D5.4

Design of an integrated EHR web app for HCP – V1

ABSTRACT

This deliverable presents the preliminary relevant aspects 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 D5.1 - Software requirements specification of an integrated EHR web app for HCP - V1 [D5.1] and reflects, at this stage of project implementation, the most important initial design features addressing the import / export data directly from/to the S-EHR on the smartphone.

Delivery Date	11th October 2019
Work Package	WP5
Task	T5.1
Dissemination Level	Public
Type of Deliverable	Report
Lead partner	SIVECO

InteropEHRate project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106.

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.

CONTRIBUTORS

	Name	Partner
Contributors	Simona Bica, Nicu Jalba, Simona Alexandru, Andrei Madalin Oanca, Adrian Bradu	SIVECO
Contributors	Gabor Bella, Simone Bocca	UNITN
Reviewers	Claudiu Baluta	SCUBA
Reviewers	Sofianna Menesidou	UBIT
Reviewers	Paul de Raeve, Juan Fernandez	EFN

LOGTABLE

Version	Date	Change	Author	Partner
0.1	16-08-19	First draft of table of contents	Simona Bica	SIVECO
0.2	06-09-19	Contribution to Chapter 4.1. Architectural Approach - First 5 architectural components and introduction; First input for Methodological approach	Andrei Madalin Oanca	SIVECO
0.2 (1)	09-09-19	Contribution to Introduction, Context and Methodological approach	Simona Bica	SIVECO
0.2 (2)	10-09-19	Contribution to Chapter 5. Designing the GUI of HCP Web App (Technical compliance insight)	Simona Alexandru	SIVECO
0.2 (3)	12-09-19	Contribution to Chapter 4.3. Interoperability approach and current findings	Gabor Bella, Simone Bocca, Subhashis Das	UNITN

0.2 (3)	13-09-19	Contribution to Chapter 5. Designing the GUI of HCP Web App (Health domain compliance insight) and Chapter 6. Conclusions and next steps	Simona Bica	SIVECO
0.2 (4)	17-09-19, 18-09-19	Final contributions to Chapter 4.1 Architectural approach and Chapter 4.2 Technical approach	Andrei Madalin Oanca, Simona Alexandru, Nicu Jalba	SIVECO
0.2 (5)	17-09-19	Final contributions to sections 4.3 and 3 on Interoperability	Gábor Bella, Simone Bocca, Subhashis Das	UNITN
0.2 (6)	18-09-19	Last contributions and updated version of mock-ups and GUI	Simona Bica, Simona Alexandru	SIVECO
0.3	20-09-19	Deliverable ready for the internal review	Simona Alexandru, Simona Bica, Adrian Bradu	SIVECO
0.4	23-09-2019	Deliverable review	Paul De Raeve, Juan Fernandez	EFN
0.5	27-09-2019	Deliverable review	Juan Fernandez	EFN
0.6	03-10-2019	Integrate comments and suggestions	Simona Bica	SIVECO
0.7	04-10-2019	Internal quality check and completion	Adrian Bradu	SIVECO
0.8	04-10-2019	Deliverable finalized and sent to final review and quality check	Simona Bica	SIVECO
1.0	07-10-2019	Quality check completed	Argyro Mavrogiorgou	UPRC
1.1	8-10-2019	Final review	Laura Pucci	ENG
1.2	10-10-19	Addressed comments	Adrian Badu	SIVECO
Vfinal	10-10-2019	Final version for submission	Laura Pucci	ENG

ACRONYMS

Acronym	Term and definition
API	Application Programming Interface
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.
WP	Work Package
UI	User Interface
IEHR	InteropEHRate
IPS	International Patient Summary
GUI	Graphical User Interface
ICT	Information and Communication Technologies
R&D&I	Research, Development and Innovation
HCP	HealthCare Professional
EHR	Electronic Health Record
S-EHR	Smart HER
D2D	Device to Device protocol
PID	Personal Identification
HTML	Hypertext Mark-up Language
XML	eXtensible Markup Language
W3C	World Wide Web Consortium
CSS	Cascading Style Sheets
HR	Health Record
UCD	User Centred Design
AMDD	Agile Model Driven Development

DAO	Data Access Object
DI	Dependency Injection
IoC	Inversion of Control
MVC	Model-View-Controller
QR code	Quick Response code
ICD-9	International Classification of Diseases, Ninth Revision
ICD-10	International Classification of Diseases, Tenth Revision

TABLE OF CONTENT

LIST OF FIGURES

1. INTRODUCTION

1.1. Scope of the document

This deliverable is the first report, produced by the InteropEHRate project within the activities related to *Incremental EHRs integration*, which covers the preliminary design requirements of HCP Web App solution.

The purpose of this deliverable is to define the design of HCP Web App solution, based on the results obtained within the previous deliverable [D5.1] Software requirements specification of an integrated EHR web app for HCP - V1. 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.

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 to 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: Depicts a representative 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. 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 first of three deliverables that will be produced as a result of the task related to *HCP web app and EHR functionalities,* dedicated to the design of HCP Web App and comprises the preliminary design elements and User Interface (UI) design of HCP Web App that will then be detailed and updated incrementally in the next deliverables - [D5.5] - Design of an integrated EHR web app for HCP - V2 and [D5.6] - Design of an integrated EHR web app for HCP - V3.

2. CONTEXT

2.1. Goals

Within the project implementation, the present deliverable depicts the most relevant aspects concerning the design of HCP Web App, based on the valued results from the project tasks in charge to define the user requirements (scenarios and user requirements presented in deliverable [D2.1] User Requirements for cross-border HR integration - V1) 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 deliverable[D4.4] Design of libraries for remote and D2D HR exchange - V1 concerning the innovative way of designing the data exchange (design of libraries for D2D HR exchange) was also considered in drafting this deliverable.

As a continuation of the preliminary considerations presented in deliverable [D5.1] Software requirements specification of an integrated EHR web app for HCP - V1, the present deliverable 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*.

Within this deliverable, HCP Web App is depicted from four significant perspectives:

- Methodological perspective (addressed in Section 3METHODOLOGICAL 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 deliverable presents the preliminary characteristics and considerations of HCP Web App design, whereas the iterative completions and updates of our design approach will be depicted in the next deliverables [D5.5] Design of an integrated EHR web app for HCP - V2 and [D5.6] Design of an integrated EHR web app for HCP - V3.

The conceptual level specification to design the HCP Web App solution, including the compatibility requirements with HL7 FHIR profile (data sets, terminologies, concepts, domain model, etc...), is based on the scientific approach from the deliverable [D2.7] FHIR profile for EHR interoperability - V1.

The GUI design requirements and the current stage of UI implementation are based on the Scenarios and User Requirements presented in deliverable [D2.1] User Requirements for cross-border HR integration - V1, as well as on the preliminary 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 activities foreseen within the InteropEHRate project related to the *Validation of results* will be also connected with the activities of the task focusing on *HCP web app and EHR functionalities,* producing the current deliverable, because the healthcare partners part of the project Consortium will be involved to explore and exploit the functionalities of the InteropEHRate innovative platform.

3. METHODOLOGICAL APPROACH

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

Agile methodology and the User-Centred Design are iterative approaches used in the implementation of software solutions, including the particular stage of the solution analysis and design. Agile and UCD are iterative approaches which perfectly suit the features and objectives of the InteropEHRate project.

The Agile software development methodology is appropriate for the ICT projects specific to the R&D&I (Research, Development and Innovation) domain, because in this particular field, for clarifying the complex situations to confront and solving various requirements might be adequate to use an iterative process.

In this stage of HCP Web App design, the Agile methodology is applied as an iterative and incremental process in order to draft *the design specifications and particularities* of the application, as they are depicted hereinafter in *Section 4. Design of the HCP Web App technical solution*.

Being a user-driven (opposite to the technology-driven) approach, the UCD methodology presumes the involvement of users from the start and in all phases of the product / service development in an iterative manner. Using this iterative approach, the application will be built incrementally by adding new features and functionality over several iterations, based on the valued end-user feedback. Taking into consideration the specific of our target group, we started 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 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*.

One of the most demanding parts of the methodological approach was to explore the most appropriate methods and techniques that meet the specific requirements of the design stage. As a result of our research, we considered that one specific approach of Agile fits the best the requirements of solution design, namely Agile Model Driven Development (AMDD).

The design of HCP Web App solution is based on the following principles:

- Simplicity
  The meaning of this principle is that it lacks any unnecessary complexity and only is displayed what it needs to do.
- Easy to use
  The meaning of this principle is that an user-friendly and intuitive interface does not require of prior technical knowledge when access / operate it.

- Adaptability to the user specific way of working
  The meaning of this principle is that users are able to customise the application according to their needs / taste.

- Extensibility
  The meaning of this principle is the solution should be designed to allow the addition of new functionalities and capabilities.

3.1 Collaboration on the use of conversion and translation services

The HCP Web App relies on the *InteropEHRate Translation Services* described in deliverables [D5.9] and [D5.11]. These services are used by the HCP Web App to translate the S-EHR to the local language. Furthermore, the HCP Web App also needs to be capable of reading S-EHRs that already use international (non-local) standards and contain translations into multiple languages.

The challenge is that the specifications of the Translation Services and of the detailed S-EHR structure (e.g., how translations are encoded in FHIR) are not yet available at the time of writing this deliverable. They will emerge through an organic process where the various project partners establish the specifications and best practices with respect to the multilingual use of FHIR, something the FHIR standard does not yet explicitly support. Also, the specification and implementation of the Translation Services is work in progress among the activities of the InteropEHRate project in its early stages.

The HCP Web App development methodology with respect to dealing with (reading, displaying) international and translated data has to take into account the limitations above. Therefore, the services to be used and the interoperability-related business logic implemented by the HCP Web App need to proceed in an iterative manner. The first iteration(s) need to be based on a high-level understanding of the complexity of dealing with multiple versions of the same data (original versions, converted versions, translated versions) as presented in detail in Section 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 InteropEHRate 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 that will ease the interaction between the Health Care Provider and the HCP Web App.

As it was presented in [D5.1], 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 InteropEHRate 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.

