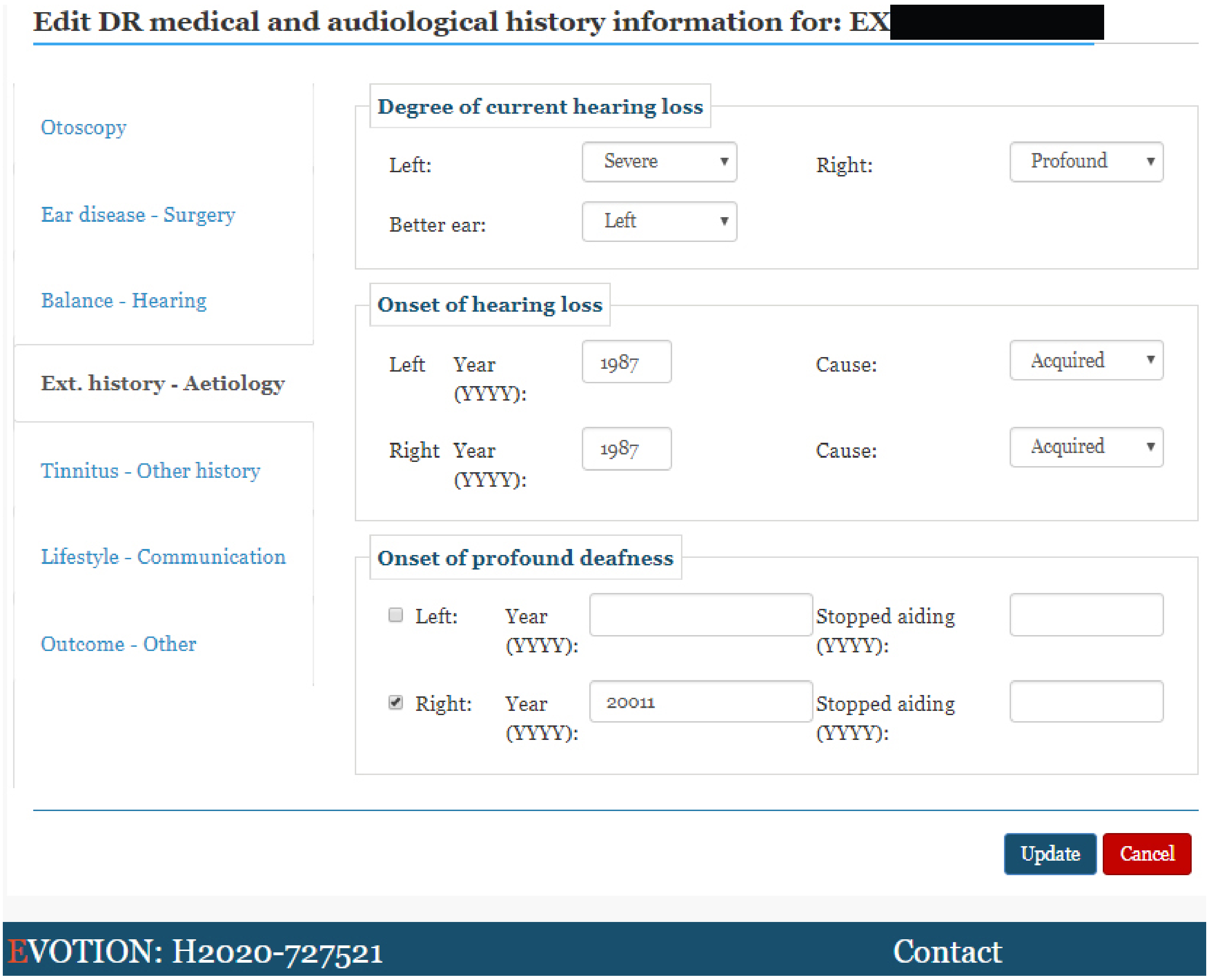
EVOTION: H2020-727521

This project has received funding from the European Union's Horizon 2020 Research & Innovation Programme under grant agreement No 727521

Contact

http://h2020evotion.eu/

Figure 21: Editing patient's medical history (Balance - hearing).



Edit DR medical and audiological history information for: EX[redacted]

Degree of current hearing loss

Left: Severe ▾ Right: Profound ▾

Better ear: Left ▾

Onset of hearing loss

Left Year (YYYY): 1987 Cause: Acquired ▾

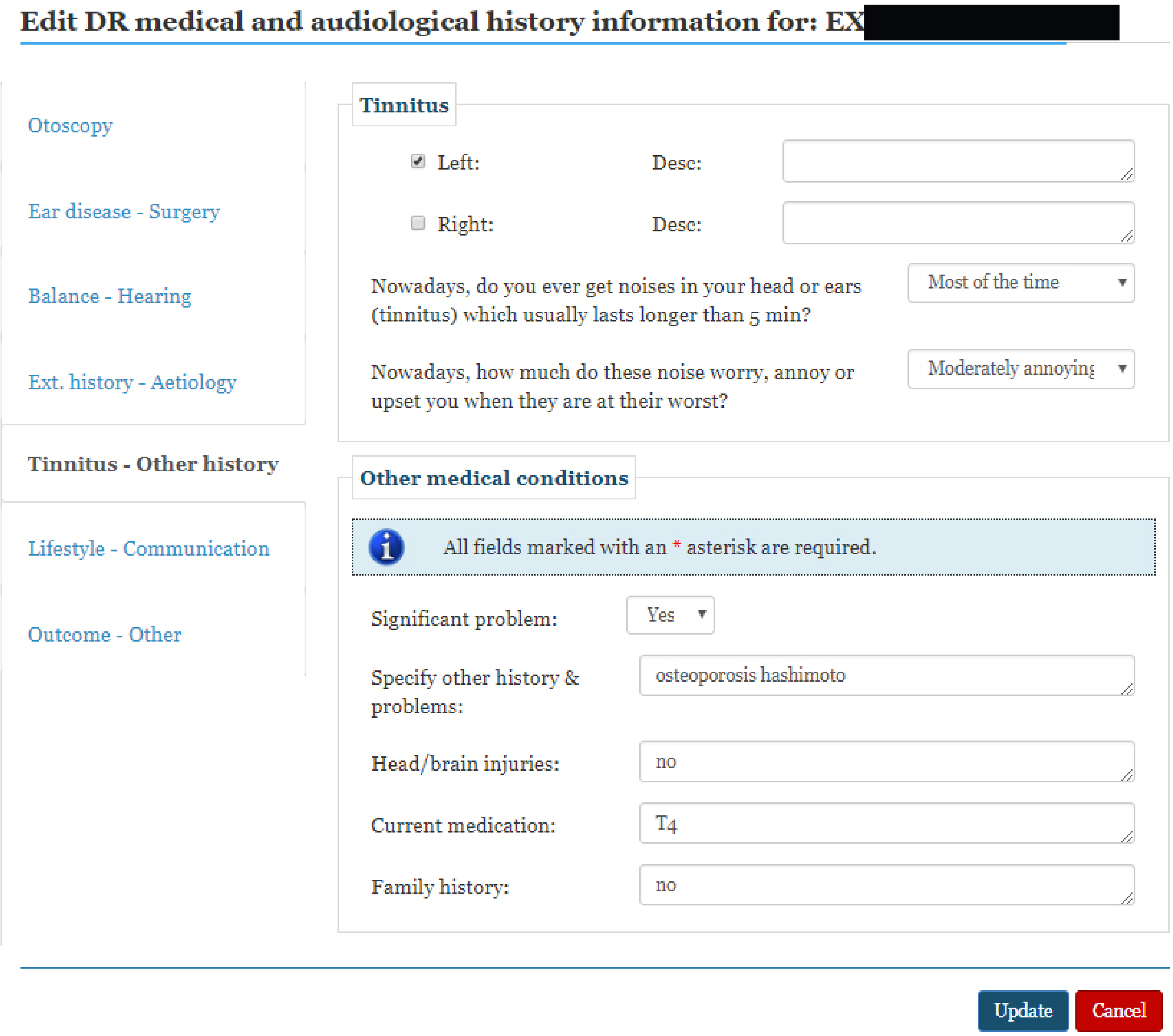
Right Year (YYYY): 1987 Cause: Acquired ▾

Onset of profound deafness

□ Left: Year (YYYY): Stopped aiding (YYYY):

☑ Right: Year (YYYY): 20011 Stopped aiding (YYYY):

Figure 22: Editing patient's medical history (Ext. history - Aetiology).



Edit DR medical and audiological history information for: EX[redacted]

Tinnitus

☑ Left: Desc:

□ Right: Desc:

Nowadays, do you ever get noises in your head or ears (tinnitus) which usually lasts longer than 5 min? Most of the time ▾

Nowadays, how much do these noise worry, annoy or upset you when they are at their worst? Moderately annoyinç ▾

Other medical conditions

All fields marked with an * asterisk are required.

Significant problem: Yes ▾

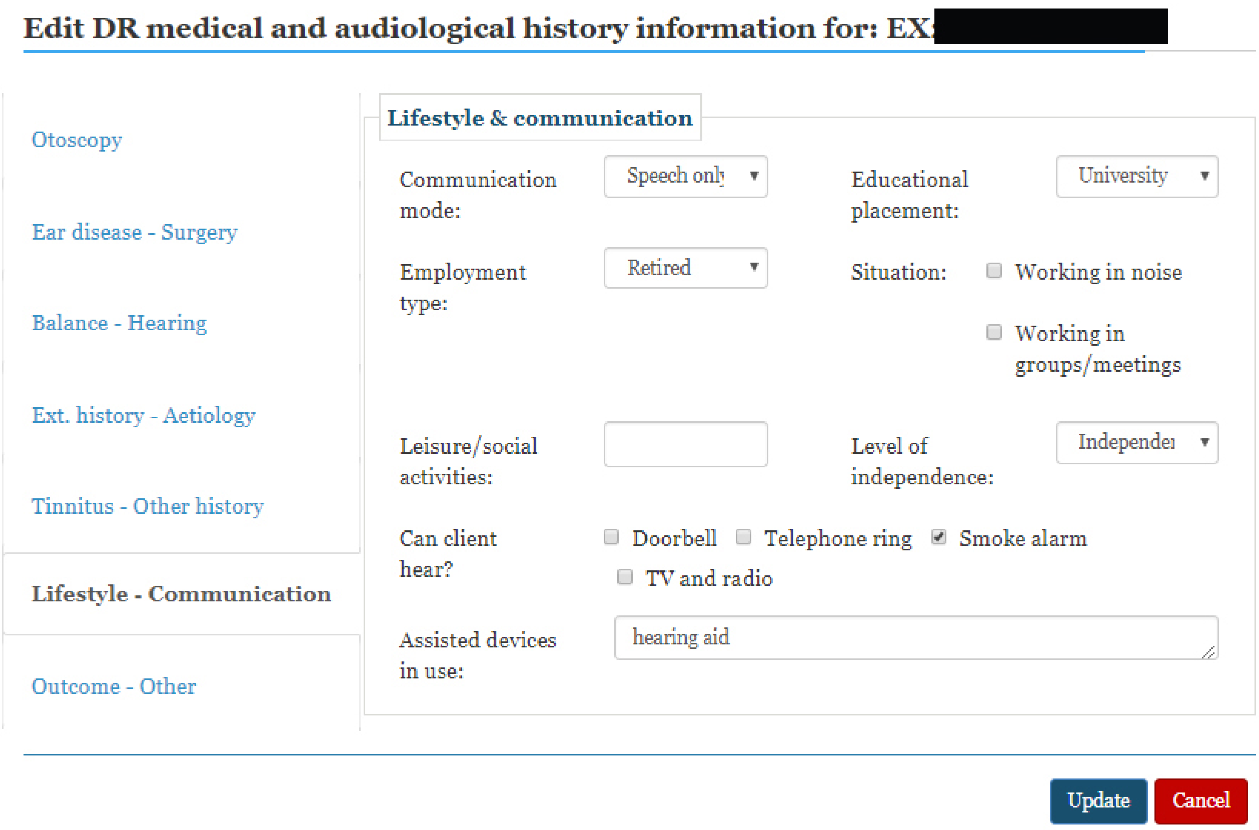
Specify other history & problems: osteoporosis hashimoto

Head/brain injuries: no

Current medication: T4

Family history: no

Figure 23: Editing patient's medical history (Tinnitus – Other history).



Edit DR medical and audiological history information for: EX[redacted]

Lifestyle & communication

Communication mode: Speech only

Educational placement: University

Employment type: Retired

Situation: □ Working in noise □ Working in groups/meetings

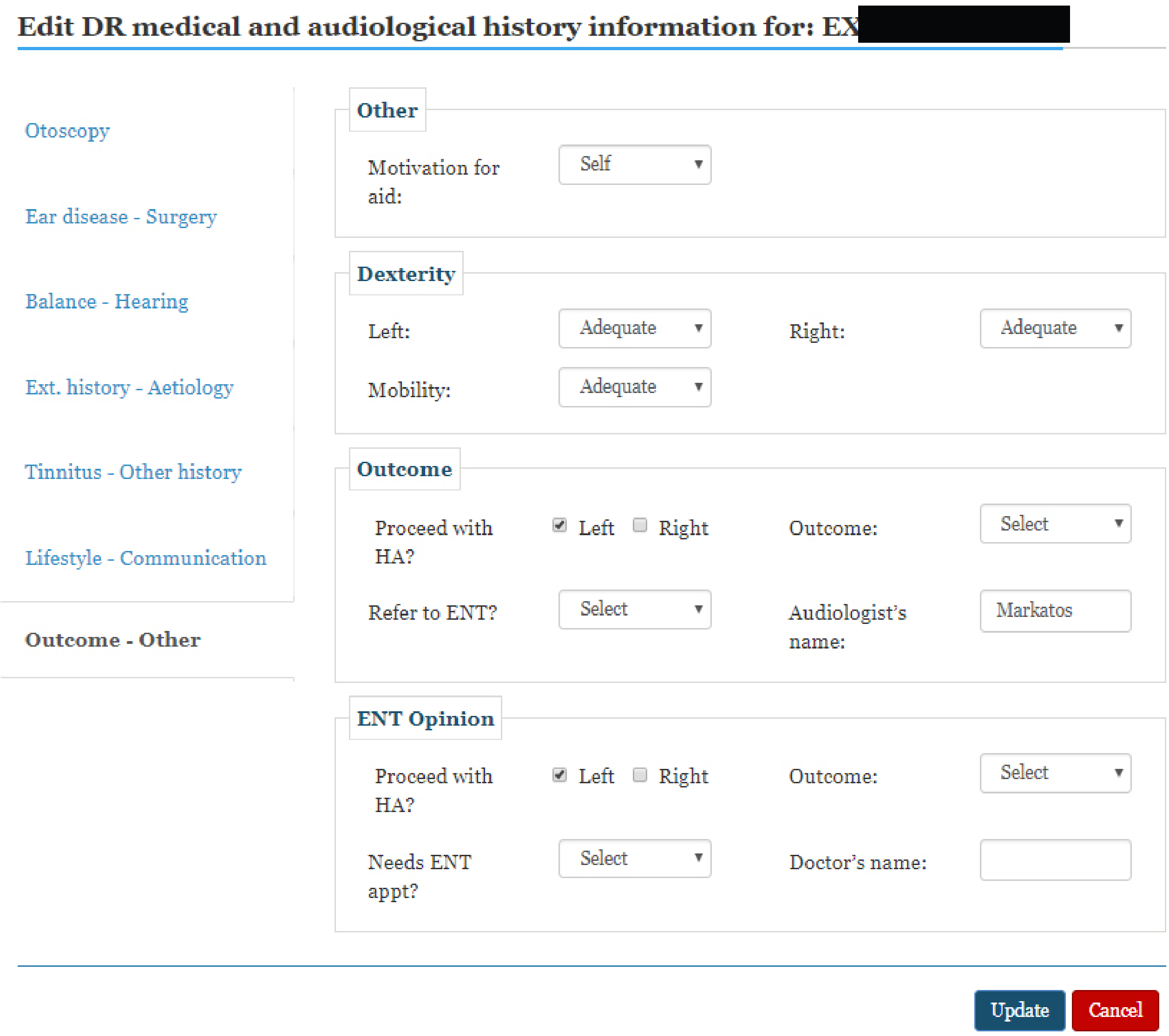
Leisure/social activities:

Level of independence: Independent

Can client hear? □ Doorbell □ Telephone ring ☑ Smoke alarm □ TV and radio

Assisted devices in use: hearing aid

Figure 24: Editing patient's medical history (Lifestyle - Communication).



