D5.1

Software requirements specification of an integrated EHR web app for HCP - V1

ABSTRACT

This deliverable assumes the elaboration of preliminary software requirements and the design of the integrated web app – Healthcare Professional Application (HCP app) used by healthcare professionals for creating and accessing health data of foreign patients, based on user requirements specified in *Work Package 2 – Architecture for cross-border HR integration* and continuous collaboration with the final users. Healthcare Professional Application will exploit the Device to Device and remote protocols defined in *Work Package 4 HR Interoperability Protocols* for implementing the following functionalities: (i) import/export data directly from/to the Smart-Electronic Health Record (S-EHR) mobile application on the smart phone, (ii) import/export data from/to S-EHR cloud and (iii) access integrated health records imported from Electronic Health Records (EHRs)/Electronic Medical Records (EMRs) of other healthcare providers.

Delivery Date	3rd July 2019
Work Package	WP5
Task	T5.1
Dissemination Level	Public
Type of Deliverable	Report
Lead partner	SIVECO

InteropEHRate project is co-funded by the European Union (EU) Horizon 2020 program under Grant number 826106.

CONTRIBUTORS

	Name	Partner
Editors / Contributors	Simona Bica, Nicu Jalba, Andrei Oanca, Cristiana Stermin, Adrian Bradu	SIVECO
Contributors	Debora Desideri, Alessio Graziani	ENG
Contributors	Gabor Bella, Simone Bocca	UNITN
Reviewer	Kotsiopoulou Christina	HYGEIA
Reviewer	Julien Henrard	Andaman 7
Reviewer	Paul De Raeve	EFN

LOGTABLE

Version	Date	Change	Author	Partner
0.1	03-05-2019	First draft of table of contents	Simona Bica	SIVECO
0.2	03-06-2019	Update of table of contents based on partners' feedback	Simona Bica	SIVECO
0.3	10-06-2019	Contribution for Section 4 - HCP app description (Technical view)	Nicu Jalba, Andrei Oanca	SIVECO
0.4	12-06-2019	Contribution for Section 1, Section 2, Section 4 and Section 5 (Contextual and business view)	Simona Bica, Cristiana Stermin	SIVECO
0.5	14-06-2019	Contribution for Section 3 - Architectural view – Integration approach and interoperability findings	Gabor Bella, Simone Bocca	UNITN
0.6	18-06-2019	Deliverable ready for internal review	Adrian Bradu	SIVECO
0.7	20-06-2019	Contribution for Section 3, Architectural view – Integration approach and interoperability findings	Debora Desideri, Alessio Graziani	ENG

1.0	21-06-2019	Internal review and gathered feedback	Julien Henrard	Andaman 7
1.1	24-06-2019	Internal review and gathered feedback	Kotsiopoulou Christina	HYGEIA
1.2	25-06-2019	Integrate comments and suggestions	Nicu Jalba	SIVECO
1.3	26-06-2019	Integrate comments and suggestions	Andrei Oanca	SIVECO
1.4	26-06-2019	Internal quality check and completion	Adrian Bradu	SIVECO
1.5	27-06-2019	Deliverable finalized and sent to final review and quality check	Adrian Bradu	SIVECO
1.6	28-06-2019	Quality check completed	Argyro Mavrogiorgou	UPRC
1.7	02-07-2019	Terminology/Content check	Paul De Raeve	EFN
Vfinal	03-07-2019	Final version for submission	Laura Pucci	ENG

ACRONYMS

Acronym	Description
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.
D2D	Device to Device Protocol
EHR	Electronic Health Record
GUI	Graphical User Interface
HCP	HealthCare Professional
HTML	Hypertext Markup Language
IHS	InteropEHRate Health Services
IHT	InteropEHRate Health Tools
IRS	InteropEHRate Research Services
JPA	Java Persistence API
MD2DI	Mobile Device to Device Interface
R2D	Remote to Device
R2DI	Remote to Device Interface
RIS	Research Interoperability Services
RSI	Research Interface
S-EHR	Smart EHR
S-EHR-C	S-EHR Cloud
TD2DI	Terminal Device to Device Interface

UCD	User Centred Design
UI	User Interface
WBA	Web Bluetooth API
CSS	Cascading Style Sheets
WP	Work Package

TABLE OF CONTENT

LIST OF FIGURES

1 INTRODUCTION

1.1 Scope of the document

The current deliverable is the first report of WP5 – Incremental EHRs integration and covers the preliminary aspects and findings related to *Task 5.1 HCP app and EHR functionalities* of the InteropEHRate project. Task 5.1 is focused on producing the software requirements specification and the design of the HCP app solution[1] used by healthcare professionals for accessing and creating health data of foreign patients.