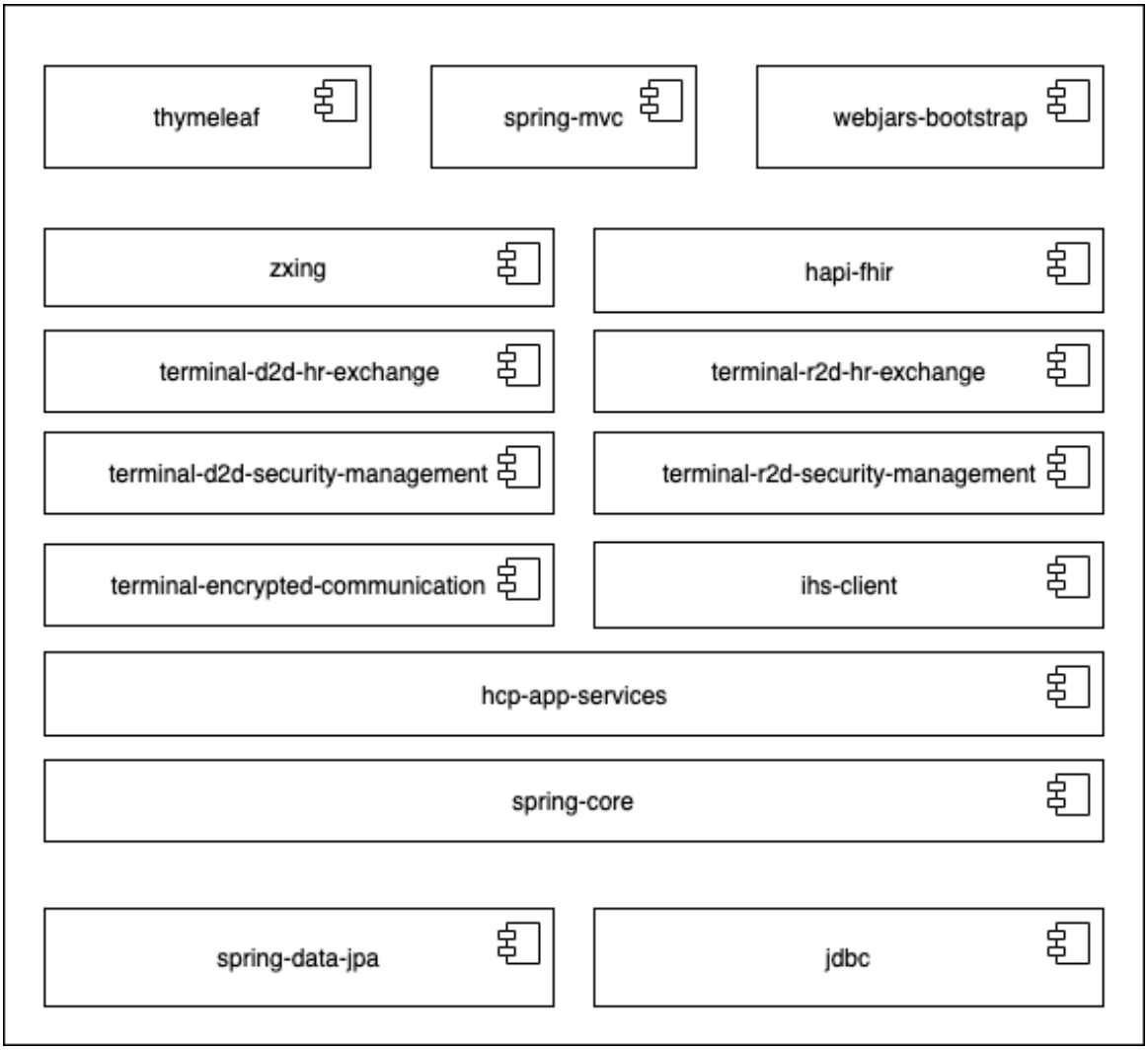


Figure 1 - Class Diagram

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 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 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 InteropEHRate project.

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

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

Hapi-fhir an open-source implementation of the FHIR specification for java applications. This library will be used by all applications developed within InteropEHRate 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 full 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.

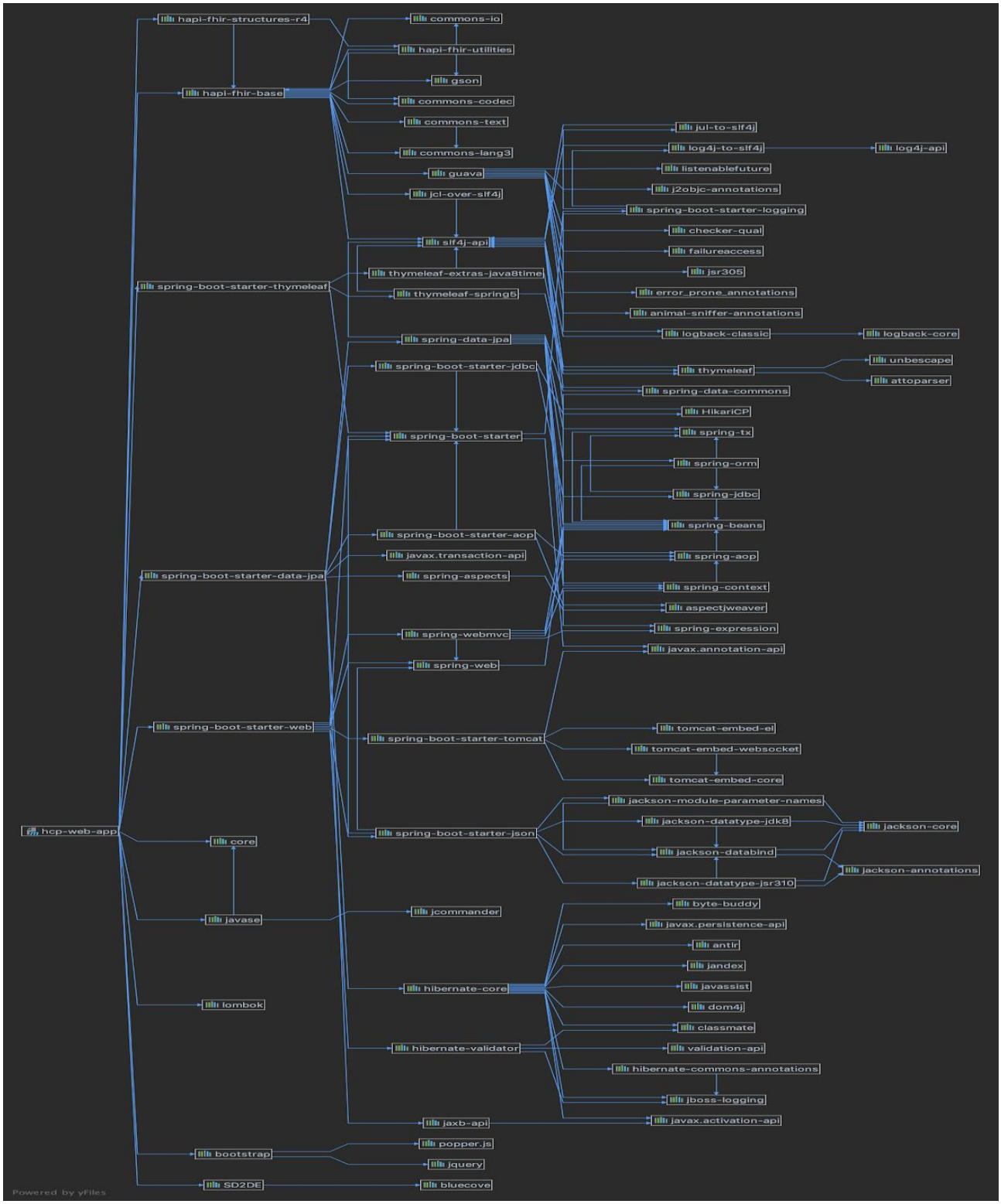


Figure 2 - Dependency tree diagram of the HCP Web App components

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 perspective that were of utmost importance 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). One consideration was of using Kotlin for the development of some modules. Next versions of the deliverable will highlight the parts of the application that are developed in Kotlin.
- 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 (V1) 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 deliverable[1].

[1] September 2019.

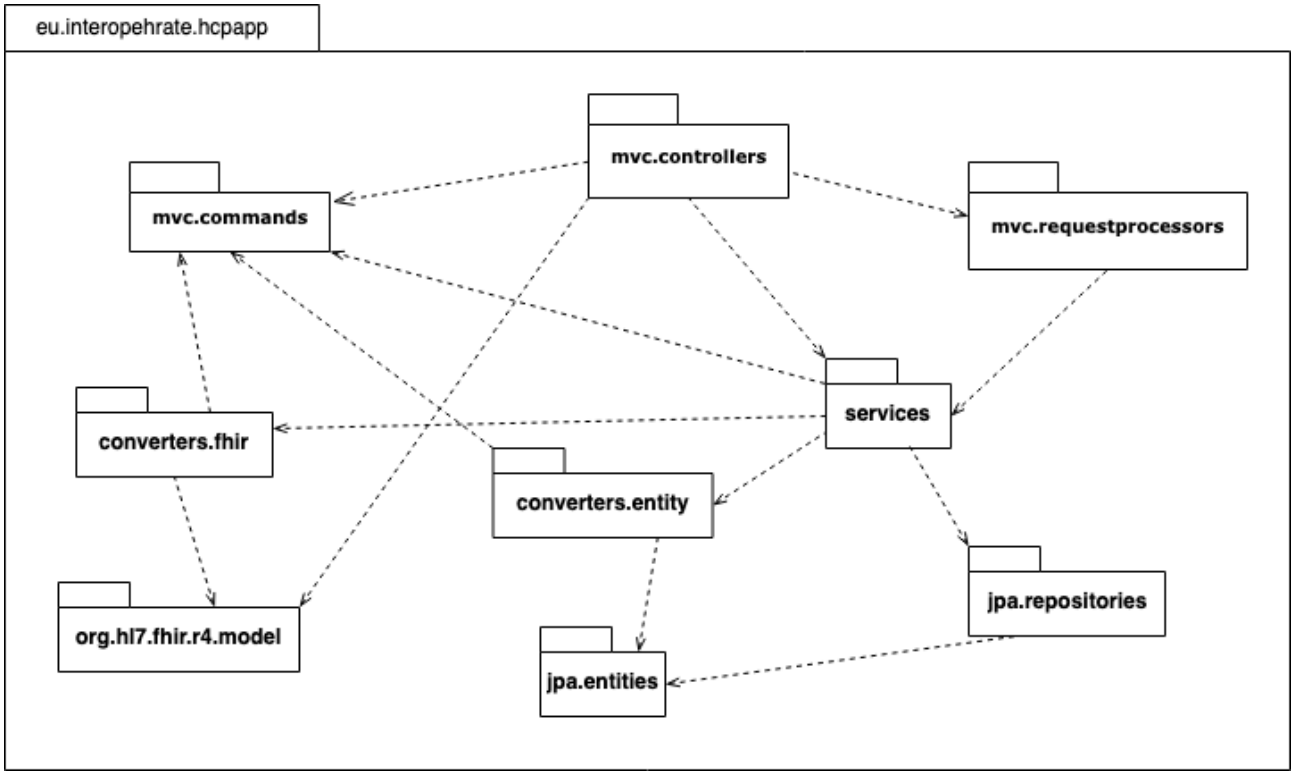


Figure 3 - HCP Web App - Main packages and their relations

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.