Edit DR medical and audiological history information for: EX[redacted]

Other

Motivation for aid: Self

Dexterity

Left: Adequate Right: Adequate

Mobility: Adequate

Outcome

Proceed with HA? ☑ Left ☐ Right Outcome: Select

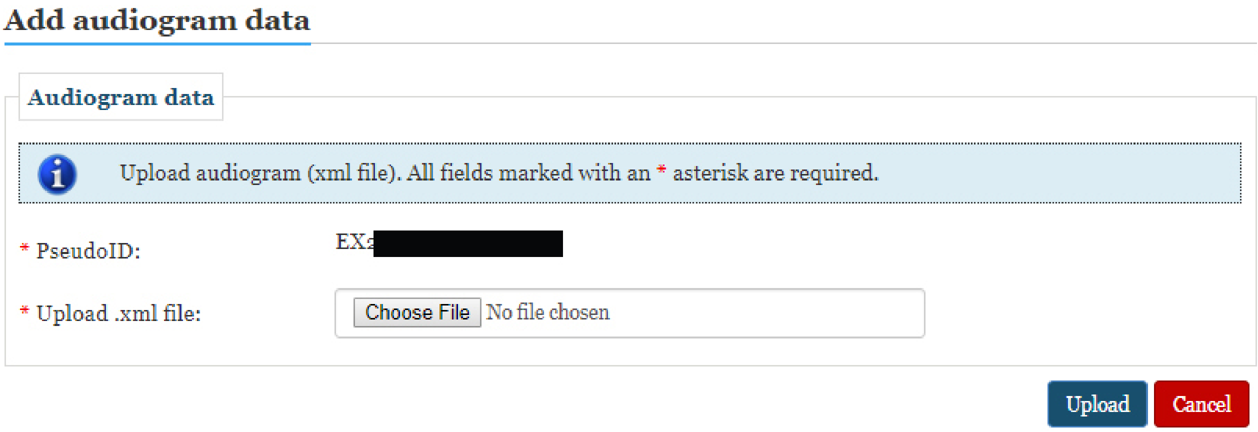
Refer to ENT? Select Audiologist's name: Markatos

ENT Opinion

Proceed with HA? ☑ Left ☐ Right Outcome: Select

Needs ENT appt? Select Doctor's name:

Figure 25: Editing patient's medical history (Outcome - Other).



Add audiogram data

Audiogram data

Upload audiogram (xml file). All fields marked with an * asterisk are required.

* PseudoID: EX[redacted]

* Upload .xml file: Choose File No file chosen

Figure 26: Adding an audiogram: selecting the proper NOAH xml file for upload.

APPENDIX 2 - Additional Use Case Scenarios for the Components' Communication

Scenario 2. Sudden Deterioration of Hearing

In this scenario we assume that an analytic workflow computing the deterioration is in progress. This triggers the transfer of the relevant data at the time of the event to the clinician and/or audiologist.

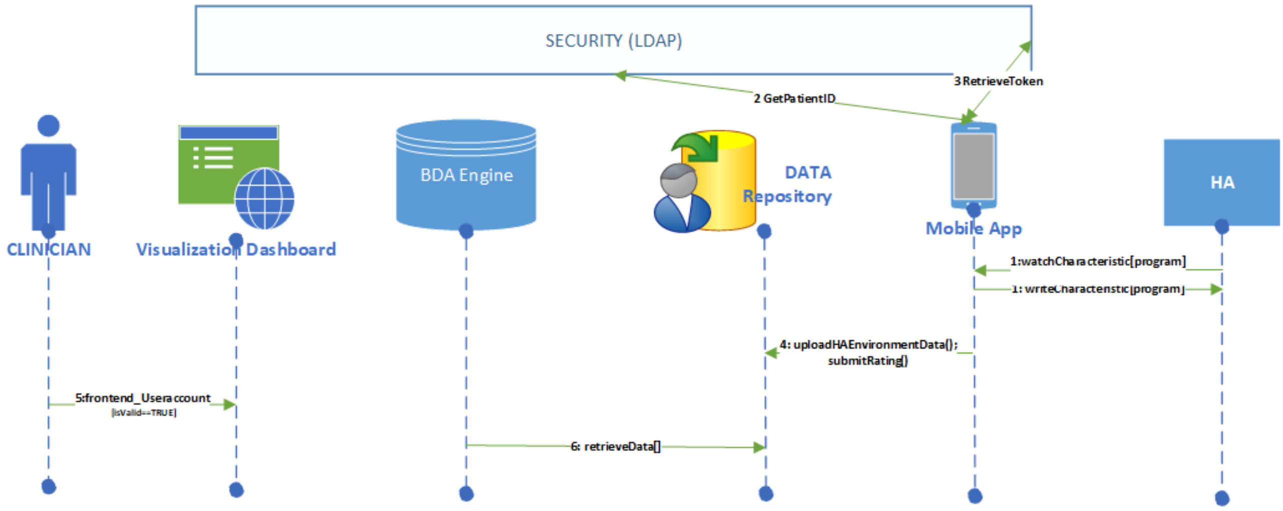


Figure 27: Scenario 2. Sudden Deterioration of Hearing

Scenario 3. Assessment of initial follow up policies

This scenario serves to identify profiles of users who would benefit from earlier remote interventions that can reduce the number of physical follow appointments, and maximise the overall satisfaction of HA users.

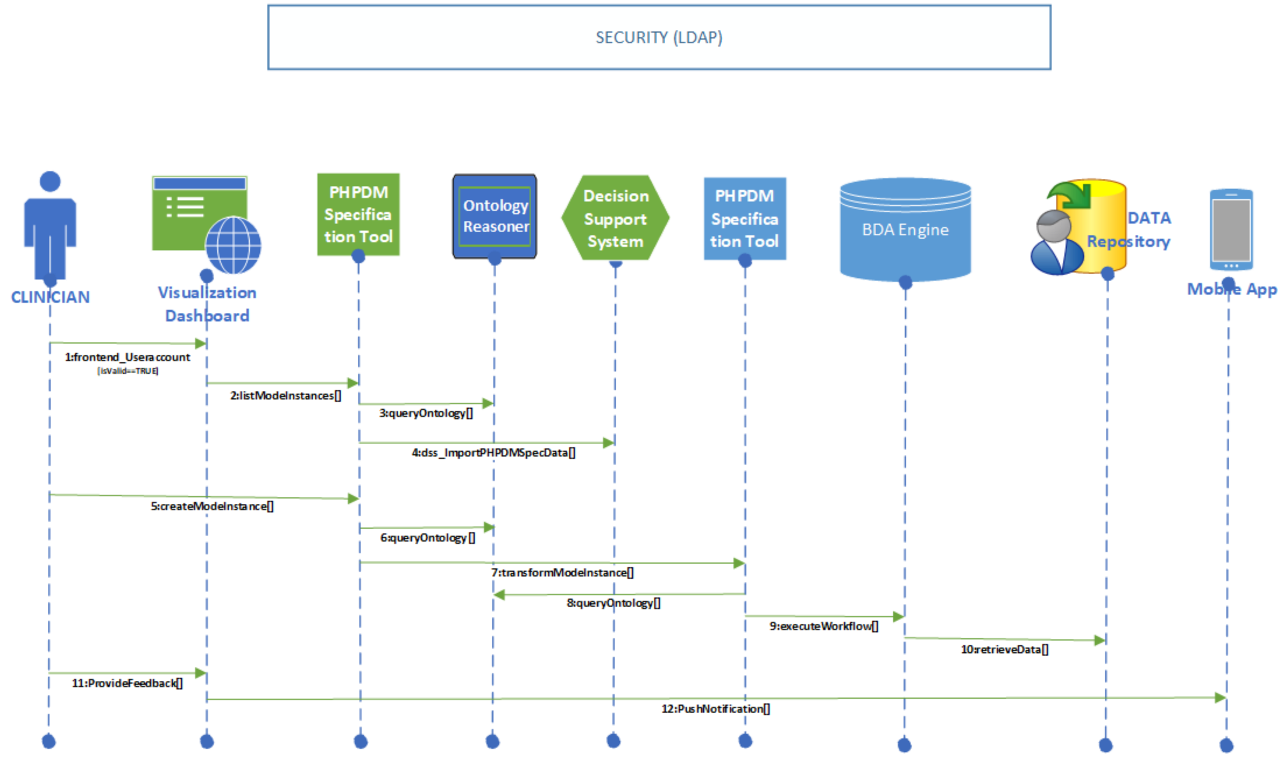


Figure 28: Scenario 3. Assessment of initial follow up policies

Scenario 4. NIHL Protection of people with hearing impairments from the harmful effects

This use case captures scenario regarding determination of individualised (environmental, physiological) factors associated with increased risk for Temporary Threshold Shift (TTS) or noise induced hearing loss (NIHL) for prevention of further TTS/NIHL episodes.

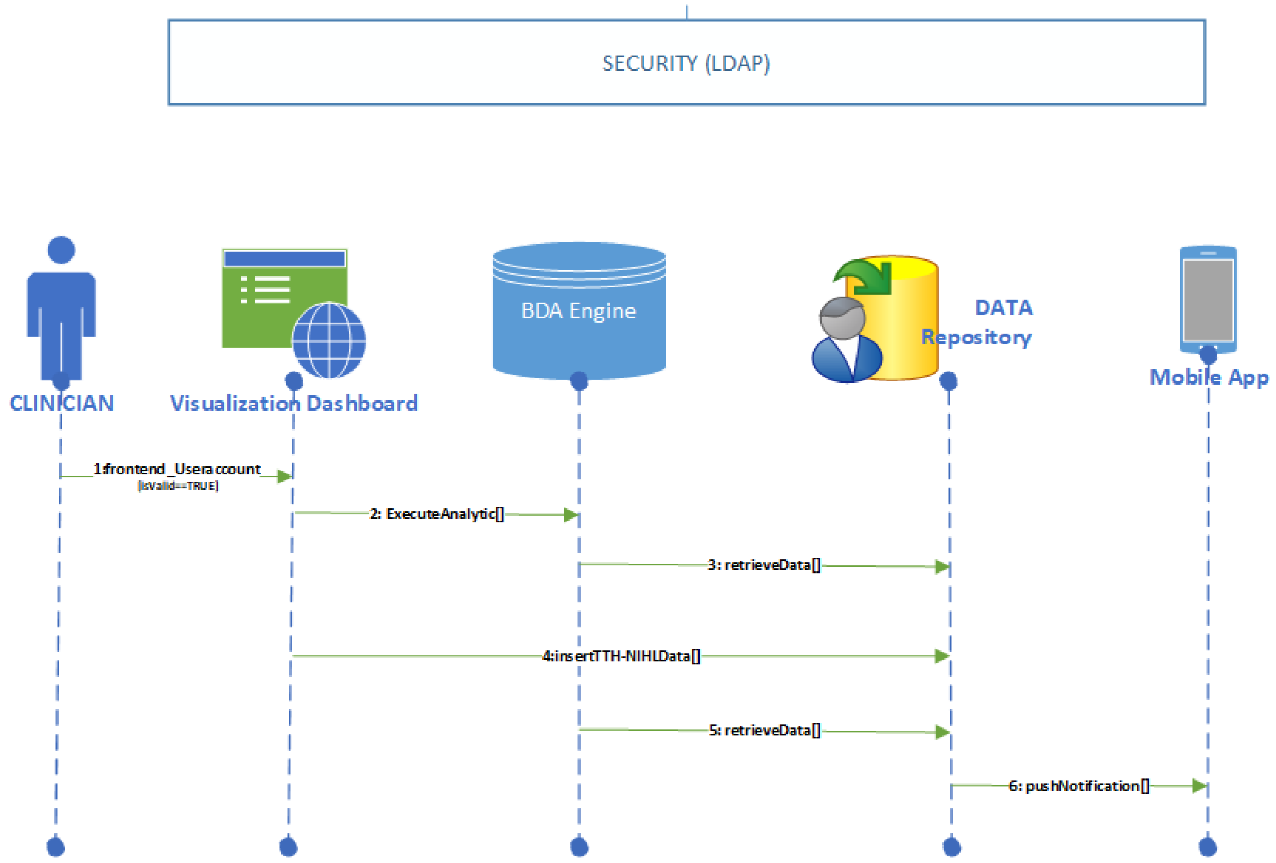


Figure 29: Scenario 4. NIHL Protection of people with hearing impairments from the harmful effects

Scenario 5. Performing audiological tests through HA
This scenario describes how audiological tests data will be recorded and stored by the EVOTION platform. Analytic workflows are pre-loaded and executed

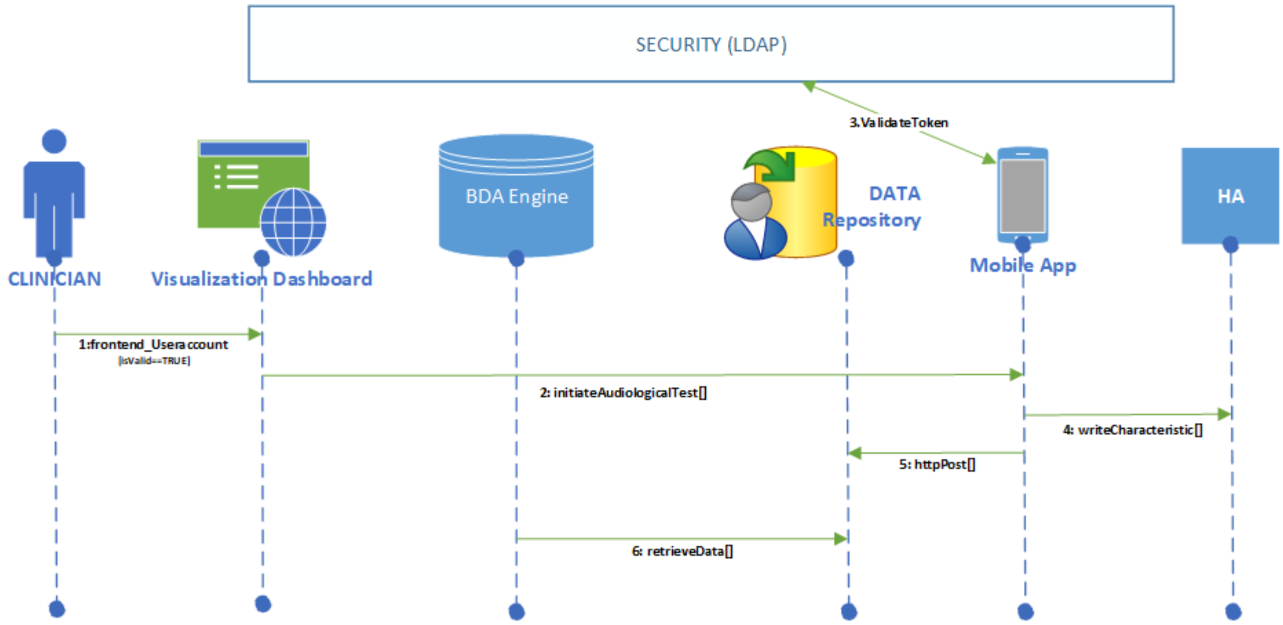


Figure 30: Scenario 5. Performing audiological tests through HA (Analytic workflows are pre-loaded and executed)

Scenario 6. Prognosis of low HA usage

This scenario is related to the formation of public health policy for exploring the potential of interventions aimed at preventing the low usage of HAs.