The definition of software requirements and the design of HCP app are based on the results obtained in WP2 and WP4, as they are presented mainly in the deliverables [D2.1], [D2.4] and [D4.1].

At this stage of project implementation, the deliverable aims to depict the most significant features of HCP app which address the following directions:

- import/export data directly from/to the S-EHR on the smart phone
- import/export data from/to S-EHR cloud
- access integrated health records imported from EHRs/EMRs of other healthcare providers.

1.2 Intended audience

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

- Technical staff: developers, IT health consultants, analysts, web designers, interested to have an overview about the design and specific implementation of HCP app
- Healthcare providers interested in how to use an application like HCP 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 report is structured in five chapters, as following:

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 representative description of relevant characteristics of this particular stage of the project implementation, addressing also the relation to other WPs and tasks.

Section 3. Architectural view – Integration approach and interoperability findings: presents the major aspects concerning the integration of HCP app in the InteropEHRate solution and the interoperability approach relevant for HCP app (e.g. compliance with HL7 FHIR profile, coding and data mapping, vocabularies ...).

[1] In the context of InteropEHRate project the term “solution” shall be read as a result component or group of components resulting from the project and answering to project requirements.

Section 4. HCP app description**:** depicts the preliminary technical aspects of the architectural design of HCP app and the software requirements specification. Relevant aspects concerning the technological approach specific to implement HCP app are also presented.

Section 5. Conclusions and next steps: presents the conclusions and next steps concerning this particular stage of elaborating HCP app and EHR functionalities.

1.4 Updates with respect to previous version (if any)

The current deliverable is the first of the three deliverables of Task 5.1 dedicated to software requirements specification and design of HCP app (D5.1 the present deliverable and D5.2 and D5.3 the further versions of the present deliverable - Software requirements specification of an integrated EHR web app for HCP) and includes the preliminary architectural design elements and User Interface (UI) design of HCP app that will then be detailed and updated incrementally in the next deliverables – [D5.4] Design of an integrated EHR web app for HCP - V1, [D5.5] Design of an integrated EHR web app for HCP - V2 and [D5.6] Design of an integrated EHR web app for HCP- V1.

2 CONTEXT

2.1 Goals

Within the project implementation, the deliverable presents the technical approach pursued in designing HCP app, based on the valuable results obtained in the stage of user needs and requirements analysis [D2.1] and the relevant aspects concerning the innovative way of data exchange depicted in deliverable [D4.1].

One purpose of this document consists on providing personalized and customizable information to end-users, based on a ***user-centric approach***. Representative aspects and details of this approach were outlined hereinafter, in *Section 4.2.2 Communication with S-EHR app*.

The HCP app is the "prototype of secure web app, used by the HCPs to securely exchange health data of their EMRs with any S-EHR and to read health data stored in federated EHRs. Hospitals may use the HCP app or may choose to evolve the Graphical User Interfaces (GUIs) of their EMRs to add the same functionalities of the HCP app."

Taking into consideration the above mentioned characteristics, HCP app is a software application designed such as to provide healthcare professionals with the ability to access and operate patients' data from S-EHR, S-EHR Cloud and EMR.

Moreover, HCP App aims of graphically illustrating through the user interfaces how relevant data about the patient condition is exchanged, thus being a valuable tool for analyzing the perception of final users about the InteropEHRate solution.

For this purpose, significant work will be put on designing compliant GUIs to accomplish the end user needs and requirements, based on standardized User Interface (UI) and User Experience (UX) design principles and methods, as presented hereinafter in *Section 4.2.2 Communication with S-EHR app*.

Within this deliverable, HCP app is described from three significant perspectives:

- Architectural perspective (addressed in Section 3 and Section 4)
- Technological perspective (addressed in Section 4)
- End User perspective (addressed in Section 4).

The deliverable presents the preliminary technical characteristics / features of HCP app, whereas the details, iterative completions and updates of the technical solution will be described in the next deliverables [D5.2] Software requirements specification of an integrated EHR web app for HCP - V2 and [D5.4] Software requirements specification of an integrated EHR web app for HCP - V3.

The current report depicts relevant aspects about the design of HCP app solution based on the results and findings gathered in the stage of user needs and requirements analysis and design of the InteropEHRate solution, registered in the deliverables [D2.1] User Requirements for cross-border HR integration - V1 and [D2.4] InteropEHRate Architecture - V1.

The defining technical elements and software requirements specification of HCP app in this stage of implementation are also based on Use Cases defined in the deliverable [D2.7] FHIR profile for EHR interoperability - V1 which deals with the formalization of health information compatible with HL7 FHIR profile (data set, data format, data semantics, terminologies, ...).

The Use Cases correspond to the three representative Scenarios detailed in deliverable [D2.1]: Device to device HR exchange (D2D), Remote to device HR exchange (R2D) and Research HR exchange, the first two Use Cases being suggestive for the HCP app design.

