

The logo for InteropEHRate features a series of colorful dots in shades of purple, blue, yellow, orange, pink, and light blue, arranged in a pattern that suggests a stylized 'I' or a cluster of data points.

InteropEHRate

D5.5

Design of an integrated EHR web app for HCP – V2

ABSTRACT

This deliverable presents the relevant increments and findings concerning the design of the Healthcare Professional Application (HCP Web App) used by healthcare professionals for creating and accessing health data of foreign patients. The preponderance of its content is derived from the deliverable [22] D5.4 – Design of an integrated EHR web app for HCP – V1 and reflects, at the second year stage of project implementation, the most important design updates addressing the import / export data directly from/to the S-EHR on the smartphone.

Delivery Date	January, 14 th 2021
Work Package	WP5
Task	T5.1
Dissemination Level	Public
Type of Deliverable	Report
Lead partner	SIMAVI



This document has been produced in the context of the InteropEHRate Project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106. All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.



This work by Parties of the InteropEHRate Consortium is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

DRAFT

CONTRIBUTORS

	Name	Partner
Contributors	Andrei Madalin Oanca, Nicu Jalba, Simona Alexandru, Adrian Bradu, Mihai Seceleanu, Simona Bica	SIMAVI
Contributors	Gabor Bella	UNITN
Reviewers	Sofianna Menesidou	UBIT
Reviewers	Radu Retezeanu	SCUBA

DRAFT

LOG TABLE

Version	Date	Change	Author	Partner
0.1	18-08-20	First draft of table of contents	Andrei Madalin Oanca	SIMAVI
0.2	22-08-20	Contribution to Chapter 1.4 Contribution to Chapter 2 Contribution to Chapter 5	Andrei Madalin Oanca	SIMAVI
0.3	14-10-20	Internal review and updates of the deliverable	Simona Bica	SIMAVI
0.4	27-10-20	Review and addition of latest design information	Simona Alexandru	SIMAVI
0.5	27-10-20	Internal Review	Adrian Bradu	SIMAVI
0.6	03-11-20	Review and send for contributions	Adrian Bradu	SIMAVI
0.7	26-11-20	Changes and additions	Gabor Bella	UNITN
0.8	01-12-20	First internal review	Sofianna Menesidou	UBIT
0.7	03.12-20	Corrections and completion after first review	Simona Alexandru	SIMAVI
0.8	03-12-20	Revisions and submission for second internal review	Adrian Bradu	SIMAVI
0.8	09-12-20	Second internal review	Radu Retezeanu	SCUBA
0.9	10-12-20	Updated and sent to QA check	Adrian Bradu	SIMAVI
1.0	14-12-20	Quality check	Argyro Mavrogiorgou	UPRC
1.1	15-12-20	Revised after quality check	Mihai Seceleanu	SIMAVI
1.2	04-01-21	Revised and sent for submission	Adrian Bradu	SIMAVI
Vfinal	13-01-21	Final check and submission	Francesco Torelli Laura Pucci	ENG

ACRONYMS

Acronym	Term and definition
AMDD	Agile Model Driven Development
API	Application Programming Interface
CSS	Cascading Style Sheets
D[N].[N]	Deliverable document referred into the text where [N] is a number. In the first position the number represents the work package number and in the second position is the ordinal number of the deliverable inside the work package.
D2D	Device to Device protocol
DAO	Data Access Object
DI	Dependency Injection
DICOM	Digital Imaging and Communications in Medicine (standard for the communication and management of medical imaging information and related data)
EHR	Electronic Health Record
GUI	Graphical User Interface
HCP	HealthCare Professional
HIS	Hospital Information System
HR	Health Record
HTML	Hypertext Mark-up Language
ICD-10	International Classification of Diseases, Tenth Revision
ICD-9	International Classification of Diseases, Ninth Revision
ICT	Information and Communication Technologies
IEHR	InteropEHRate
IoC	Inversion of Control
IPS	International Patient Summary
JDBC	Java DataBase Connector
JDK	Java Development Kit
M9	Month 9
MVC	Model-View-Controller
PACS	Picture Archiving and Communication System
PHR	Personal Health Record
PID	Personal Identification
QR code	Quick Response code
R&D&I	Research, Development and Innovation
RIS	Radiological Information System
S-EHR	Smart HER
UCD	User Centred Design
UI	User Interface
W3C	World Wide Web Consortium

WP	Work Package
XML	eXtensible Markup Language

DRAFT

TABLE OF CONTENT

1.	INTRODUCTION	1
1.1.	Scope of the document	1
1.2.	Intended audience	1
1.3.	Structure of the document	1
1.4.	Updates with respect to previous version (if any)	2
2.	CONTEXT	4
2.1.	Goals	4
2.2.	Relation with other project deliverables and project results	5
3.	METHODOLOGICAL APPROACH	6
4.	DESIGN OF THE HCP WEB APP TECHNICAL SOLUTION	7
4.1.	Architectural approach	7
4.2.	Technical approach	11
4.2.1	Coding conventions	12
4.2.2	Package diagram	12
4.2.3	Entity diagram	14
4.3.	Interoperability approach	16
4.3.1.	Interoperable and multilingual information within health data	17
4.3.2.	Reading and writing health records	19
4.3.3.	Display of health records	21
4.4.	Display medical images	22
4.5.	Emergency scenario - integration with S-EHR Cloud	23
5.	DESIGNING THE GUI OF HCP WEB APP	26
5.1.	End-user viewpoints and perspectives	26
5.2	Mockups of HCP Web App. Second iteration for updating and improving the UI design	27
5.3.	Current stage of UI implementation	34
6.	Automated testing	57
6.1.	Unit testing - Continuous Integration with Gitlab Runner, Junit and Maven	57
6.2.	User interactions scripts with Selenium	60
7.	CONCLUSIONS AND NEXT STEPS	61

LIST OF FIGURES

Figure 1: - Component Diagram	8
Figure 2: Dependency tree diagram of the HCP Web App components	10
Figure 3: HCP Web App - Main packages and their relations	13
Figure 4: HCP Web App - Main diagram of the entities involved	16
Figure 5: Sequence diagram for underlying InteropEHRate Health Services - HCP Interactions	20
Figure 6: Electrocardiogram in Weasis	22
Figure 7: Skull - magnetic resonance image in Weasis	23
Figure 8: Home page with 2 options available for the HCP	23
Figure 9: Emergency Service interface	25
Figure 10: - Emergency Service implementation	25
Figure 11: IPS - get, decrypt, parse	25
Figure 12: New main menu	29
Figure 13: Demographic information about patient	30
Figure 14: Current Diseases	30
Figure 15: Pathology history	31
Figure 16: Allergies	31
Figure 17: Current Medications	32
Figure 18: Document Hstory Consultation	32
Figure19 : Laboratory Tests	33
Figure20: Vital Signs	33
Figure21 : Conclusion	34
Figure 22: Home page: D2D connection QR code for Medical Visit and option to switch to Emergency	35
Figure 23: Emergency Scenario selected	36
Figure24:- Emergency Scenario: Input page for the QR code information	36
Figure 25: Active Emergency scenario and general info about the patient	37
Figure 26: Patient General Data Page with the Certificate details and the button Invalid Digital Identity	38
Figure 27:-Error message	39
Figure 28: CurentPatient - General Data Page	39
Figure 29: INTERNATIONALIZATION - Italian labels selected	40
Figure 30: Current Diseases	40
Figure 31: Pat History	41
Figure 32: Allergies View Page	41
Figure 33: Allergies Add page	42
Figure 34: Prescription screen	42
Figure 35: Current Medications	43
Figure 36: Add page for Prescription	44
Figure 37: Added Prescription in a separated table	44
Figure 38: Uploaded Prescription to S-EHR	45
Figure 39: Document History Consultation	45
Figure 40: Laboratory Results, Observation blood tests	46
Figure 41: Laboratory Results, X-ray Image Report	47
Figure 42: MRI (Magnetic Resonance Imaging) in Weasis application	48
Figure 43: ECG result in Weasis application	48
Figure 44: Vital Signs	49
Figure 45: Add Vital Signs	49
Figure 46: Add Vital Signs Information details	50
Figure 47: Outpatient report screen with no data to synchronize	50

Figure 48: Outpatient report screen with data to synchronize but with no connection to the S-EHR mobile app 51

Figure 49: Outpatient report screen with data to synchronize..... 52

Figure 50 : Practitioner details and health care organisation 52

Figure 51: Upload Organization Certificate 53

Figure 52: Practitioner details 53

Figure 53: Upload Practitioner Certificate..... 53

Figure 54: Nomenclature administration 54

Figure 55: Group of buttons for translation and conversion services (Original selected) 55

Figure 56: Current Diseases screen with original (selected) content received 56

Figure 57: Current Diseases screen with translated (selected) and converted content received 56

Figure 58: HCP app Runners 58

Figure 59: YML configuration for HCP app 59

Figure 60: Build lifecycle phases..... 59

Figure 61: Gitlab Runner output console 60

DRAFT

1. INTRODUCTION

1.1. Scope of the document

The deliverable is the second report, produced by the InteropEHRate project within the activities related to *Incremental EHRs integration*, which covers the design requirements of HCP Web App solution, specifically the design requirements corresponding to second year of project implementation.

The purpose of this deliverable is to present the design of HCP Web App solution, based on the results obtained within the previous deliverable [22] [D5.2] Software requirements specification of an integrated EHR web app for HCP - V2. At this stage of project implementation, the deliverable aims to depict the major features and principles of designing the HCP Web App solution, addressing essentially the import/export data directly from/to the S-EHR App on the smartphone and the retrieval of emergency information from the S-EHR Cloud.

1.2. Intended audience

The document is intended to different categories of professionals, such as:

- Technical staff: developers, consultants, analysts, web designers, interested to have an overview about the specific design of HCP Web App;
- Healthcare providers interested in how to use an application like HCP Web App from the perspective of end-users.

Both categories could be interested in participating in co-design sessions during each development cycle, in order to improve and enrich the solution capabilities.

1.3. Structure of the document

The deliverable is structured in six chapters, as follows:

Section 1. Introduction: Presents a summary concerning the purpose and objectives of the deliverable, its structure and relation to other tasks and deliverables.

Section 2. Context: Presents a description of relevant characteristics of this particular stage of the project implementation, addressing also the relation to other project deliverables and project results.

Section 3. Methodological approach: Presents the significant aspects regarding the appropriate methodology (Agile) and principles (UCD) applied in this stage of designing the HCP Web App solution.

Section 4. Design of the HCP Web App technical solution: Presents the major aspects concerning the pertinent architectural approach and technical approach for this stage of designing the HCP Web App solution. Relevant aspects concerning the interoperability approach and current findings specific to implement HCP Web App are also presented.

Section 5. Designing the GUI of HCP Web App: Presents the significant aspects regarding the design (concept, principles, conventions, “look and feel” approach) and the specific requirements of the user interfaces.

Section 6. Testing the HCP Web App: Presents the two approaches regarding the automated testing of the application functionalities:

- Unit and integration testing
- By simulating User interaction.

Section 7. Conclusions and next steps: Presents the conclusions and next steps concerning this particular stage of implementing HCP Web App.

1.4. Updates with respect to previous version (if any)

The current deliverable is the second out of three ([22] [D5.4], [D5.5] - the current deliverable and [D5.6]) which is dedicated to the design of HCP Web App. This deliverable depicts the increments that have emerged from the first deliverable, [22] [D5.4].

The main additions to the HCP Web App in this stage are the redefined design elements and the changes that have emerged at the User Interface (UI) level coming from the co-design sessions. The elements presented in this deliverable will be updated in the last deliverable on this topic, [D5.6] - Design of an integrated EHR web app for HCP - V3.

Below is a summary of the features designed in the version 2 of the application. These functionalities are detailed in the following chapters:

- New medical information categories downloaded from S-EHR Mobile App through D2D connection:
 - Vital Signs:
 - View
 - Add/Edit/Delete in local database
 - Upload to S-EHR Mobile App
 - Pathology History:
 - View
 - Laboratory Results:
 - View (including limits and abnormalities)
 - Current Diseases:
 - View
 - Document history

- View
- Prescriptions:
 - View (with pagination, sort and filter functionalities)
 - Add/Edit/Delete in local database
 - Upload to S-EHR Mobile App
- Emergency scenario
 - Establish connection with S-EHR Cloud
 - Interrogate S-EHR Cloud API for downloading patient's IPS
 - Decrypt, extract and display the information from IPS
- Security:
 - Displaying the patient's digital identity (from his digital certificate)
 - Possibility to close the D2D connection if the digital identity does not match
- Display Image Report
- Display DICOM images
- Translation and conversion
 - Machine translation of the medical content
 - Concept translation of the medical content
 - I18N (internationalization) of UI
 - Code conversion (ICD9->ICD10, ICD10->ICD9)
- Reorganizing the application main menu (according to co-design sessions).

2. CONTEXT

2.1. Goals

Within the project implementation, the present deliverable depicts the most relevant updates concerning the design of HCP Web App, which have been brought in the second year of implementation. The deliverable is based on the valued results from the project tasks in charge to define the user requirements (scenarios and user requirements presented in deliverables [15] [D2.1] and [16] [D2.2]) and support from the healthcare partners of the consortium in proposing the most significant *data categories, data sets of IPS and standard terminology* to implement in this stage. Significant input from deliverables [19] [D4.4] Design of libraries for remote and D2D HR exchange - V1 and [D4.5] Design of libraries for remote and D2D HR exchange - V2 concerning the innovative way of designing the data exchange (design of libraries for D2D HR exchange) was also considered in drafting this deliverable.

The base of this document is the deliverable [22] [D5.4] Design of an integrated EHR web app for HCP -V1 which contains the preliminary decisions regarding the design of HCP Web App. This deliverable comes as a continuation of the considerations presented in deliverable [21] [D5.2] Software requirements specification of an integrated EHR web app for HCP - V2, and it aims to provide *personalized and customizable information to end-users, based on the user-centric approach*. Representative aspects and details of this approach were outlined hereinafter, in *Section 5. Designing the GUI of HCP Web App*.

As in the last deliverable, HCP Web App is depicted from four significant perspectives with the visibly improvements that came with the second year of development:

- Methodological perspective (addressed in Section 3 METHODOLOGICAL APPROACH)
- Architectural perspective (addressed in Section 4 DESIGN OF THE HCP WEB APP TECHNICAL SOLUTION)
- Technical perspective (addressed in Section 4 DESIGN OF THE HCP WEB APP TECHNICAL SOLUTION)
- End User (i.e. healthcare professionals) perspective (addressed in Section 5 DESIGNING THE GUI OF HCP WEB APP).

The improvements to the conceptual level specifications for the design of the HCP Web App solution, including the compatibility requirements with HL7 FHIR profile (data sets, terminologies, concepts, domain model, etc...), were thoroughly analyzed and considered on the scientific approach from the deliverable [18] [D2.8] FHIR profile for EHR interoperability - V2.

The GUI design requirements and the current updates at the UI level are based on the Scenarios and User Requirements presented in deliverable [16] [D2.2] User Requirements for cross-border HR integration - V2, as well as on the findings resulted from the co-design sessions and co-creation iterations with the healthcare partners. Within the deliverable, these specific requirements are illustrated in Section 5 through representative mock-ups and print screens from the HCP Web App.

2.2. Relation with other project deliverables and project results

The *Validation of results* is connected with the actual Task 5.1 and the other tasks of WP5 because the healthcare partners of the project consortium will be involved in WP7 to explore and exploit the functionalities of the InteropEHRate innovative platform.

DRAFT

3. METHODOLOGICAL APPROACH

This section refers to the relevant aspects of the methodological approach corresponding to stage of designing the HCP Web App solution, comprising Agile methodology and UCD (User-Centred Design) specific for the software development.

As presented in the previous deliverable [22] [D5.4], Agile and UCD perfectly suit the features and objectives of the InteropEHRate project. In this stage of updating and improving the design of HCP Web App, the Agile Model Driven Development (AMDD) was further applied. In this stage of HCP Web App design, the AMDD is applied as an iterative and incremental process in order to draft *the design specifications and particularities* of the application, as they are presented hereinafter in *Section 4. Design of the HCP Web App technical solution*.

The UCD methodology was also used in this stage of designing the HCP Web App solution, allowing to build the application in a incremental way by adding new features and functionality over several iterations, based on the valuable feedback of end-users. Considering the specific of our target group, we optimised the design of the user interface *from the perspective of healthcare professionals*, based on the feedback gathered from the end users within iterative co-design / co-creation sessions. Details reflecting the iterative approach of GUI design are presented hereinafter in *Section 5. Designing the GUI of HCP Web App*.

The HCP Web App relies on the *InteropEHRate Conversion and Translation Services* (described in deliverables D5.9 [23] and D5.11 [25], respectively) in order to extract the meanings of coded values and terminology from text entered by the HCP, as well as to translate natural language text inside health data to the language of the HCP and/or the patient. As these services are being specified and implemented in parallel to work on the HCP Web App, the development methodology of the interoperability-related business logic implemented by the HCP Web App needs to proceed in an iterative manner. The specifications for dealing with these services and with multiple (original, converted, translated) versions of the same data are presented in detail in **Chapter 4.3** below.

4. DESIGN OF THE HCP WEB APP TECHNICAL SOLUTION

4.1. Architectural approach

The main architectural approach of the HCP Web App was developed in such a way to satisfy all the main requirements of the IEHR project. This includes the exchange of information with the S-EHR mobile App, the display of different types of information, the connection with S-EHR Cloud, the addition and modification of the aforementioned information and other requirements. Of course, the HCP Web App provides a Graphical User Interface (GUI) that will ease the interaction between the Health Care Provider and the HCP Web App.

As it was presented in D5.1 [20], the HCP Web App uses Java technologies being developed using Spring framework; thus resulting in a broader range of compatibility of the software solution that will be developed for the IEHR project. In this deliverable, the following figures will explain the main components of the HCP Web App that are part of the solution.