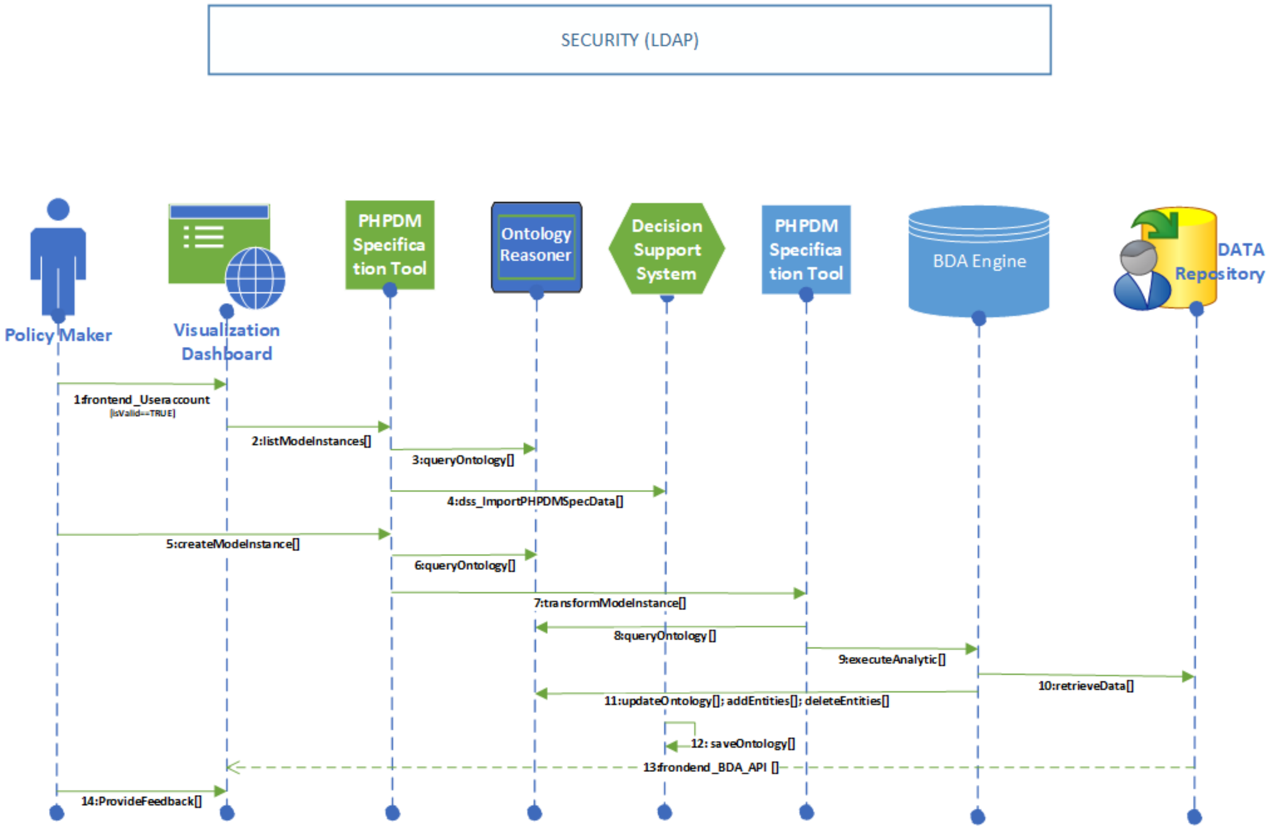


Figure 31: Scenario 6. Prognosis of low HA usage

Scenario 7. Mobile hearing coach

The Mobile Hearing Coach is like a personal trainer towards hearing fitness. It makes the patients train, self-manage and involve other people to their condition.

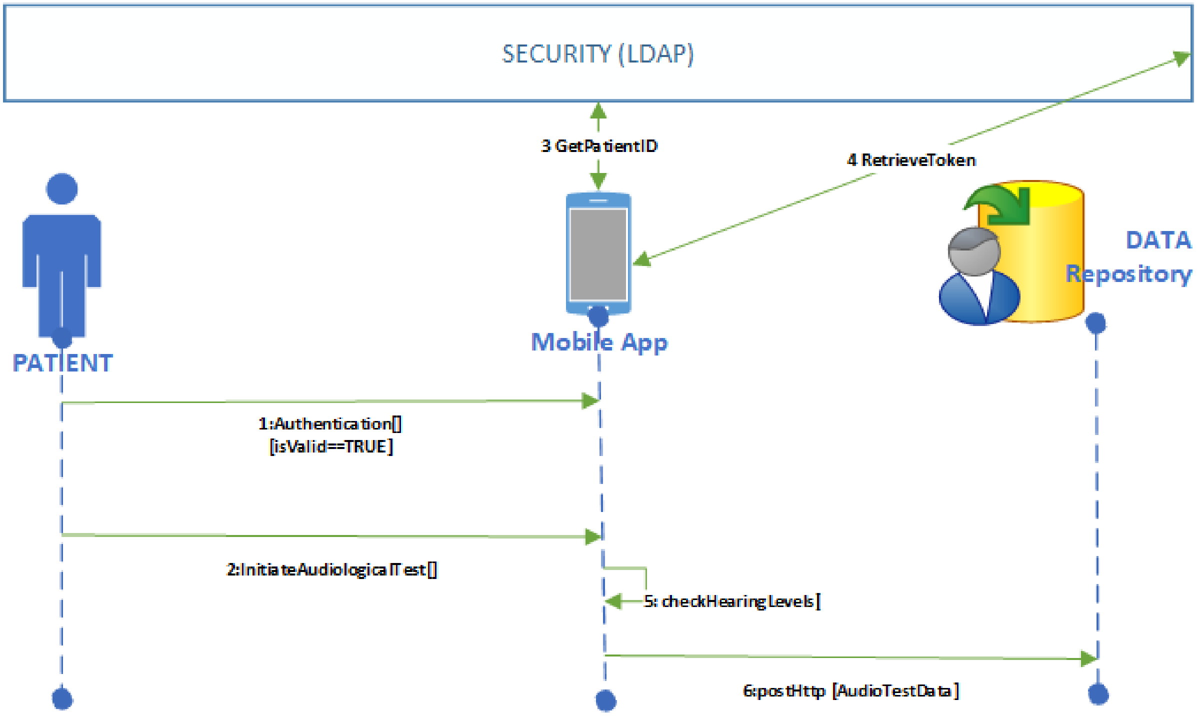


Figure 32: Scenario 7. Mobile hearing coach

Scenario 8. Protection of people with hearing impairments from the harmful effects of loud noise

This scenario captures scenarios regarding determination of individualized (environmental, physiological) factors associated with increased risk for Temporary Threshold Shift (TTS) or noise induced hearing loss (NIHL) for prevention of further TTS/NIHL episode

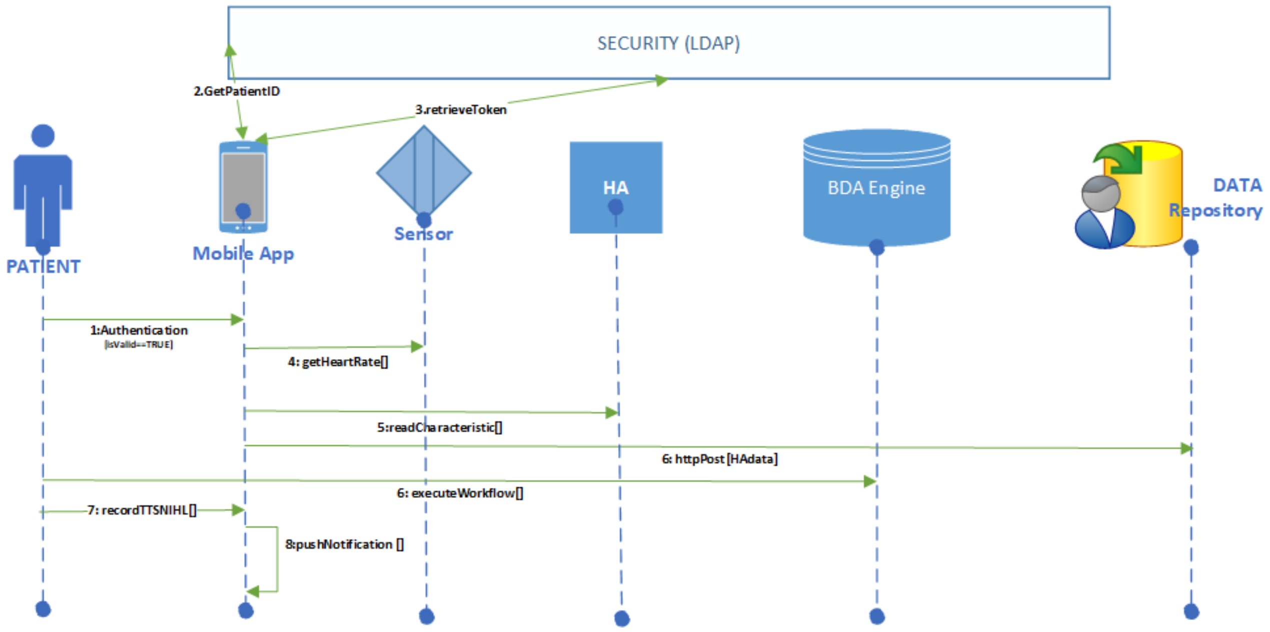
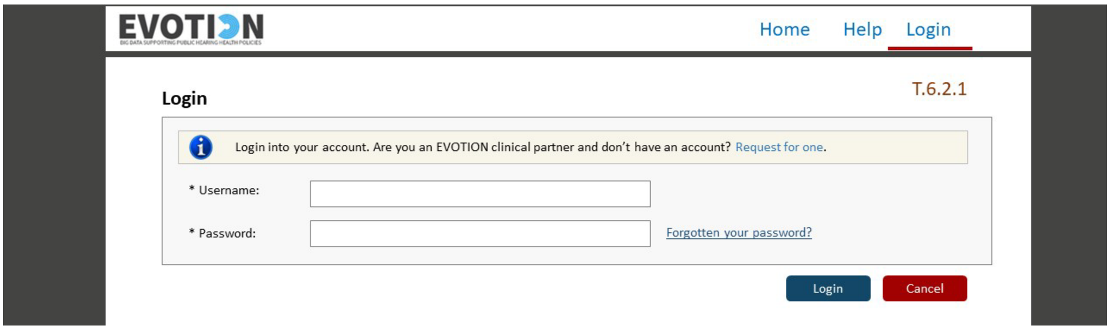


Figure 33: Scenario 8. Protection of people with hearing impairments from the harmful effects of loud noise

APPENDIX 3 - EHS Detailed web interface design



Login

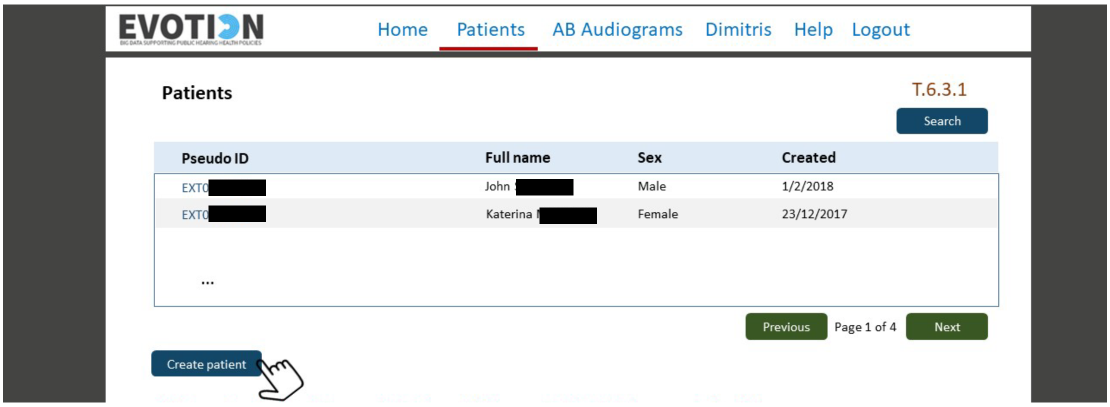
Login into your account. Are you an EVOTION clinical partner and don't have an account? Request for one.

* Username:

* Password:

Forgotten your password?

Figure 34: End-user's log in.

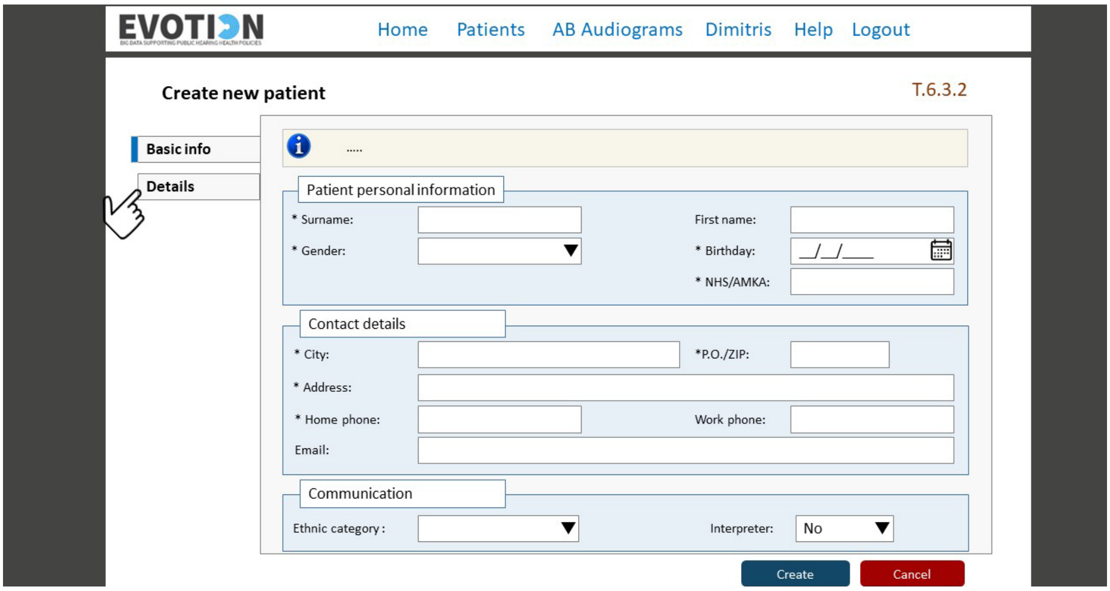


Patients

Pseudo ID	Full name	Sex	Created
EXT0[redacted]	John [redacted]	Male	1/2/2018
EXT0[redacted]	Katerina [redacted]	Female	23/12/2017
...			