The following phases outlined in the Use Cases for D2D and R2D were considered in drafting HCP App:

- Determining the security context
- Setting up the data exchange framework
- Determining the policies for data access (based on the patient consent)
- Illustrating the access to data (Graphical design of user interfaces).

Three categories of relevant information / data were considered as valued scientific input and analyzed when defining the technical specifications (software requirements specification and design findings) of HCP app in the particular version described in this deliverable:

- Information concerning the *methodological approach* relevant for HCP app design (input from deliverable [D2.1] User Requirements for cross-border HR integration - V1– Section 2. Approach for requirement analysis and Section 6.5. Users Focus Groups)
- Information concerning the *functional approach* relevant for HCP app design (input from deliverable [D2.1] – Section 5. Reference Scenarios and Section 6. User Requirements)
- Information concerning the *end-user perspective* relevant for HCP app design (input from deliverable [D2.1] – Section 4. S-EHR Content, Section 5. Reference Scenarios and Section 6.5. Users Focus Groups).

The architectural view and technological approach depicted hereinafter, in Section 3 and Section 4, are based on the technical input from the deliverables [D2.4] and [D4.1].

The conceptual approach to designing the HCP app solution, including compatibility requirements with HL7 FHIR profile, is based on the scientific input from the deliverable [D2.7] FHIR profile for EHR interoperability - V1.

Moreover, the design of HCP app is based on the preliminary information about the D2D and R2D protocols, depicted in the deliverable [D4.1] (v1). The particular software requirements specification regarding the use of D2D protocol to provide the secure data exchange between HCP app and S-EHR are figured by UML charts in *Section 4.2.1. User flow diagrams (D2D)*.

The GUI design requirements for HCP app are based on the Scenarios and User Requirements presented in deliverable [D2.1], as well as on the findings resulted from the Focus Groups assessment in the same deliverable. Within the deliverable, in the section dedicated to ***User perspective*** in Section 4.2.2, these requirements are illustrated through representative mock-ups that meet the user needs.

2.2 Relation with other WPs/Tasks

The current document is the first deliverable of Task 5.1 within *WP5 – Incremental EHRs integration* and is based on the valuable work and input from WP2 - [D2.1], [D2.4] and WP4 - [D4.1]. The preliminary results gathered in WP2 and WP4 represent the scientific and technical basis for WP5.
The deliverable is based on the results and findings obtained mainly in Task 2.1 User requirements, Task 2.2 Architecture specification, Task 2.3 Interoperability profile and Task 4.1 Remote and D2D healthcare protocols.

The *WP7 – Validation of results* will be also connected with the actual Task 5.1 and the others tasks of WP5 because different healthcare partners (FTGM, CHU, HYG, SCUBA, EFN) of the project consortium will be involved in WP7 to explore and exploit the functionalities of InteropEHRate innovative platform.

3 ARCHITECTURAL VIEW – INTEGRATION APPROACH AND INTEROPERABILITY FINDINGS

HCP App is a software application designed to provide medical staff with the ability to access and operate patients' data from S-EHR Mobile App, S-EHR Cloud and EHR of the Healthcare Organization. In other words, the HCP App is an application used by the HCPs to securely exchange health data of their EHRs with any S-EHR Mobile App and to read health data stored in S-EHR Cloud using the InteropEHRate protocols.

HCP App provided by the project is an example of software application for HCPs that supports the InteropEHRate protocols. The objective of this prototype is to demonstrate concretely how the HCP can use InteropEHRate protocols and how can exploit the InteropEHRate Health Services (IHS[2]) to interact with the content of an existing EHR.

In this respect this HCP App implementation will provide functionalities for:

- Import data from S-EHR and export them back using the D2D protocol (TD2DI[3], MD2DI);
- Import data by the remote protocol (R2DI[4]) from the S-EHR cloud;
- Import health data from current systems (EHRs) within healthcare organization using the IHS interface (IHSI) provided by IHS component.

In the following picture the interfaces provided and required by HCP app are shown (interfaces that are part of the InteropEHRate protocol are depicted in green).