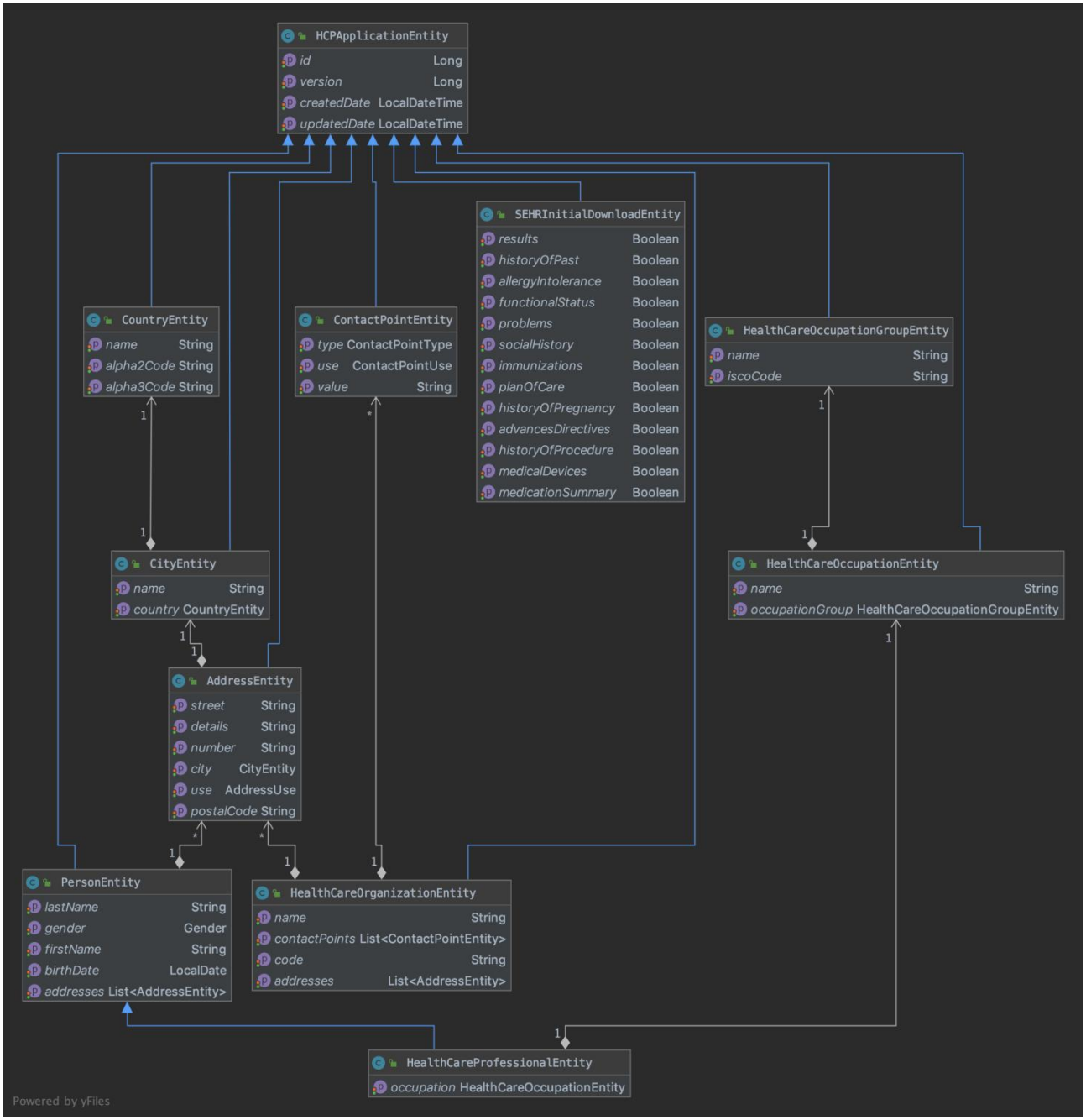


Figure 4 - HCP Web App - Main diagram of the entities involved

- **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;
- **HealthCareOrganizationEntity** - nomenclature with health care occupations.

4.3. Interoperability approach and current findings

4.3.1. Representation of interoperable and multilingual information within the S-EHR

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

- the language used within the S-EHR is different from the language of the HCP;
- the S-EHR uses encoding systems and/or terminology that are locally not used and/or not well known;
- the S-EHR structures the EHR in a way that is (more or less) different from local data schemas and may be more or less understandable to HCPs.

Due to the difficulty of harmonising EHRs across countries, the IEHR specifications foresee three levels of interoperability, in increasing order of complexity:

1. **secure:** the original EHR documents (such as PDF files, images, etc.) are embedded into a SEHR container, and no conversion or translation are applied to them;
2. **syntactic:** the original *structured* (e.g., tabular) EHR is mapped to the FHIR data schema format, 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 a level 3 S-EHR contains the following information:

- the EHR in its intact, original, unstructured, document-based form (always);
- the EHR in its original structured form (whenever available);
- the EHR 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.

- the EHR in FHIR format in a target language.

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

"La diagnosi primaria è l'occlusione e stenosi dell'arteria basilare"

meaning

"the primary diagnosis is occlusion and stenosis of basilar artery."

After conversion to SEHR and translation to English (e.g., because the patient is British and has requested such a translation) the following information will be found within a level-3 semantic S-EHR:

- the original unstructured EHR, in Italian:
  - "La diagnosi primaria è l'occlusione e stenosi dell'arteria basilare";
- the original structured EHR, in Italian:
  - diagnosi_primaria = 433.00 (using ICD-9 coding, which may be implicit);
- the International (structured) EHR, in Italian:
  - fhir:Condition.condition = I65.1 (using ICD-10), annotated as "Occlusione e stenosi dell'arteria basilare";
- the International EHR, in English:
  - fhir:Condition.condition = I65.1 (using ICD-10), annotated as "Occlusion and stenosis of basilar artery".

The precise way these three versions are represented is yet to be defined, as the current FHIR 4.0.0 specification does not provide adequate support for representing multilingual information. One of the following approaches may be used:

- as FHIR provides explicit support for languages on the resource level only, a solution is to duplicate FHIR resources (e.g., fhir:Condition), each instance providing all information on a given language (in the example above, there would be one Condition resource in Italian and one in English);
- specify FHIR extensions on the level of individual attributes, allowing the representation of attribute values in multiple languages.

4.3.2. Reading and producing interoperable S-EHRs from the HCP Web App

The HCP Web App receives, using the D2D or R2D protocols, a S-EHR that was extracted from another hospital and converted to the international FHIR-based format as described in the section above. This S-EHR will always contain the original version(s) of the EHR in the original language, and may also contain translations of it to other languages. Typically, 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 a high enough quality.

For this reason, upon reception of the SEHR, the HCP Web App needs to check whether it needs to be translated to the local language, and if this is the case, it needs to use the *Translation Services* within *InteropEHRate Health Services*, to be defined in deliverable [D5.11]. The Translation Services will add a

new, translated version of the EHR to the S-EHR container, which the HCP Web App can read and display for the local HCPs.

Since data values go through automated conversions (to international standards) and translations, they may be incorrect with a certain probability, even if they are assumed to be of generally high quality. For this reason, it is crucial for the HCP Web 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.

The S-EHR will contain a metadata flag that will indicate for each FHIR data value whether it went through a conversion or a translation process. The HCP App can then take display decisions based on the value of this flag.

5. DESIGNING THE GUI OF HCP WEB APP

5.1. End-user viewpoints and perspectives

The most significant and intuitive perspective concerning the communication between HCP Web App and S-EHR app is the visual illustration of different features of HCP Web App accessible to healthcare professionals. In this context, the section presents the actual stage of designing the GUI of HCP Web App, based on the end user valued input.

As presented above in *Section 3. Methodological approach*, we started 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.

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 in this stage of HCP Web App design, 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 users' requirements (V1 selection related to HCP app), as they were described in deliverable *[D2.1]* - *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.
    - An HCP can view the Patient Summary (a portion of it) shared with him/her by a Citizen using the HCP app.
    - An HCP 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 the "look and feel" display mode (preliminary relevant data sets and corresponding fields, specific for the patient health records) in ***HCP Web App*** according to the HL7 FHIR requirements, as they were described in the deliverable *[D2.7] (Interoperability profile – conceptual level specification) - Chapter 4. Specification of concepts*, namely:
    - IPS Section (required): Medication
    - IPS Section (required): Allergies and Intolerances
    - IPS Section (required): Active Problems
    - IPS Section (recommended): Results (Laboratory results and Observational results).

- 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 July 2019 and August 2019, with relevance for testing the D2D communication protocol in this stage of InteropEHRate implementation, such as:
    - Displaying Allergies and Intolerances: e.g. Penicillin
    - Displaying Active Problems: e.g. Ischemic heart disease
    - Displaying Medication: e.g. Mexiletine / Mexitil
    - Displaying Results:
        - Laboratory results – Blood analysis: e.g. Calcium (serum)
        - Observational results - Weight, height, systolic and diastolic pressure.

5.2. 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 smart phone (specifically, *the first stage of D2D library development and the corresponding UI design and implementation in HCP Web App*).

The following section summarises the progress and results of the first development iterations, comprising the preliminary "look and feel" version of HCP Web App for displaying the most relevant preliminary healthcare categories and data sets from S-EHR.

As presented in the previous Paragraph 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 deliverable [D2.7] - 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.

5.2.1. Mockups of HCP Web App. First iterations

This section depicts the preliminary 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. We mention that visual design was excluded in the first iterations especially to emphasize the clear focus on functionality and testing of data exchange (D2D protocol) between HCP Web App and S-EHR.