Page 1 of 4

Figure 35: All available patients presented in Zebra striping tabular form. Selecting the desired link ID from the list will automatically make it the active selection (patient's specific information to be displayed).



Create new patient

.....

Patient personal information

* Surname:

First name:

* Gender:

* Birthday: __/__/____

* NHS/AMKA:

Contact details

* City:

*P.O./ZIP:

* Address:

* Home phone:

Work phone:

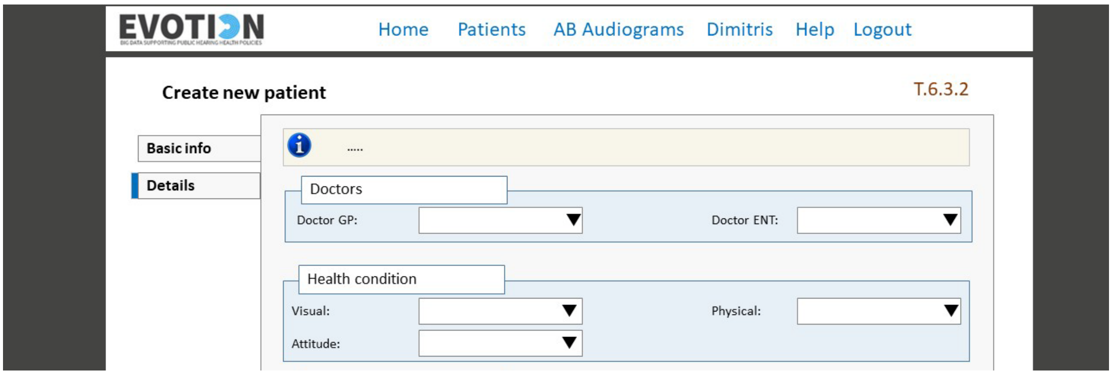
Email:

Communication

Ethnic category :

Interpreter: No

Figure 36: Creation of a new patient (part A: basic info).



Create new patient

T.6.3.2

.....

Doctors

Doctor GP: ▼

Doctor ENT: ▼

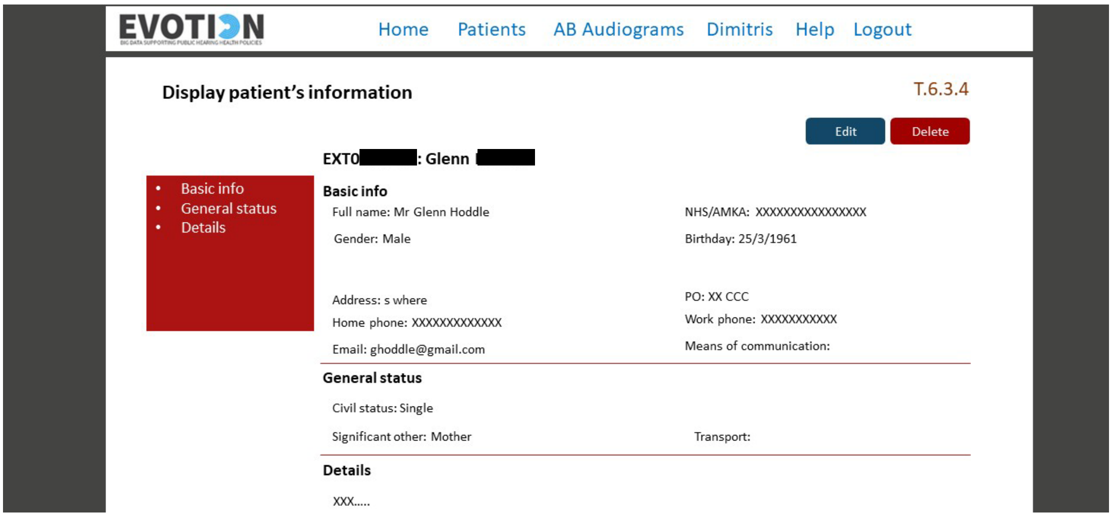
Health condition

Visual: ▼

Physical: ▼

Attitude: ▼

Figure 37: Creation of a new patient (part B: details)



Display patient’s information

T.6.3.4

EXT0[redacted]: Glenn [redacted]

Basic info

Full name: Mr Glenn Hoddle

NHS/AMKA: XXXXXXXXXXXXXXX

Gender: Male

Birthday: 25/3/1961

Address: s where

PO: XX CCC

Home phone: XXXXXXXXXXXXX

Work phone: XXXXXXXXXXX

Email: ghoddle@gmail.com

Means of communication:

General status

Civil status: Single

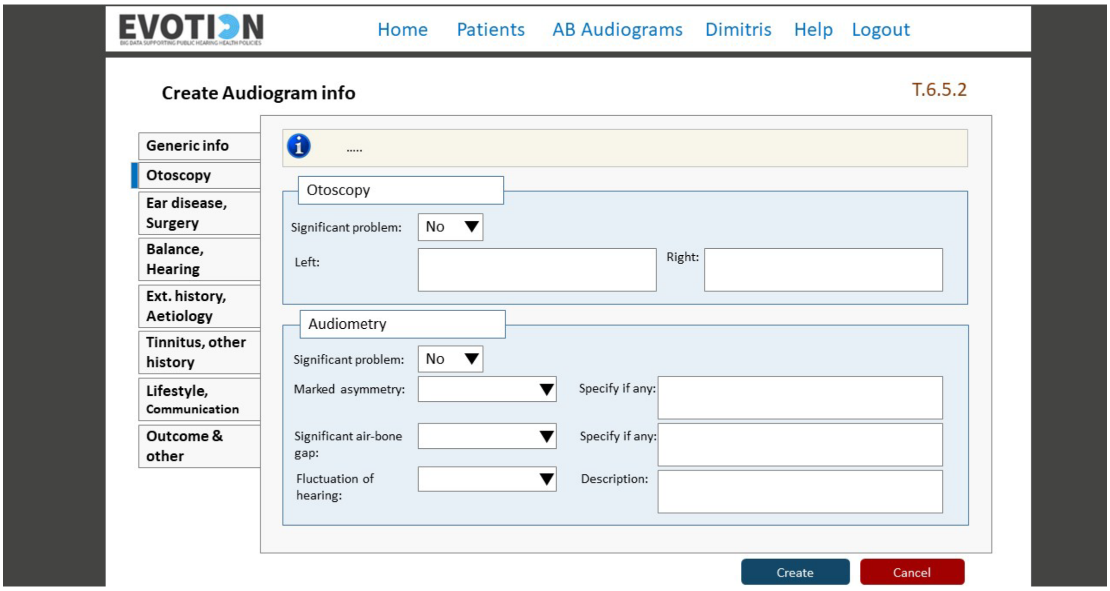
Significant other: Mother

Transport:

Details

XXX.....

Figure 38: Portion of current patient’s personal details. Key identification properties will be blanked out to preserve the patient's anonymity.



Create Audiogram info

T.6.5.2

.....

Otoscopy

Significant problem: No ▼

Left:

Right:

Audiometry

Significant problem: No ▼

Marked asymmetry: ▼

Specify if any:

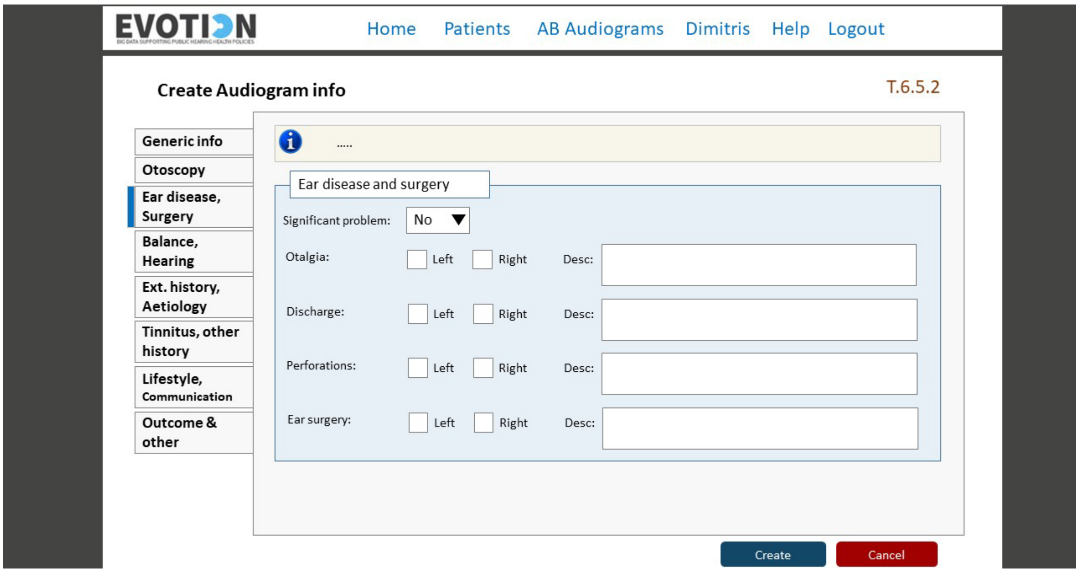
Significant air-bone gap: ▼

Specify if any:

Fluctuation of hearing: ▼

Description:

Figure 39: Editing patient’s medical history (Otoscopy).



Create Audiogram info

T.6.5.2

.....

Ear disease and surgery

Significant problem: No ▼

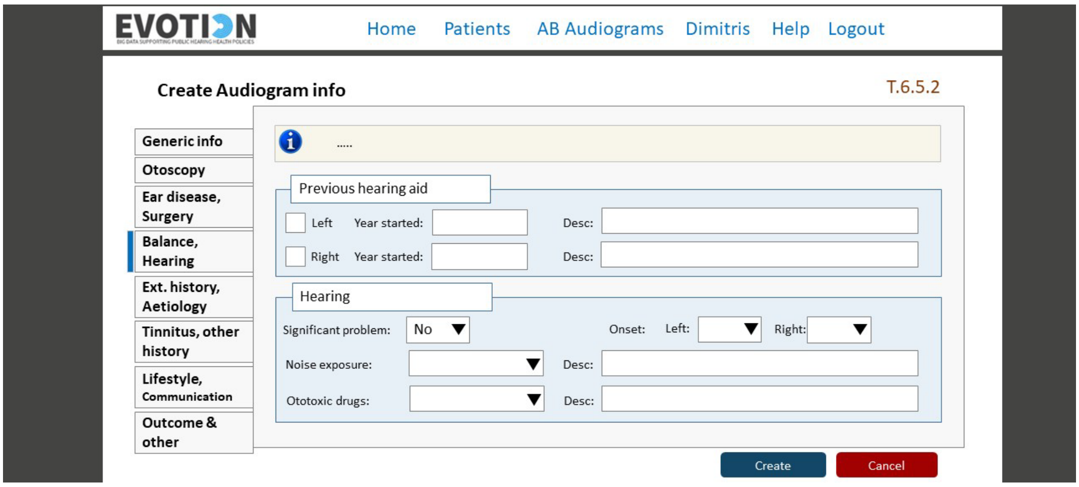
Otalgia: □ Left □ Right Desc:

Discharge: □ Left □ Right Desc:

Perforations: □ Left □ Right Desc:

Ear surgery: □ Left □ Right Desc:

Figure 40: Editing patient's medical history (Ear disease - surgery).



Create Audiogram info

T.6.5.2

.....

Previous hearing aid

□ Left Year started: Desc:

□ Right Year started: Desc:

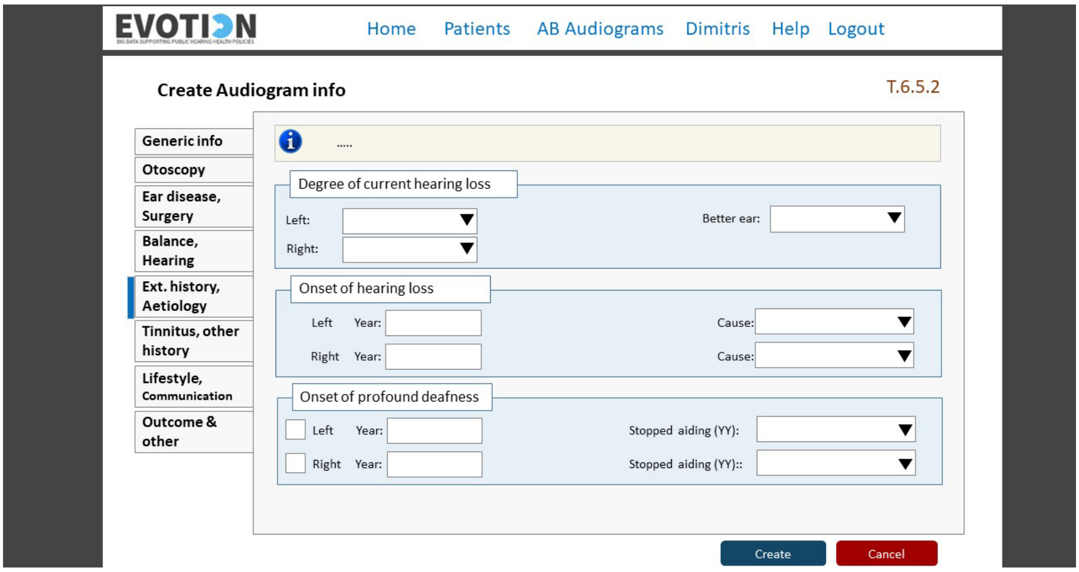
Hearing

Significant problem: No ▼ Onset: Left: ▼ Right: ▼

Noise exposure: ▼ Desc:

Ototoxic drugs: ▼ Desc:

Figure 41: Editing patient's medical history (Balance - hearing).



Create Audiogram info

T.6.5.2

.....