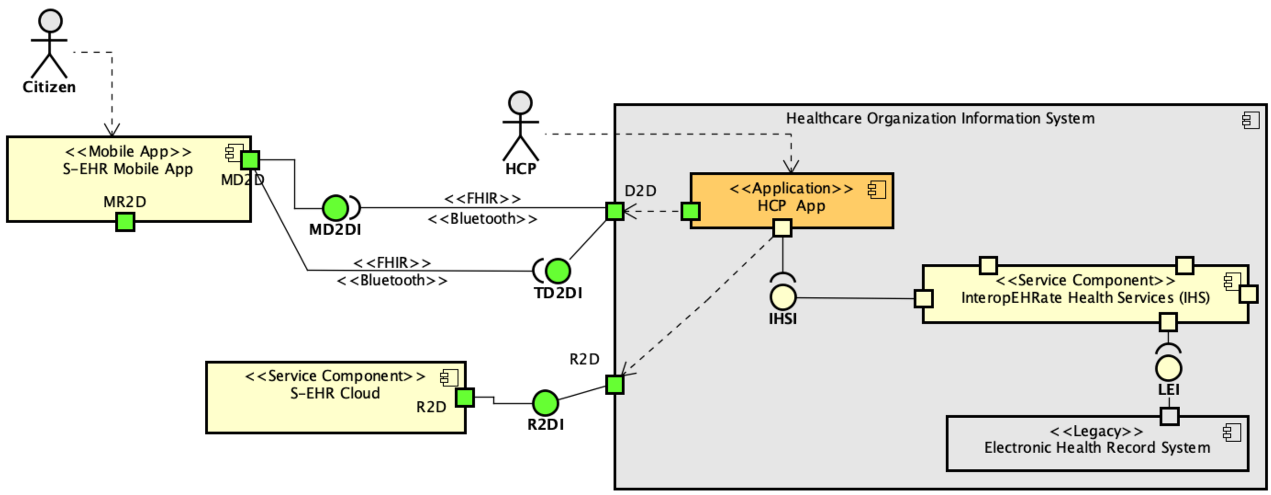

Figure 1: Interfaces of HCP app

[2] InteropEHRate Health Services (IHS) offers runtime functions for data conversions and translation and it exposes the interfaces for R2D protocol to interact with the S-EHR cloud. It interacts with existing legacy EHR systems through the LEI interface that allows the import of health data from the legacy systems. The IHS can convert structured data from legacy to S-EHR and vice versa and uses an external service to translate free text to the local language and/or to the citizen language.

[3] Terminal Device to Device Interface: offered any application used by HCPs to support the D2D protocol, i.e. to exchange health data with citizen's S-EHRs at short distance, without using internet.

[4] Remote to device interface: offered by the HealthCare Organization to support the R2D protocol, that is secure communication protocol (and API), using internet, for cross-border exchange of health data.

The HCP App interacts with the *InteropEHRate Health Services* (IHS) component, specified in deliverables [D2.4] and [D5.9], in two use cases:

- For the download of a S-EHR from a patient's mobile device to the HCP App, in the purpose of localizing it into the language and standards of the health institution where the visit is taking place;
- For the conversion and upload of a S-EHR from the legacy information system of a hospital to the patient's mobile device.

3.1 S-EHR Download Use Case

After the S-EHR has been downloaded from the patient's mobile phone to the HCP App, its contents need to be localized with respect to the local health institution. The term *localization* covers the following operations:

1. Translation of the general schema of the FHIR-based S-EHR to the language of the institution (*target language* in the following);
2. Translation of free text contained within the health data into the target language, using machine translation methods;
3. Retrieval of HCP-friendly natural language labels in the target language for medical codes contained in the S-EHR (e.g., disease or procedure codes);
4. If required by local practices, mapping of international codes to locally used ones.

The IHS services that execute the operations above are automatic and, as such, by definition cannot provide localizations that are 100% accurate in all circumstances. Typically, for cases 1, 3, and 4 above, accuracy is expected to be high as translations are based on a fixed set of mappings specified by domain experts. In case 2, inaccuracies are expected to be more common, depending on the quality of the underlying machine translation service. For these reasons, for each data value localized with uncertainty, the HCP App will put in evidence to the HCP the use of automated methods through the GUI, in order to notify the HCP about potential inaccuracies. The IHS takes charge of marking up all such data items as part of the localization process.

In the download scenario the HCP App uses the IHS's method called *localize(sehr, targetLang)* passing as parameters the S-EHR object received from the mobile app, and a string that specifies the language in which the data have to be translated. The IHS check if the language of the data in the S-EHR object is the same as targetLang parameter, if this check is false, so the languages are different, IHS uses the method *translateSehrResource(resource, targetLang, level)* to invoke the translation component. This method is executed for each resource present in S-EHR object. The translation component uses an external component for the machine translation calling the method *externalMachineTranslation(text, targetLang)* that returns the text passed by parameter translated into the language specified by targetLang parameter. Once all resource in S-EHR object has been translated in this way, the S-EHR object is returned to the HCP App to be read in the desired language.