The following Component Diagram contains the major architectural components highlighting how the application is constructed at the level of each layer. The code needed for developing the HCP Web App will be written around these components, respecting the rules and conventions imposed by them. Thus, the HCP Web App will be implemented using the best practices and well known patterns on which these components are based.

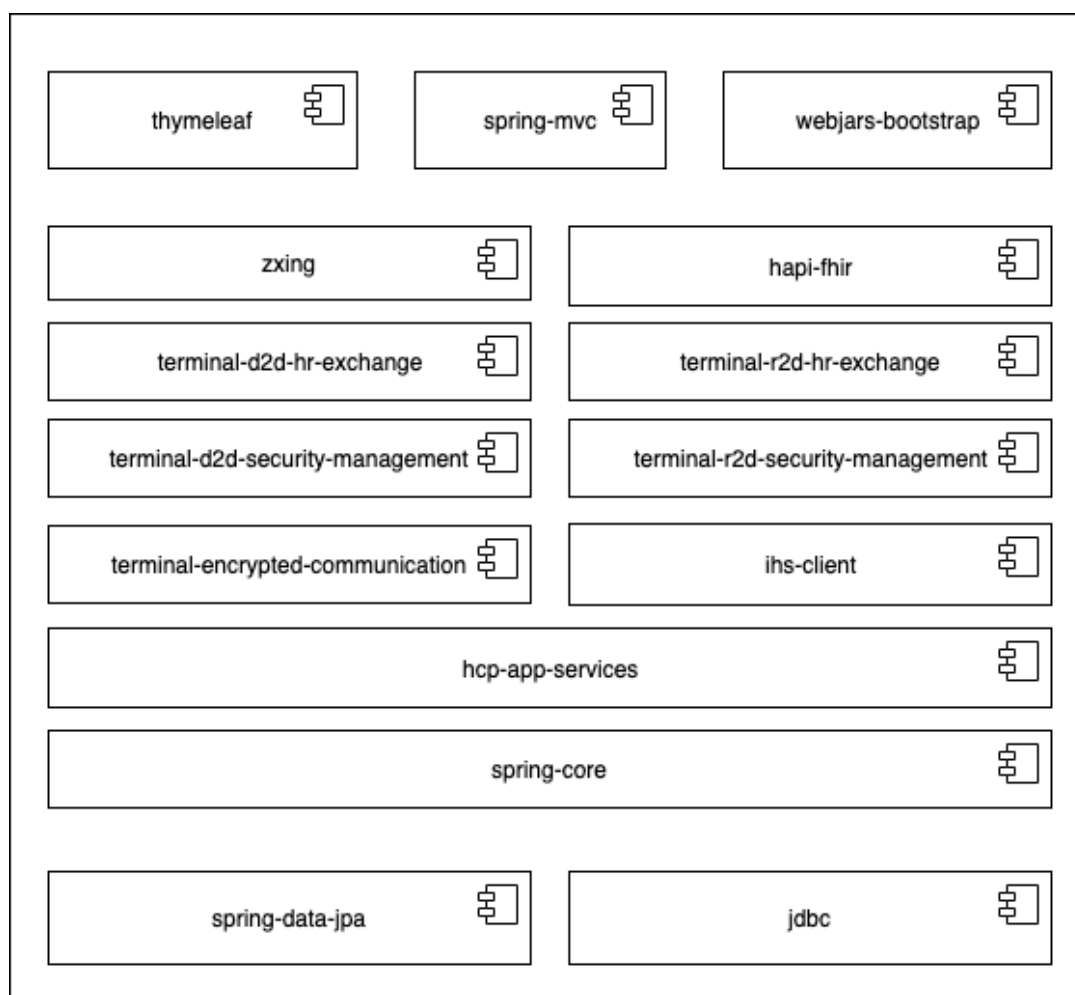


Figure 1: - Component Diagram

Spring-Data-JPA is a module of Spring framework that allows us developers to simply create JPA based repositories. These repositories are useful as they provide CRUD (Create, Read, Update, Delete) operations on the many entities that the project contains by speeding the traditional way of creating DAOs [10] for Object Relational Mapping.

The second component necessary for manipulating data stored in relational databases is **JDBC**. JDBC is necessary to connect to a Database directly and running SQL commands. Spring JDBC Framework is in charge of working on the low-level details starting from opening the connection, preparing and executing the SQL statement, processing exceptions, handling transactions, and finally closing the connection.

The **Spring-Core** is situated at the base of the rest of the components. Spring-Core is the foundation module on which the rest of the components rely on. It provides support for developing applications using Dependency Injection (DI) and Inversion of Control (IoC).

The **hcp-app-services** is the next component which covers a broader range of important class components for the HCP Web App. Some examples include BluetoothConnectionService, SEHRInitialDownloadService and others. This component will contain all the services needed in providing the functionality in agreement with the requirements of the HCP Web App.

The **IHS client** is the component through which the HCP Web App will work with IHS Services. IHS Services provides localization functionalities, being another application that composes the IEHR architecture.

The **Terminal-Encrypted-Communication** is one of the main security oriented components of the HCP Web App. The main functionality that this component will provide is the key agreement and encryption of the main information needed for creating the communication bridge between the S-EHR App and HCP Web App. This will provide both security and trustability of the shared information of the devices that are involved in creating a connection for sharing medical data. This component will be developed within the IEHR project.

Terminal d2d security management - implements the main security functionalities required by the D2D protocol. This component will act as a plug-in for the D2D library. This component will be developed within the IEHR project.

Terminal r2d security management - implements the main security functionalities required by the R2D protocol. This component will act as a plug-in for the R2D library. This component will be developed within the IEHR project.

Terminal d2d hr exchange focuses on data exchange between a citizen and involved healthcare professionals, and the S-EHR Mobile App and HCP Web App, respectively, during a patient visit. This component will be developed within the IEHR project.

Terminal r2d hr exchange enables data exchange in emergency situations. Therefore, the citizen is not actively involved in the transactions taking place between the components HCP Web App and S-EHR Cloud. This component will be developed within the IEHR project.

Hapi-fhir is an open-source implementation of the FHIR specification for java applications. This library will be used by all applications developed within the IEHR project that will work with FHIR resources.

Zxing is a library, implemented in Java, for barcode and QR code image processing. It will be used to generate the QR code containing the Bluetooth address of the terminal running the HCP Web App.

Spring MVC is another module of spring framework, designed in order to develop web applications using the well-known pattern Model-View-Controller (MVC).

Thymeleaf is a template engine for developing web and non-web applications, being mostly used for implementing the view part of the MVC based web applications. Spring Framework has a dedicated module for working with Thymeleaf.

Webjars-bootstrap. Bootstrap CSS is an open source framework for developing HTML, CSS and JS. This framework is included in the architecture of HCP Web App as a jar archive known in the community as WebJars. These Webjars are designed to work with most JVM containers and web frameworks.

The following picture illustrates the first level of dependency tree on which the HCP Web App is based. The components described above are included in this tree alongside the dependencies needed by these main components - transitive dependency [5] . Transitive dependencies will not be described in the deliverables.

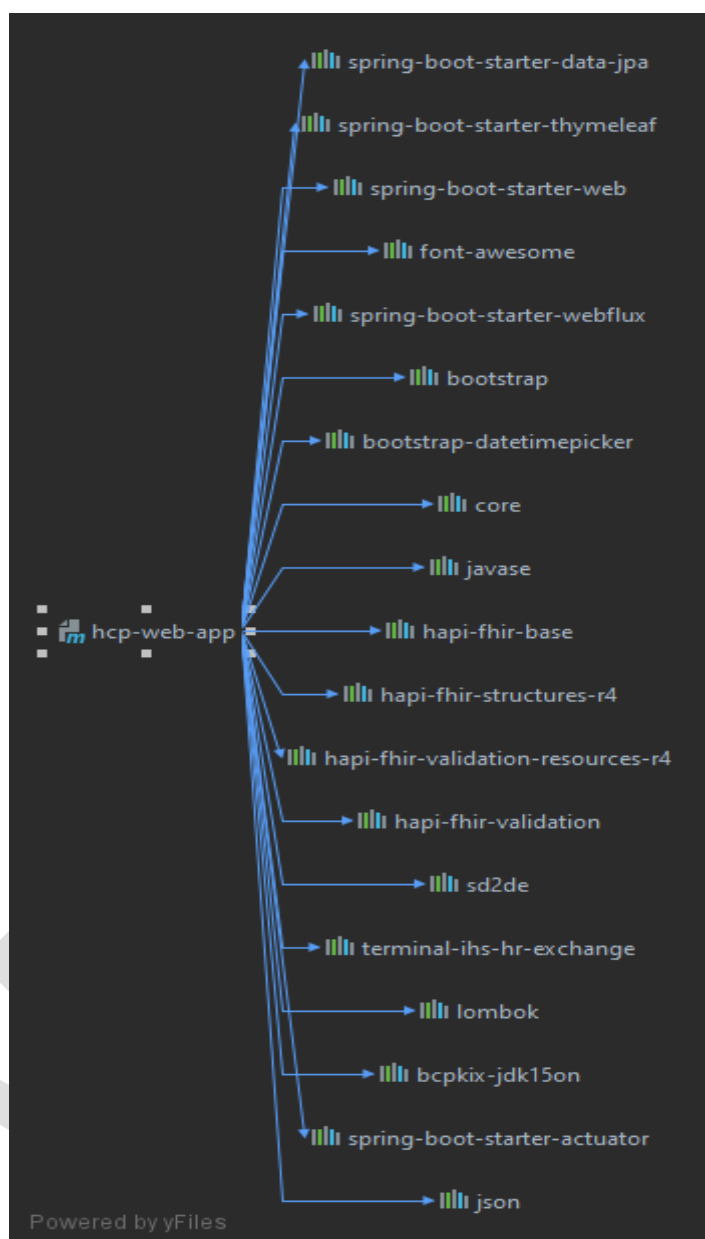


Figure 2: Dependency tree diagram of the HCP Web App components

Following can be found every dependency detailed:

- The first important dependency needed to run the HCP application is **Spring Boot Starter Data JPA**. This dependency is responsible for enabling the usage of Spring data JPA using the Hibernate framework. Hibernate ORM (or simply Hibernate) is an object-relational mapping tool for the Java programming language.
- **Spring Boot Starter Thymeleaf** dependency enables for HCP app the Thymeleaf engine template. It is suited for serving XHTML/HTML5 at the view layer of MVC-based web applications. Thymeleaf also provides full Spring Framework integration.
- **Spring Boot Starter Web** is the starter responsible for building web, including RESTful, applications using Spring MVC. It uses **Tomcat** as the default embedded container.

- On the HPC app, developers can use the **org.webjars Font Awesome** for displaying specific icons in the future.
- **Spring Boot Starter WebFlux** is the starter for building WebFlux applications using Spring Framework's Reactive Web support. WebFlux is useful for creating IO live streaming of data.
- **Bootstrap** dependency provides the functionality of using the Bootstrap framework to simplify the work on designing the Graphical User Interface.
- **Bootstrap DateTimePicker** dependency is a JWebSwing implementation that comes with pre designed interfaces for the user to choose dates.
- The **com.google.zxing Core** dependency contains the Core barcode encoding/decoding library.
- The **com.google.zxing JavaSE** dependency contains the Java SE-specific extensions to core ZXing library.
- The **Hapi Fhir Base** dependency contains HAPI FHIR Core Library.
- The **Hapi Fhir Structures R4**, **Hapi Fhir Validation Resources** and **Hapi Fhir Validation** dependencies are self explanatory.
- **SD2DE** dependency is the HCP side D2D exchange library needed for using the device to device Bluetooth connection. This is developed by UPRC partner. SD2DE includes the SD2DSM as a sub-component for ensuring the security management for D2D communications. The SD2DSM module is developed by UBIT partner.
- **Terminal IHS HR Exchange** is responsible with the Translation services and Cloud access. In this library the developers are UNITN and BYTE partners.
- The **Lombok** dependency enables the functionalities provided by the Project Lombok that simplifies a lot of repetitive code writing.
- The **org.bouncycastle bcpkix-jdk15on** dependency represents the Bouncy Castle Java APIs for CMS, PKCS, EAC, TSP, CMP, CRMF, OCSP, and certificate generation. This jar contains APIs for JDK 1.5 and up. The APIs can be used in conjunction with a JCE/JCA provider such as the one provided with the Bouncy Castle Cryptography APIs.
- **Spring Boot Starter Actuator** provides production ready features to help developers monitor and manage their application.
- **JSON** dependency is a package that contains files which implement JSON encoders/decoders in Java. The package also includes the capability to convert between JSON and XML, HTTP headers, Cookies, and CDL.

4.2. Technical approach

In the following subsections of the 4.2 chapter, the technical approach of the HCP Web App design stage will be presented. The following paragraphs will contain details about development viewpoints and perspectives that were of utmost importance discussed and decided by the SIMAVI team in the development process of the HCP Web App. Of course, the main step of defining the technical approach is to present the main used technologies.

The HCP Web App is a Java Web Application developed using the Java framework SpringBoot as the main framework. The dependencies of other libraries that are used in the development are managed by Maven.

The keywords that characterize the technical approach for HCP application development are:

- Java Development Kit (JDK);
- Spring Boot Framework as the backbone of all the components that make up the application;
- Thymeleaf as template engine for generating html pages;
- Bootstrap CSS and jQuery for the frontend development;
- Java Database Connectivity (JDBC) the application programming interface (API) used by the applications developed in java to access relational databases;
- H2 database a relational database management system which in this version (V2) of the design will be embedded in HCP Web App as a in-memory database;
- JUnit for unit testing the components and services developed within the project.

4.2.1 Coding conventions

Coding conventions are a set of guidelines for a specific programming language that contain the best approach for programming style and practices for a program written in that language. The main code standards that were used are the following ones:

- Oracle Java Code Conventions for the Java modules
- Google HTML/CSS Style Guides
- W3C Extensible Markup Language (XML) 1.0 (Fifth Edition).

4.2.2 Package diagram

This section presents the main packages and relations between them of the HCP Web App as they were designed at the time of writing this deliverable¹.

¹ September 2019.

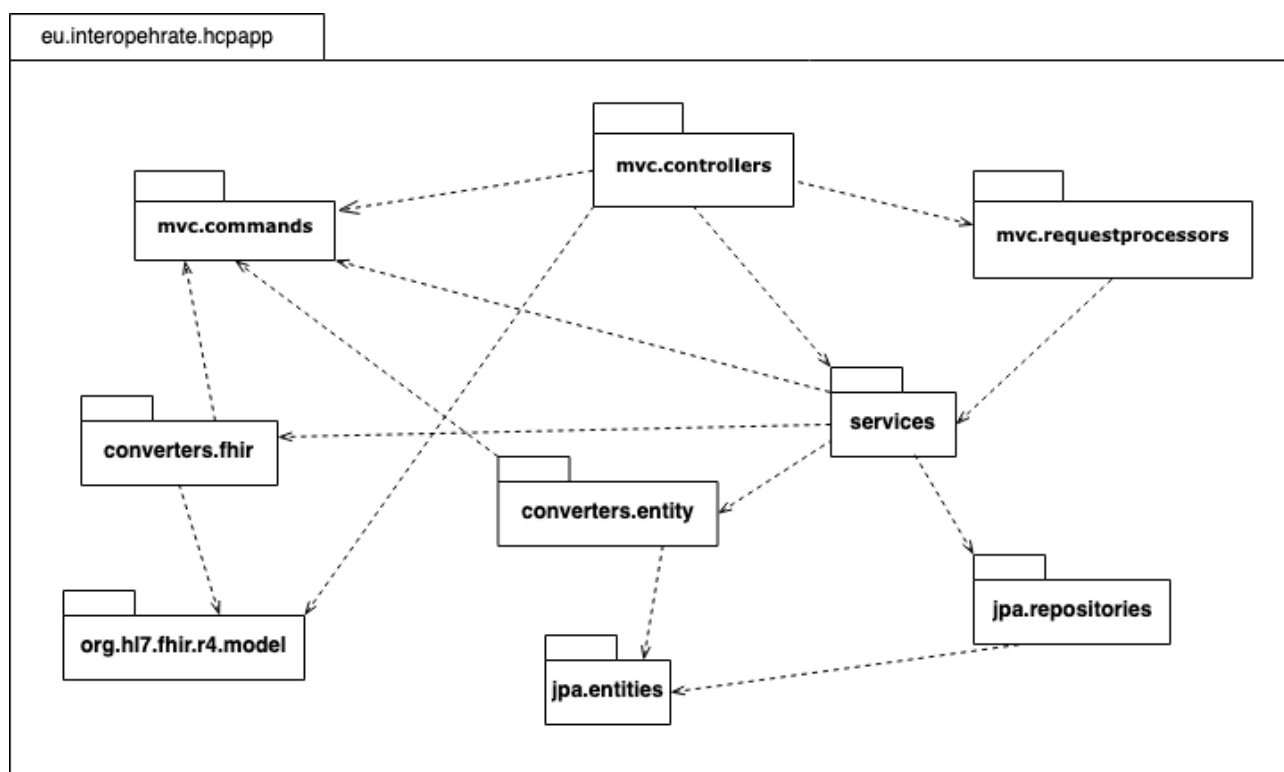


Figure 3: HCP Web App - Main packages and their relations

These packages can be presented considering the three layers of the application.

Packages for implementing the presentation layer

eu.interopehrate.hcpapp.mvc.controllers - contains classes marked with the stereotype `@Controller` responsible to generate responses to the requests that come via HTTP (Hyper Text Transfer Protocol). *Controller* is part of the MVC pattern [9].

eu.interopehrate.hcpapp.mvc.commands - classes used to implement the *Model* part of the MVC pattern [9]. These can be seen as Data Transfer Objects (DTO) [11] but we are considering the Command pattern [12] in order to have more flexibility.

eu.interopehrate.hcpapp.mvc.requestprocessors - helper classes used by the controller classes written in order to simplify the code. Controllers will delegate the execution to these processors.