Degree of current hearing loss

Left: ▼ Better ear: ▼

Right: ▼

Onset of hearing loss

Left Year: Cause: ▼

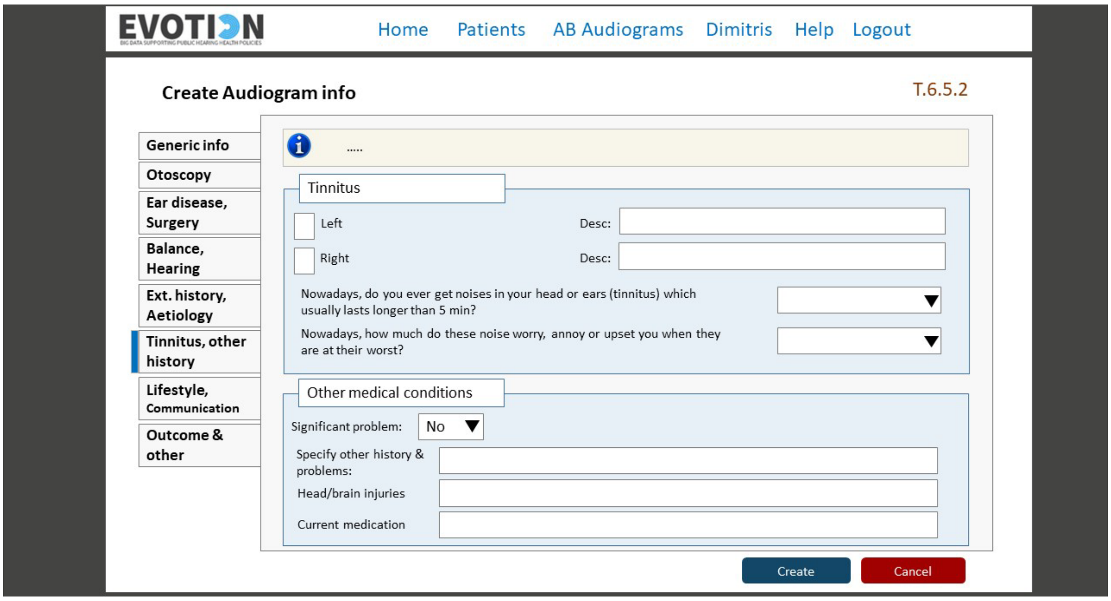
Right Year: Cause: ▼

Onset of profound deafness

□ Left Year: Stopped aiding (YY): ▼

□ Right Year: Stopped aiding (YY):: ▼

Figure 42: Editing patient's medical history (Ext. history - Aetiology).



Create Audiogram info

T.6.5.2

.....

Tinnitus

□ Left Desc:

□ Right Desc:

Nowadays, do you ever get noises in your head or ears (tinnitus) which usually lasts longer than 5 min? ▼

Nowadays, how much do these noise worry, annoy or upset you when they are at their worst? ▼

Other medical conditions

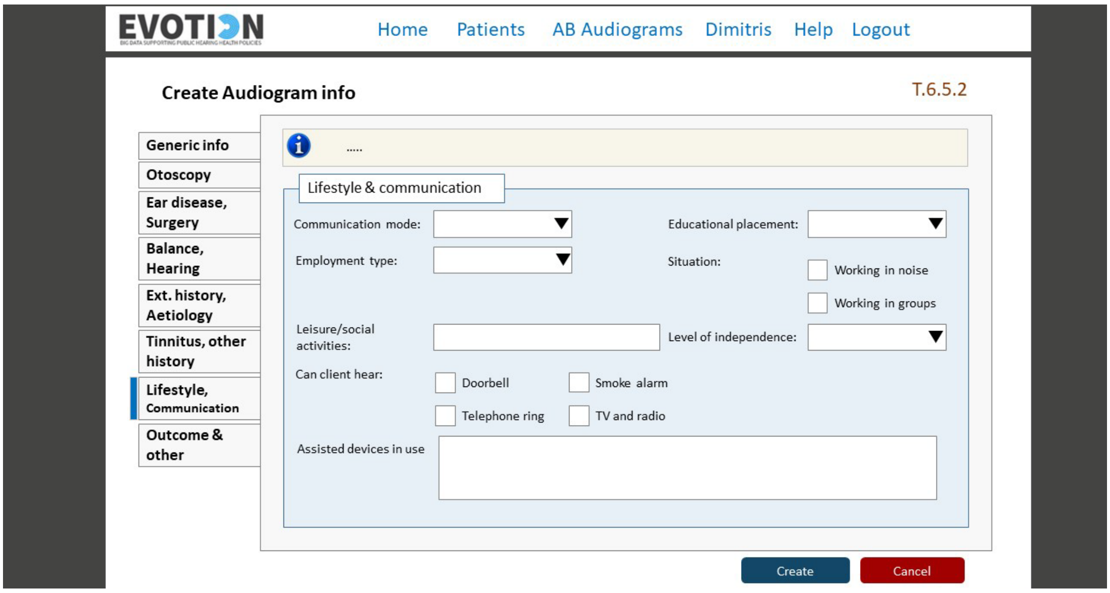
Significant problem: No ▼

Specify other history & problems:

Head/brain injuries

Current medication

Figure 43: Editing patient's medical history (Tinnitus – Other history).



Create Audiogram info

T.6.5.2

.....

Lifestyle & communication

Communication mode: ▼ Educational placement: ▼

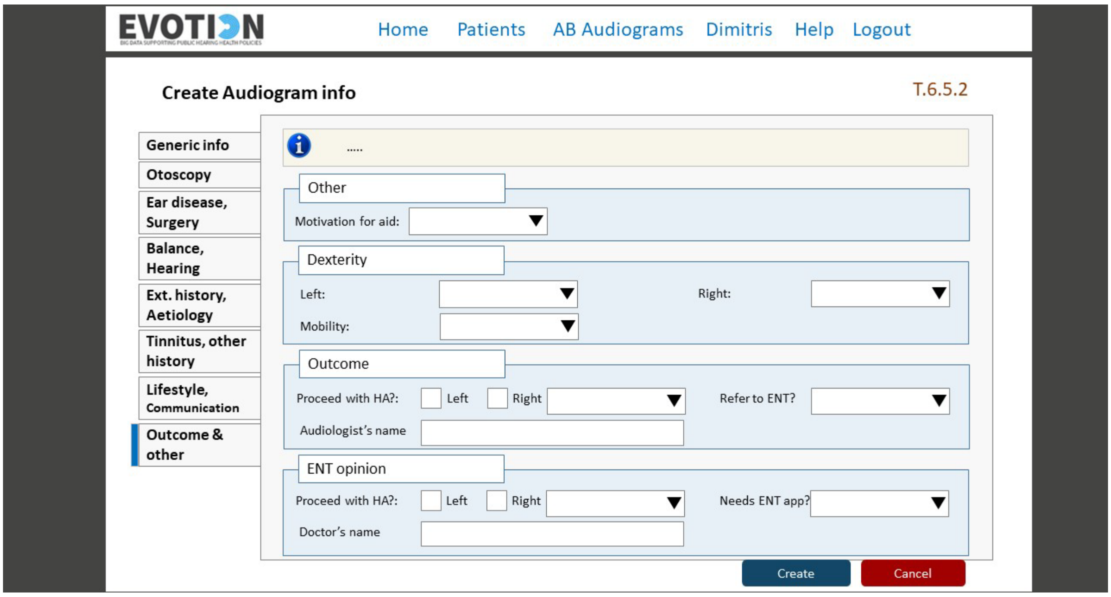
Employment type: ▼ Situation: □ Working in noise □ Working in groups

Leisure/social activities: Level of independence: ▼

Can client hear: □ Doorbell □ Smoke alarm □ Telephone ring □ TV and radio

Assisted devices in use

Figure 44: Editing patient's medical history (Lifestyle - Communication).

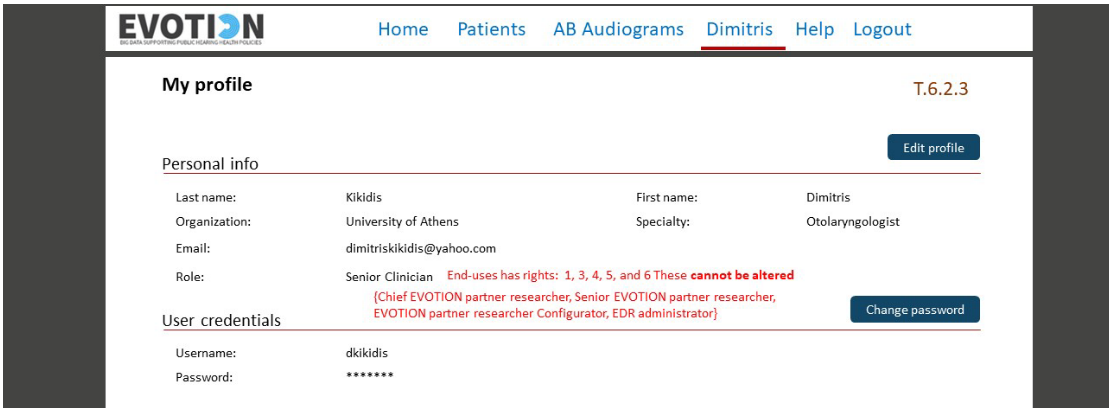


Create Audiogram info

T.6.5.2

......

Figure 45: Editing patient's medical history (Outcome - Other).



My profile

T.6.2.3

Personal info

Last name:	Kikidis	First name:	Dimitris
Organization:	University of Athens	Specialty:	Otolaryngologist
Email:	dimitriskikidis@yahoo.com		
Role:	Senior Clinician		

End-uses has rights: 1, 3, 4, 5, and 6 These **cannot be altered** {Chief EVOTION partner researcher, Senior EVOTION partner researcher, EVOTION partner researcher Configurator, EDR administrator}

User credentials

Username:	dkikidis
Password:	*******

Figure 46: End-user's log in info.

APPENDIX 4- EVOTION API in details

Data Repository API

Evotion Hospital System (EHS) API provides basic services for the audiology clinic management.

a. API Roles:

0: EDR administrator → EHS admin
1: Chief Clinician
2: Senior Clinician
3: Research Clinician
4: Clinician 8: Configurator

Speciality (specialtyid)
0: Undefined
1: General practitioner
2: Otolaryngologist 3: ENT clinician

Organization (organizationid)
0: Undefined
1: University College London
2: University of Athens
3. Guys Hospital
4. City University
5. James Paget Hospital
6. Oticon
7. Iatriko Hospital

ETHNIC_ORIGINS: undefined list
1: Greek people by ethnic or national origin, 9: Other

b. EHS RESTful Web Services:

$hostname = http://hospital.ip.address/ehs/

Operation Description	End user login	
Rest API URL	($hostname)/login/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"uname": "",   "xxxxx": ""
}
	Application/JSON	

Output parameters		
Name	**Type**	**Description**
“token”	String	On success insertion, this function returns the security token.

Operation Description	Create new end user	
Rest API URL	($hostname)/rest/api/createuser/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"fname": "",
"lname": "",
"organizationid”:,
"specialtyid”:,
"email": "",
"uname": "",
"password": "",
"mobile”:, "role_id":
}
	JSON Object	Fname: End user First Name
		Lname: End user Last name
		Organizationid: Hospital/Organization
		Specialtyid: End user Speciality
		Email: End user email
		Uname: End user username
		Password: End user password
		Mobile: End Users Mobile
		Role_id: End Users Role

Output parameters		
Name	**Type**	**Description**
Status	Integer	On success insertion this function returns the status code.
		Status: {1: pending registration, 2: accepted, 3: rejected}

Operation Description	View all registrations

Rest API URL	($hostname) /rest/api/users/registration/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**
Array of [{ “userID”:, "fname": "", "lname": "", "record_date": "organizationid": , "specialtyid": , "role_id": , “approver_id”: } ….]	JSON Array	Only EHS administrator "role_id": 0 may trigger this function. It returns all the registered end users.

Operation Description	View all registered end users of a specific organization	
Rest API URL	($hostname) /rest/api/users/viewall/{organizationid}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**

Array of [{ “userID”:, "fname": "", "lname": "", "record_date": "organizationid": , "specialtyid": , "role_id": , “approver_id”: } ….]	JSON Array	Only EHS administrator "role_id": 0 may trigger this function. It returns all the registered end users for a specific organization.

Operation Description	Update End User registration	
Rest API URL	($hostname) /rest/api/users/register/update/{userid}/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
“userID”:,
"fname": "",
"lname": "",
"organizationid": ,
"specialtyid": ,
"email": "",
"mobile":,
"role_id":,
“status”:,
}
	JSON Object	-

Output parameters		
Name	**Type**	**Description**
userID	Integer	Only EHS administrator "role_id": 0 may trigger this function. It returns the userID.

Operation Description	Update End Use’s information	
Rest API URL	($hostname) /rest/api/users/update/{userid}/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
“userID”:,
"fname": "",
"lname": "",
"organizationid": ,
"specialtyid": ,
"email": "",
"uname": "",
"password": "",
"mobile":
}
	JSON Object	Only accepted end users (status = 2) may trigger this function.

Output parameters		
Name	**Type**	**Description**
status	Integer	Status: {1: pending registration, 2: accepted, 3: rejected}

Operation Description	View End User’s information	
Rest API URL	($hostname) /rest/api/users/view/{userid}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
Output parameters		
Name	**Type**	**Description**

{
"user_id":,
"fname": "",
"lname": "",
"organizationid": ,
"specialtyid": ,
"email": "",
"uname": "",
"password": "",
"record_date":,
"mobile":,   "role_id": ,
“approver_id” :
}
	JSON Object	End User Information

Operation Description	View End User’s information	
Rest API URL	($hostname) /rest/api/users/view/uname/{uname}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**

{
"user_id":,
"fname": "",
"lname": "",
"organizationid": ,
"specialtyid": ,
"email": "",
"uname": "",
"password": "",
"record_date":,
"mobile":,   "role_id": ,
“approver_id” :
}
	JSON Object	End User Information

Operation Description	Create Evotion Patient
Rest API URL	($hostname) /rest/api/patients/create/{userid}/

Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"user_id" the end user who created him
"sys_createstamp"
"title"
"firstname"
"lastname"
"secondarycode"
"dateofbirth"
"sex"
"city"
"postalcode"
"address"
"workphone"
"email"
"ethnic_origin"
"interpr_req"
"interpr_lng_id"
"doctor_gp_id"
"doctor_ent_id"
"conditionvisual"
"conditionphysical"
"conditionattitude"
}
	JSON Object	Creates patient's data. An end-user with proper rights may trigger this function
		(e.g., "role_id" = {2}

Output parameters		
Name	**Type**	**Description**
"patient_id": "" "civilstatus":	JSON Object	Returns newly created civilstatus

Operation Description	Anonymise patient's ID	
Rest API URL	($hostname) /rest/api/patients/anonymise/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"patient_id”:, "userid":
}
	JSON Object	Replaces “patient_id” value with “civilstatus” (the pseudoID) and removes any personal data (first name, last name, birthday, addresses, and phone numbers). An end-user with proper rights may trigger this function (e.g.,
		"role_id" = {2})

Output parameters		
Name	**Type**	**Description**

{
"civilstatus":, "userid":
}
	JSON Object	On success, returns HTTP code 200.

Operation Description	Get all patients	
Rest API URL	($hostname) /rest/api/patients/getall/{user_id}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**

[{ [0] => Array ([patient_id] => EHS10 [guid] => [lastname] => Ye [address] => 6th Ring Road [address2] => [alert] => [alert_synchronized] => [annual_pmt_free] => [annual_pmt_method] => [annualcard_datedue] => [annualcard_number] => [city] => Beijing [civilstatus] => EX20******* [code_verif_by_pas] => [code_verif_dttime] => [code_verif_stat_id] => [code_verif_user_id] => [color_id_l] => [color_id_r] => [comm_hearing_aid] => [conditionattitude] => 3 [conditionphysical] => 3 [conditionvisual] => 3 [country] => [cust_group_id] => [customer_id] => [dateofbirth] => 331776000 [dateofdeath] => [dateofdeathreg] => [doctor_ent_id] => 3003 [doctor_gp_id] => 2002 [email] => someother@gmail.com [ethnic_origin] => 9 [exchange_approved] => [external_subm_num] => [external_system_id] => [extraname] => [firstname] => Bin [funding_progr_id] => [hl7_address1] => [hl7_address2] => [hl7_address3] => [hl7_address4] => [hl7_address5] => [homephone] => 23456789 [immun_date] => [immun_status_id] => [immunisation_id] => [implant_keyworker] => [import_error] => [insurance1] => [insurance2] => [interpr_lng_id] => 7 [interpr_req] => 0 [invoice_receiver] => [nhs_issue] => [note] => [other1] => [other2] =>	JSON Array	Return patients list and **a portion of all patient's data** within organization. An end-user with proper rights may trigger this function (e.g., "role_id" = {2})

[phone_mobile] => [physician] => [postalcode] => 54544 [postcode_id] => [pref_comm] => [primarycode] => [province] => [referral] => [relatives] => [salutation] => [secondarycode] => 55555 [sex] => M [significantother] => [sms_consent_no] => [sms_consent_yes] => [surg_consent_date] => [surg_consent_id] => [sys_createstamp] => 1520981303 [sys_createuser] => [sys_d_alert_modify] => [sys_d_alert_user] => [sys_deleted] => [sys_modifystamp] => [sys_modifyuser] => [sys_pas_update] => [title] => Mr [transport] => [user_data_field1] => [user_data_field2] => [user_dropdown1] => [user_dropdown2] => [user_id] => 3003 [visitaddress] => [war_pensioner] => [workphone] => 879654111 [zip] =>) } ….]		