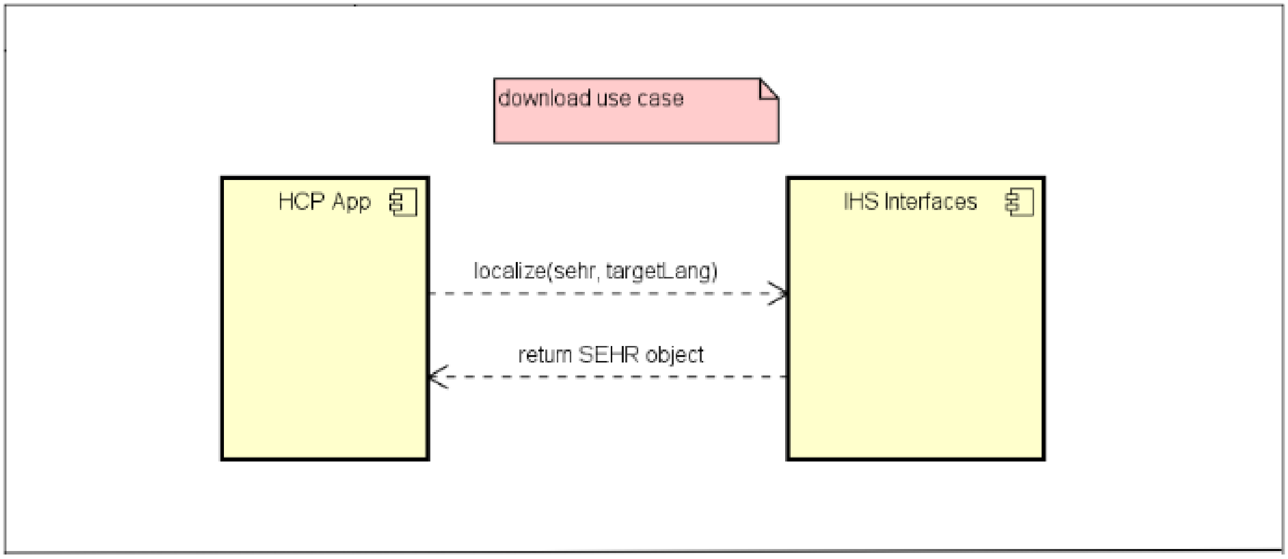

Figure 2: Download use case - localize method of the HCP App

3.2 S-EHR Extraction Use Case

Upon request from a patient to obtain his/her S-EHR on his/her mobile phone, the HCP App makes a request to the IHS for the extraction of the EHR from the legacy database of the hospital and its subsequent conversion to the S-EHR representation. In case the patient's own preferred language is different from the original language of the EHR, the conversion process mentioned above also involves translation to the patient's language.

The process described above is initiated by the HCP App by calling the IHS method *getResources(patientID, patientLang)* to find the resource corresponding to the patient specified by patientID parameter. If the language of the EHR object retrieved for the patient is different from the language specified by patientLang parameter, then IHS makes a translation as explained in the previous use case. After that the S-EHR object is returned to the mobile app and it will be available for the citizen.

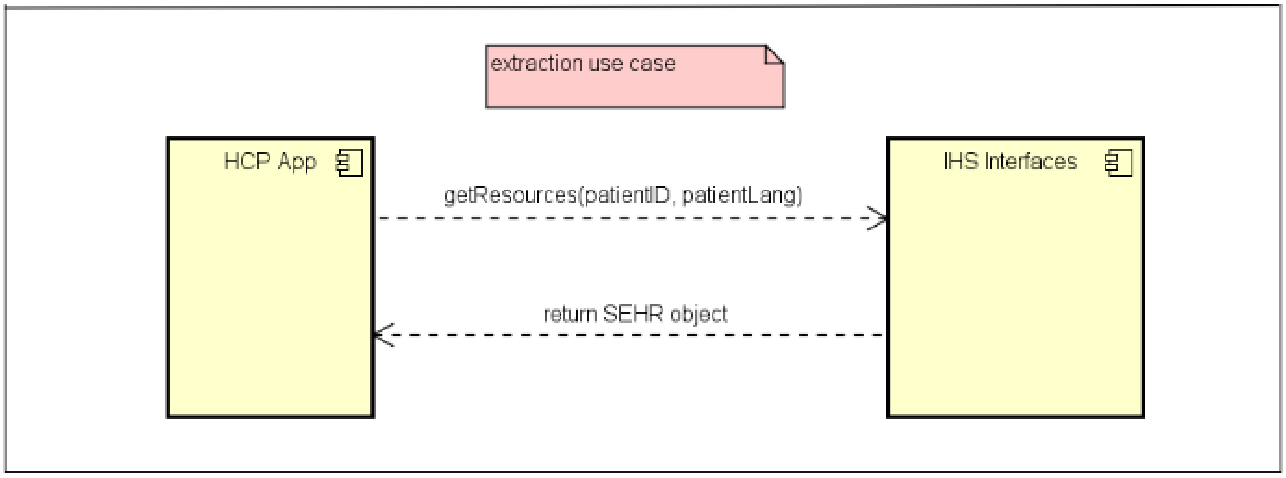

Figure 3: Extraction use case – getRessources method of the HCP App

4 HCP APP DESCRIPTION

4.1 Architectural design of HCP app

The design of the HCP App was made in order to implement the main functionalities specified in the previous section (exchange information with S-EHR Mobile App, S-EHR Cloud and IHS) having in mind to implement a friendly Graphical User Interface that the HCPs will interact with.