Packages for implementing the logic layer

eu.interopehrate.hcpapp.services - contains the classes that implements the business logic of the HCP app. Classes from this package are annotated with the stereotype `@Service`.

eu.interopehrate.hcpapp.converters.entity - The classes from this package performs conversions between entities and commands and vice versa, namely, conversion between the model of the presentation layer and the model of the data layer. These classes are used by the services and are annotated with the stereotype `@Component`.

eu.interopehrate.hcpapp.converters.fhir - The classes from this package performs conversions between FHIR model used in the project (third party) and commands and vice versa. These classes are used by the services and are annotated with the stereotype *@Component*.

org.hl7.fhir.r4.model - These classes are not developed within the project, they are part of the FHIR implementation included in HCP app and this implementation is called HAPI-FHIR [6]. These classes are used by the services and are annotated with the stereotype *@Component*

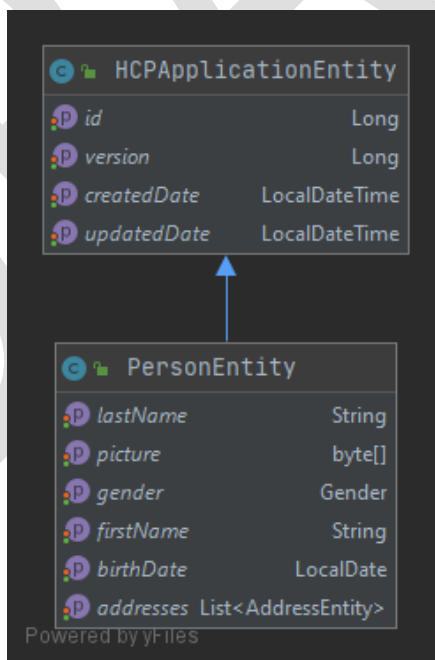
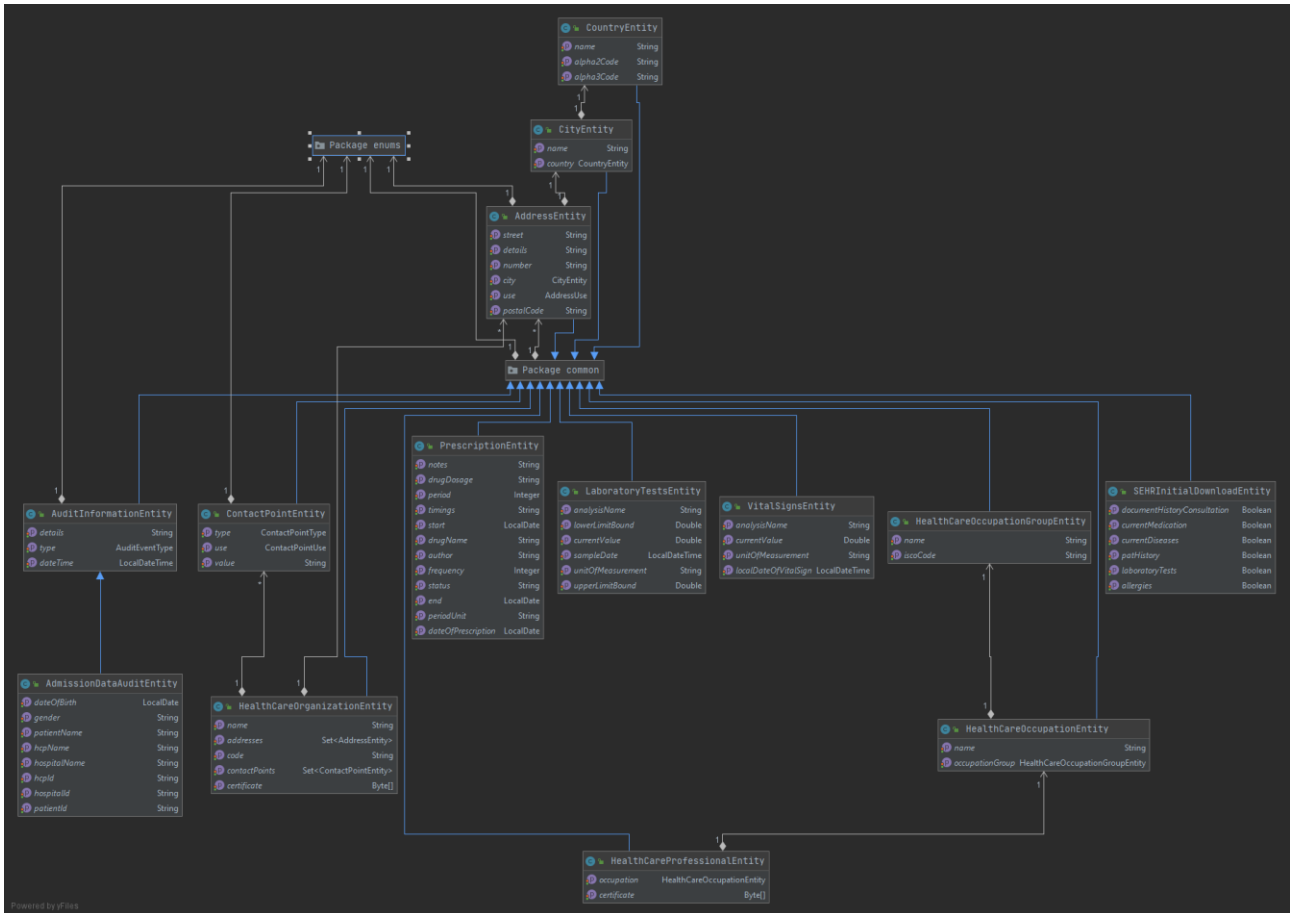
Packages for implementing the data layer

eu.interopehrate.hcpapp.entities - The classes from this package are the Java Persistence API (JPA) [7] entities that are mapped on the tables from the relational database. These classes are mainly used by the repositories and are annotated with *@Entity*.

eu.interopehrate.hcpapp.repositories - This package contains mainly interfaces for which the implementations will be generated at runtime by the Spring Framework. These interfaces are annotated with the stereotype *@Repository* being an implementation of the Repository pattern [8].

4.2.3 Entity diagram

The diagram below represents the diagram of the entities involved in the HCP App. All entities are derived from a mapped super class - HCPApplicationEntity.



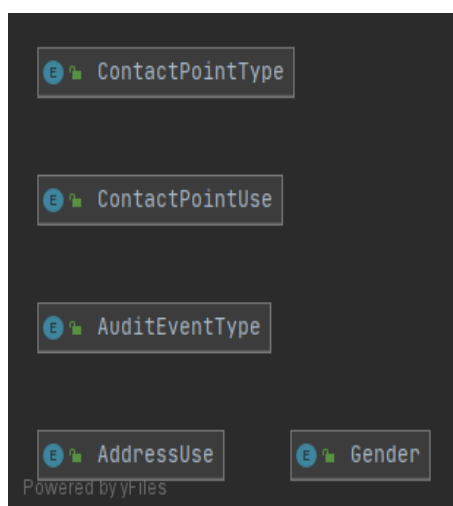


Figure 4: HCP Web App - Main diagram of the entities involved

- **HCPApplicationEntity** representing the common superclass.
- Entities for modeling addresses:
 - **CountryEntity** - stores countries;
 - **CityEntity** - stores cities;
 - **AddressEntity** - stores addresses
- **HealthCareOrganizationEntity** - stores information about health care organizations;
- **PersonEntity** - stores information about people. Is super class for more specialized entities, for instance health care professionals;
- **HealthCareProfessionalEntity** - stores information about health care professionals;
- **SEHRInitialDownloadEntity** - stores what medical data category will be downloaded when the D2D connection is established;
- **ContactPointEntity** - contact points for organizations or persons;
- **HealthCareOccupationGroupEntity** - nomenclature with health care occupation groups;
- **HealthCareOccupationEntity** - nomenclature with health care occupations;
- **AuditInformationEntity** - stores audit information:
- **AdmissionDataAuditEntity** - stores patient information;
- **LaboratoryTestsEntity** - stores laboratory results information;
- **PrescriptionEntity** - stores prescription information;
- **VitalSignsEntity**- stores vital signs information.

4.3. Interoperability approach

This section specifies how the HCP App deals with FHIR-based health data that follows the Interoperability Profile as defined by the InteropEHRRate project. Such data may be expressed in multiple languages and may

have undergone data conversion and translation for interoperability reasons. In case multiple versions or languages of the same data are available to the HCP App, it needs to show the most appropriate version to HCPs, in a way that they are made aware whenever converted or translated data is being shown to them.

4.3.1. Interoperable and multilingual information within health data

A major functionality of the HCP App is to display health records that originate from other healthcare organisations (hospitals). Such a feature is subject to numerous challenges:

- the language used within the health record is different from the language of the HCP;
- the health record uses encoding systems and/or terminology that are locally not used and/or not well known by the local HCP;
- the health record is structured in a way that is usually different from local data schemas and may be more or less understandable to local HCPs.

Due to the difficulty of harmonising EHRs across countries, InteropEHRate foresees three levels of interoperability, in increasing order of complexity:

1. **secure:** the original health record (potentially consisting of data structures as well as unstructured PDF files, images, etc.) are embedded into a FHIR-based container, but no conversion or translation are applied to them, and are presented to the HCP as is;
2. **syntactic:** the original *structured* (e.g., tabular) health record data are mapped to the appropriate FHIR resources, but data values (terminology, codes, text) remain unchanged;
3. **semantic:** beyond syntactic mapping, local codes and terminology are mapped to international ones, and natural language text is translated to the language of the HCP using the App.

The three levels are cumulative, meaning that level n also supports level $n-1$. In practice, this means that level 3 “Smart Health Data” may contain all of the following information:

- the health record in its intact, original, unstructured, document-based form (always);
- the health record in its original structured form (whenever available);
- the health record in FHIR format in the original language, where certain data values are present in multiple forms:
 - in their original form (always);
 - in their international mapped representation;
 - when the original form contains a coded value, a natural-language interpretation in the original language may be added;
 - when the original representation contains natural language text, a new version of the text enriched with concepts (meanings of terms) may be added.
- translations of the health data included as FHIR extensions.

For example, let us suppose that an Italian health record contains the information in both structured and unstructured form:

“La diagnosi primaria è l’occlusione dell’arteria basilare”

meaning “the primary diagnosis is the occlusion of the basilar artery.”

A level-1 (“secure interoperability”) health record may contain:

- the original unstructured health data, in Italian: “La diagnosi primaria è l’occlusione al livello dell’arteria basilare”;
- if the original data was structured, it could express the same information in the following form (using ICD-9 coding, which may not be indicated explicitly):

diagnosi_primaria
433.0

A level-2 (“syntactic interoperability”) health record will be converted to FHIR; however, its contents may not use terminology and codes as defined in the Interoperability Profile²:

Condition.code			
Coding			Text
code	system	display	occlusione dell'arteria basilare
433.0	ICD-9		

A level-3 (“semantic interoperability”) health record will also be in FHIR but will also be using international standards as defined in the Interoperability Profile:

Condition.code			
Coding			text
code	system	Display	occlusione dell'arteria basilare
I65.1	ICD-10	occlusione e stenosi dell'arteria basilare	
433.0	ICD-9	occlusione e stenosi dell'arteria basilare	

² The FHIR data representations are illustrative and may not fully conform to FHIR specifications.

Note that it is possible, although not mandatory, to include multiple codings, as shown above. Note also that the contents of the “display” attributes come from the ICD-9 and ICD-10 classifications (inserted by the Health Data Integration Platform when generating the FHIR dataset), while the content of the “text” attribute (if present) was provided by the original health data itself.

Finally, translations of codes and text are incorporated into FHIR as extensions. The full syntax of extensions is defined as part of the Interoperability Profile [18] [D2.8], the example below is a simplified representation.

Condition.code				text	_text																
Coding																					
code	system	display	_display																		
I65.1	ICD-10	occlusione e stenosi dell'arteria basilare	<table border="1"> <tr> <td colspan="2">extension: translation</td> </tr> <tr> <td>language</td> <td>Eng</td> </tr> <tr> <td>value</td> <td>occlusion and stenosis of basilar artery</td> </tr> <tr> <td>provenance</td> <td>ICD-10</td> </tr> </table>	extension: translation		language	Eng	value	occlusion and stenosis of basilar artery	provenance	ICD-10	occlusione dell'arteria basilare	<table border="1"> <tr> <td colspan="2">extension: translation</td> </tr> <tr> <td>language</td> <td>eng</td> </tr> <tr> <td>value</td> <td>occlusion of the basilar artery</td> </tr> <tr> <td>provenance</td> <td>Google Translate</td> </tr> </table>	extension: translation		language	eng	value	occlusion of the basilar artery	provenance	Google Translate
			extension: translation																		
language	Eng																				
value	occlusion and stenosis of basilar artery																				
provenance	ICD-10																				
extension: translation																					
language	eng																				
value	occlusion of the basilar artery																				
provenance	Google Translate																				
433.0	ICD-9	occlusione e stenosi dell'arteria basilare																			

4.3.2. Reading and writing health records

The HCP App receives, using the D2D or R2D protocols, Smart Health Data that was extracted from another hospital, where it has been converted to the international FHIR-based format as described in the section above. This data will always contain the original version(s) of the health records in the original language, but may also contain translations of it to other languages. Typically for a cross-border scenario, however, it will *not* contain the translation into the local language (of the local HCPs) as such a translation can only be done locally (i.e., in the target country) with high-enough quality. Therefore, the HCP App needs to use locally deployed *InteropEHRRate Health Services* in order to translate textual data content to the language of local HCPs. In case the HCPs extend the patient’s health data by new content, this content may also be translated to the patient’s own language upon his/her request, in case it is different from the local language. Newly entered content may also need to be processed in order to detect and convert

terminology and codes entered by the HCP as natural-language strings into formal knowledge, in order to enable their automated conversion to international standards. The *Figure 5* below shows the process by which the HCP App uses the underlying InteropEHRate Health Services in order to implement data conversion and translation.

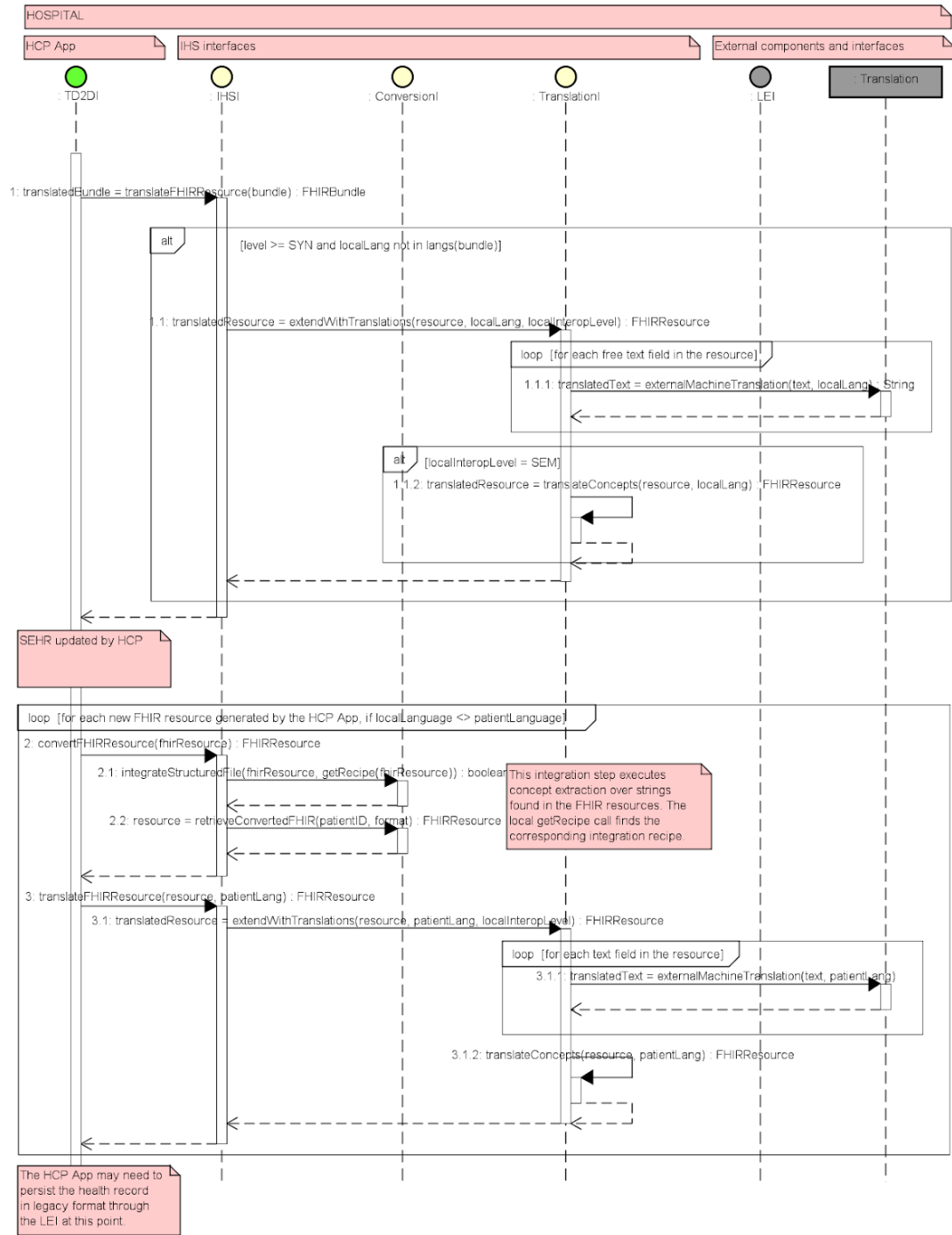


Figure 5: Sequence diagram for underlying InteropEHRate Health Services - HCP Interactions

1. The HCP App verifies the language(s) of the incoming FHIR-based health record. In case it differs from the local language, the HCP App calls the *translateFHIRResource* IHS translation API endpoint to translate the entire FHIR bundle.
2. While the internal details of the translation process are shown in the figure above, they are hidden from the HCP App. The translation call returns a FHIR bundle with translations added as FHIR extensions, in the way described in [25] [D5.11].
3. At this point, the HCP App is ready to display the translated health data to the HCP user.
4. The HCP may enter new content into the health record. It is assumed that all such content will be implemented as newly created FHIR resources, and it is the HCP App that creates them according to the Interoperability Profile [18] [D2.8].
5. If the HCP entered free text in his/her own language, individual healthcare terms, or codes, then these texts and codes may need to be represented according to the Interoperability Profile. Furthermore, upon specific request from the patient, free text entered in the local language may need to be translated to the patient's own language. These functionalities are again provided by the IHS as follows.
6. The HCP App first applies conversion to each newly created FHIR resource (either separately or bundled together) using the Conversion Services, by calling *integrateStructuredFile* and then retrieving the result from the Data Integration Platform through a *retrieveConvertedFHIR* call. These service endpoints are described in detail in deliverable [24] [D5.10].
7. Then, upon explicit request from the patient, already converted resources are extended by translations into the patient's own language (using the *extendWithTranslations* call), if such translation is provided by the local IHS. This translation could (for example) be triggered by the HCP, indicating the target language of the translation.
8. It is the responsibility of the HCP App to reassemble the FHIR bundle from the resources mentioned above before uploading this bundle onto the S-EHR App.