Operation Description	View patients data	
Rest API URL	($hostname) /rest/api/patients/view/{userid}/{Patient_id}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**

{ "patient_id": "100" the real PatientID generated by the system "civilstatus": EXT0000232 "User_id": 201 "sys_createstamp": 813157200000 "title" "firstname" "lastname" "secondarycode" "dateofbirth" "sex" "city" "postalcode" "address" "workphone" "email" "ethnic_origin" "interpr_req" "interpr_lng_id" "doctor_gp_id" "doctor_ent_id" "conditionvisual" "conditionphysical" "conditionattitude" Other fields will be ignored …. }	JSON Object	Return the full set of patient's data (where patient == Patient_id). An end-user with proper rights may trigger this function (e.g., "role_id" = {2})

Operation Description	Update patient data	
Rest API URL	($hostname)/rest/api/patients/update/{userid}/{patient_id}/	
Rest API Type	PUT	
Input parameters		
Body	**Type**	**Description**

{
"patient_id": "100" the real PatientID generated by the system
"civilstatus": EXT0000232
"User_id": 201 "Sys_createstamp": 813157200000
"title"
"firstname"
"lastname"
"secondarycode"
"dateofbirth"
"sex"
"city"
"postalcode"
"address"
"homephone"
"workphone"
"email"
"ethnic_origin"
"interpr_req"
"interpr_lng_id"
"doctor_gp_id"
"doctor_ent_id"
"conditionvisual" "conditionphysical"
"conditionattitude"
}
	JSON Object	Updates the full set of patient’s data (where "civilstatus" == pseudoID). An end-user with proper rights may trigger this function (e.g., "role_id" = {2})

Output parameters		
Name	**Type**	**Description**
200	HTTP Code	

Operation Description	Search patient data	
Rest API URL	($hostname)/rest/api/patients/search/{userid}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**

[{ "patient_id": "100", PatientID "civitstatus": EXT0000232, Critical: use this field to store pseduID "secondarycode": null, "title": "EAV", "lastname": "EAV", "firstname": "EAV", "sex": "E", "user_id": null, "dateofbirth": -813157200000, "city":null "country": null "doctor_ent_id": null, "doctor_gp_id": null, } ....]	JSON Array	Return patients list and a portion of their data within organization. An end-user with proper rights may trigger this function (e.g., "role_id" = {2})

Operation Description	Create Audiogram info for existing pseudoID	
Rest API URL	($hostname) /rest/api/patients/audiogram/create/{userid}/{pa tient_id}/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{ “patient_id”: varchar(50) not null, “create_date”: timestamp (created by web gui) "civilstatus": EXT0000232, **Critical: pseudoID** //"user_id": 3003, for future version “otos_problem”: X, small int, Q_DRMED.OTOSCOPY_PROB “otos_l”: “text”, char(120), Q_DRMED.Q_LEFT “otos_r”: “text”, char(120), Q_DRMED.Q_RIGHT “audiometry_prob”: X, small int, Q_DRMED.AUDIOMETRY_PROB “asymmetry”: {Null, Yes, No, N/A}, CHAR(5), Q_DRMED.ASYMMETRY “assymetry_desc”: “text”, CHAR(160), Q_DRMED.ASYMMETRY_DESC “airbone”: {Null, Yes, No, N/A}, CHAR(5), Q_DRMED.AIRBONE “airbone_desc”: “text”, CHAR (160), Q_DRMED.AIRBONE_DESC “fluct_status”: {Null, Yes, No, N/A}, CHAR(5), Q_DRMED.FLUNCTUATION_STATUS “fluct_desc”: “text”, CHAR (160), Q_DRMED.FLUNCTUATION_DESC “Other_comment”: “text”, CHAR(254), Q_DRMED.OTHER_COMMENT //not used “ear_disease_prob”: 0-1, small int, EAR_DISEASE_PROB “otalgia_l”: 0-1, small int, Q_DRMED.OTALGIA_L “otalgia”: 0-1, small int, Q_DRMED.OTALGIA “otalgia_desc”: “text”, CHAR (220) Q_DRMED.OTA_DESC “discharge_l”: 0-1, small int, DISCHARGE_L “discharge”: 0-1, small int, DISCHARGE “discharge_desc”: “text”, CHAR (220), Q_DRMED.DISCHARGE_DESC “perforations_l”: 0-1, small int, Q_DRMED.PERFORATIONS_L “perforations”: 0-1, small int, Q_DRMED.PERFORATIONS “perforations_desc”:	JSON Object	Creates a full audiogram record for a pseudoID. An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

“text”, CHAR (220), Q_DRMED.PERFORATIONS_DESC “ear_surgery_l”: 0-1, small int, Q_DRMED.EAR_SURGERY_L “ear_surgery”: 0-1, small int, Q_DRMED.EAR_SURGERY “ear_surgery_desc”: “text”, CHAR (220), Q_DRMED.EAR_SURGERY_DESC “previous_aid_l”: 0-1, small int, Q_DRMED.PREVIOUS_AID_L “hi_prevha_ystart_l”: 1988, small int, Q_DRMED.HI_PREVHA_YSTART_L “previous_aid_descl”:” text”, char(240), Q_DRMED.PREVIOUS_AID_DESCL “previous_aid”: 0-1, small int, Q_DRMED.PREVIOUS_AID “hi_prevha_ystart_r”: 1988, small int, Q_DRMED.HI_PREVHA_YSTART_R “previous_aid_desc”:”t ext”, char(240), Q_DRMED.PREVIOUS_AID_DESC “hearing_prob”: X, small int, HEARING_PROB “onset_l”: {Null, From birth, Gradual, Sudden, N/A}, CHAR(20), Q_DRMED.ONSET_L “onset”: {Null, From birth, Gradual, Sudden, N/A}, CHAR(20), Q_DRMED.ONSET “noise_exposure”: {Null, Leisure, Gunfire, Occupational, N/A}, CHAR(20), NOISE_EXPOSURE “noise_expo_desc”: “text”, CHAR(250), NOISE_EXPO_DESC “ototoxic_drug”: {Null, Quinine, Diuretics, Aminoglycoside antibiotics, N/A}, CHAR(30), OTOTOXIC_DRUG “ototoxic_drug_desc”: “text”, CHAR(200), OTOTOXIC_DRUG_DESCZ “hi_degree_currhl_l”: {Null, Mild, Moderate, Severe, Profound}, CHAR(20), HI_DEGREE_CURRHL_L		

“hi_degree_currhl_r”: {Null, Mild, Moderate, Severe, Profound}, CHAR(20), HI_DEGREE_CURRHL_R “hi_better_ear”: {Null, Left, Right, N/A}, CHAR(10), HI_BETTER_EAR “hi_onset_ystart_l”: “year!”, small int, HI_ONSET_YSTART_L “hi_onset_couse_l”: {Null, Congenital, Acquired, Unknown}, varchar(20), HI_ONSET_COUSE_L “hi_onset_ystart_r”: “year”, small int, HI_ONSET_YSTART_R “hi_onset_couse_r”: {Null, Congenital, Acquired, Unknown}, varchar(20), HI_ONSET_COUSE_R “hearing_desc”: “text”, CHAR(250), HEARING_DESC //not used “hi_prdeaf_l”: 0-1, small int, HI_PRDEAF_L “hi_prdeaf_ystart_l”: “text”, small int, HI_PRDEAF_YSTART_L “hi_prdeaf_ystop_l”: “text”, small int, HI_PRDEAF_YSTOP_L “hi_prdeaf_r”: 0-1, small int, HI_PRDEAF_L “hi_prdeaf_ystart_r”: “text”, small int, HI_PRDEAF_YSTART_R “hi_prdeaf_ystop_r”: “text”, small int, HI_PRDEAF_YSTOP_R “hi_comment”: “text”, longvar, HI_HISTOR_COMMNT // not used “tinnitus_l”: 01, small int, TINNITUS_L “tinnitus_desc_l”: “text”, long var, TINNITUS_DESC_L “tinnitus”: 01, small int, TINNITUS_R “tinnitus_desc”: “text”, long var, TINNITUS_DESC_R		

“question1”: {Null, N/A, No, never, Some of the time, Most of the time}, char(50), Q_DRMED.QUESTION1 “question2”: {Null, N/A, No, Do not get noise in head or ears, Not at all annoying, Slightly annoying, Moderately annoying, Severely annoying}, char(50), Q_DRMED.QUESTION2 “other_medical_prob”: 0-1, small int, OTHER_MEDICAL_PROB “other_medical_desc”: “text”, long var, OTHER_MEDICAL_DESC “oth_head_injur”: “text”, long var, OTH_HEAD_INJUR “oth_curr_medic”: “text”, long var, OTH_CURR_MEDIC “familyhistory”: “text”, long var, FAMILYHISTORY “ls_com_mode”: {Null, Sign (BSL), Sign and Speech, Speech only}, varchar(20), LS_COMM_MODE “ls_educ_placem”: {Null, Pre-school, Mainstream, Resource base/unit for HI, Resource base/unit for other, School for deaf, Special school other, Further education mainstream (e.g., A levels, Further education technical/apprentice, Further education specialist provision for deaf, University, No longer in education other}, varchar(50), LS_EDUC_PLACEM “ls_empl_type”: {"", Regular, Part-time, Retired, Not working, Supported regular, Supported part-time}, varchar(50), LS_EMPL_TYPE “ls_empl_sit1”: 0-1, small int, LS_EMPL_SIT1 “ls_empl_sit2”: 0-1, small int, LS_EMPL_SIT2 “ls_leisure_socact”: “text”, long var, LS_LEISURE_SOCACT “independence_level”: {“”, Home carer, Independent, Independent with carers, Residential/Nursing, With family), varchar (50), INDEPENDENCE_LEVEL “ls_canhear1”: 0-1, small int, LS_CANHEAR1 “ls_canhear2”: 0-1, small int, LS_CANHEAR2		

“ls_canhear3”: 0-1, small int, LS_CANHEAR3 “ls_canhear4”: 0-1, small int, LS_CANHEAR4 “ls_canhear_commnt”: “text”, long var, LS_CANHEAR_COMMNT // not used “ls_assist_devinuse”: “text”, long var, LS_ASSIST_DEVINUSE “motivation_for_aid”: {"", Self, Family/Friends, GP, Carers, Other}, varchar(50), MOTIVATION_FOR_AID “dexterity_left”: {Null, Adequate, Poor}, varchar(50), DEXTERITY_LEFT “dexterity_right”: {Null, Adequate, Poor}, varchar(50), DEXTERITY_RIGTH “mobility”: {Null, Adequate, Poor, Wheelchair}, varchar(50), MOBILITY “outcome_proceed_l”: 0,1, smallint, OUTCOME_PROCEED_L “outcome_proceed_r”: 0,1, smallint, OUTCOME_PROCEED_R “outcome_proceed”: {Yes, No, N/A}, char(3), OUTCOME_PROCEED “outcome_refer”: {Yes, No, N/A}, char(3), OUTCOME_REFER “audiologist_name”: “text”, char(40), AUDIOLOGIST_NAME “ent_proceed_l”: 0,1, smallint, ENT_PROCEED_L “ent_proceed_r”: 0,1, smallint, ENT_PROCEED_R “**ent**_proceed”: {Yes, No, N/A}, char(3), ENT_PROCEED “ent_appt”: {Yes, No, N/A}, char(3), ENT_APPT “doctor_name”: “text”, char(40), DOCTOR_NAME **“sys_createuser”: 1**		

(creation status) }		

Output parameters		
Name	**Type**	**Description**
-	-	-

Operation Description	Update Audiogram info for existing pseudoID	
Rest API URL	($hostname) /rest/api/patients/audiogram/update/{userid}/{patient_id}/	
Rest API Type	PUT	
Input parameters		
Body	**Type**	**Description**

{
“id”: “patient_id”: varchar(50) not null,
“create_date”: timestamp (created by web gui)
"civilstatus": EXT0000232, Critical: pseudoID
//"user_id": 3003, for future version
...All data as previous
}
	JSON Object	Updates the audiogram record for a pseudoID. An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})= {2, 3})

Output parameters		
Name	**Type**	**Description**
-	-	-

Operation Description	Update Audiogram info for existing pseudoID

Rest API URL	($hostname) /rest/api/patients/audiogram/viewall/{userid}/{patient_ id}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**

[
{
“id” : int,
“patient_id”: EHS system ID
"civilstatus": EXT0000232,
"user_id": 3003,
“investdate”: timestamp
“sys_createuser ”: 1 (constrain)
...rest of data
}
]
	JSON Array	An end-user (userid) with proper rights may trigger this function (e.g.,
		"role_id" = {2, 3})

Operation Description	View all Audiogram info for a patient	
Rest API URL	($hostname) /rest/api/patients/audiogram/viewall/{userid}/{patient_ id}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		

Name	Type	Description

[
{
“id” : int,
“patient_id”: EHS system ID
"civilstatus": EXT0000232,
"user_id": 3003, “investdate”: timestamp
“sys_createuser”: 1 (constrain) ...rest of data
}
]
	JSON Array	An end-user (userid) with proper rights may trigger this function (e.g.,
		"role_id" = {2, 3})

Operation Description	View a specific Audiogram info	
Rest API URL	($hostname) /rest/api/patients/audiogram/viewall/{userid}/{sys_create stamp}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**

{ "id" : int, "patient_id" : EHS system ID "civilstatus ": EXT0000232, "user_id": 3003, "investdate" : timestamp ...rest of data }	JSON Object	An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

Operation Description	Delete Audiogram Infor for existing patient	
Rest API URL	($hostname) /rest/api/patients/audiogram/delete/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"id":1
"patient_id":"EHS1",
"sys_createstamp":1524227754630 }
	Application/json	Delete an existing patient

Output parameters		
Name	**Type**	**Description**
-	-	-

Operation Description	Create Audiogram data for existing pseudoID
Rest API URL	($hostname) /rest/api/patients/audiogramdata/ create/{userid}/{patient_id}/{organizarion_id}/
Rest API Type	POST
Input parameters	

Body	Type	Description
{ "file": Path to xml file	Application/json	Creates a Audiogram for a pseudoID by sending an XML
}		file. An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3}).
Output parameters		
Name	**Type**	**Description**
200	HTTP Code	

Operation Description	View all Audiograms for a patient	
Rest API URL	($hostname) /rest/api/patients/audiogramdata/viewall/{userid}/{patie nt_id}/	
Rest API Type	GET	
Input parameters		
Body	**Type**	**Description**
-	-	-
Output parameters		
Name	**Type**	**Description**

[{ "patient_id": int, "firstname" "lastname": "userid": 203, "created_time ": longint "curve": xx "investdate": , "GUID" : }]	JSON Array	Return the audiograms

Operation Description	Delete Audiogram data	
Rest API URL	($hostname) /rest/api/patients/audiogramdata/delete/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"patient_id":
int,
"created_date": longint
"curve": xx,
"investdate":
date, "guid":""
}
	JSON Object	-

Output parameters		
Name	**Type**	**Description**
200	HTTP Code	-

Operation Description	Create Referral for existing pseudoID

Rest API URL	($hostname) /rest/api/patients/referral/create/{userid}/{pseudoID}/	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
“refdate”:,
“refcreatedate”:,
“treatmenttype”:,
“reason”:,
“customer”:,
“medicalpriorityid”:
,
“information”: ,
“interviewnote”:
}
	JSON Object	Creates a referral for a pseudoID. An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

Output parameters		
Name	**Type**	**Description**
200	HTTP Code	-

Evotion Hospital System (EHS) API

Bellow follows a breakdown of all the REST API services that have been built for the EHS system. To make this somewhat more comprehensive the order that has been used attempts to convey the grouping of the resources of the system’s database. For example, the topmost services are the ones that conduct operations against the basic entity of the system, i.e., the patients. Subsequently, we present all the properties of the patient e.g. the patient’s personal information, the patient’s Audiogram info (i.e., AuditBase object for handling patient’s medical history) and data (i.e., AuditBase object for handling patient’s audiogram), etc.