According to [D4.1] the D2D protocol defined in the project will have a reference implementation, also developed within the project as a reusable library, having Bluetooth as communication protocol. The integration of this library is the requirement with the biggest impact in the architecture of HCP App. The first option considered for implementing D2D protocol was “Web Bluetooth API” **[WBA]**. Because the specifications of this API are not yet completed, the libraries that implement D2D protocol will be developed as standalone components that will run natively on HCPs terminals. This will require installing the HCP App as a desktop application on health care professional’s workstation (terminal).

From a development point of view, HCP App will use java technologies, thus ensuring Operating System independence and will have direct communications with S-HER Mobile App and IHS as is illustrated in the following figure. The figure also contains the libraries for exchanging information with S-EHR Mobile App and S-EHR Cloud, developed within the project as reference implementation; thus, this implementation of HCP App will show how these libraries can be integrated in any other HCP Application, developed from scratch or by adding new modules to existing ones.

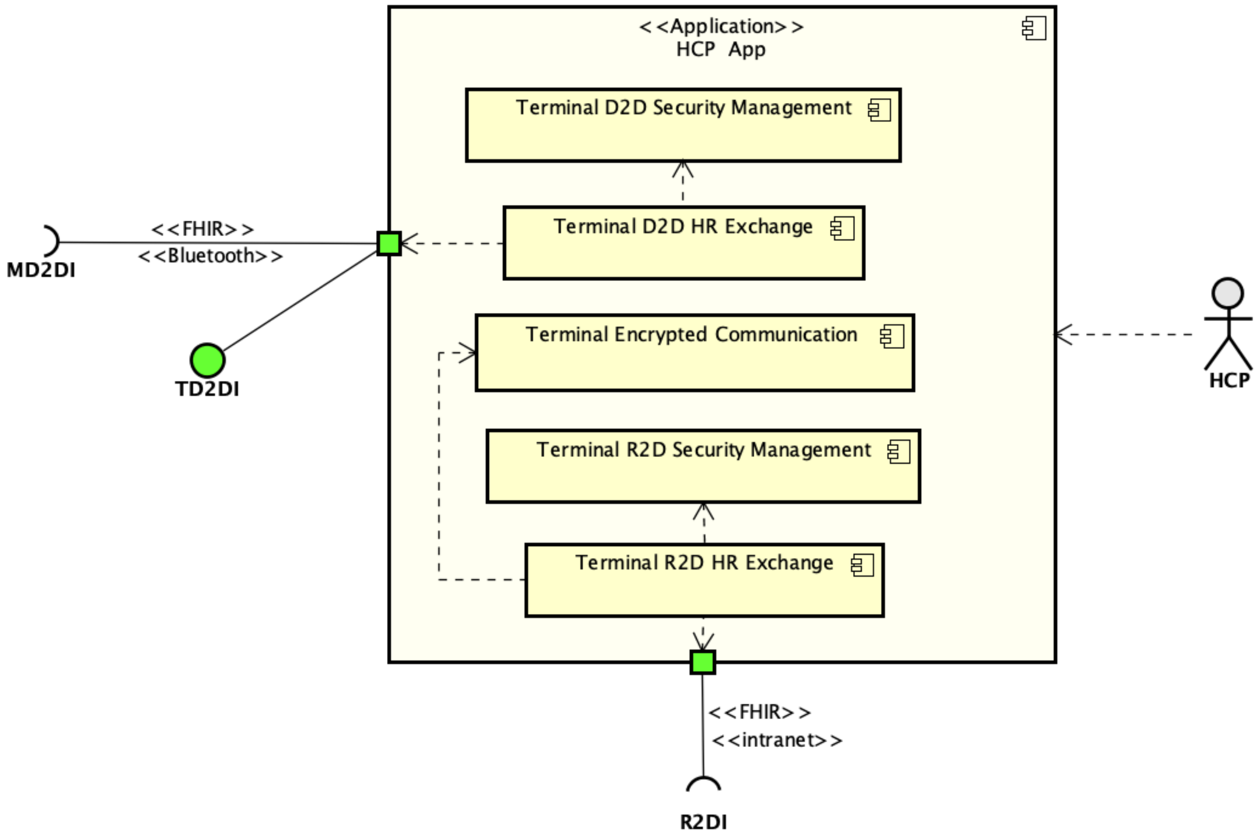

Figure 4: Libraries and interfaces for communication with S-EHR Mobile App and IHS