Like the mockups, the visual design will be tested with the end users in the next stage of development and further optimised once the first functional version of HCP Web App is implemented.

Hereinafter, are presented a series of mock-ups which reflect the outcome of two co-design iterations of July 2019 and August 2019 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.

The following graphical illustration (suite of mock-ups) meets the users' requirements and the *proposed fields* to display, corresponding to specific data sets for a patient (Current Patient) compliant with HL7 FHIR profile, as follows:

General data (Patient Identification)

Fields: Name, Surname, Date of birth, Location of birth, Gender, Country of residence, Social security number / PID.

Health data

- Allergies and Intolerances
- Fields (Allergy / Intolerance): Identifier, Name, Clinical status, Type, Category, Criticality
- Problems
- Fields (Main chronic condition): Identifier, Name, Clinical status, Date.
- Medication
- Fields (Drug / Medicine): Code / Identifier, INN / DCI (Name of ingredient), Manufacturer (Commercial name), Concentration (Medication form), Dose (Instructions), Start date, Status.
- Diagnostic results ----> Observational results
- Fields: Weight, Height, Blood pressure (Systolic pressure / Diastolic pressure), Date.
- Diagnostic results ----> Laboratory results (Blood tests)
- Fields: Type / Category of analysis (e.g. Biochemistry), Code analysis, Name, Value, Date, Normal values interval.

The tool used for creating mock-ups is Microsoft Visio, one of the most recognized for the graphical features and capabilities. In this particular case of graphical representation the elements used are buttons, drop-down menu, menu bar and application form.

The graphics elements were chosen such as to satisfy the specific requirements presented in the previous section, namely:

- Set of user requirements of D2.1, limited to V1 selection related to HCP app
- Major sections of IPS defining the patient EHR
- Initial health data set
- Patient profile (Current Patient).

The way the screens were created is using the drag and drop capability. The main screen contains the major sections (e.g. Home, Current Patient, Test D2D Library and Administration) and relevant fields for each particular section as presented hereafter.

Figure 5 - Home: Request access S-EHR – Open the connection between HCP app and S-EHR app