End User

End user login
POST $hostname. "/login/

```
{
  "uname": "test01",
  "password": "xxxxx",
}
```

On success insertion this function returns userID.
This record is associated with "status": 1 (default = pending registration)
System automatically generates the Timestamp "record_date": 1450450438296

approver_id: {1:pending registration, 2: accepted, 3: rejected}

End user registration

POST $hostname. "/rest/api/users/registration/create/

```
{
  "fname": "test",
  "lname": "test",
  "organizationid": 2,
  "specialtyid": 3,
  "email": "xxx@xxx.xxx",
  "uname": "test",
  "password": "xxxxx",
  "mobile": null,
  "role_id": 2
}
```

On success insertion this function returns userID.
This record is associated with "status": 1 (default = pending registration)
System automatically generates the Timestamp "record_date": 1450450438296
Status: {1:pending registration, 2: accepted, 3: rejected}

View all registrations

GET $hostname. "/rest/api/users/registration/

Only EHS administrator "role_id": 0 may trigger this function

Array of

```
[
{
  “userID”: XXXX,
  "fname": "XXXX",
  "lname": "XXXX",
  "record_date": 1450450438296
  "organizationid": 2,
  "specialtyid": 3,
  "role_id": 2,
  “approver_id”: 1}
….
]
```

View all registered endusers of an organization

GET $hostname. "/rest/api/users/viewall/{organizationid}/

```
Array of
[
{
  “userID”: XXXX,
  "fname": "XXXX",
  "lname": "XXXX",
  "record_date": 1450450438296
  "organizationid": 2,
  "specialtyid": 1, 2 or 3,
  "role_id": 2,
  “approver_id”: 2}
….
]
```

Update End user registration

POST $hostname. "/rest/api/users/register/update/{userid}/

Only EHS administrator "role_id": 0 may trigger this function and alter "role_id" and “status” and other info

```
{
  “userID”: XXXX,
  "fname": "XXXX",
  "lname": "XXXX",
  "organizationid": 2,
  "specialtyid": 3,
  "email": " xxx@xxx.xxx ",
  "mobile": null,
  "role_id": 2,
  “status”: 2 or 3,
}
```

Update of end user’s info

POST $hostname. "/rest/api/users/update/{userid}/

Only accepted end users (status = 2) may trigger this function

```
{
  "user_id": 203,
  "fname": "XXXX",
  "lname": "XXXX",
  "organizationid": 2,
  "specialtyid": 3,
  "email": " xxx@xxx.xxx ",
  "uname": "XXXX",
  "password": " XXXX ",
  "mobile": null
}
```

View end user’s info

GET $hostname. "/rest/api/users/view/{user_id}/

```
{
  "user_id": 203,
  "fname": "XXXX",
  "lname": "XXXX",
  "organizationid": 2,
  "specialtyid": 3,
  "email": " xxx@xxx.xxx ",
  "uname": "XXXX",
  "password": " XXXX ",
  "record_date": 1450450438296,
  "mobile": null,
  "role_id": 2,
  “approver_id”: 2
}
```

Patients

Create patient

POST $hostname. “/rest/api/patients/create/{userid}/

Creates patient’s data. An end-user with proper rights may trigger this function (e.g., "role_id" = {2})
{
"user_id" the end user who created him
"sys_createstamp" 813157200000, longint type = now()
"title"
"firstname"
"lastname"
"secondarycode"
"dateofbirth"
"sex"
"city"
"postalcode"
"address"
"workphone"
"email"
"ethnic_origin"
"interpr_req"
"interpr_lng_id"
"doctor_gp_id"
"doctor_ent_id"
"conditionvisual"
"conditionphysical"
"conditionattitude"
}

Get all patients

GET $hostname. “/rest/api/patients/getall/{user_id}/

Return patients list and **a portion of all patient's data** within organization. An end-user with proper rights may trigger this function (e.g., "role_id" = {2})

Repositioned some of the parameters to indicate ones to be utilised

```
[
{
    [0] => Array ( [patient_id] => EHS10 [guid] => [lastname] => XXXX [address] => 6th Ring Road [address2] =>
[alert] => [alert_synchronized] => [annual_pmt_free] => [annual_pmt_method] => [annualcard_datedue] =>
[annualcard_number] => [city] => Beijing [civilstatus] => EX*********** [code_verif_by_pas] =>
[code_verif_dttime] => [code_verif_stat_id] => [code_verif_user_id] => [color_id_l] => [color_id_r] =>
[comm_hearing_aid] => [conditionattitude] => 3 [conditionphysical] => 3 [conditionvisual] => 3 [country] =>
[cust_group_id] => [customer_id] => [dateofbirth] => 331776000 [dateofdeath] => [dateofdeathreg] =>
[doctor_ent_id] => 3003 [doctor_gp_id] => 2002 [email] => someother@gmail.com [ethnic_origin] => 9
[exchange_approved] => [external_subm_num] => [external_system_id] => [extraname] => [firstname] => XXX
[funding_progr_id] => [hl7_address1] => [hl7_address2] => [hl7_address3] => [hl7_address4] => [hl7_address5]
=> [homephone] => 23456789 [immun_date] => [immun_status_id] => [immunisation_id] => [implant_keyworker]
=> [import_error] => [insurance1] => [insurance2] => [interpr_lng_id] => 7 [interpr_req] => 0 [invoice_receiver] =>
[nhs_issue] => [note] => [other1] => [other2] => [phone_mobile] => [physician] => [postalcode] => 54544
[postcode_id] => [pref_comm] => [primarycode] => [province] => [referral] => [relatives] => [salutation] =>
[secondarycode] => 55555 [sex] => M [significantother] => [sms_consent_no] => [sms_consent_yes] =>
[surg_consent_date] => [surg_consent_id] => [sys_createstamp] => 1520981303 [sys_createuser] =>
[sys_d_alert_modify] => [sys_d_alert_user] => [sys_deleted] => [sys_modifystamp] => [sys_modifyuser] =>
[sys_pas_update] => [title] => Mr [transport] => [user_data_field1] => [user_data_field2] => [user_dropdown1] =>
[user_dropdown2] => [user_id] => 3003 [visitaddress] => [war_pensioner] => [workphone] => 879654111 [zip] =>
)
}
….
]
```

View patient data

GET $hostname. "/rest/api/patients/view/{userid}/{Patient_id}/

Return the full set of patient’s data (where patient == Patient_id). An end-user with proper rights may trigger this function (e.g., "role_id" = {2})

returns:

```
{
"patient_id": "100" the real PatientID generated by the system
"civilstatus": EXT0000232
"User_id": 201
"sys_createstamp": 813157200000
"title"
"firstname"
"lastname"
"secondarycode"
"dateofbirth"
"sex"
"city"
"postalcode"
"address"
"workphone"
"email"
"ethnic_origin"
"interpr_req"
"interpr_lng_id"
"doctor_gp_id"
"doctor_ent_id"
"conditionvisual"
"conditionphysical"
"conditionattitude"


Other fields will be ignored
….
}
```

Update patient data

PUT $hostname. “/rest/api/patients/update/{userid}/{patient_id}/

Updates the full set of patient's data (where "civilstatus" == pseudoID). An end-user with proper rights may trigger this function (e.g., "role_id" = {2})

Input
{
"patient_id": "100" the real PatientID generated by the system
"civilstatus": EXT0000232
"User_id": 201
"Sys_createstamp": 813157200000
"title"
"firstname"
"lastname"
"secondarycode"
"dateofbirth"
"sex"
"city"
"postalcode"
"address"
"homephone"
"workphone"
"email"
"ethnic_origin"
"interpr_req"
"interpr_lng_id"
"doctor_gp_id"
"doctor_ent_id"
"conditionvisual"
"conditionphysical"
"conditionattitude"
}

Returns 200

Delete patient data

DELETE $hostname. "/rest/api/patients/delete/{userid}/{patient_id}/

200

Audiogram info (Q_DRMED)

Create Audiogram info for existing pseudoID

POST $hostname. "/rest/api/patients/audiogram/create/{userid}/{patient_id}/

Creates a full audiogram record for a pseudoID. An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

```
{
     “patient_id”: varchar(50) not null,
     “create_date”: timestamp (created by web gui)
    "civilstatus": EXT0000232, Critical: pseudoID
    //"user_id": 3003, for future version


    “otos_problem”: X, small int, Q_DRMED.OTOSCOPY_PROB
    “otos_l”: “text”, char(120), Q_DRMED.Q_LEFT
    “otos_r”: “text”, char(120), Q_DRMED.Q_RIGHT
    “audiometry_prob”: X, small int, Q_DRMED.AUDIOMETRY_PROB
    “asymmetry”: {Null, Yes, No, N/A}, CHAR(5), Q_DRMED.ASYMMETRY
    “assymetry_desc”: “text”, CHAR(160), Q_DRMED.ASYMMETRY_DESC
    “airbone”: {Null, Yes, No, N/A}, CHAR(5), Q_DRMED.AIRBONE
    “airbone_desc”: “text”, CHAR (160), Q_DRMED.AIRBONE_DESC
    “fluct_status”: {Null, Yes, No, N/A}, CHAR(5), Q_DRMED.FLUNCTUATION_STATUS
    “fluct_desc”: “text”, CHAR (160), Q_DRMED.FLUNCTUATION_DESC
    “Other_comment”: “text”, CHAR(254), Q_DRMED.OTHER_COMMENT //not used


    “ear_disease_prob”: 0-1, small int, EAR_DISEASE_PROB
    “otalgia_l”: 0-1, small int, Q_DRMED.OTALGIA_L
    “otalgia”: 0-1, small int, Q_DRMED.OTALGIA
    “otalgia_desc”: “text”, CHAR (220) Q_DRMED.OTA_DESC
    “discharge_l”: 0-1, small int, DISCHARGE_L
    “discharge”: 0-1, small int, DISCHARGE
    “discharge_desc”: “text”, CHAR (220), Q_DRMED.DISCHARGE_DESC
    “perforations_l”: 0-1, small int, Q_DRMED.PERFORATIONS_L
    “perforations”: 0-1, small int, Q_DRMED.PERFORATIONS
    “perforations_desc”: “text”, CHAR (220), Q_DRMED.PERFORATIONS_DESC
    “ear_surgery_l”: 0-1, small int, Q_DRMED.EAR_SURGERY_L
    “ear_surgery”: 0-1, small int, Q_DRMED.EAR_SURGERY
    “ear_surgery_desc”: “text”, CHAR (220), Q_DRMED.EAR_SURGERY_DESC

    “previous_aid_l”: 0-1, small int, Q_DRMED.PREVIOUS_AID_L
    “hi_prevha_ystart_l”: 1988, small int, Q_DRMED.HI_PREVHA_YSTART_L
    “previous_aid_descl”:”text”, char(240), Q_DRMED.PREVIOUS_AID_DESCL
    “previous_aid”: 0-1, small int, Q_DRMED.PREVIOUS_AID
    “hi_prevha_ystart_r”: 1988, small int, Q_DRMED.HI_PREVHA_YSTART_R
    “previous_aid_desc”:”text”, char(240), Q_DRMED.PREVIOUS_AID_DESC


    “hearing_prob”: X, small int, HEARING_PROB
    “onset_l”: {Null, From birth, Gradual, Sudden, N/A}, CHAR(20), Q_DRMED.ONSET_L
    “onset”: {Null, From birth, Gradual, Sudden, N/A}, CHAR(20), Q_DRMED.ONSET
    “noise_exposure”: {Null, Leisure, Gunfire, Occupational, N/A}, CHAR(20), NOISE_EXPOSURE
    “noise_expo_desc”: “text”, CHAR(250), NOISE_EXPO_DESC
    “ototoxic_drug”: {Null, Quinine, Diuretics, Aminoglycoside antibiotics, N/A}, CHAR(30), OTOTOXIC_DRUG
    “ototoxic_drug_desc”: “text”, CHAR(200), OTOTOXIC_DRUG_DESCZ


    “hi_degree_currhl_l”: {Null, Mild, Moderate, Severe, Profound}, CHAR(20), HI_DEGREE_CURRHL_L
    “hi_degree_currhl_r”: {Null, Mild, Moderate, Severe, Profound}, CHAR(20), HI_DEGREE_CURRHL_R
    “hi_better_ear”: {Null, Left, Right, N/A}, CHAR(10), HI_BETTER_EAR


    “hi_onset_ystart_l”: “year!”, small int, HI_ONSET_YSTART_L
    “hi_onset_couse_l”: {Null, Congenital, Acquired, Unknown}, varchar(20), HI_ONSET_COUSE_L
    “hi_onset_ystart_r”: “year”, small int, HI_ONSET_YSTART_R
    “hi_onset_couse_r”: {Null, Congenital, Acquired, Unknown}, varchar(20), HI_ONSET_COUSE_R

    “hearing_desc”: “text”, CHAR(250), HEARING_DESC //not used
```

Update Audiogram info for existing pseudoID

PUT $hostname. "/rest/api/patients/audiogram/update/{userid}/{patient_id}/

Updates the audiogram record for a pseudoID. An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

```
{
     “id”:
     “patient_id”: varchar(50) not null,
    “create_date”: timestamp (created by web gui)
   "civilstatus": EXT0000232, Critical: pseudoID
   //"user_id": 3003, for future version


   ...All data as previous
}
   "timestamp": 813157200000, longint is generated by the system
```

View all Audiogram info(s) (all records) for a patient

GET $hostname. "/rest/api/patients/audiogram/viewall/{userid}/{patient_id}/

An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

```
[
{
   “id” : int,
   “patient_id”: EHS system ID
   "civilstatus": EXT0000232,
   "user_id": 3003,
   “investdate”: timestamp
   “sys_createuser”: 1 (constrain)
   ...rest of data
}
]
```

View a specific Audiogram info (all records)

GET $hostname. "/rest/api/patients/audiogram/view/{userid}/{sys_createstamp}/

sys_createstamp= longint
An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

```
{
   “id” : int,
   “patient_id”: EHS system ID
   "civilstatus": EXT0000232,
   "user_id": 3003,
   “investdate”: timestamp


   ...rest of data
}
```

Delete Audiogram info for existing patient

POST $hostname. "rest/api/patients/audiogram/delete/

```
{
“id”:1
"patient_id":"EHS1",
"sys_createstamp":1524227754630
}
```

Audiogram Data

Create Audiogram data for existing pseudoID

POST $hostname. "/rest/api/patients/audiogramdata/create/{userid}/{patient_id}/{organizarion_id}/

Creates a Audiogram for a pseudoID by sending an XML file. An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3}).

```
{
   “file”: Path to xml file
}
```

System should generate an id for every newly created record, as well as a Created_time timestamp.