The libraries developed within the project that will be included in this sample implementation of HCP App are:

- Terminal D2D Security Management – implements the main security functionalities required by the D2D protocol;
- Terminal D2D HR Exchange – extends the D2D security library to offer an implementation of the D2D protocol for the exchange of health data on bluetooth [D4.1];
- Terminal R2D Security Management – implements the main security functionalities (Identity Management, Consent Management, Authorization Management) required by the R2D protocol;
- Terminal R2D HR Exchange – extends the R2D security libraries to offer an implementation of the R2D protocol for the exchange of health data on the internet [D4.1];
- Terminal Encrypted Communication – offers useful functionalities for encrypted exchange of health data;

The entire list of libraries developed in the project as reusable components are described in [D2.4].

Even if the development of libraries that implement the D2D protocol is not based on the WBA, the HCP App developed as an example of implementation will be web based having a three tier architecture being ready to integrate a possible future implementation of D2D protocol based on WBA.
The following figure illustrates the application architecture, including the components needed to meet user requirements.

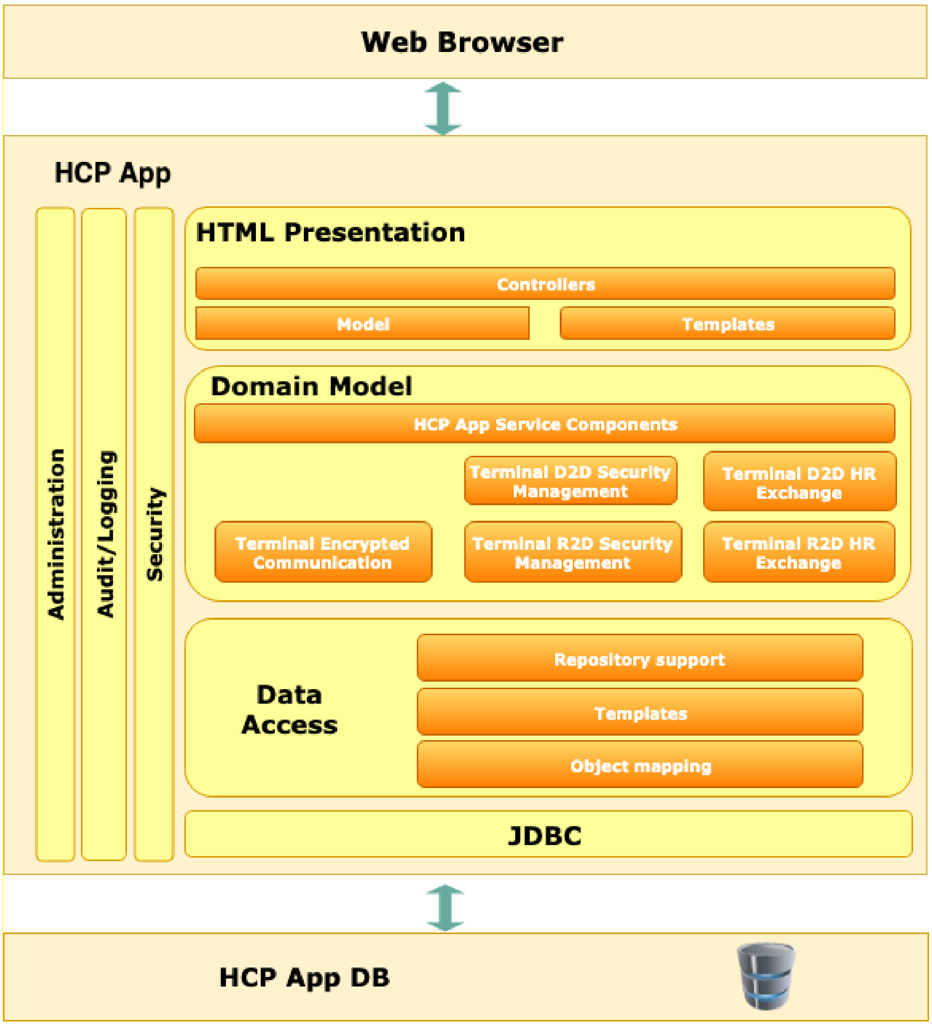

Figure 5: HCP App main architecture layers

HCP App has a multi layer architecture, each implementing specific functionalities, thus:

- HTML Presentation – builds dynamic html pages using a Model View Controller (MVC) implementation and natural templates;
- Domain Model – implements the logic that covers user requirements. The layer offers services to presentation layer integrating at the same time the libraries that implements D2D protocol and R2D protocol;
- Data Access – provides support for accessing database server with Java Persistence API (JPA) and repository pattern;

There will be components for implementing security, auditing and administration.