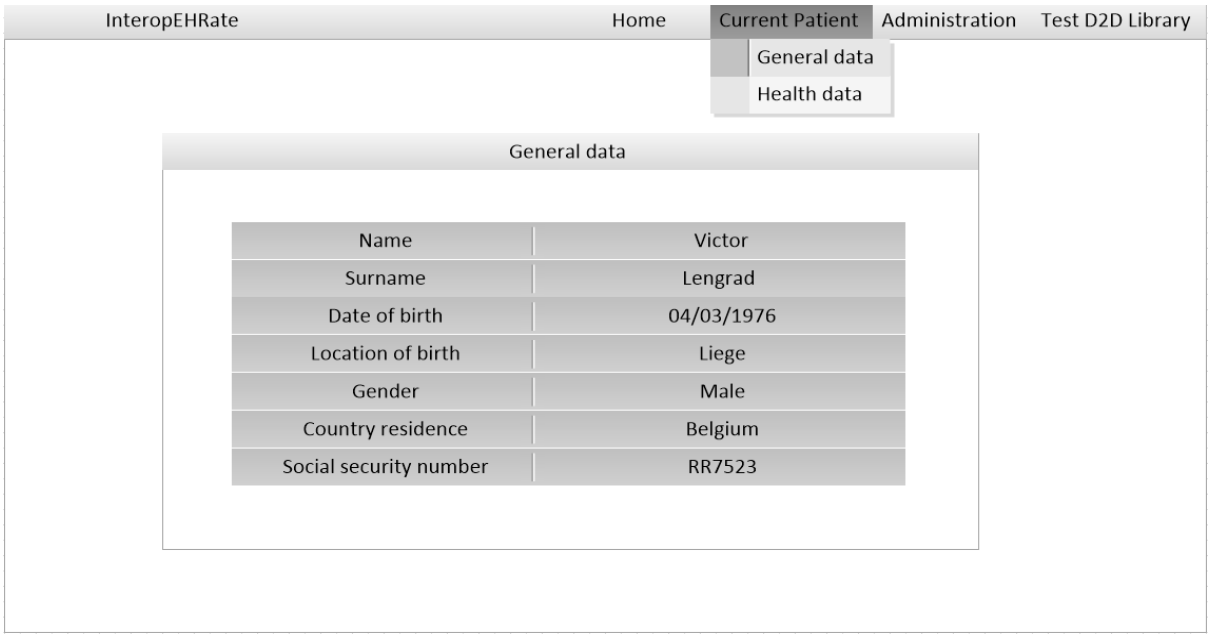


Figure 6 - Current Patient: General data

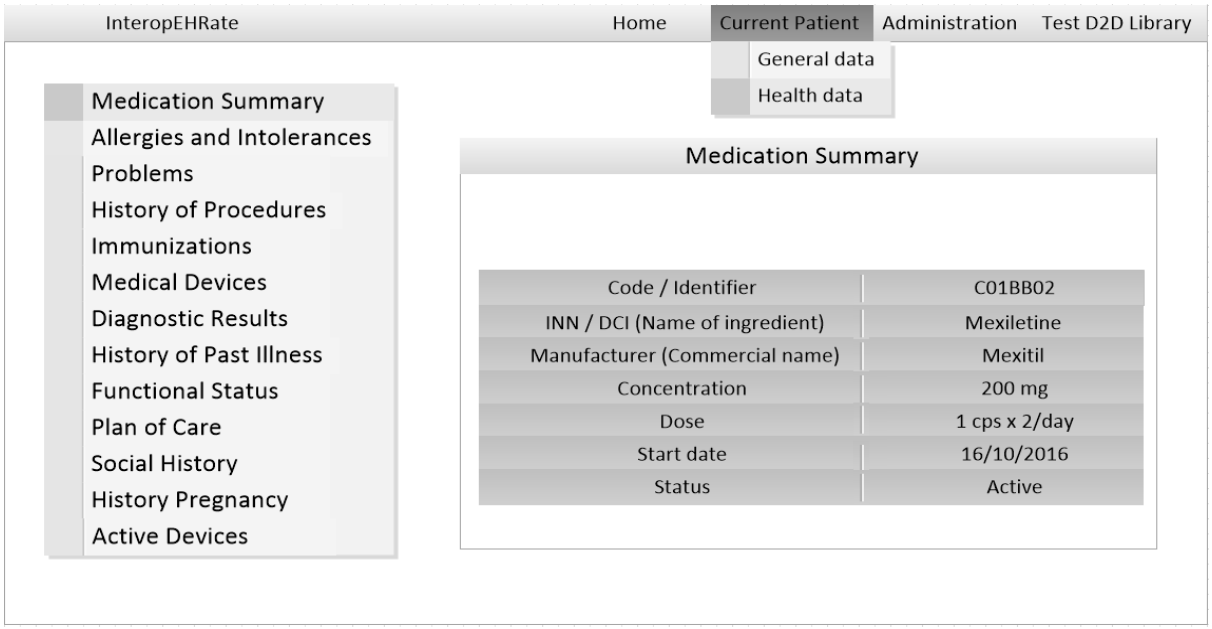


Figure 7 - Current Patient: Health data – Medication

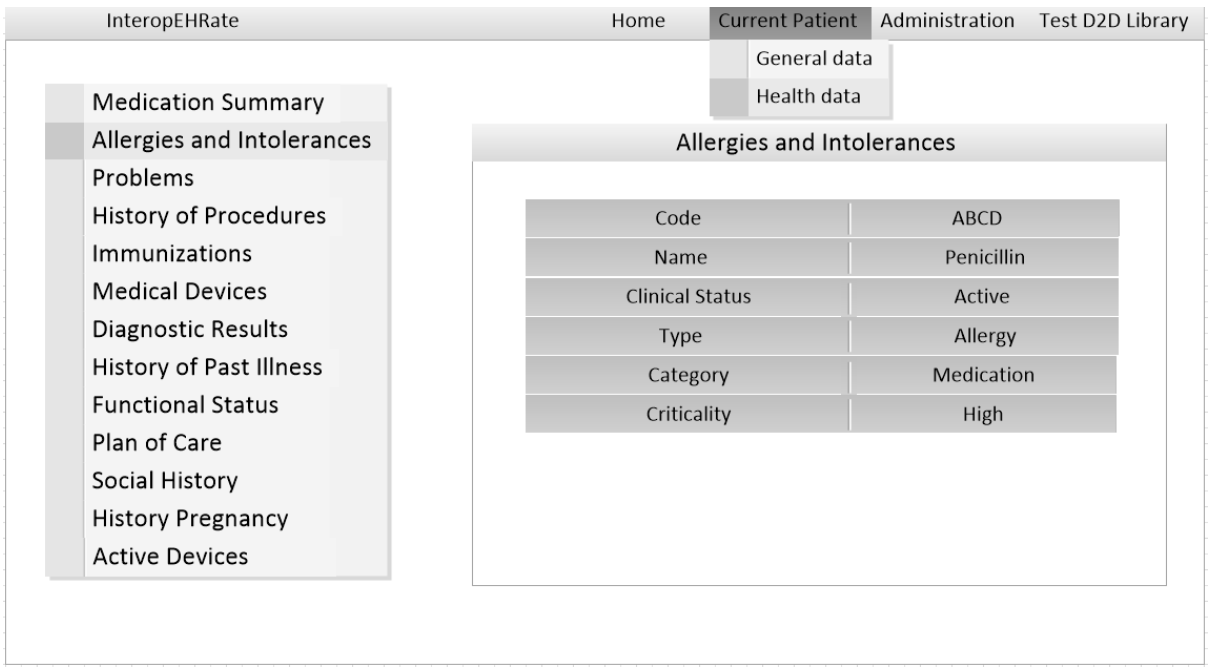


Figure 8 - Current Patient: Health data – Allergies and Intolerances

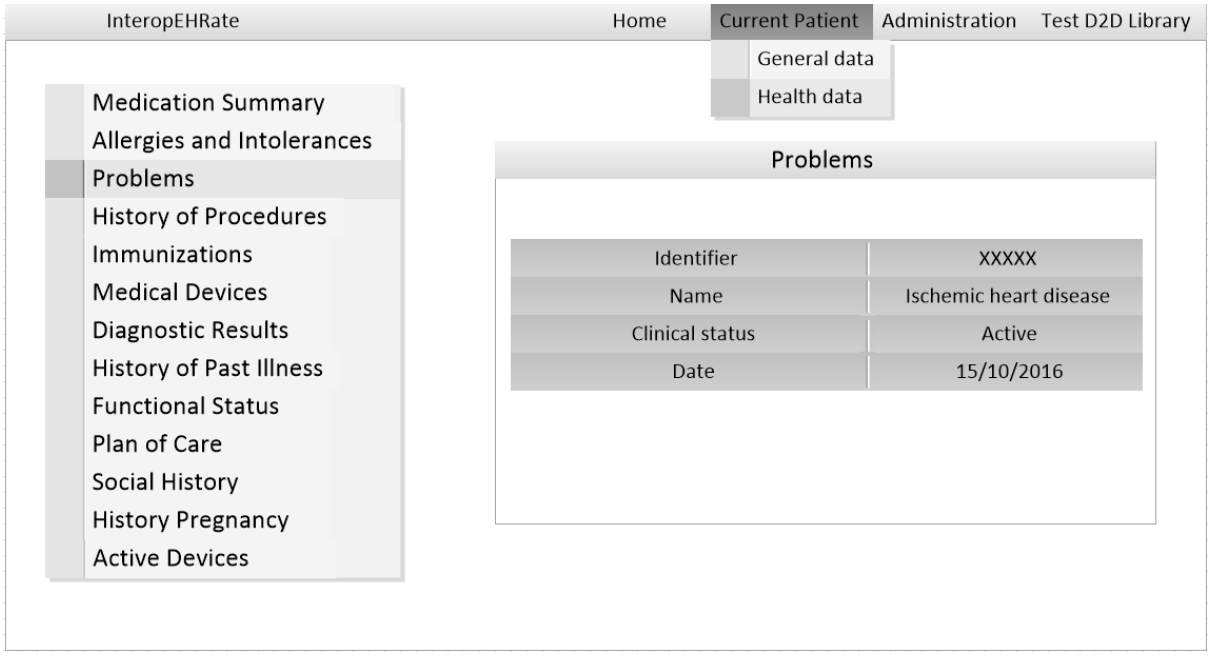


Figure 9 - Current Patient: Health data – Problems / Main chronic condition

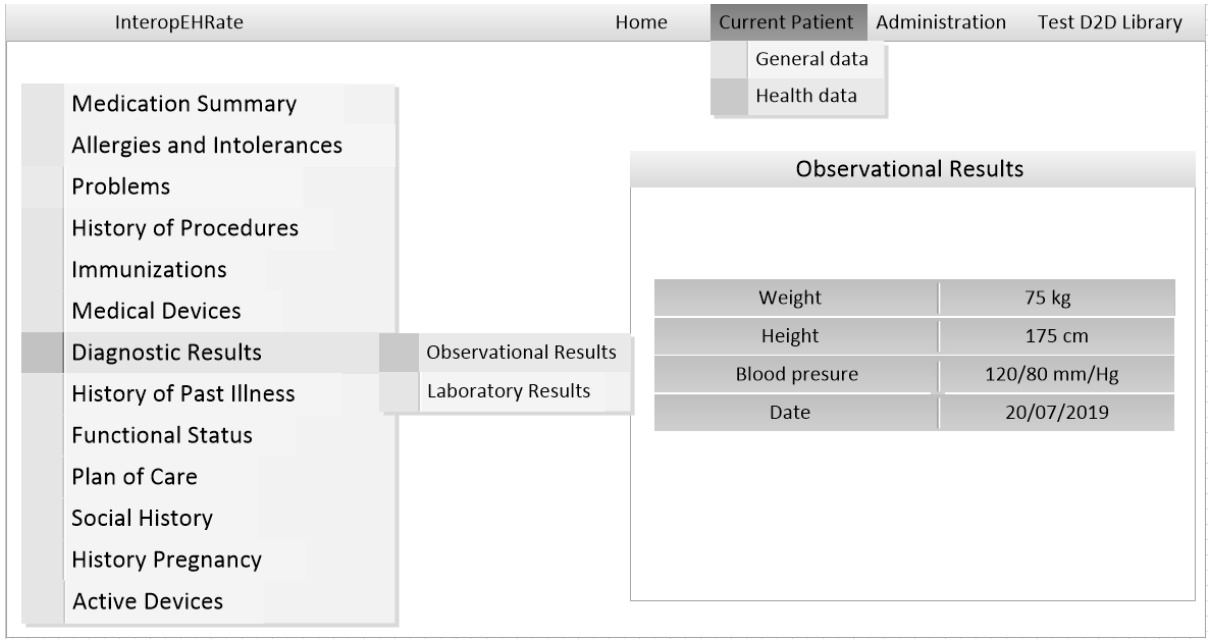


Figure 10 - Current Patient: Health data – Diagnostic Results / Observational Results

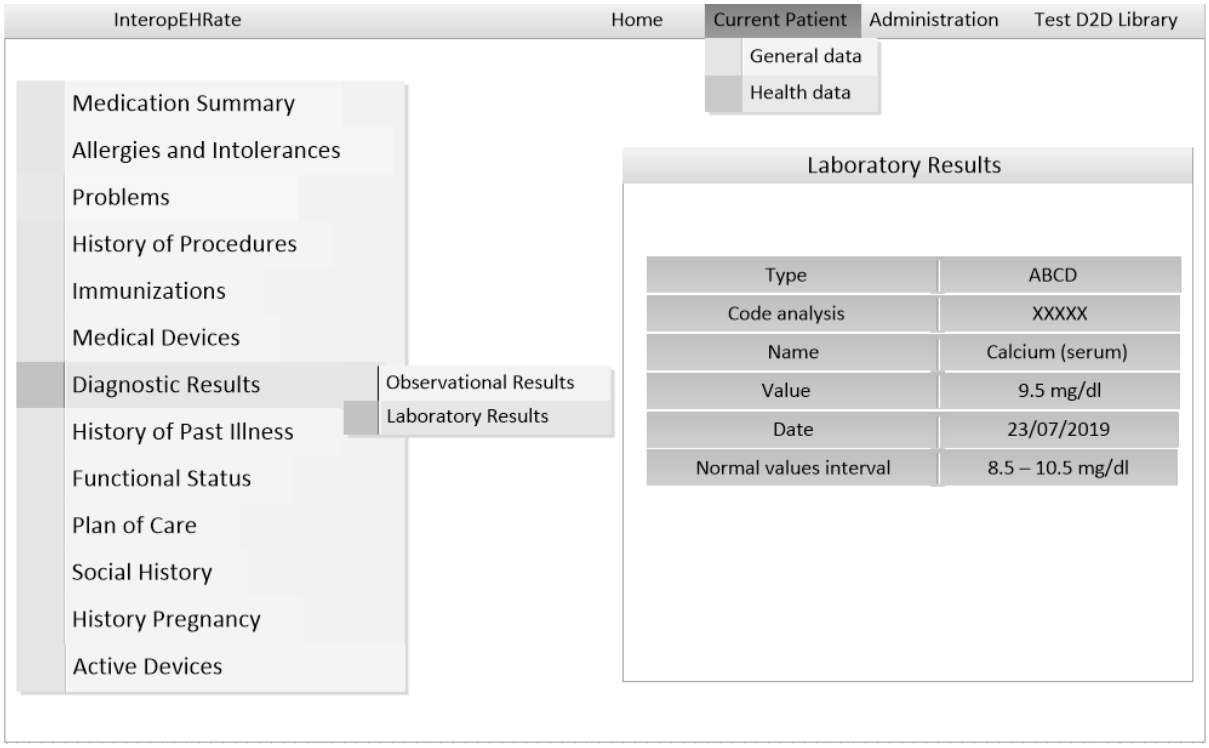


Figure 11 - Current Patient: Health data – Diagnostic Results / Laboratory Results