System automatically inserts → audindex = 2 (organizationid) and guid = 1

On success, returns 200.

View all Audiograms for a patient

GET $hostname. "/rest/api/patients/audiogramdata/viewall/{userid}/{patient_id}/

An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

```
[
{
   “patient_id”: int,
   “firstname”
   “lastname”:
   “userid”: 203,
   “created_time”: longint
   “curve”: xx
   “investdate”: longint 2017-12-22!!!
     Guid = 1 (constrain)
}
]
```

View Audiogram data for existing pseudoID

GET $hostname . "/rest/api/patients/audiogramdata/view/{userid}/{patient_id}/{Curve}/{created_time}/

An end-user (userid) with proper rights may trigger this function (e.g., "role_id" = {2, 3})

```
{
  “patient_id”: int,
   “Firstname”
   “lastname”:
   “userid”: 203,
   “created_time”: longint
   “curve”: xx
   “investdate”: timestamp
   Rest of Data
}
```

Delete Audiogram data

DELETE $hostname. "/rest/api/patients/audiogramdata/delete/{userid}/{id}

200

Big Data Analytics Engine API [1]

id AddWorkflow(EDAW, params)

AddWorkflow	It permits to add an EDAW into the Workflow Catalogue.	
Input parameters		
Name	**Type**	**Description**
EDAW	Structured type.	It is the Workflow Catalogue entry including executable/orchestrated workflow as provided by the PHPDM Transformation tool, scheduling preferences, formal parameters, etc.
Params	List of Structured types representing each Tasks' actual parameters.	It contails all the requested parameters for each Task in the EDAW. The EDAW is structured to take parameters at Task level.
Output parameters		
Name	**Type**	**Description**
Id	Numeric	It represents the id of the loaded EDAW as stored in the Workflow Catalogue

<list of EDAWs> FindWorkflowModel(DAWID)

FindWorkflowModel	Finde all the EDAW referred to a given PHPDM Model Instance Data Analytic Workflow (DAWID) from the Catalogue.	
Pre-conditions	DAWID must refers to an existing Model Instance in the Ontology Manager.	
Input parameters		
Name	**Type**	**Description**
DAWID	String	Unique identifier for a DAW Model Instance
Output parameters		
Name	**Type**	**Description**
list of EDAWs	List of Structured type	The list of EDAWS that refer to a specific DAWID in the Catalogue

EDAW FindWorkflowInstance(EDAWID)

FindWorkflowInstance	Find a given EDAW from the Catalogue.	
Pre-conditions	EDAWID must refers to a loaded EDAW.	
Input parameters		
Name	**Type**	**Description**

EDAWID	numeric	Unique identifier for a loaded EDAW
Output parameters		
Name	**Type**	**Description**
EDAW	Structured type	The EDAW as it is in the Catalogue

ret ModifyWorkflow (EDAW,params)

ModifyWorkflow	It permits to modify a given EDAW, like for instance updating the formal parameters, the schedule preference , the EWAD code, of a loaded EDAW. It allows to modify also the actual parameters.	
Pre-conditions	The EDAW must be already loaded into the Catalogue. If the EDAW is running, it is first stopped, modified and then re-executed.	
Post-conditions	Ret corresponds to a valid output message	
Input parameters		
Name	**Type**	**Description**

EDAW	Structured type.	It is the Workflow Catalogue entry including executable/orchestrated workflow as provided by the PHPDM Transformation tool, scheduling preferences, formal parameters, etc.
Params	List of Structured types representing each Tasks' actual parameters	It contails all the requested parameters for each Task in the EDAW. The EDAW is structured to take parameters at Task level.
Output parameters		
Name	**Type**	**Description**
Ret	boolean	Positive in case of success, negative in case of error (e.g., EDAW is not available in the Catalogue)

ret RemoveWorkflowModel(DAWID)

RemoveWorkflowModel	Remove all the EDAW referred to a given PHPDM Model Instance Data Analytic Workflow (DAWID) from the Catalogue.	
Pre-conditions	DAWID must refers to an existing Model Instance in the Ontology Manager.	
Post-conditions	Ret correspond to a valid output message.	
Input parameters		
Name	**Type**	**Description**
DAWID	String	Unique identifier for a DAW Model Instance
Output parameters		
Name	**Type**	**Description**
Ret	boolean	Negative if the given DAWID is not present in the Catalogue, positive otherwise. boolean

ret RemoveWorkflowInstance(EDAWID)

RemoveWorkflowInstance	Remove a given EDAW from the Catalogue.	

Pre-conditions	EDAWID must refers to a loaded EDAW.	
Post-conditions	Ret correspond to a valid output message.	
Input parameters		
Name	**Type**	**Description**
EDAWID	numeric	Unique identifier for a loaded EDAW
Output parameters		

Name	**Type**	**Description**
Ret	boolean	Negative if the given EDAWID is not present in the Catalogue, positive otherwise. boolean

ExecuteWorkflow(EDAWID)

ExecuteWorkflow	Execute a loaded EDAW given its EDAWID according to the scheduler preference. It can be also used to force re-execution of an EDAW coherently with the schedule preference.	

Pre-conditions	EDAWID must refers to a loaded EDAW	
Post-conditions	The Job id given by the Workflow Scheduler is saved into the Workflow Catalogue entry related to EDAWID to manage the running/scheduled EDAW.	
Input parameters		
Name	**Type**	**Description**
EDAWID	numeric	Unique identifier for a loaded EDAW

ret StopWorkflow(EDAWID)

StopWorkflow	Stop the execution of an EDAW and perform the relative rollback activities if required.	
Pre-conditions	EDAWID must refers to an EDAW under execution	
Post-conditions	Ref correspond to a valid output message	
Input parameters		
Name	**Type**	**Description**
EDAWID	numeric	Unique identifier for loaded EDAW

Output parameters		
Name	**Type**	**Description**
Ret	boolean	Positive if the EDAWID can be stopped, negative otherwise.

id AddTask (ImplDAWTask)

AddTask	It allows to add an implementation of a given DAW Task (ImplDAWTask) into the Task Catalogue.	
Pre-conditions	This API is for internal use only and it requires BDA Administration authorizations to be executed.	
Input parameters		
Name	**Type**	**Description**
ImplDAWTask	A structured Type	It represents an entry of the Task Catalogue
Output parameters		

Name	Type	Description

Id	numeric	It represent the id of the Implemented Task as it is inserted into the Task Catalogue

Ret ModifyTask (ImplDAWTask)

ModifyTask	It allows to modify any of the Task related attributes, like pointing to different implementation, modifying the reference library to name but a few.	
Pre-conditions	This API is for internal use only and it requires BDA Administration authorizations to be executed. ImplDAWTask must refers to an Implemented Task available in the Task Catalogue.	
Input parameters		
Name	**Type**	**Description**
ImplDAWTask	Structured Type	It represents the entry of the Task Catalogue for which we want to modify some of its attributes
Output parameters		

Name	Type	Description
Ret	boolean	Positive in case of successful modification, negative in case of error (e.g., EDAW not available in the Catalogue)

Ret RemoveTaskModel (DAWTaskID)

RemoveTaskModel	It allows to remove all the implementation of a given a DAW task model	
Pre-conditions	This API is for internal use only and it requires BDA Administration authorizations to be executed. DAWTaskID must refer to a DAW Task present in the EVOTION Ontology Manager	
Input parameters		
Name	**Type**	**Description**
DAWTaskID	String representing a DAW Task ID in the Ontology Manager.	It is an unique identifier referring to a Class in the Ontology of the PHPDM language.
Output parameters		

Name	Type	Description
Ret	boolean	Negative if the given DAWTaskID is not present

RemoveTaskInstance (ImplDAWTaskID)

RemoveTaskInstance	It allows to a specific implementation of a given Task.	
Pre-conditions	This API is for internal use only and it requires BDA Administration authorizations to be executed. ImplDAWTaskID must refer to an Implemented Task available in the Task Catalogue.	
Input parameters		
Name	**Type**	**Description**
ImplDAWTaskID	Numeric. It represents an Implemented DAW Task. It is optional.	It represents the entry of the Task Catalogue. If present the deletion refers just to this ID otherwise to all the implemented Task related to the DAWTaskID
Output parameters		

Name	**Type**	**Description**
Ret	boolean	Negative if the given ImplDAWTaskID is not present in the Catalogue, positive otherwise

<ImplementedTasks> FindTask(DAWTaskID)

FindTask	It permits to search for an implementation of a given DAW Task ID. It returns the details needed to generate an EDAW involving the retrieved Implemented Task.	
Input parameters		
Name	**Type**	**Description**
DAWTaskID	A String representing a DAW Task ID in the Ontology Manager.	It is an unique identifier referring to a Class in the Ontology of the PHPDM language.
Output parameters		
Name	**Type**	**Description**

ImplementedTasks	List of Implemented Tasks as they are specified in the Task Catalogue (i.e., the list of corresponding Catalogue Entries)	It contains the Implemented Analytic Tasks' details relative to a given DAW Task ID.

JobID AddEDAWJob (EDAWID)

AddEDAWJob	It adds an EDAW to the schedule according to the preference.	
Pre-conditions	This API is for internal use only and it requires BDA Administration authorizations to be executed. EDAWID must refer to an EDAW available in the Workflow Catalogue.	
Input parameters		
Name	**Type**	**Description**
EDAWID	Numeric Type representing an EDAW	It represents the entry of the Workflow Catalogue for which we want set up a schedule.
Output parameters		
Name	**Type**	**Description**

JobID	Job identifier	It represents the identifier related to the EDAW in execution as it is represented by the Execution Manager/Scheduler

RemoveEDAWJob (EDAWID)

RemoveEDAWJob	It removes an EDAW from the scheduler. If the EDAW is in running, it stops the EDAW job before the removal.	
Pre-conditions	This API is for internal use only and it requires BDA Administration authorizations to be executed. EDAWID must refer to an EDAW available in the Workflow Catalogue.	
Input parameters		
Name	**Type**	**Description**
EDAWID	Numeric Type representing an EDAW in the Workflow Catalogue	It represents the entry of the Workflow Catalogue for which we want to modify a schedule.

<EDAWID> ListRunningEDAWJob

ListRunningEDAW	It lists all the EDAW that are running.	

Pre-conditions	This API is for internal use only and it requires BDA Administration authorizations to be executed.	
Output parameters		
Name	**Type**	**Description**
<EDAWID>	List of EDAWID	It contains the list of all the EDAWID that are running

StopRunningEDAWJob(EDAWID)

ListRunningEDAW	It stops specific EDAWID.	
Pre-conditions	This API is for internal use only and it requires BDA Administration authorizations to be executed. EDAWID must refer to a running EDAW.	
Input parameters		
Name	**Type**	**Description**
EDAWID	Numeric Type representing an EDAW in the Workflow Catalogue	It represents the entry of the Workflow Catalogue wich is under execution and for which we want to sto

Security API

Operation Description	**SecurityEngine.Login**	
Rest API URL	$hostname:443 + "/security/login" (Content Type: application/json)	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"username": "XXXXX",
"password" : " xxxxx "
}
	Application/json	In order to login you should send a post request with json body containing user and password fields

Output parameters		
Name	**Type**	**Description**
200	HTTP Code	Successful Response
Body	**Type**	**Description**

{
"token":" token "
}
	Application/json	Token to be used for functions that need authorization
		Note: token expires in 6 hours

Operation Description	**SecurityEngine.Validate**	
Rest API URL	$hostname:443 + "/security/validate" (Content Type: application/json)	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"token":"XXXXXXX…XXXXX"
}
	Application/json	In order to validate a token you should send a post request with json body containing the token.

Output parameters		
Name	**Type**	**Description**
200	HTTP Code	Successful Response
Body	**Type**	**Description**

{
"user" : "XXXXXXX",
“role” : “configurator”
}
	Application/json	json body contains the user that holds the valid token

Operation Description	**SecurityEngine.Logout**
Rest API URL	$hostname:443 + "/security/logout” (Content Type: application/json)

Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"token":" XXXXXXX…XXXXX"
}
	Application/json	In order to logout and invalidate a token you should send a post request with json body containing the token.

Output parameters		
Name	**Type**	**Description**
200	HTTP Code	Successful Response with text/plain message: "User *USER* logged out successfully!" *Note: Each user has a role. Each asset has two permissions (read/write) for the moment. These two permissions hold a list of the available roles. For more information refer to the LDAP tables below.

Operation Description	**SecurityEngine.Authorize**	
Rest API URL	$hostname:443 + "/security/authorize" (Content Type: application/json)	
Rest API Type	POST	
Input parameters		
Body	**Type**	**Description**

{
"token":" XXXXXXX…XXXXX"
"permission": "read",
"asset": "dashboard"
}
	Application/json	In order to ask if a user is authorized to use an asset you should send a post request with json body containing the token that the user holds, the permission of that defines the action the user wants to take and the specific asset.

Output parameters		
Name	**Type**	**Description**
200	HTTP Code	Successful Response with text/plain message: “Successful Authorization! User: *USER* is authorized.“

Evotion LDAP database (OpenLDAP)

In order to connect to LDAP database and add/delete/modify users and tables:
dn suffix: ‘dc=evotion,dc=com’
rootdn: ‘cn=admin,dc=evotion,dc=com’
rootpw:XXXX

Table 4: Tables of Evotion LDAP database

Configurators table	stores all the configurators of the mobiles
dn: ou=configurators,dc=evotion,dc=com	

Equipment table	stores all the registered mobile devices
dn: ou=equipment,dc=evotion,dc=com	

CityHospital table	stores all the clinicians of City Hospital

dn: ou= cityHospital,dc=evotion,dc=com

AthensHospital table	stores all the clinicians of Athens Hospital
dn: ou= athensHospital,dc=evotion,dc=com	

Assets table	stores all the assets/components of Evotion Platform. Used for authorization. Each asset has allowed permissions (read/write) from certain roles
dn: ou= assets,dc=evotion,dc=com	

Below a dummy entry is included to indicate the needed struct of .ldif files in order to add new entries.

```
dn: uid=evotionAdminTech,ou=configurators,dc=evotion,dc=com
objectclass: configurator
cn: Evotion Administrator Technician
uid: dummyusername
userpassword: dummypassword
role: admin
description: Administrator Technician of Evotion LDAP
```