4.3.3. Display of health records

The FHIR health record will indicate for each FHIR data value whether it went through a conversion or a translation process. The HCP App can then display data based on the value of this flag. Since data values go through automated conversions (to international standards) and translations, due to imperfections inherent to these processes and their underlying resources, they may provide incorrect results with a certain probability, even if they are assumed to be of generally high quality. For this reason, it is crucial for the HCP App:

- explicitly to show to the HCP all data values that went through a conversion or translation step;
- to display the original (unconverted, not translated) values alongside the new ones.

In the InteropEHRate reference implementation, converted codes, terms, and the translations of their human-readable descriptions originate from expert-curated knowledge (term bases, classifications) and can generally be considered as accurate. Machine-translated text is translated using generic technology and should not be considered as accurate nor entirely relied upon. Thus, it may be desirable for the HCP App to distinguish between these two cases in the way the converted/translated data values are signalled to the HCP or, as a more neutral solution, to provide provenance information alongside the converted/translated value (such as "Google Translate" or "ICD-10" in the translations from the example in the previous section).

4.4. Display medical images

In the second version (V2) of HCP Web App, a new module dedicated to displaying DICOM images [26] was designed. DICOM - Digital Imaging and Communications in Medicine — is the international standard for medical images and related information. It defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use [19] .

One of the best known applications that implements best practices in displaying DICOM images is Weasis [27] . Weasis DICOM viewer is a cross-platform, free, multi-language open source software, which allows a flexible integration to PACS, RIS, HIS or PHR. Weasis is a very popular solution used by hospitals and provides web-based access to radiological images.

HCP Web App integrates Weasis and it must be installed on the HCP's terminal. HCP Web App starts Weasis when image display functionalities are accessed.

Weasis can manage images in 2 ways, in a client-server architecture or direct integration with another application. Given the specific of the HCP Web App, where the medical images are received from S-EHR Mobile App or S-EHR Cloud, Weasis has to be integrated directly. HCP Web App receives images as byte array and invokes Weasis in order to display them.

Below are two screenshots with Weasis displaying images provided by the HCP Web App.

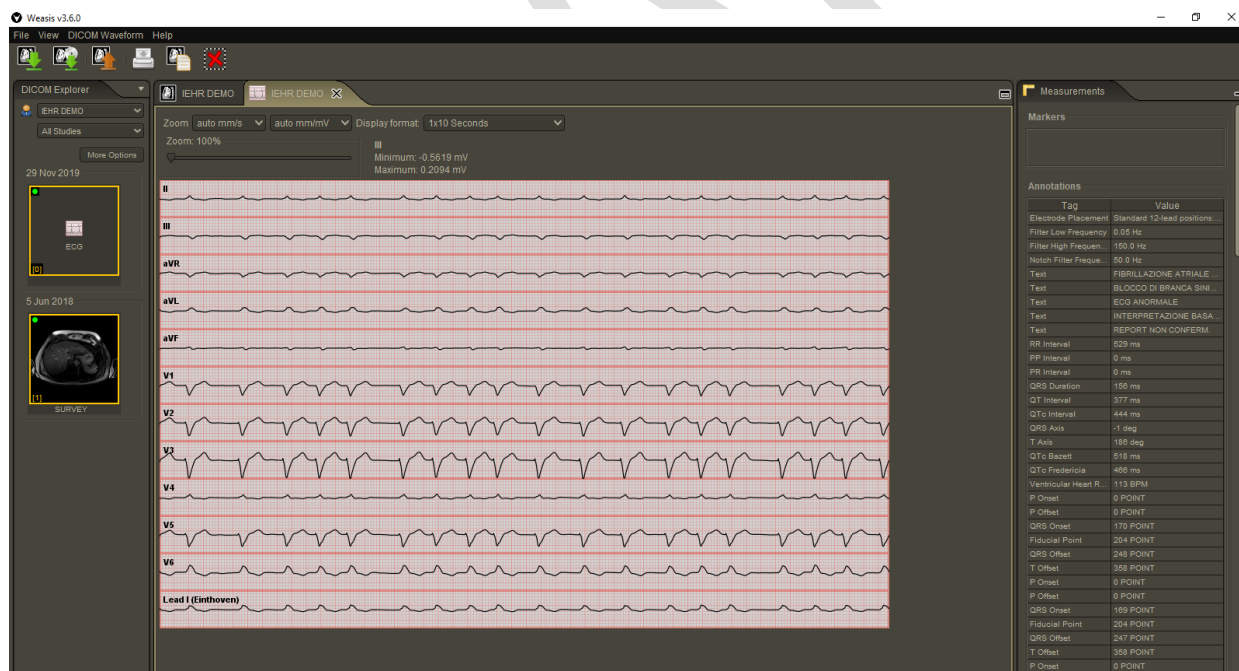


Figure 6: Electrocardiogram in Weasis

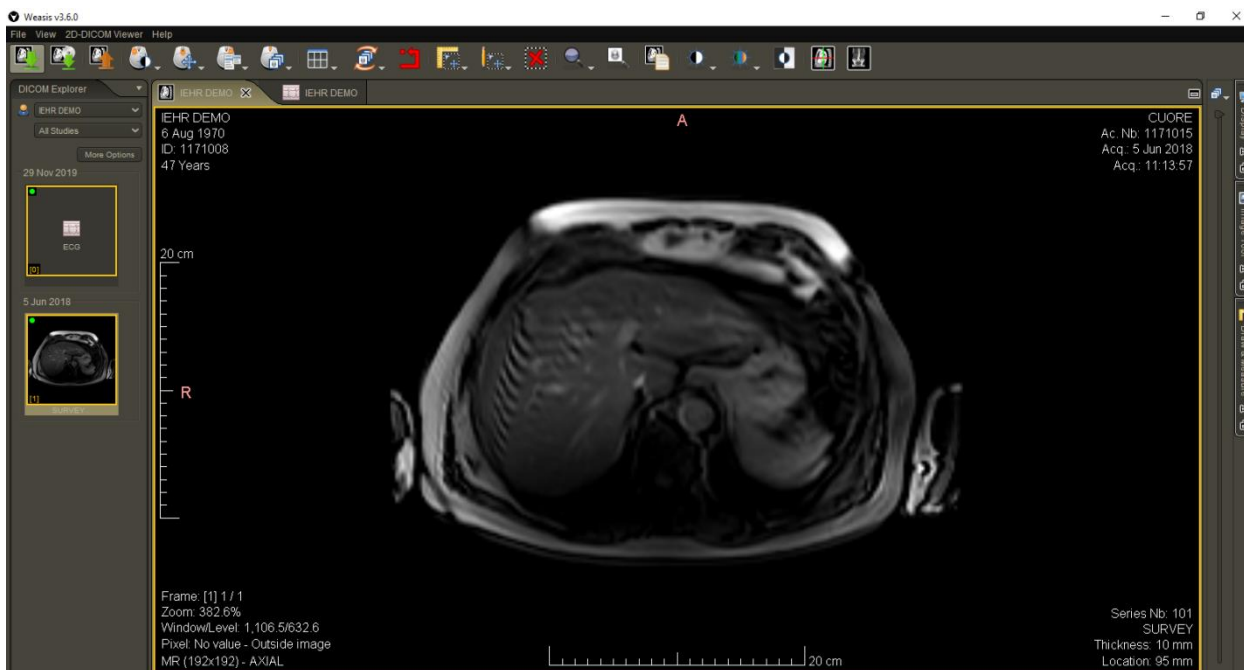


Figure 7: Skull - magnetic resonance image in Weasis

4.5. Emergency scenario - integration with S-EHR Cloud

The HCP application can now access data directly from S-EHR Cloud services via scanning the patient’s dedicated QR Code, according to the emergency scenario. On the Homepage of the app, the HCP (Healthcare professional) can select one of two connection types with the patient’s data. He/she can select the Medical Visit scenario option (connection with S-EHR mobile app via Bluetooth) or the Emergency scenario option (connection with the patient’s S-EHR Cloud):

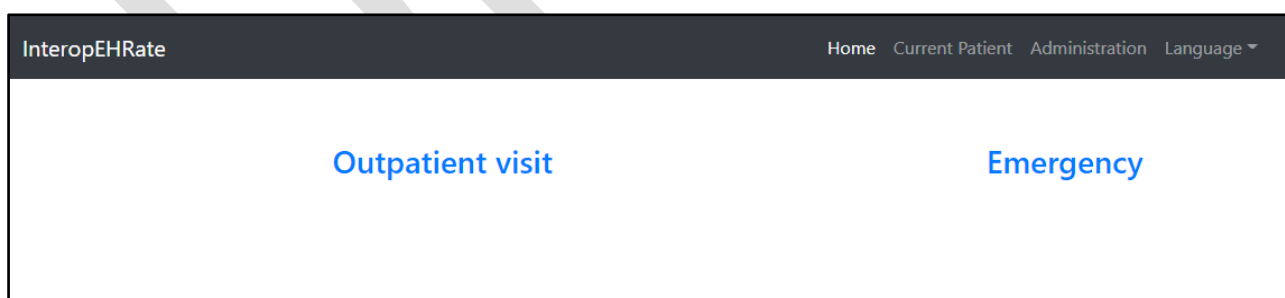


Figure 8: Home page with 2 options available for the HCP

According with the latest CoDesign sessions, the Medical Visit option was renamed Outpatient Visit.

From the perspective of security there are three issues that need to be addressed when HCP App establishes a connection with S-EHR Cloud to exchange information:

- Patient identification;

- HCP identification;
- Encryption / decryption of exchanged information.

Details about the security solution will be provided in [D6.7] Design of a service for cloud storage of S-EHR content. From the HCP App perspective it has to access the API exposed by the S-EHR Cloud with the required input data. Currently (in version V2) the only information that is exchanged is the download of International Patient Summary (IPS). This can be obtained by a HTTP request of type GET with two parameters:

- The patient's token;
- The HCP's token.

An example of such a request is presented below.

```
GET http://213.249.46.253:5000/download-ips?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Im1hcmlvLnJvc3NpIiwicGFzc3dvcmQiOiJpbnRlcm9wZWWhyYXR1IiwiaXhwIjozNTk3NDE2MTA4fQ.18b_KSrzCvJKfRTQs1UVtbhOKXJ_17fIw7_PYZznN94&hcp_token=TBD
```

```
Accept: application/json
```

The response to this request is the IPS in JSON format, encrypted by using a symmetric algorithm. Then the JSON string is decrypted, information is extracted from the IPS and displayed in the specific sections.

The information about the patient's token, the symmetric key and the Cloud service address is coming from the QR code held by the patient and scanned by a QR code reader connected to the HCP application. The content of the QR code is a string in JSON format with the following structure:

```
{
  "symKey": "UiYmJk/iWopGS2n0YIhZsgp1auRVyahR7sgFOtEC5r4\u003d",
  "token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Im1hcmlvLnJvc3NpIiwicGFzc3dvcmQiOiJpbnRlcm9wZWWhyYXR1IiwiaXhwIjozNTk3NDE2MTA4fQ.yg3-NsG9UWJ8lczTkS62bNTIR5K9ERq3JEt9Qje_fCg",
  "uri": "http://213.249.46.253:5000/"
}
```

- symKey - The symmetric key for decrypting the content received from S-EHR Cloud;
- token - The patient's token;
- uri - The address of S-EHR Cloud.

Below are presented code snippets which illustrate the design of the sub module dedicated to the interaction between S-EHR Cloud and the HCP App:

- Module's interface - defines a single method that returns the IPS as a bundle based on the content read from the QR code.

```

4
5 public interface EmergencyService {
6     Bundle getIps(String qrCode);
7 }

```

Figure 9: Emergency Service interface

- Interface implementation - the JSON content of the QR code is parsed and the information needed for querying the S-EHR Cloud is extracted.

```

@Override
public Bundle getIps(String qrCode) {
    if (qrCode != null) {
        try {
            JSONObject qrCodeJSON = new JSONObject(qrCode);
            if (qrCodeJSON.has(SYM_KEY_KEY) && qrCodeJSON.has(TOKEN_KEY) && qrCodeJSON.has(URI_KEY)) {
                String symKey = qrCodeJSON.getString(SYM_KEY_KEY);
                String token = qrCodeJSON.getString(TOKEN_KEY);
                String uri = qrCodeJSON.getString(URI_KEY);
                return getIps(uri: uri + CONTEXT, token, symKey, UUID.randomUUID().toString());
            }
        } catch (JSONException e) {
            throw new RuntimeException(e);
        }
    }
    return null;
}

```

Figure 10: - Emergency Service implementation

- The cloud is queried, the content is decrypted and then the result is parsed using the HAPI library.

```

private Bundle getIps(String uri, String citizenToken, String symKey, String hcpToken) {
    UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl(uri)
        .queryParams("token", citizenToken)
        .queryParams("hcp_token", hcpToken);

    HttpHeaders headers = new HttpHeaders();
    headers.setAccept(Collections.singletonList(MediaType.APPLICATION_JSON));

    HttpEntity<String> entityRequest = new HttpEntity<>(headers);
    ResponseEntity<String> jsonIpsResponse = restTemplate.exchange(uriBuilder.toUriString(), HttpMethod.GET, entityRequest, String.class);

    String jsonIpsDecrypted;
    try {
        jsonIpsDecrypted = EncryptedCommunicationFactory.create().decrypt(jsonIpsResponse.getBody(), symKey);
    } catch (Exception e) {
        throw new RuntimeException(e);
    }

    FhirContext fhirContext = terminalFhirContext.getContext();
    IParser parser = fhirContext.newJsonParser();
    return (Bundle) parser.parseResource(jsonIpsDecrypted);
}

```

Figure 11: IPS - get, decrypt, parse

The Graphical User Interface dedicated to the connection with S-EHR Cloud is presented hereinafter in **Chapter 5.2.2. UI design in HCP Web App. Second GUI Iteration.**

5. DESIGNING THE GUI OF HCP WEB APP

5.1. End-user viewpoints and perspectives

As the healthcare professionals are highly interested in the demonstration of the communication between the HCP WEB App and the S-EHR, the most immediate and intuitive way is through visual illustration of different features of HCP Web App that will be accessible to healthcare professionals. To reach this goal, the section presents the updated version of the design of the GUI of HCP Web App, based on the end user's valuable input.

As presented above in *Section 3. Methodological approach*, we continued to draft and optimise the design of the user interface from the perspective of healthcare professionals, based on valuable feedback gathered from the end users involved in the project within iterative co-design / co-creation sessions, applying the principles of UCD methodology. The new feedback from the co-design sessions held this year led us to redefine the design presented in the previous deliverable [22] [D5.4]. These changes will be shown during the next subchapter.

Applying the UCD methods during the HCP Web App UI design, we aim to obtain an increased usability, accessibility, user satisfaction and comfort. Furthermore, applying a user-centric approach every time we apply a design change to the HCP Web App, we aim to ensure also the visibility and readability of the specific content, and the visual consistency of information displayed.

More specifically, our work focused on the *end user layer* design was mainly based on the following aspects:

- Designing the particular user interfaces in **HCP Web App** corresponding to the first year and second year users' requirements (V1 and V2 selection related to HCP app), as they were described in deliverable [15] [D2.1] and [16] [D2.2] - *Chapter 5. User requirements*, namely:
 - HCP uses the software "HCP App", able to access an S-EHR app by using a Device to Device connection
 - The HCP can see the identification data of the Citizen on the HCP app in order to confirm the Citizen's identity
 - The HCP can configure what are the Citizen's health data that will be automatically downloaded at connection time of the HCP app with the S-EHR app. Depending on the situation (e.g. a regular consultation or an emergency) the HCP will be able to choose if she/he wants to see all the patient health data available, or only a glimpse of it.
 - The set of Citizen's health data, previously configured by the HCP, are automatically downloaded on the HCP app at connection time of the HCP app with the S-EHR app. If some of this data is hidden by the patient, they are not downloaded.
 - A Doctor can view the Patient Summary (a portion of it) shared with him/her by a Citizen using the HCP app.
 - A doctor can view vital signs and other measurements using the HCP app

- A portion of Patient Summary is automatically downloaded at connection time of the HCP app with the S-EHR app
 - The vital signs and other measures taken during previous visits are downloaded from the S-EHR at the request of the HCP app.
-
- Establishing an initial set of patient health data (provided by S-EHR app) to illustrate in the **HCP Web App** within the discussions and co-design sessions with the healthcare partners of the project in May 2020, June 2020, July 2020 and August 2020, with relevance for testing the D2D communication protocol in this stage of InteropEHRate implementation, such as:
 - Displaying Demographic Information
 - Displaying Allergies
 - Displaying Current Diseases
 - Displaying Current Medications
 - Displaying Laboratory tests
 - Displaying Pat history
 - Displaying Document History Consultation
 - Displaying Vital Signs
 - Displaying Visit Data
 - Displaying Reason
 - Displaying Reason
 - Displaying Ph Exam
 - Displaying Conclusion
 - Displaying Prescriptions.