Figure 12 - Administration: Health Care Organization

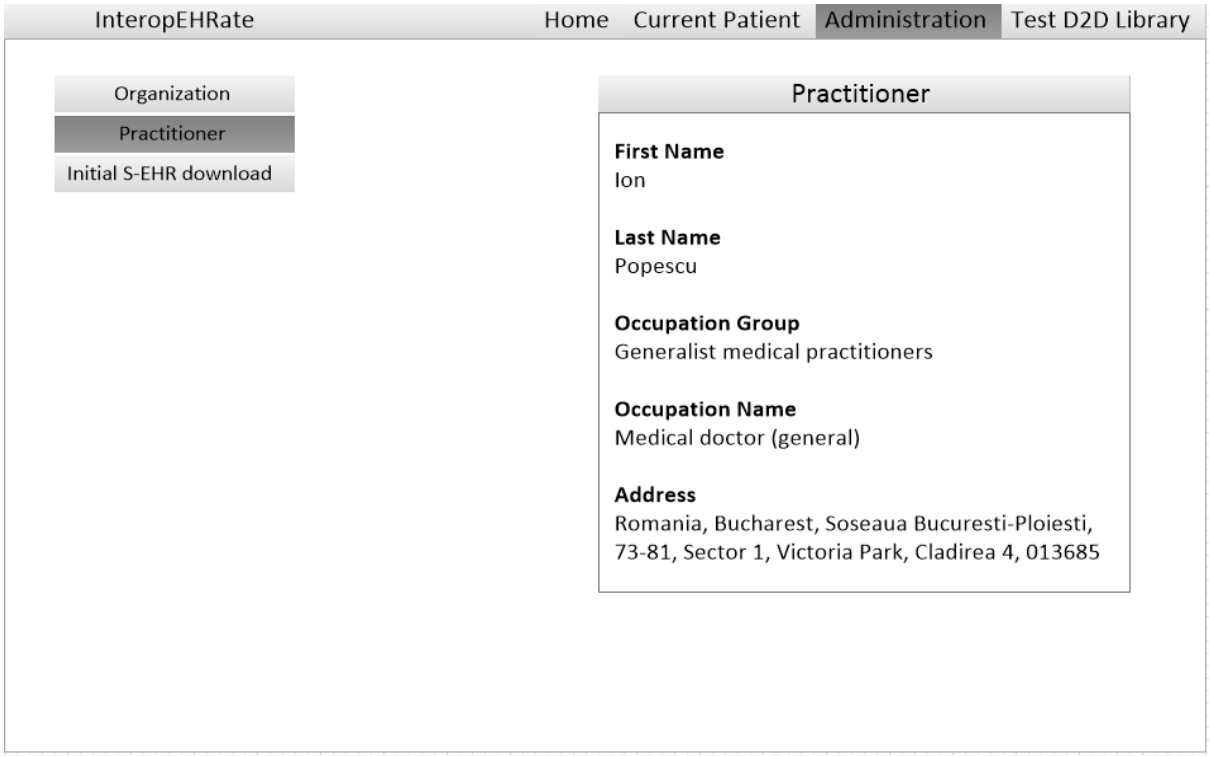


Figure 13 - Administration: Practitioner

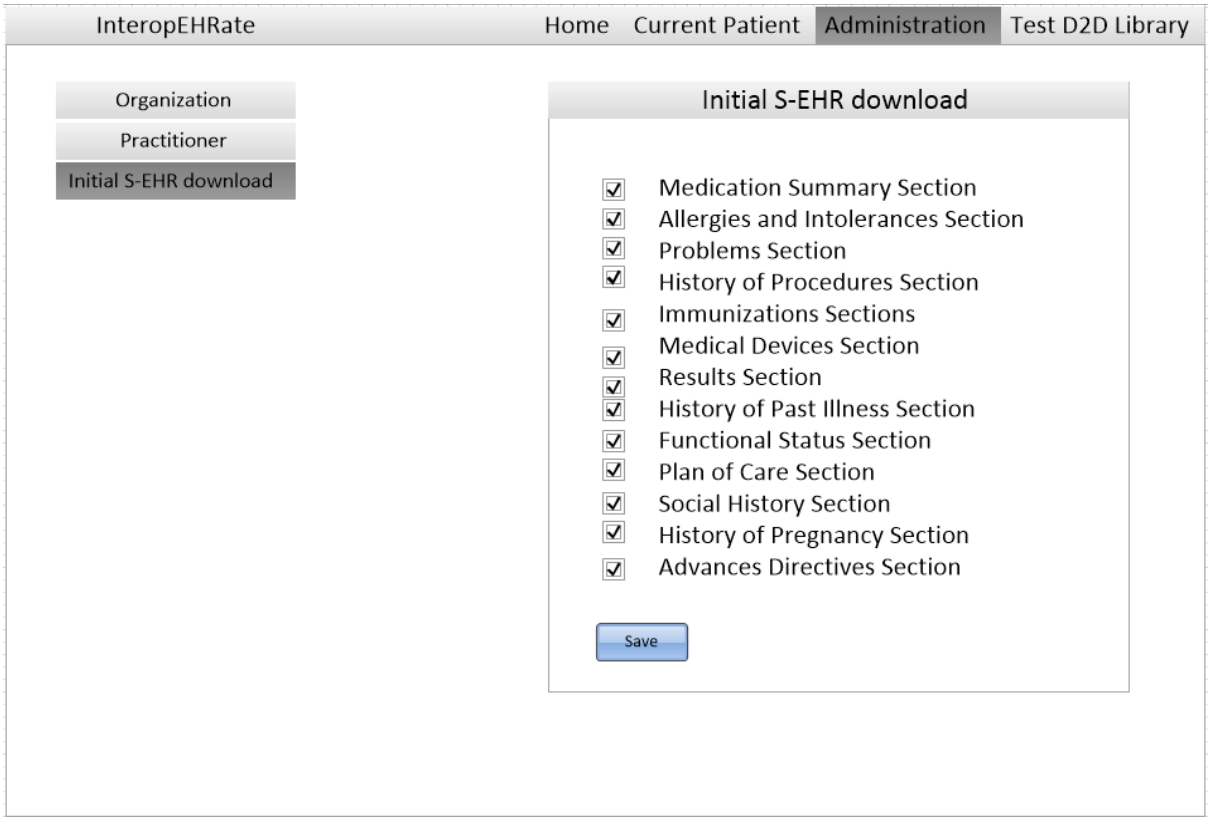


Figure 14 - Administration: Initial S-EHR download

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.

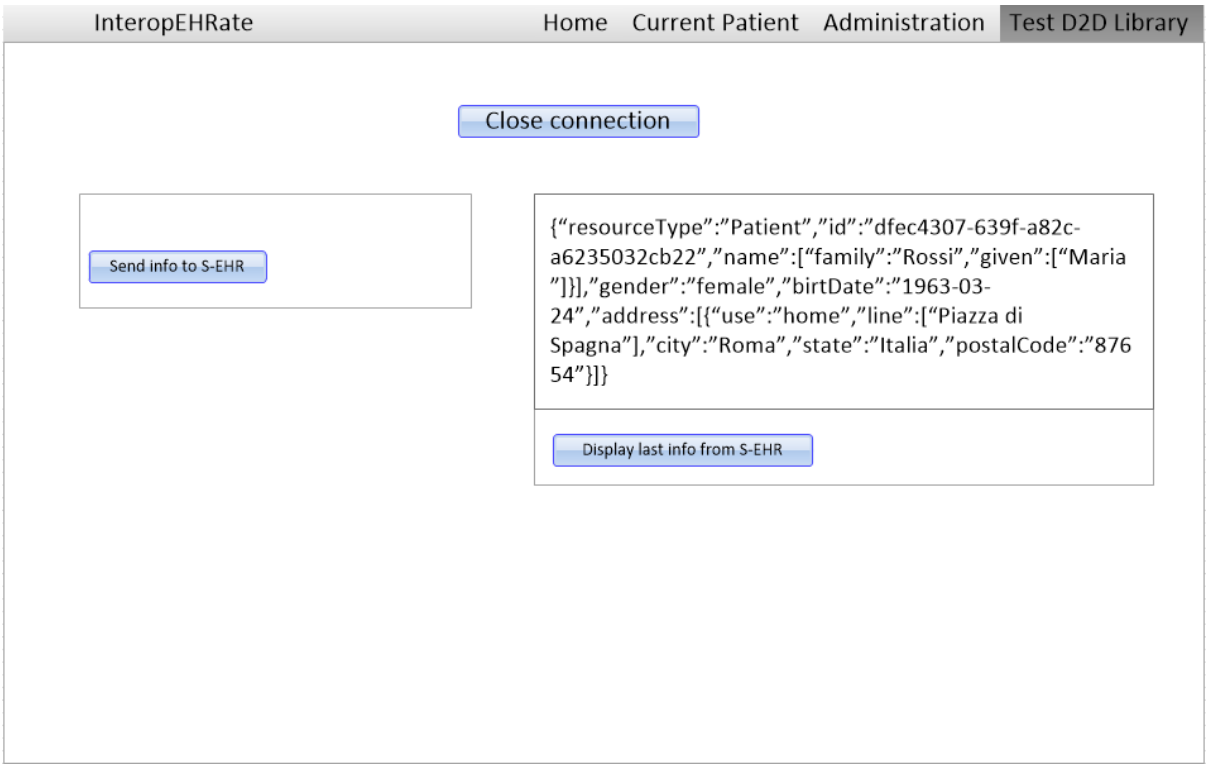


Figure 15 - Test D2D Library

One of the most significant functionality in this stage of implementation is testing of data exchange (D2D protocol) between HCP Web App and S-EHR – "Test D2D Library".

The connection is established at the time when the laptop of HCP and the patient' smart phone have the Bluetooth turned on.

When the connection between HCP Web App and S-EHR app is established, two buttons appear / are activated in HCP Web App: <Send information to S-EHR> and < Display last information from S-EHR>.

Once the connection is established, the data exchange is starting and relevant data from S-EHR are displayed in HCP Web App.

In S-EHR App will be displayed information about the health organization and in HCP Web App will be displayed data about the patient. Data from S-EHR is sent via Bluetooth in HCP Web App and will be displayed when push the Display button. When the Bluetooth connection is closed the two buttons from HCP Web App are hidden.

In S-EHR App will be displayed similarly as in HCP Web App two buttons: <Lineread> and <SendData>; SendData will send the data in HCP Web App and Lineread will read data from the HCP Web App.

5.2.2. UI design in HCP Web App. Current GUI representation

This section presents the current state of implementing the GUI at the application level.

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.

Hereinafter, are presented a series of print screens which reflect the outcome of the current stage of GUI implementation. These preliminary design characteristics and features of GUI will be refined in the next deliverables [D5.5] and [D5.6].

The print screens show how the user interfaces were designed, including also, for a concrete illustration, how they were implemented (programing code) in Bootstrap CSS.

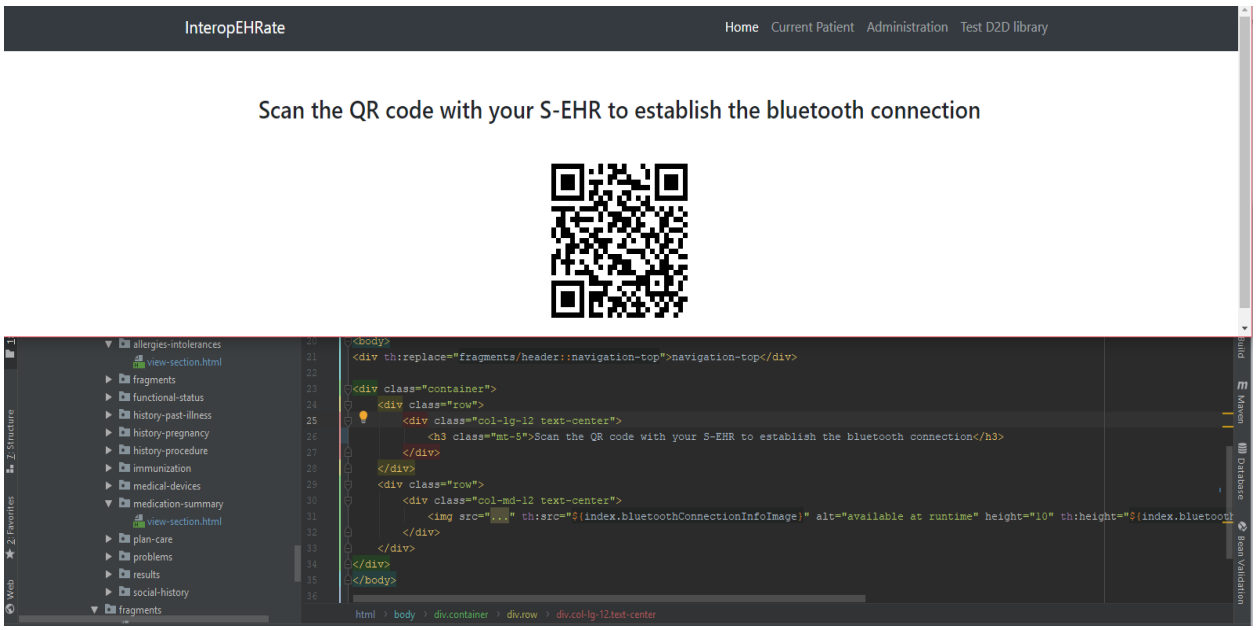


Figure 16 - Home page: Establish D2D connection / display information about current connection

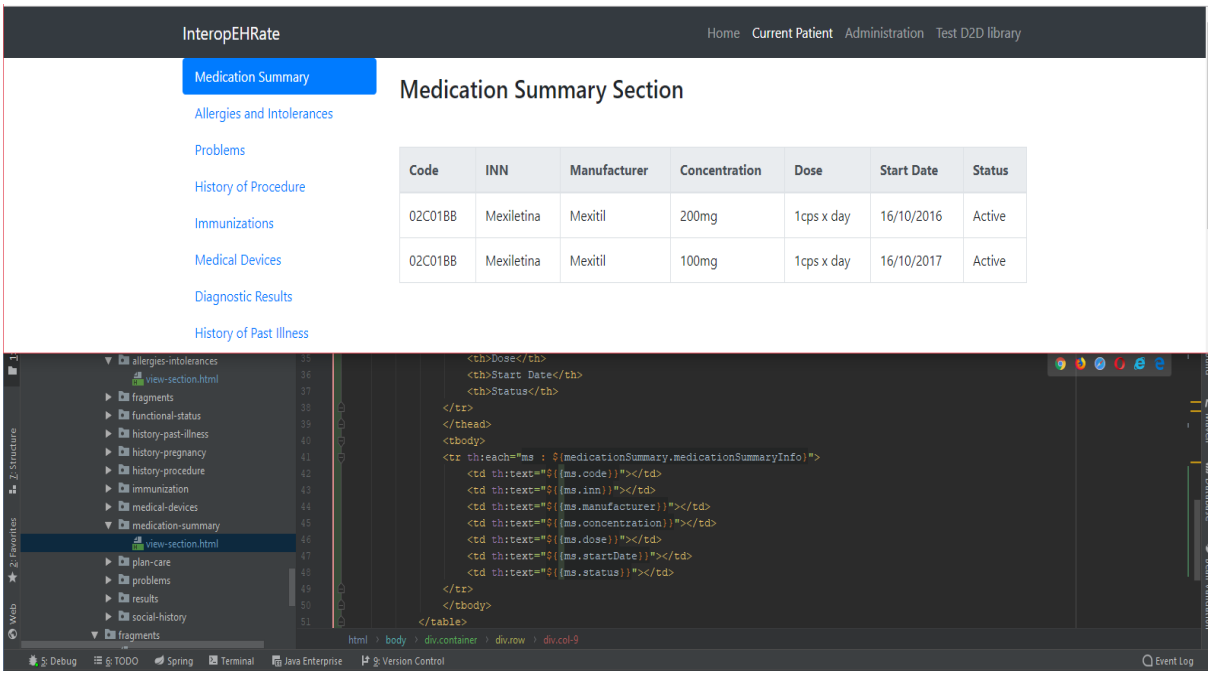


Figure 17 - Current Patient page: Display the information imported from S-EHR app corresponding to the current connection - relevant health data for Medication

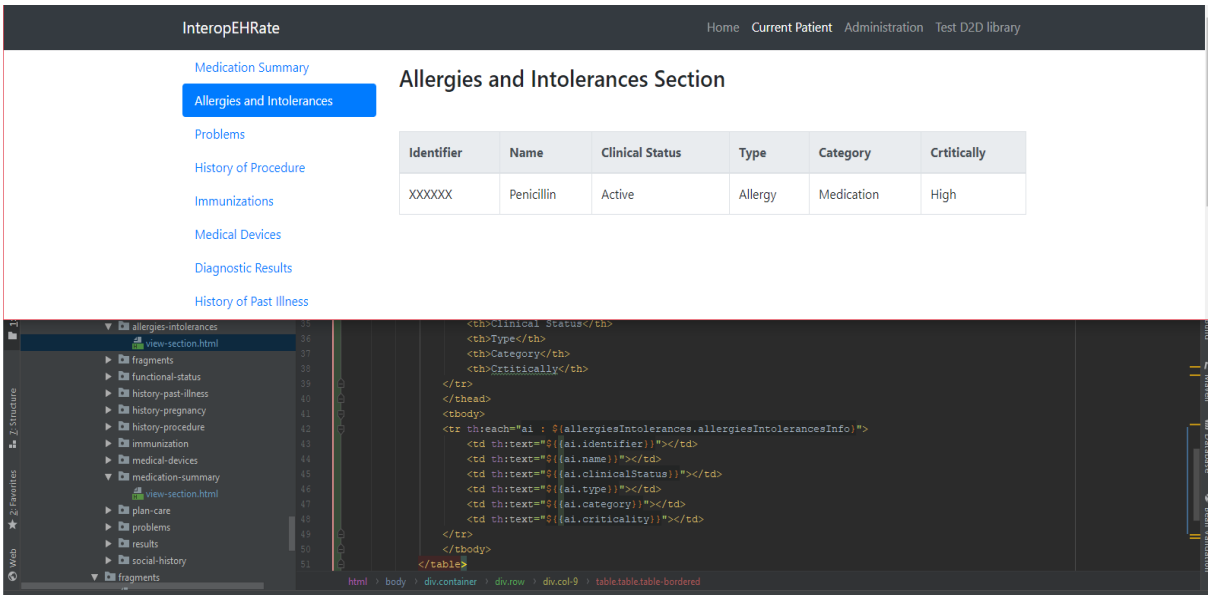
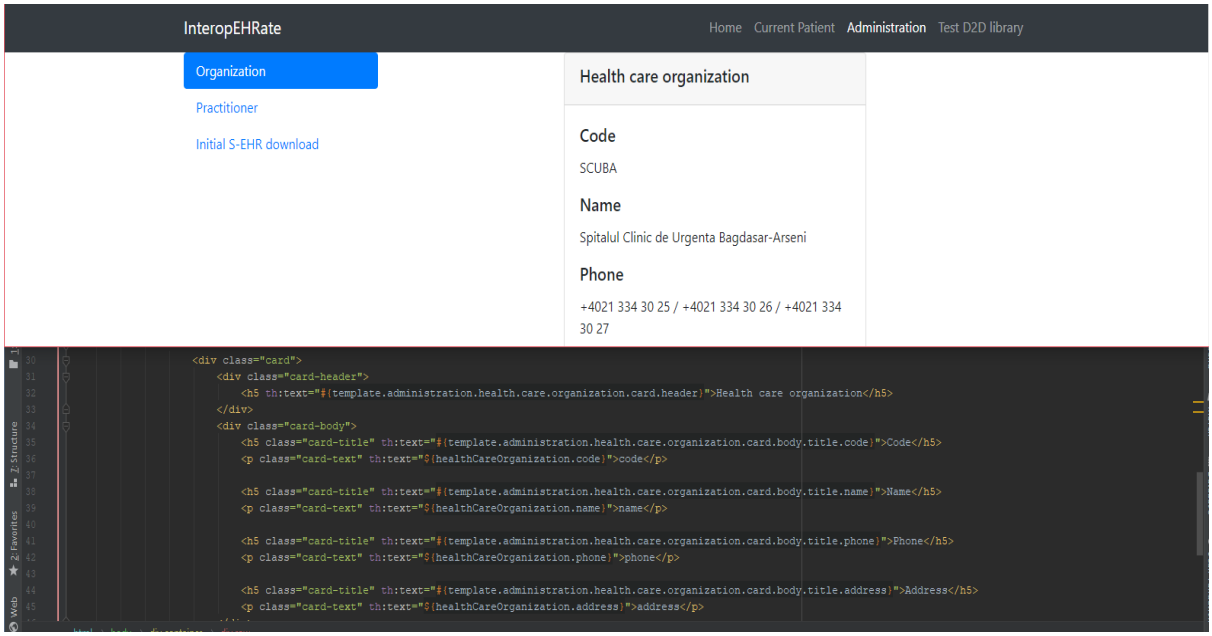


Figure 18 - Current Patient page: Display the information imported from S-EHR app corresponding to the current connection - relevant health data for Allergies and Intolerances

***Administration of HCP Web App**:*

Display the options of configuring the current instance, providing in this stage of implementation the following options: Organization, Practitioner and Initial S-EHR download.



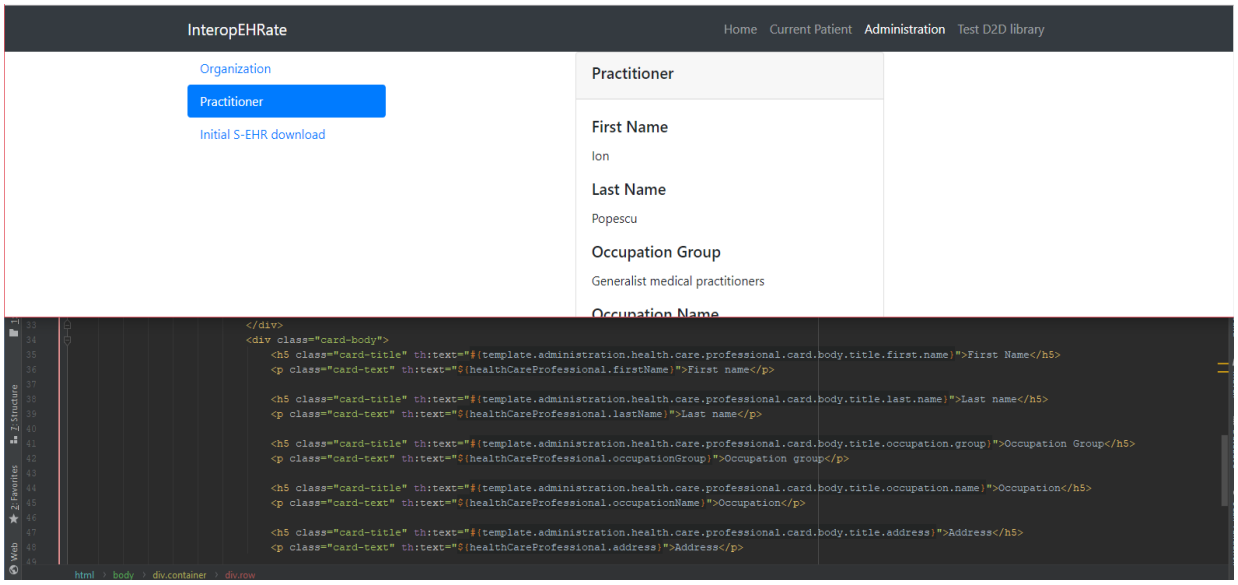

```
<div class="card">
    <div class="card-header">
        <h5 th:text="#{template.administration.health.care.organization.card.header}">Health care organization</h5>
    </div>
    <div class="card-body">
        <h5 class="card-title" th:text="#{template.administration.health.care.organization.card.body.title.code}">Code</h5>
        <p class="card-text" th:text="${healthCareOrganization.code}">code</p>

        <h5 class="card-title" th:text="#{template.administration.health.care.organization.card.body.title.name}">Name</h5>
        <p class="card-text" th:text="${healthCareOrganization.name}">name</p>

        <h5 class="card-title" th:text="#{template.administration.health.care.organization.card.body.title.phone}">Phone</h5>
        <p class="card-text" th:text="${healthCareOrganization.phone}">phone</p>

        <h5 class="card-title" th:text="#{template.administration.health.care.organization.card.body.title.address}">Address</h5>
        <p class="card-text" th:text="${healthCareOrganization.address}">address</p>
```