5.2 Mockups of HCP Web App. Second iteration for updating and improving the UI design

This section depicts the updates and improvements of UI design specifications of HCP Web App, based on the relevant aspects mentioned previously. Important design decisions are documented, too.

Specific mock-ups were designed to transfer the requirements of HCP Web App into a visual representation.

The simplified sketches focus in this stage of design on functional elements. Like the mockups, the visual design will be tested with the end users in the next stage of development and further optimised once this functional version of HCP Web App is implemented.

Hereinafter, are presented a series of mock-ups which reflect the outcome of ten co-design sessions from May 2020, June 2020, July 2020 and August 2020 where the healthcare partners of the project were involved. The proposed graphical illustration meets the users' requirements and the specific health data organization and structure compliant with HL7 FHIR profile.

Within these 10 co-design sessions together with the final users it was decided to make changes regarding the menu left hand and the categories of data. The new main menu contains new sections of data, but also the old categories with a different name.

The new sections of data are:

- Current Diseases
- Pathology History
- Document History Consultation
- Visit Data
- Vital Signs
- Reason
- Ph Exam
- Conclusion

For the other sections the name was changed:

- Demographic Information represents the old section General data
- Allergies represents the old section Allergies and Intolerances
- Laboratory tests represents the old section Diagnostic results

A new section – Visit data has been added, which is a drop-down menu. This section has other sections with patient data, namely: Vital signs, Reason, Ph Exam, Conclusion and Prescriptions.

According to the co-design sessions we depict in the following the mock-ups of the sections presented above.

The Main menu of HCP Web App

InteropEHRate	Home	Current Patient	Administration
Demographic Information		Name: Mario Rossi	Age: Country:
Current Diseases			
Pat history			
Allergies			
Current Medications			
Document History Consultation			
Laboratory Tests			
Visit Data			
Vital Signs			
Reason			
Ph Exam			
Conclusion			
Prescriptions			

Figure 12: New main menu

Demographic Information

InteropEHRate		Home	Current Patient	Administration
Demographic Information		Name: Mario Rossi		Age: Country:
Current Diseases	Name:	GP/Referral physician name:		
Pat history	Surname:	GP/Referral physician phone:		
Allergies	Birth date:	Country of residence:		
Current Medications	Place of birth:	E-mail physician:		
Document History Consultation	Gender:	Referral nurse name:		
Laboratory Tests	Spoken language:	Referral nurse phone:		
Visit Data	Country of residence:	Country of residence:		
Vital Signs	Social security number/PID:	E-mail nurse:		
Reason	Next of kin:			
Ph Exam	Emergency contact:			
Conclusion	Patient phone:			
Prescriptions	E-mail patient:			

Figure 13: Demographic information about patient

Current Diseases

InteropEHRate		Home	Current Patient	Administration									
Demographic Information		Name: Mario Rossi		Age: Gender:									
Current Diseases	<table border="1"> <thead> <tr> <th>Disease</th> <th>Date of diagnosis</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>Chronical Heart Failure</td> <td>22.03.2020</td> <td>EF= 25% ACD device implanted in 2006</td> </tr> <tr> <td>Psoriasis</td> <td>19.05.1998</td> <td>Head, neck</td> </tr> </tbody> </table>				Disease	Date of diagnosis	Comment	Chronical Heart Failure	22.03.2020	EF= 25% ACD device implanted in 2006	Psoriasis	19.05.1998	Head, neck
Disease	Date of diagnosis	Comment											
Chronical Heart Failure	22.03.2020	EF= 25% ACD device implanted in 2006											
Psoriasis	19.05.1998	Head, neck											
Pat history													
Allergies													
Current Medications													
Document History Consultation													
Laboratory Tests													
Visit Data													
Vital Signs													
Reason													
Ph Exam													
Conclusion													
Prescriptions													

Figure 14: Current Diseases

Pathology history

InteropEHRRate Home **Current Patient** Administration

Demographic Information Name: Mario Rossi Age: Country:

Current Diseases

Pat history

Allergies

Current Medications

Document History Consultation

Laboratory Tests

Visit Data

Vital Signs

Reason

Ph Exam

Conclusion

Prescriptions

Diagnosis	Year of diagnosis	Comments

Risk factor

Hypertension

Diabetes

Dyslipidemia

Current/Past smoker

Family History of CVD

Patient History

Social History

Family History

Figure 15: Pathology history

Allergies

InteropEHRRate Home **Current Patient** Administration

Demographic Information Name: Mario Rossi Age: Country:

Consents

Current Diseases

Pat history

Allergies

Current Medications

Document History Consultation

Laboratory Tests

Visit Data

Vital Signs

Reason

Ph Exam

Conclusion

Prescriptions

Language of the Section

Category	Name	Type	Symptoms

Comments

Figure 16: Allergies

Current Medications

Figure 17: Current Medications

Document history consultation

A filter has been added and data can be filtered by specialty (e.g. Cardiology or Psychiatry).

Figure 18: Document History Consultation

Laboratory Tests

The analyses can be sorted alphabetically.

The screenshot shows the 'Laboratory Tests' section of the InteropEHRRate application. The top navigation bar includes 'Home', 'Current Patient', and 'Administration'. Patient information is displayed as 'Name: Mario Rossi', 'Age:', and 'Country:'. The left sidebar lists various medical categories, with 'Laboratory Tests' selected. Below the sidebar, there are tabs for 'Laboratory Results', 'Pathology Results', 'Radiology Results', and 'Others Results'. A 'Language of the Section' dropdown is set to 'Original'. The main content area displays a table of laboratory results for 'Observation Laboratory'.

Analysis	5/17/20, 1:46 PM	5/17/20, 1:46 PM	5/17/20, 1:46 PM
SARS-Cov	-	-	-
Eritrociti	-	-	-

Figure19 : Laboratory Tests

Vital Signs

The screenshot shows the 'Vital Signs' section of the InteropEHRRate application. The top navigation bar and patient information are identical to the previous screenshot. The left sidebar lists various medical categories, with 'Vital Signs' selected. Below the sidebar, there are tabs for 'Original' and 'Translated'. The main content area displays a table of vital signs for 'Observation Laboratory'.

Analysis	5/17/20, 1:46 PM	5/17/20, 1:46 PM
Heart rate [bpm]	-	-
Breathing rate [b\min]	-	-
Weight [kg]	-	-
Temperature [C]	-	-
Height [cm]	-	-
Blood pressure [mmHg]	-	-
SPO2 [%]	-	-
AVPU and alertness	-	-

Figure20: Vital Signs

Conclusion

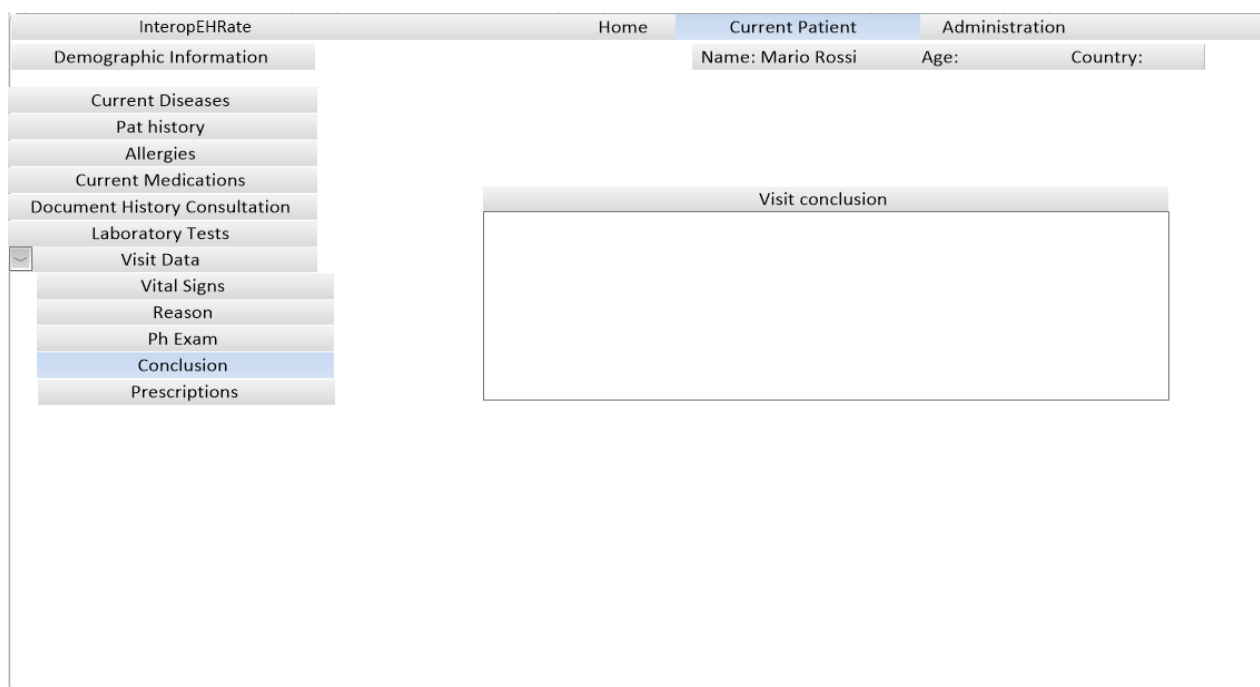


Figure21 : Conclusion

5.3. Current stage of UI implementation

At this stage of project implementation, the section depicts the major features and principles of designing the HCP Web App solution, addressing essentially the import/export data directly from/to the S-EHR on the smartphone (specifically, *the actual stage of D2D library development and the corresponding UI design and implementation in HCP Web App*). Because this is the second year of implementation, this section will depict the major features of the HCP Web App that have been refined during this year and the improvements that have been brought to the features already done last year.

The following section summarises the progress and results of the first and second development iterations, comprising the preliminary and upgraded “look and feel” version of HCP Web App for displaying, sending, manipulating (add, remove, edit) and translating the most relevant healthcare categories and data sets from and to S-EHR mobile application. We also included in this deliverable the Emergency Scenario design which is seemingly integrated inside the Home Screen of the HCP Web App.

As presented in the previous Chapter 5.1, the major sections of the IPS form the basis of the patient’s EHR. Moreover, the *data sets corresponding to the HL7 FHIR standard* (Appendix: Mapping of HL7 FHIR profiles and resources for scenario requirements of deliverables [17] [D2.7] and [18] [D2.8] - Interoperability profile – implementable level specification) were the origin of defining the fields characteristic for the patient's health records, as illustrated in the following chapters dedicated to the UI design.

This section presents the current state at which the implemented GUI is at the second year mark.

The front-end technology used for the application design (UI implementation) is Bootstrap CSS, one of the most recognized for the graphical features and capabilities. Bootstrap CSS is a framework for designing web applications easier and faster.

In this particular case of graphical implementation, we used the framework to create the buttons, tables and pages of the HCP Web App, based on HTML and CSS templates for tables, buttons and other graphic elements.

The GUI design is based on the same structure and visual elements as the mockups presented in the previous section.

As a result of changes that were made to the HL7 FHIR profile and from the discussions with the end users during the co-design sessions, the General data and Health data information have been changed, thus resulting in changes at the design level as well. More examples of data have been received, resulting in new layouts of the presentation of those data, which includes text, tables and even images.

The next paragraphs contain a series of screenshots which show the current stage of GUI implementation.

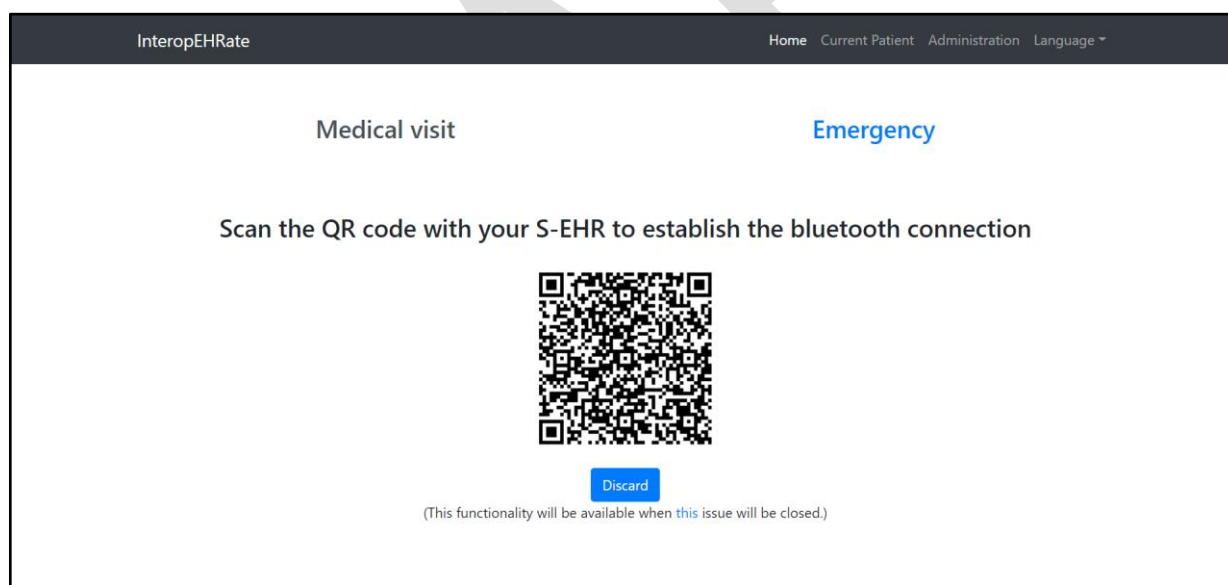


Figure 22: Home page: D2D connection QR code for Medical Visit and option to switch to Emergency

The main addition of the Home page in this new version of HCP Web App is the possibility of the healthcare users to easily switch between medical and emergency components that the HCP Web App provides.

Emergency scenario

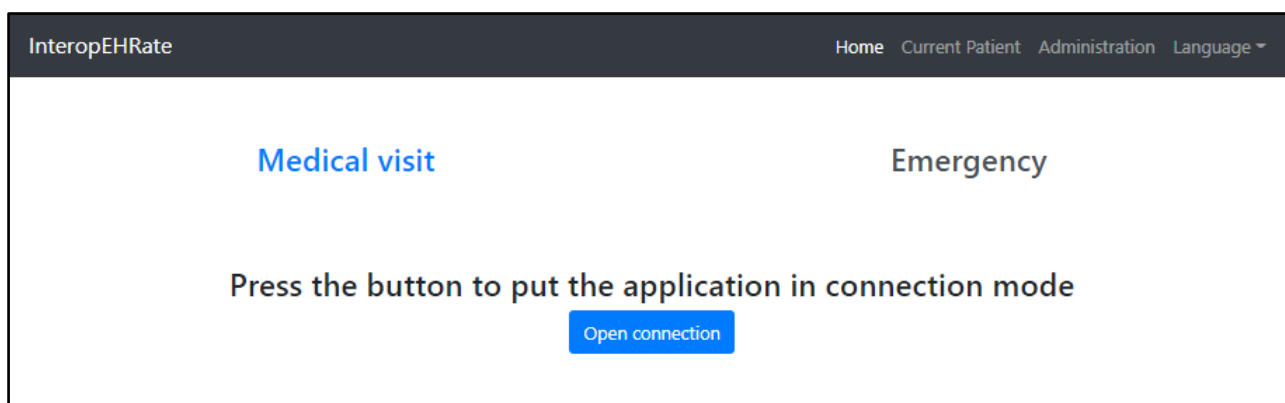


Figure 23: Emergency Scenario selected

Currently, both Emergency and Medical visit scenarios are implemented, with the exception of the HCP being able to read the QR code of the patient during the Emergency scenario. For the time being, the only way to activate the Emergency scenario is that the user to enter as text the QR code decrypted info into a text box which allows the user to proceed with the aforementioned scenario. So, after clicking on the “Open connection” button, the next text box is displayed where the user needs to type the text related to the Cloud (the text is available in **Chapter 4.5**):

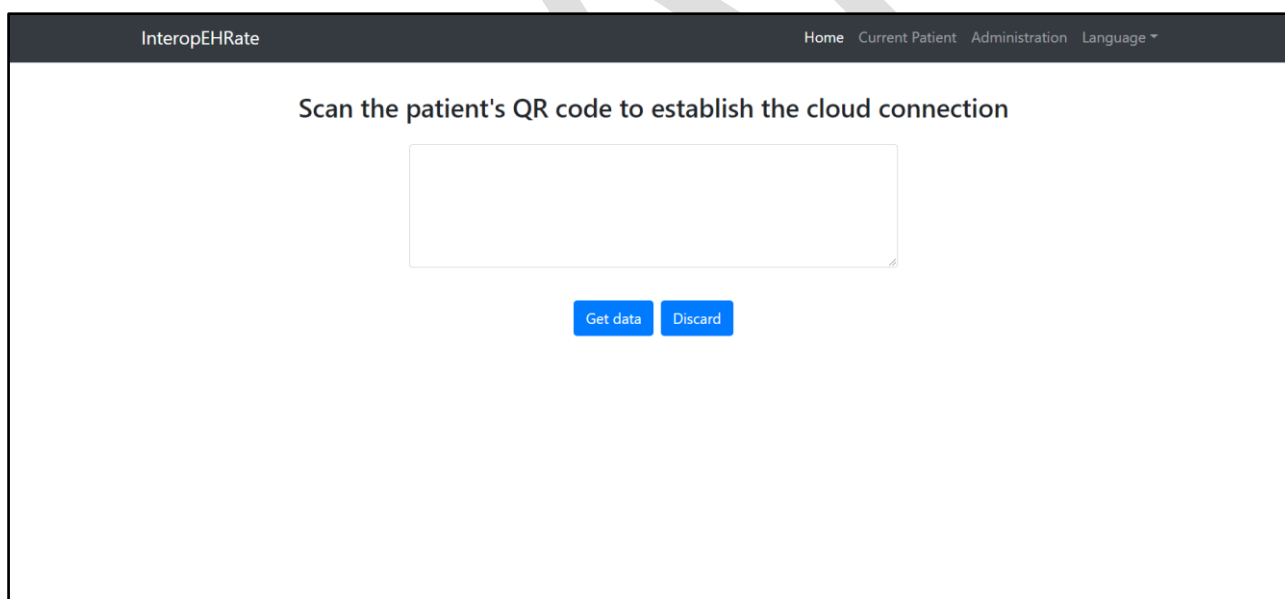


Figure24:- Emergency Scenario: Input page for the QR code information

Now the connection with the Cloud is established and the Emergency text is displayed on the top left corner under the Project title (see image below).

The screenshot displays the InteropEHRate web application interface. At the top, there is a navigation bar with the application name 'InteropEHRate' on the left and links for 'Home', 'Current Patient', 'Administration', and 'Language' on the right. Below the navigation bar, a light blue banner contains the text 'Emergency' on the left and 'Name: Mario Rossi Age: 60 Country: -' on the right. The main content area features a large 'Connected' message in the center, with a blue button labeled 'Close connection' directly below it. Underneath this message is a white box titled 'General info about patient' which contains the following details: First Name (Mario), Last Name (Rossi), Gender (MALE), Birth date (1960-01-01), and ID Patient (http://213.249.46.205:8080/NCP/fhir/Patient/31/_history/1).

Figure 25: Active Emergency scenario and general info about the patient

Below is described the Medical Visit scenario with the Bluetooth connection with S-EHR mobile application.

Medical Visit scenario

After a successful connexion of the S-HER App with the HCP Web App and the consent given from the patient, a page with general data about the patient will be displayed. A green or red message will appear confirming/disallowing the connexion depending on the correctness of the medical IPS received. A band containing the name and the age of the patient will also appear on every page that the HCP enters from this point forward.

InteropEHRate Home Current Patient Administration Language ▾

Medical visit Name: Mario Rossi Age: 60 Country: -

Connected
Transmission completed!
Close connection

General info about patient

First Name
Mario

Last Name
Rossi

Gender
UNKNOWN

Birth date
1960-01-01

ID Patient
Patient/5fd921a7-53df-4868-ba9b-9104ec530c0d

Consent
I have read and understood InteropEHRate's Privacy Policy. I hereby give permission to the recipient health care provider to process (view, store, edit etc.) the personal data stored in my Personal Health Record on this application for the purpose of medical diagnosis and/or treatment. I understand that my consent will remain valid for these purposes unless I affirmatively withdraw it. I have the right to withdraw this consent at any time.

Certificate details
UID=0f3e03e0-b4ca-4a76-b21d-bdef16267ed0,CN=Mario Rossi,OU=InteropEHRate Certificate,O=InteropEHRate,C=IT

Invalid Digital Identity

Figure 26: Patient General Data Page with the Certificate details and the button Invalid Digital Identity

The Red button called “Invalid Digital Identity” is for the practitioner to deny the connection if the patient’s Certificate doesn’t match the patient identity. By clicking it, the connection with the S-EHR is stopped.

In case that something went wrong while the data from S-EHR is transferred to the HCP app, a red marker will appear on the home screen and the connection with S-EHR is interrupted (see below):

InteropEHRate Home Current Patient Administration Language ▾

Medical visit Name: Mario Rossi Age: 60 Country: -

Connected
Transmission error, please retry!
Close connection

General info about patient

First Name
Mario

Last Name
Rossi

Gender

Figure 27:-Error message

As the patient has given consent to share his/her health data, the HCP can begin to look at the information it contains and can begin adding/editing information. As stated earlier, from the discussion with the focus group this year, it led to significant changes at the level of the Health Data categories. In this new version, the HCP can now choose between three default languages that were selected as more relevant for the project implementation. The first topic of Health Data is Demographic Information. Here, the HCP can see the patient's personal data.

InteropEHRate Home Current Patient Administration Language ▾

Medical visit Name: Mario Rossi Age: 60 English
Romanian
Italian

Demographic Information

Current Diseases
Pat History
Allergies
Current Medications
Document History Consultation
Laboratory tests
Visit Data ▾

Demographic Information

Patient details	
Name	Rossi
Surname	Mario
Birth Date	1960-01-01
Place of Birth	-
Gender	UNKNOWN
Native Language	-
Country of Residence	-
Social Security Number/PID	-
Next of Kin	-

Figure 28: CurentPatient - General Data Page

Dettagli del paziente	
Nome	Rossi
Cognome	Mario
Data di nascita	1960-01-01
Luogo di nascita	-
Genere	MALE
Lingua nativa	-
Paese di residenza	-
Numero di previdenza sociale / PID	-
parente più vicino	-

Figure 29: INTERNATIONALIZATION - Italian labels selected

A new category that has been added is the Current Diseases. In this category, the HCP will be able to add different diseases that the patient is suffering from. The translate service is also fully functional in this category. A future update will provide the possibility for the HCP to also add notes to the patient directly.

Disease	Date of Diagnosis	Comment
Other restrictive cardiomyopathy	30-03-2017	
Diabetes	31-08-2020	

Figure 30: Current Diseases

Pat (Pre-Admission Testing) History is the patient's health category which contains the tests that are done before the admission. The implementation of this screen is still in progress.

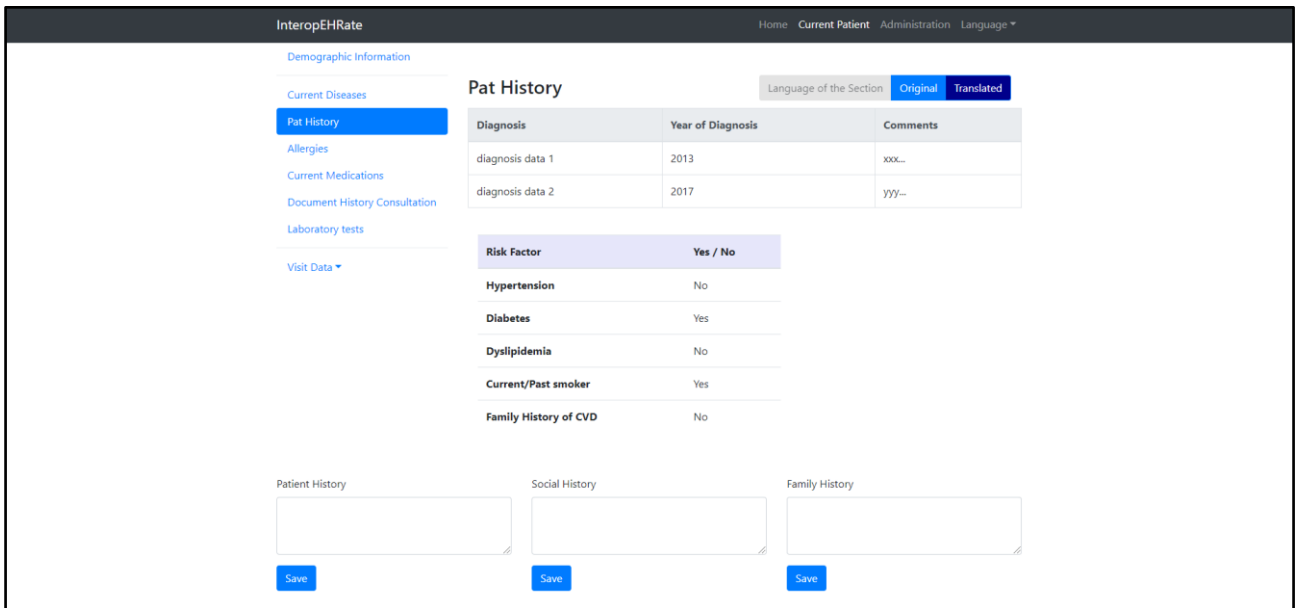


Figure 31: Pat History

The Allergies category of the IPS is also found in this version of the design. Any users of the HCP Web App can see the allergies that have been sent from the S-EHR and can add any additional ones to that list (see below). The HCP is able to translate any allergies of the patient that are already in the list sent from S-EHR and any allergies that have been added afterwards.

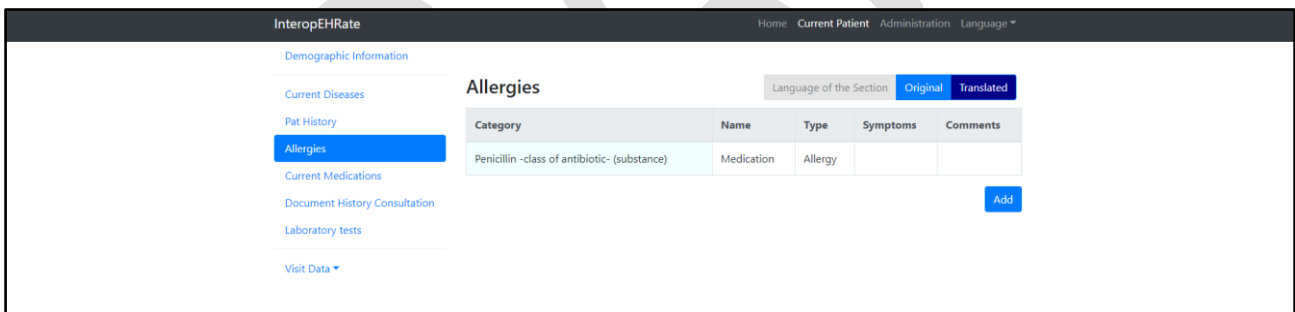


Figure 32: Allergies View Page

Figure 33: Allergies Add page

The Current Medications category contains information about the medication that the patient takes at the moment and information about future meds that will be prescribed by the HCP. To achieve this, the HCP can view, can add, edit, remove, filter and save to database every detail about the medication his/her patient receives. The Translation component of the InteropEHRate is also functional for this category. Sorting is automatically done by status field (active > on-hold > stopped):

Drug Name	Drug Dosage	Timings	Status	Action
Teofillina 200mg	1 tablet	2 times per day	Active	Medication Details

Figure 34: Prescription screen

Every prescription record has a button “Medication Details” that can show more info about the specific medication:

The screenshot shows the InteropEHRate web application interface. At the top, there is a navigation bar with 'Home', 'Current Patient', 'Administration', and 'Language'. Below this, a header bar displays 'Medical visit' and patient information: 'Name: Mario Rossi', 'Age: 60', and 'Country: -'. A sidebar on the left contains navigation links: 'Demographic Information', 'Current Diseases', 'Pat History', 'Allergies', 'Current Medications' (highlighted in blue), 'Document History Consultation', 'Laboratory tests', and 'Visit Data'. The main content area is titled 'Current Medications' and includes a 'Language of the Section' dropdown with 'Original' and 'Translated' options. Below this, there are two tabs: 'Prescription' and 'Medication Statement'. The 'Medication Statement' tab is active, showing a list of medications. The first medication, 'Teofillina 200mg', is highlighted, and its details are displayed in a modal window. The details include: 'Date of Prescription' (2019-03-01), 'Author' (Ion Popescu), 'Drug Dosage' (1 tablet), 'Timings' (2 times per day), 'Start date' (2020-01-16), 'End date (optional)' (2020-01-20), 'Notes' (Assumere durante i pasti), and 'Status' (Active). A 'Back' button is located at the bottom right of the modal window.

Figure 35: Current Medications

Figure 36: Add page for Prescription

Drug Name	Drug Dosage	Timings	Status	Action
Teofillina 200mg	1 tablet	2 times per day	Active	Medication Details

Drug Name	Drug Dosage	Timings	Status	Actions
Aspirin	1 tablet	2 times per day	Active	Medication Details Edit Delete

Figure 37: Added Prescription in a separated table

After the new prescription record was added in the HCP local database, it is displayed in a separate table to be ready for upload back to S-EHR mobile application. To send the new added prescription to the S-EHR app, the practitioner has to click on the “Send to S-EHR” button in the bottom left corner (see picture

above). After the prescription is successfully uploaded into S-EHR through the D2D library, it is automatically added in the prescription table (see picture below):

InteropEHRate Home Current Patient Administration Language ▾

Medical visit Name: Mario Rossi Age: 60 Country: -

Demographic Information

Current Diseases

Pat History

Allergies

Current Medications

Document History Consultation

Laboratory tests

Visit Data ▾

Current Medications

Prescription Medication Statement

Language of the Section Original Translated

Filter ▾

Drug Name	Drug Dosage	Timings	Status	Action
Teofillina 200mg	1 tablet	2 times per day	Active	Medication Details
Aspirin	1 tablet	2 times per day	Active	Medication Details

Add

Figure 38: Uploaded Prescription to S-EHR

The Prescription screen supports filtering data by Drug Name field.

Inside the Document History Consultation category, the HCP can see different documents that are related to the patient's health. A filtering option is available as well. This option allows to filter data by the speciality (i.e. Cardiology or Psychiatry).

InteropEHRate Home Current Patient Administration Language ▾

Demographic Information

Current Diseases

Pat History

Allergies

Current Medications

Document History Consultation

Laboratory tests

Visit Data ▾

Document History Consultation

Language of the Section Original Translated

Filter by Speciality ▾

Date	Location	Exam	View
02/08/2020 10:51 AM	Bucharest	Visit	Open
05/06/2020 10:31 AM	Timisoara	Visit	Open

Figure 39: Document History Consultation

One of the categories that the final users considered as highly important is the Laboratory tests. In this category, the HCP can see any tests for a patient; this includes blood tests, RMI, X-rays and other laboratory

tests. For the blood tests, an abnormality tooltip will contain the normal limit value(s) of these analyses. A colour code system has been also introduced, with red signifying that the analysis is not within the normal limits. Another important component that has been designed in this stage is the image viewer of the Imagistics Laboratory component.

InteropEHRate Home Current Patient Administration Language ▾

Demographic information

Current Diseases

Pat History

Allergies

Current Medications

Document History Consultation

Laboratory tests

Visit Data ▾

Laboratory tests Language of the Section Original Translated

Laboratory results ▾ Pathology results ▾ Radiology results ▾ Other results

Observation (Results: laboratory)

Analysis	6/18/19, 9:00 AM	6/25/19, 8:30 AM
Albumina	4.7 g/dL	4.0 g/dL
Ampiezza curva distribuzione eritrociti (RDW SD)	39.3 fL	39.1 fL
Ampiezza distribuzione piastrine (PDW)	13.3 fL	12.9 fL
Ampiezza distribuzione volume eritrocitario (RDW CV)	13.5 %	13.3 %
Azotemia	41.0 mg/dL	43.0 mg/dL
BNP	36.0 ng/L	- 30 - 228
Basofili	0.02 x10 ³ /µL	0.04 x10 ³ /µL
Basofili %	0.2 %	0.6 %
Bilirubina Totale Reflex	0.7 mg/dL	0.2 mg/dL
Calcemia	9.8 mg/dL	9.5 mg/dL
Concentrazione corpuscolare media HGB (MCHC)	34.2 g/dL	34.0 g/dL
Contenuto corpuscolare medio HGB (MCH)	27.6 Pg	27.4 Pg

Figure 40: Laboratory Results, Observation blood tests

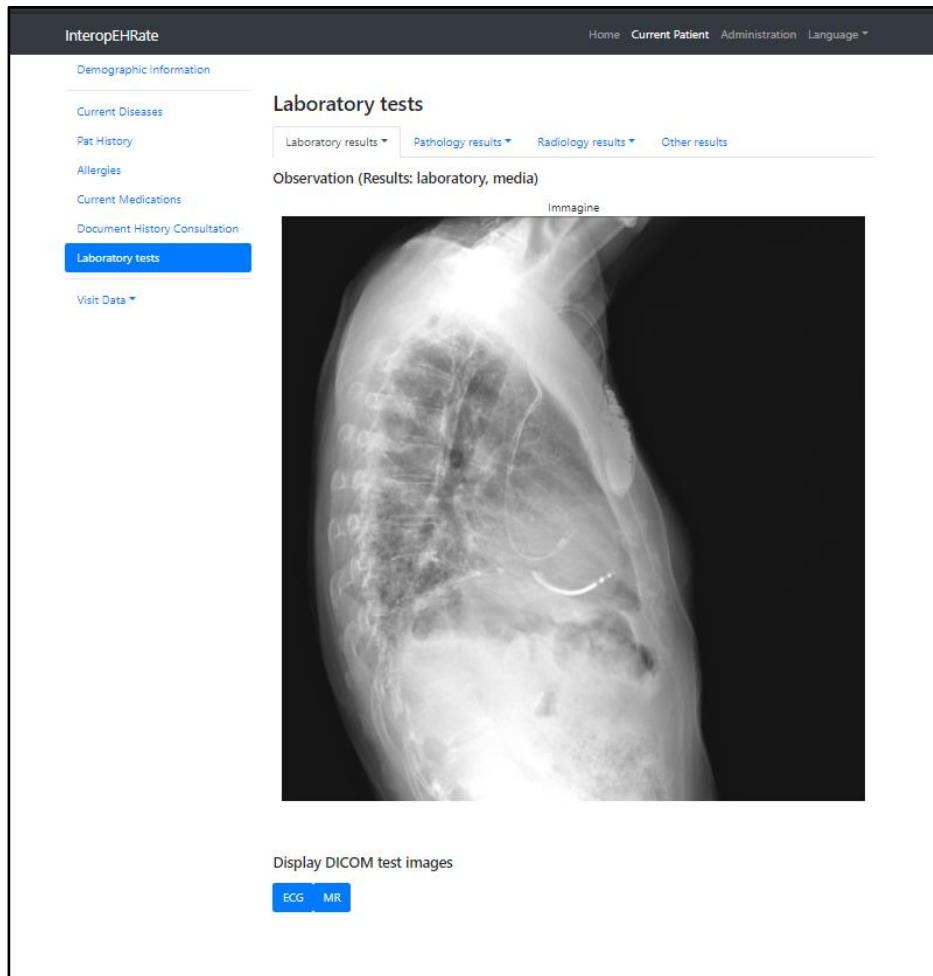


Figure 41: Laboratory Results, X-ray Image Report

The HCP app has implemented Weasis to allow the practitioner to access DICOM images. Below are presented two images accessible in Weasis:

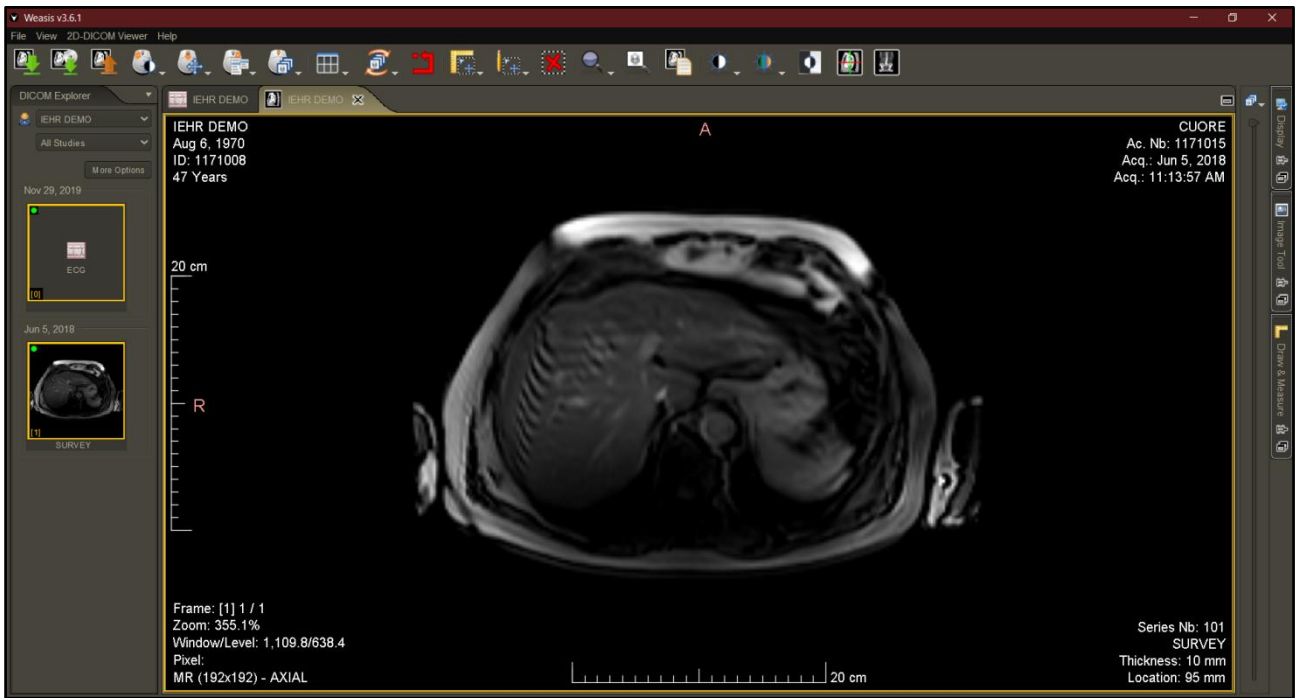


Figure 42: MRI (Magnetic Resonance Imaging) in Weasis application



Figure 43: ECG result in Weasis application

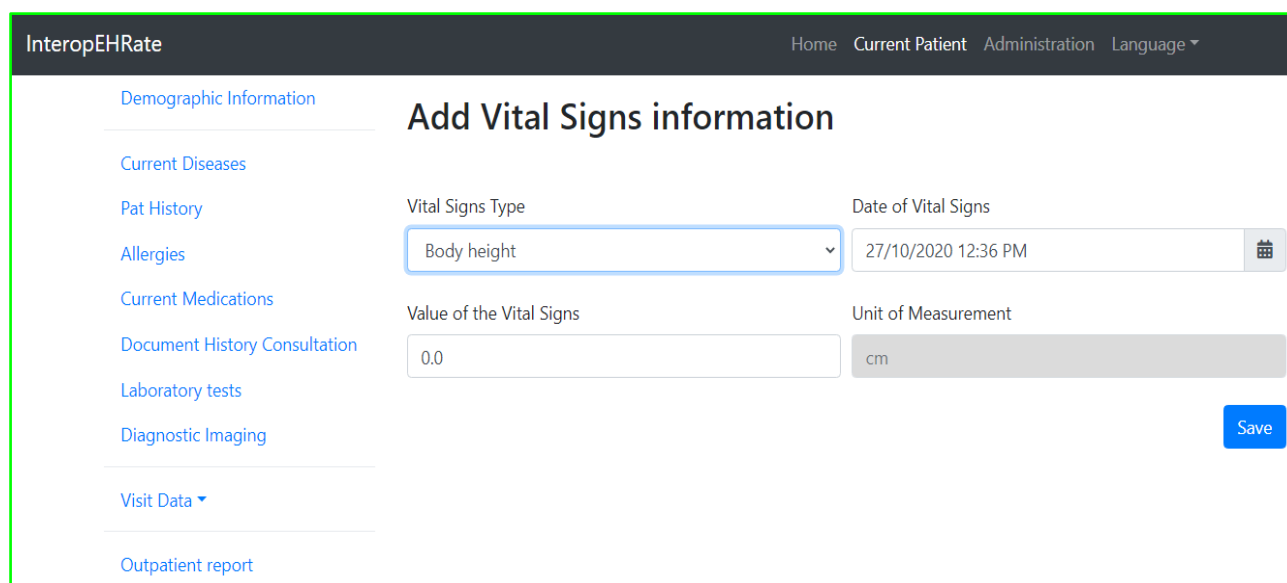
The last category that has been implemented in this stage inside the Visit Details the Vital Signs category. Inside this category, the HCP can register in HCP app the patient's vital sign values taken during the

consultation. This data includes heart rate, blood pressure, oxygen pressure etc. The HCP can view, add and translate data from the Vital Signs category.

Figure 44: Vital Signs

Figure 45: Add Vital Signs

Following the resumption of the co-design sessions, in the Vital Signs section the type of vital signs was synchronized with the unit of measurement and unit of measurement was completed automatically when selecting a Vital Sign type.



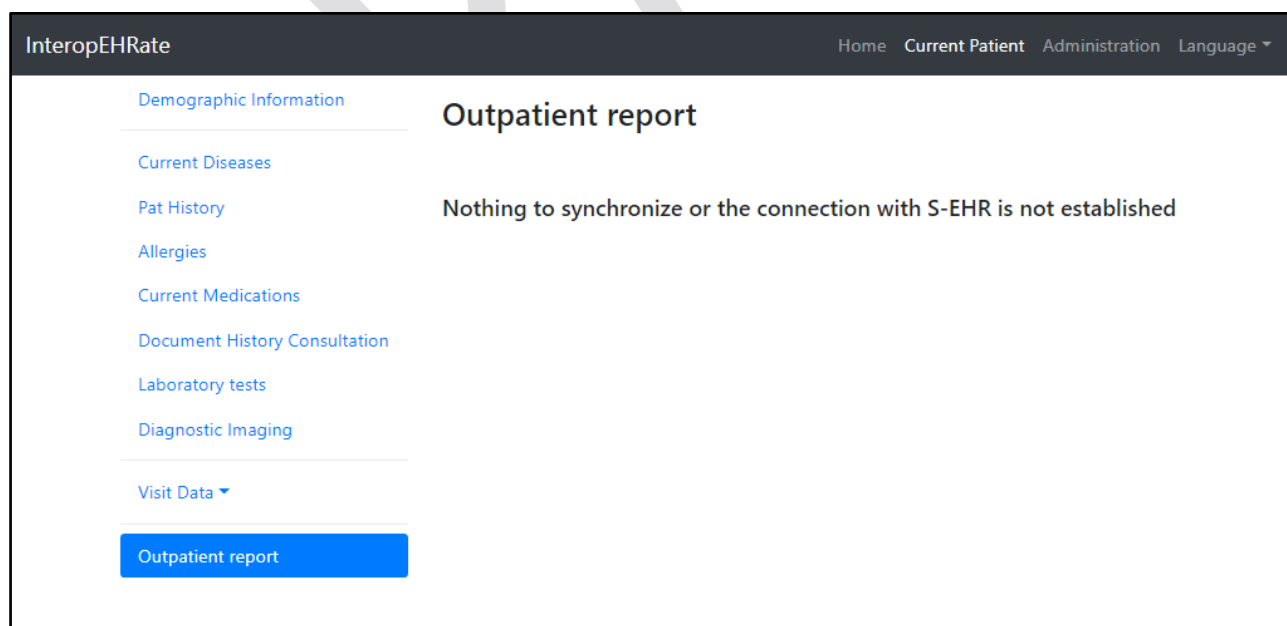
The screenshot shows the 'Add Vital Signs information' form in the InteropEHRate web application. The form is titled 'Add Vital Signs information' and is located in the 'Current Patient' section. The form contains the following fields:

- Vital Signs Type:** A dropdown menu with 'Body height' selected.
- Date of Vital Signs:** A date and time input field showing '27/10/2020 12:36 PM'.
- Value of the Vital Signs:** A text input field containing '0.0'.
- Unit of Measurement:** A dropdown menu with 'cm' selected.

A 'Save' button is located at the bottom right of the form. The left sidebar contains a list of menu items: Demographic Information, Current Diseases, Pat History, Allergies, Current Medications, Document History Consultation, Laboratory tests, Diagnostic Imaging, Visit Data, and Outpatient report.

Figure 46: Add Vital Signs Information details

In addition to the left hand side Current Patient menu, a new entry is created, according with the latest co-design sessions, called Outpatient Report:



The screenshot shows the 'Outpatient report' screen in the InteropEHRate web application. The screen is titled 'Outpatient report' and is located in the 'Current Patient' section. The main content area displays the message: 'Nothing to synchronize or the connection with S-EHR is not established'. A blue button labeled 'Outpatient report' is located at the bottom left of the screen. The left sidebar contains a list of menu items: Demographic Information, Current Diseases, Pat History, Allergies, Current Medications, Document History Consultation, Laboratory tests, Diagnostic Imaging, Visit Data, and Outpatient report.

Figure 47: Outpatient report screen with no data to synchronize

The final users requested the synchronization with S-EHR to be done from a single place, so this screen is created for the final synchronization with all new records entered by the HCP from all sections. Currently above is the screen with no new data added by the HCP, so a message according to it is displayed.

Below can be found a screenshot with the new data added by the healthcare professional:

InteropEHRRate Home Current Patient Administration Language ▾

Demographic Information

Current Diseases

Pat History

Allergies

Current Medications

Document History Consultation

Laboratory tests

Diagnostic Imaging

Visit Data ▾

Outpatient report

Outpatient report

Prescription to upload in S-EHR

Drug Name	Drug Dosage	Timings	Status	Action
Aspirine	1 tablet	2 times per day	Active	Medication Details

Vital signs to upload in S-EHR

Analysis	10/27/20, 12:56 PM
Body Weight	80.0 kg
Heart rate	75.0 /min

Nothing to synchronize or the connection with S-EHR is not established

Figure 48: Outpatient report screen with data to synchronize but with no connection to the S-EHR mobile app

In the above case, the HCP forgot to make the connection with S-EHR first, so the info message is still displayed on the page.

After the connection is made, the screen displays the Synchronization button:

InteropEHRRate Home Current Patient Administration Language ▾

Outpatient visit Name: Mario Rossi Age: 60 Country: -

Demographic Information

Current Diseases

Pat History

Allergies

Current Medications

Document History Consultation

Laboratory tests

Diagnostic Imaging

Visit Data ▾

Outpatient report

Outpatient report

Prescription to upload in S-EHR

Drug Name	Drug Dosage	Timings	Status	Action
Aspirine	1 tablet	2 times per day	Active	Medication Details

Vital signs to upload in S-EHR

Analysis	10/27/20, 12:56 PM
Body Weight	80.0 kg
Heart rate	75.0 /min

Send to S-EHR

Figure 49: Outpatient report screen with data to synchronize

By hitting the “Send to S-EHR” button, the new data is transferred to the mobile phone and the screen cleans itself.

Another implemented feature addresses the possibility to upload certificates for Organization and practitioner, as presented below:

InteropEHRRate Home Current Patient Administration Language ▾

Organization

Practitioner

Initial S-EHR download

Audit Information

Health care organization

Code
SCUBA

Name
Spitalul Clinic de Urgenta Bagdasar-Arseni

Phone
+4021 334 30 25 / +4021 334 30 27 / +4021 334 30 26

Address
Romania, Bucharest, Șoseaua Berceni, 12, Sector 4, 041915

Upload Certificate

Figure 50 : Practitioner details and health care organisation

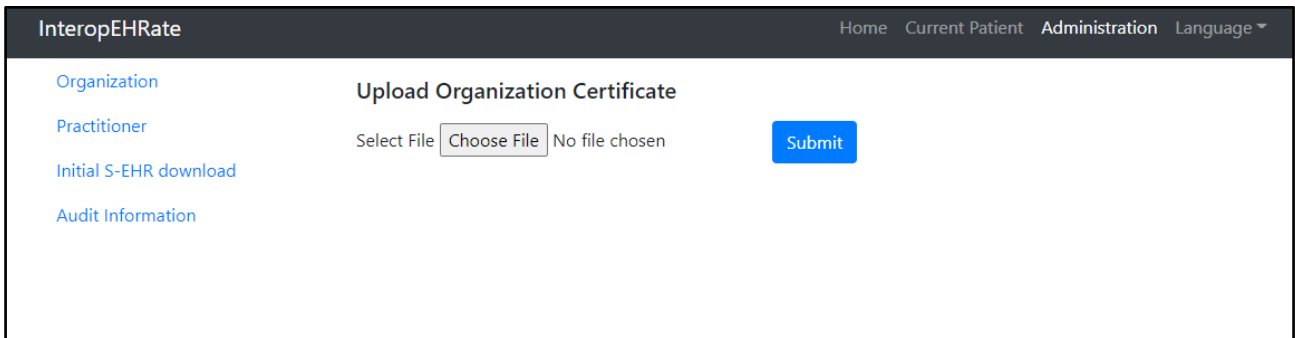


Figure 51: Upload Organization Certificate

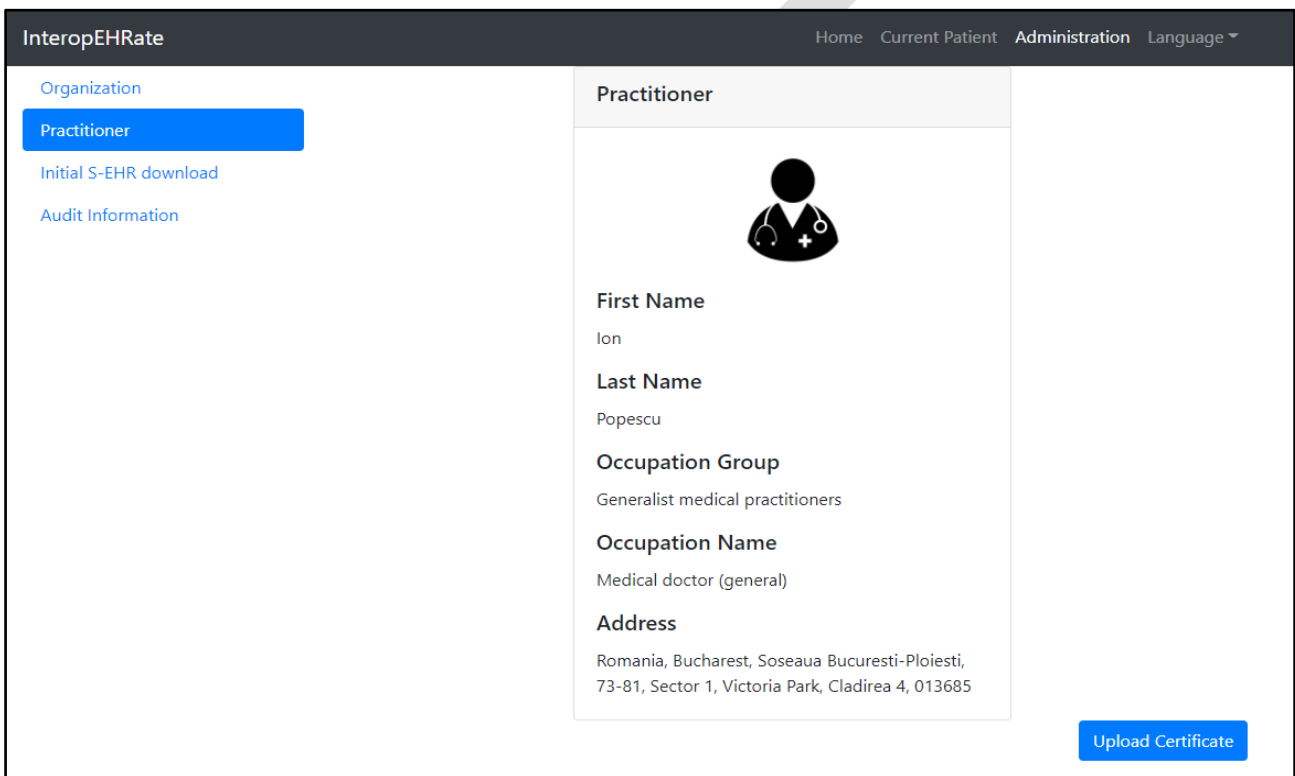


Figure 52: Practitioner details

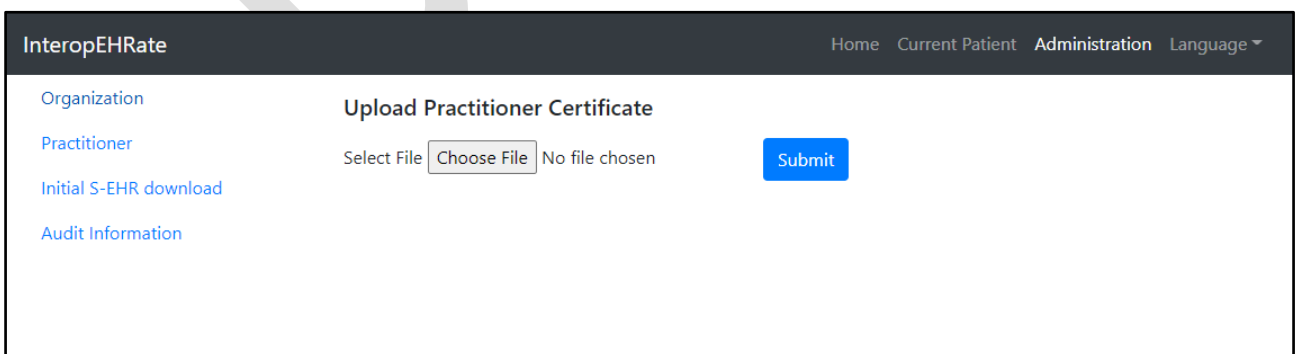


Figure 53: Upload Practitioner Certificate

Following the resumption of the co-design sessions, it was decided to add a nomenclature in the Administration section for Vital Signs. A Vital Sign can be added only by the administrator.