Figure 19 - Administration page: Organization




```
    </div>
    <div class="card-body">
        <h5 class="card-title" th:text="#{template.administration.health.care.professional.card.body.title.first.name}">First Name</h5>
        <p class="card-text" th:text="${healthCareProfessional.firstName}">First name</p>

        <h5 class="card-title" th:text="#{template.administration.health.care.professional.card.body.title.last.name}">Last name</h5>
        <p class="card-text" th:text="${healthCareProfessional.lastName}">Last name</p>

        <h5 class="card-title" th:text="#{template.administration.health.care.professional.card.body.title.occupation.group}">Occupation Group</h5>
        <p class="card-text" th:text="${healthCareProfessional.occupationGroup}">Occupation group</p>

        <h5 class="card-title" th:text="#{template.administration.health.care.professional.card.body.title.occupation.name}">Occupation</h5>
        <p class="card-text" th:text="${healthCareProfessional.occupationName}">Occupation</p>

        <h5 class="card-title" th:text="#{template.administration.health.care.professional.card.body.title.address}">Address</h5>
        <p class="card-text" th:text="${healthCareProfessional.address}">Address</p>
```

Figure 20 - Administration page: Practitioner

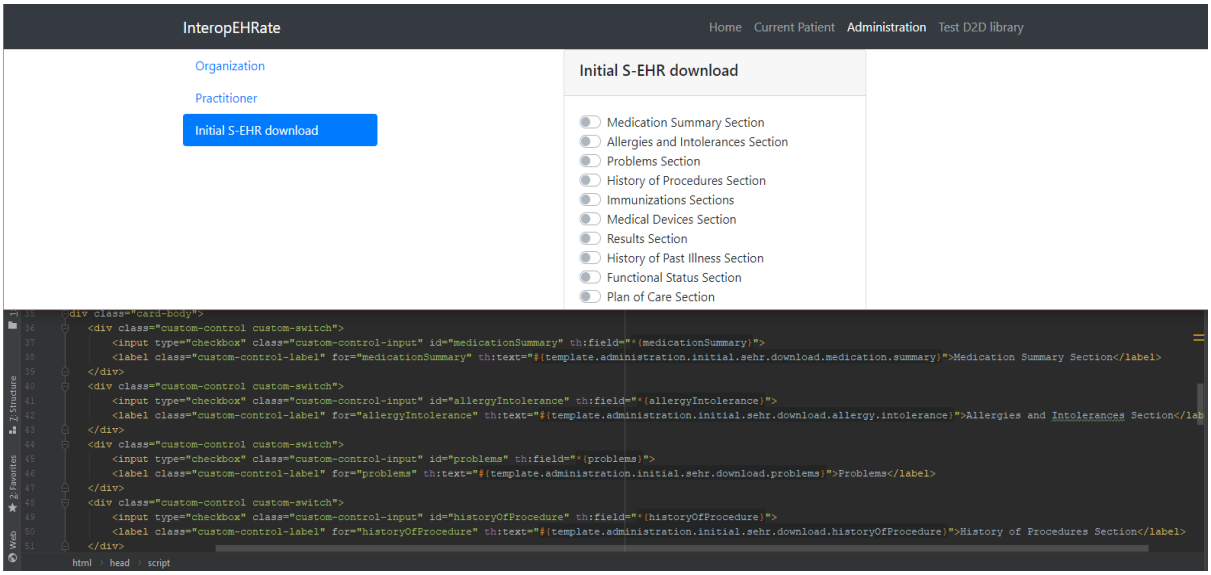


Figure 21 - Administration page: Initial S-EHR download - Specify what information will be downloaded from S-EHR App when the D2D connection is established

Test D2D library page: Performing integration tests with D2D library, as presented in the following printscreens.

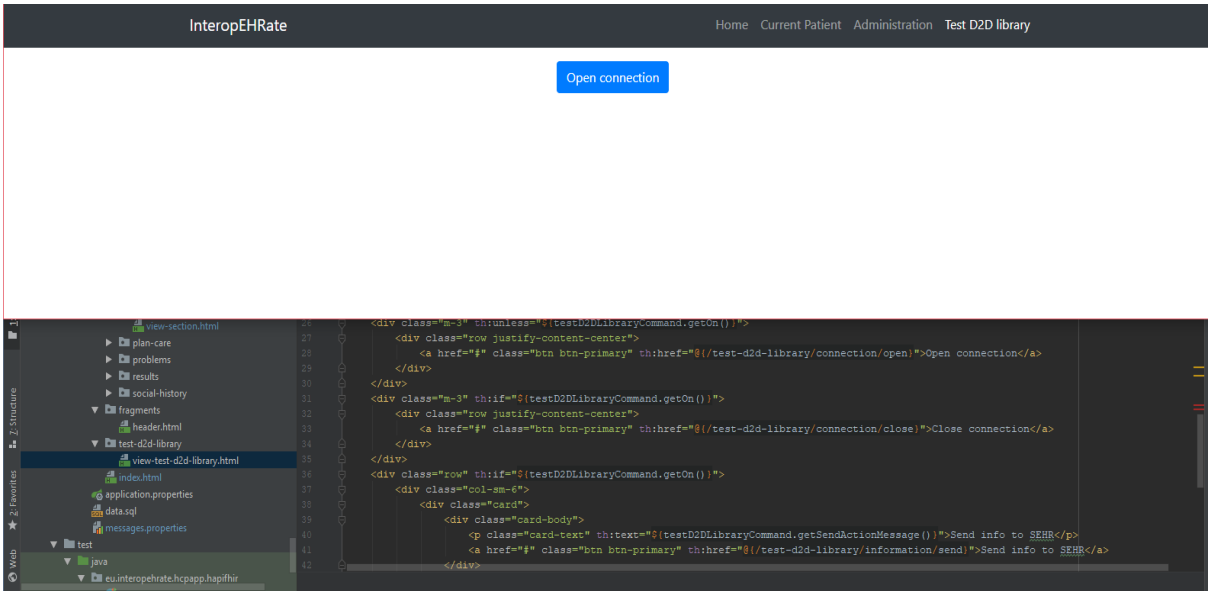


Figure 22 - Integration tests with D2D library (1)

Figure 23 - Integration tests with D2D library (2)

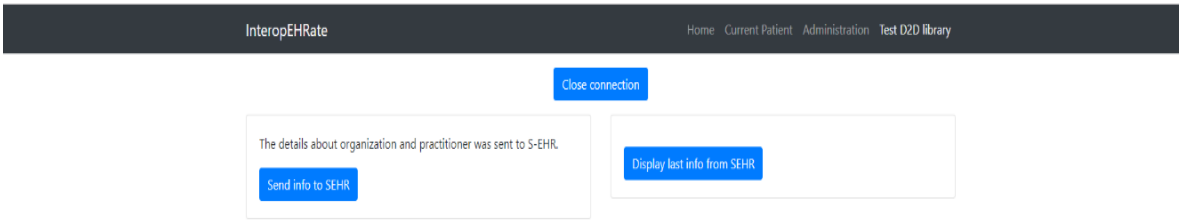


Figure 24 - Integration tests with D2D library (3)

6. CONCLUSIONS AND NEXT STEPS

The current deliverable aims to present the preliminary design specifications of the 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 [D5.1]. 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 smart phone.

Considering the particular requirements of the project tasks focusing on HCP web app and EHR functionalities, the deliverable encompasses the architectural design specifications and preliminary elements of UI design of HCP Web App that will then be detailed and updated incrementally in the next deliverables – [D5.5] and [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 preliminary 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 deliverables [D5.5] 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] Introduction to the Dependency Mechanism in Maven

[6] FHIR implementation used in the project

[7] Java Persistence API

[8] 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 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 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 contributors. (2018, February 26). Data transfer object. In *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Data_transfer_object&oldid=827785385https://en.wikipedia.org/wiki/Data_transfer_object

[12] Wikipedia contributors. (2019, September 23). Command (computing). In *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Command_(computing)&oldid=917376655

[13] http://www.agilemodeling.com

[14] https://www.agilebusiness.org

[15] **[D2.1]** InteropEHRate Consortium, User Requirements for cross-border HR integration - V1, 2019.https://www.interopehrate.eu/resources/

[16] **[D2.7]** InteropEHRate Consortium, FHIR profile for EHR interoperability - V1 (conceptual level specification, implementable level specification), 2019.https://www.interopehrate.eu/resources/

[17] **[D4.4]** InteropEHRate Consortium, Design of libraries for remote D2D HR exchange - V1, 2019.https://www.interopehrate.eu/resources/

[18] **[D5.1]** InteropEHRate Consortium, Software Requirements specification of an integrated EHR web app for HCP - V1, 2019. https://www.interopehrate.eu/resources/.

[19] **[D5.9]** InteropEHRate Consortium, Design data mapper and converter to FHIR - V1, September 2019. https://www.interopehrate.eu/resources/.

[20] **[D5.11]** InteropEHRate Consortium, Design of information extractor and natural language translator - V1, September 2020 https://www.interopehrate.eu/resources/.