InteropEHRate
Home [Current Patient](#) [Administration](#) [Language](#) ▾

- [Organization](#)
- [Practitioner](#)
- [Initial S-EHR download](#)
- [Audit Information](#)
- Vital Signs Nomenclature

Vital Signs Nomenclature

Name	Comment	Unit of measure	Range
FC	Frequenza cardiaca	bpm	0-300
FR	Frequenza respiratoria	att/min	0-60
Sat Oxy	Oxygen saturation in Arterial blood by Pulse oximetry	%	0-100
Weight	Weight	kg	
Heart rate	Heart rate	/min	
SpO2	SpO2	%	
Mean Arterial pressure	Mean Arterial pressure	mm[Hg]	
Body height	Body height	cm	
Creatinine	Creatinine	mg/dL	
Systolic Blood Pressure	Systolic Blood Pressure	mmHg	0-300
Body Weight	Body Weight	kg	0-300
Diastolic Blood Pressure	Diastolic Blood Pressure	mmHg	0-300
Glucose [Mass/volume] in Serum or Plasma	Glucose [Mass/volume] in Serum or Plasma	mg/dL	

Add vital sign

Name

Comment

Unit of measure

Range

Figure 54: Nomenclature administration

Implementing Translation and conversion services

Translation and conversion services are developed and designed by UNITN. For the *translate function* there are two separate services:

- Concept Translation service -> the first service provided by UNITN that can translate from Italian to English some labels from the json format which has the “coding” field and can support “extension” field;
- Machine Translation service -> implement a second service which is more advanced and can translate any missing labels that haven't the “coding” field in the json format of data received from S-EHR mobile app or S-EHR Cloud.

The conversion feature can be realized by the Codes Conversion Service also designed by UNITN.

In the HCP application, there is a group of buttons on every section with two options available:

- Original: means that the data (from S-EHR mobile app or S-EHR Cloud) displayed in the HCP app is not modified by any translation or conversion service;
- Translated: means that the data is translated and converted before being displayed in the application interface.

In the picture below can be seen this group of buttons with the Original button pressed:

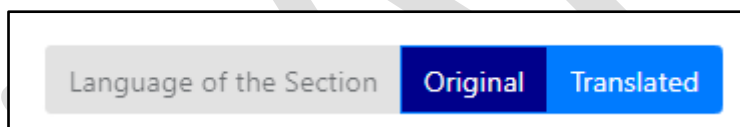


Figure 55: Group of buttons for translation and conversion services (Original selected)

Currently in the HCP app, the Translated button translates and also converts the data. This can be a little bit confusing for now. But in the future co-design sessions, the group of buttons and the translation and conversion services will be discussed.

A solution to this problem might be the changing of Translated button name into something more abstract that includes both translation and conversion functions (i.e. Localized).

Another solution might be the creation of an extra button called Converted that only converts the data. This means that the Translated button will only translate the data. So the group of buttons will have 3 options: Original, Translated and Converted (ON/OFF).

Below, is shown an example on how the translation and conversion services are used in the HCP application:

The screenshot shows the 'Current Diseases' section of the InteropEHRate application. The patient's name is Mario Rossi, age 60, and country is unknown. The 'Language of the Section' is set to 'Original'. The table below shows one entry:

Code	Disease	Date of Diagnosis	Comment
http://hl7.org/fhir/sid/icd-9:425.4	Altre Cardiomiopatie Primitive	30-03-2017	

There is an 'Add' button to the right of the table and a 'Save' button at the bottom. A 'Notes' text area is also present.

Figure 56: Current Diseases screen with original (selected) content received

The screenshot shows the 'Current Diseases' section of the InteropEHRate application. The patient's name is Mario Rossi, age 60, and country is unknown. The 'Language of the Section' is set to 'Translated'. The table below shows one entry:

Code	Disease	Date of Diagnosis	Comment
icd10:i42.5	Other restrictive cardiomyopathy	30-03-2017	

There is an 'Add' button to the right of the table and a 'Save' button at the bottom. A 'Notes' text area is also present.

Figure 57: Current Diseases screen with translated (selected) and converted content received

The code conversion can be seen in the first column (from ICD 9 to ICD 10). Also the translation can be seen in the second column of the table (from Italian to English).

Note that in the actual Current Diseases section, according to the co-design sessions, the Code column is hidden. The Code column is visible in the above pictures only to show the Codes Conversion Service in action.

6. AUTOMATED TESTING

In software application development, automated testing represents a testing technique through which the results generated by a certain module (actual) are compared with known results (expected). To achieve this, test scripts are developed and are then executed by specialized applications, called automated testing tools. Test automation is useful when certain test activities are performed repeatedly to ensure that the functionalities already developed remain correct when changes are made to adjacent modules.

6.1. Unit testing - Continuous Integration with Gitlab Runner, Junit and Maven

The HCP application uses Gitlab Runner as an automated testing method. This software automatically runs all the unit tests developed in the HCP app when a push is done on GitLab. The tests from the HCP app are created using JUnit and Mockito frameworks.

JUnit is a framework for writing unit tests for applications developed using java. It is part of the xUnit architecture for unit testing frameworks.

Mockito is an open source mocking framework for Java applications. Mockito is used to mock implementations for some functionality in order to facilitate testing especially when accessing external resources to the application. In GitLab is a component responsible for executing jobs, on the developer's machine, with the transmission of results to the server. GitLab Runner is closely related to GitLab Continuous Integration / Continuous Deployment in order to automatically execute unit and integration tests at each push operation, as specified above.

The following picture shows the runners currently registered in GitLab for the HCP application:

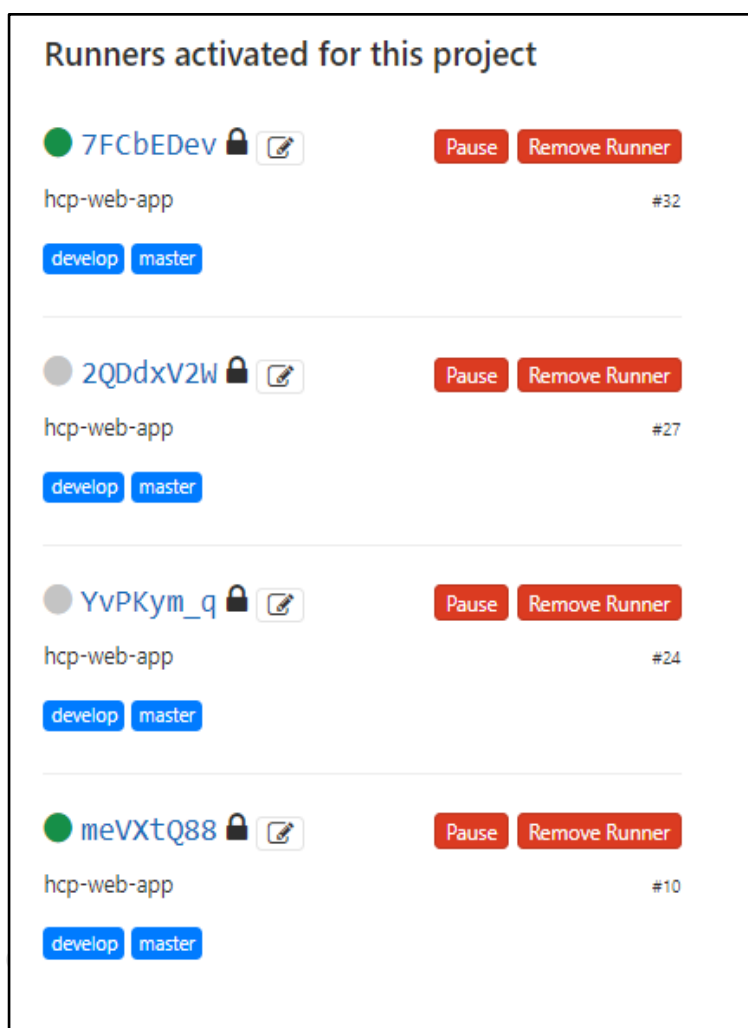
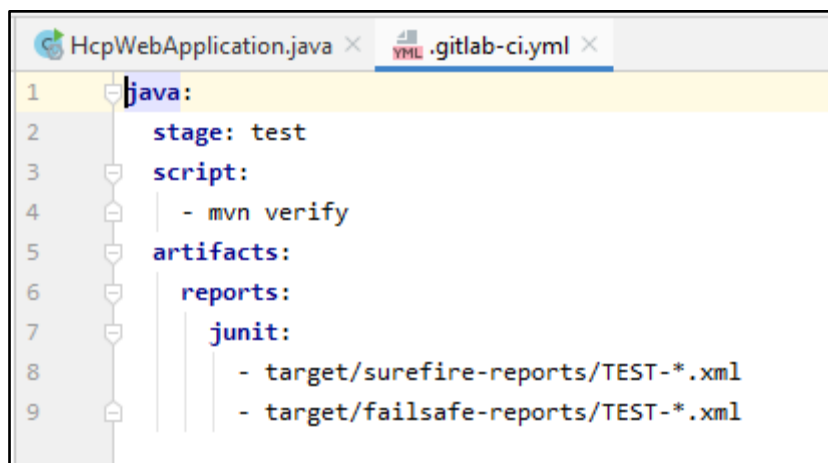


Figure 58: HCP app Runners

To configure the execution of unit tests with GitLab runner for HCP Application the following steps are required (the example is for a development environment on a windows platform):

- Download and install the GitLab Runner client;
- Register the gitlab-runner client in the GitLab server:
- Create and add .gitlab-ci.yml file in the application's project.

The content of the gitlab-ci.yml file for the HCP application is shown below.



```

1  java:
2    stage: test
3    script:
4      - mvn verify
5    artifacts:
6      reports:
7        junit:
8          - target/surefire-reports/TEST-*.xml
9          - target/failsafe-reports/TEST-*.xml

```

Figure 59: YML configuration for HCP app

The GitLab runner will execute the command `mvn verify`. Verify is the fifth phase of the build lifecycle which will also trigger the test phase as illustrated below.

- `validate` - validate the project is correct and all necessary information is available
- `compile` - compile the source code of the project
- `test` - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
- `package` - take the compiled code and package it in its distributable format, such as a JAR.
- `verify` - run any checks on results of integration tests to ensure quality criteria are met
- `install` - install the package into the local repository, for use as a dependency in other projects locally
- `deploy` - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

Figure 60: Build lifecycle phases

Hereafter is a screenshot from GitLab with the console output after an execution of Maven command triggered by GitLab runner; 14 tests were run, all successfully:

```
2020-04-12 23:54:01.129 INFO 21484 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2020-04-12 23:54:01.134 INFO 21484 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 14, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.1.2:jar (default-jar) @ hcp-web-app ---
[INFO] Building jar: C:\my-program-files\gitlab-runner\builds\z3SHLpeb\0\interopehrate\reference-hcp-app\hcp-app\target\hcp-web-app-1.0.0-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.2.4.RELEASE:repackage (repackage) @ hcp-web-app ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:00 min
[INFO] Finished at: 2020-04-12T23:54:04+03:00
[INFO] -----
▼ Saving cache
▼ Uploading artifacts for successful job
Version: 12.9.0
Git revision: 4c96e5ad
Git branch: 12-9-stable
GO version: go1.13.8
Built: 2020-03-20T13:02:39+0000
OS/Arch: windows/amd64
Uploading artifacts...
Runtime platform arch=amd64 os=windows pid=15568 revision=4c96e5ad version=12.9.0
target/surefire-reports/TEST-*.xml: found 6 matching files
WARNING: target/failsafe-reports/TEST-*.xml: no matching files
Uploading artifacts to coordinator... ok id=228 responseStatus=201 Created token=7D1xjt0m
Job succeeded
```

Figure 61: Gitlab Runner output console

6.2. User interactions scripts with Selenium

Selenium is an automated testing framework designed to check web applications across multiple browsers.

Selenium uses *test scripts* that can be written in multiple programming languages like Java, Python, C#, PHP etc. and then integrated for automated execution with Junit, Maven, TestNG or Jenkins.

By performing tests using Selenium, the HCP App will be checked for proper operation on different browsers (cross-browser) and browser versions.

In this stage of project implementation, the HCP App is not tested with Selenium, the start of the development of such tests being planned for the 4th quarter of 2020.

7. CONCLUSIONS AND NEXT STEPS

The current deliverable aims to present the updates and improvements of UI design specifications of HCP Web App solution used by healthcare professionals for accessing and creating health data of foreign patients within the InteropEHRate project.

The specific design of the HCP Web App solution was drafted based on the results obtained within the previous deliverable D5.4 [22]. At this particular stage of project implementation, the deliverable presents the relevant features and principles of designing the HCP Web App solution, addressing essentially the import/export data directly from/to the S-EHR on the smartphone. Considering the particular requirements of the project tasks focusing on HCP web app and EHR functionalities, the deliverable encompasses the architectural design specifications and the current relevant elements of UI design of HCP Web App that will then be detailed and updated incrementally in the next deliverable [D5.6].

Within the deliverable, HCP Web App is depicted from four major perspectives:

- Methodological view
- Architectural view
- Technical view
- End User view.

The deliverable presents the current design characteristics and features of HCP Web App, while the details and iterative completions of the solution design, following a similar structure of content, will be described in the next deliverable and [D5.6].

REFERENCES

- [1] **[AHMED 2005]** Ahmed Seffah, Jan Gulliksen, Michel C. Desmarais. "Human-Centered Software Engineering – Integrating Usability in the Software Development Lifecycle", 2005, Springer, <https://books.google.ro/>.
- [2] **[DEAN 2010]** Dean Leffingwell. "Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise", [Addison-Wesley Professional](#) 2010.
- [3] **[MARK 2001]** Mark Grand. "Java Enterprise Design Patterns: Patterns in Java – Wiley" 2001.
- [4] **[DEAN 2006]** Dean Leffingwell, Ryan Martens, Mauricio Zamora.. "Principles of Agile Architecture - Intentional Architecture in Enterprise-Class Systems", www.rallydev.com.
- [5] **[Apache]** Apache Maven, [Introduction to the Dependency Mechanism in Maven](#). <https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html>
- [6] **[FHIR]** FHIR implementation used in the project <https://hapifhir.io/>
- [7] **[Wikipedia]** Java Persistence API. https://en.wikipedia.org/wiki/Jakarta_Persistence
- [8] **[Wikipedia]** Wikipedia contributors. (2019, October 7). Software repository. In *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Software_repository&oldid=920020874
- [9] **[Wikipedia]** Wikipedia contributors. (2019, September 29). Model–view–controller. In *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/w/index.php?title=Model%E2%80%93view%E2%80%93controller&oldid=918659027>
- [10] **[Wikipedia]** Wikipedia contributors. (2019, April 25). Data access object. In *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Data_access_object&oldid=894006019
- [11] **[Wikipedia]** Wikipedia contributors. (2018, February 26). Data transfer object. In *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Data_transfer_object
- [12] **[Wikipedia]** Wikipedia contributors. (2019, September 23). Command (computing). In *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/wiki/Command_\(computing\)](https://en.wikipedia.org/wiki/Command_(computing))
- [13] **[Agile]** Agile Modelling - Effective Practices for Modeling and Documentation. <http://www.agilemodeling.com>
- [14] **[Agile Business Consortium]** <https://www.agilebusiness.org/page/Papers>
- [15] **[D2.1]** InteropEHRate Consortium, User Requirements for cross-border HR integration - V1, 2019. <https://www.interopehrate.eu/resources/>

- [16] [D2.2] InteropEHRate Consortium, User Requirements for cross-border HR integration – V2, 2019. <https://www.interopehrate.eu/resources/>
- [17] [D2.7] InteropEHRate Consortium, FHIR profile for EHR interoperability - V1 (conceptual level specification, implementable level specification), 2019. <https://www.interopehrate.eu/resources/>
- [18] [D2.8] InteropEHRate Consortium, FHIR profile for EHR interoperability - V2 (conceptual level specification, implementable level specification), 2020. <https://www.interopehrate.eu/resources/>
- [19] [D4.4] InteropEHRate Consortium, Design of libraries for remote D2D HR exchange - V1, 2019. <https://www.interopehrate.eu/resources/>
- [20] [D5.1] InteropEHRate Consortium, Software Requirements specification of an integrated EHR web app for HCP - V1, 2019. <https://www.interopehrate.eu/resources/>
- [21] [D5.2] Software requirements specification of an integrated EHR web app for HCP - V2, 2020. <https://www.interopehrate.eu/resources/>
- [22] [D5.4] Design of an integrated EHR web app for HCP – V1, <https://www.interopehrate.eu/resources/>
- [23] [D5.9] Design data mapper and converter to FHIR - V1, <https://www.interopehrate.eu/resources/>
- [24] [D5.10] InteropEHRate Consortium, Design of the data mapper and converter - v2, 2020. <https://www.interopehrate.eu/resources/>
- [25] [D5.11] InteropEHRate Consortium, Design of the information extractor and natural language translator - v1, 2020. <https://www.interopehrate.eu/resources/>
- [26] [DICOM] — Digital Imaging and Communications in Medicine <https://www.dicomstandard.org/>
- [27] [WEASIS] WEASIS MEDICAL VIEWER <https://nroduit.github.io/en/>
- [28] [GitLab] GitLab Runner documentation <https://docs.gitlab.com/runner/>