HCP App developed as an example of implementation will be distributed as a microservice with all components included, installation on the health care professional's terminal consisting of copying a single file.

4.2 Software requirements specification

For this stage of implementation the features and software requirements are based on results coming from [D2.1], [D2.4], [D2.7] and [D4.1].

4.2.1 User flow diagrams (D2D)

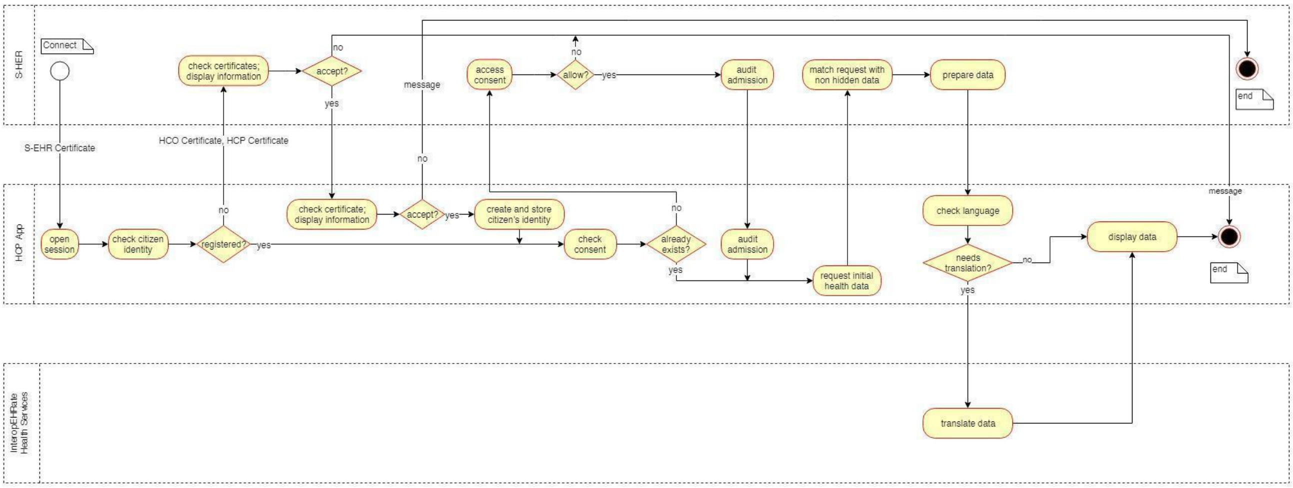

Figure 6: HCP App main architecture layers

This user flow diagram displays schematically how HCP App receives health information from S-EHR Mobile App with prior checking of security information. In order to prove the truth of the information regarding the patient and HCP, will be used some certificates issued by trusted authorities. These certificates will be imported into the two applications as part of the configuration process.

The User Flow Diagram is constructed according to the Scenario S1 from the D2.1. Because it is not decided whether the storage for the application will be a central database or a local database, the following description will only mention DB and the technical aspects of this decision will be mentioned in the following iteration.

The Patient sends his S-EHR Certificate to the HCP App. The HCP App opens a new session and checks whether the received certificate is already registered in the DB. If not, then HCP App will send the current HCP's certificate to the S-EHR Mobile App user and he has a decision of whether to accept the authentication or not. If he denies it, then the flow ends. If he accepts it, then the current HCP will receive a notification asking whether to authenticate the current patient or not. This process is done manually by the HCP by a comparison between the data that the HCP App has received and the identification data that the patient shows to the HCP. If the current HCP denies the authentication, then flow ends. If the HCP accepts the authentication, then a new entrance containing the citizen S-EHR data will be created and stored in the DB. If the user was already registered, then these steps are skipped.

The next step is for the HCP App to check if the consent of receiving the full (or partial) S-EHR health data has already been granted in the last 24 h and from the same terminal that the current HCP App instance is run on. If not, then the flow continues with requesting access to those data by the HCP App. If this request is denied, the flow ends. If it is accepted, then both HCP app and S-EHR App receive a confirmation messaging of audit admission. The request for the initial health data is started. This step is also the next one in the case that the consent was already given in the last 24 h by the same terminal that the current HCP App instance is run on.

The non-hidden data that the HCP App requested will be selected and then sent to the HCP App where they firstly will be checked to correspond to the language(S) the HCP speaks. If translation is needed then the IHS will translate the received data and will display it in the HCP App. If the data does not need translation, then this last step is skipped and the data is displayed directly.

4.2.2 Communication with S-EHR app

This section presents relevant aspects regarding the communication between HCP app and S-EHR app, answering to the specific requirement of Task 5.1: “import / export data directly from/to S-EHR on the smartphone”.
An important part of this Section is equally dedicated to the users' perspective.

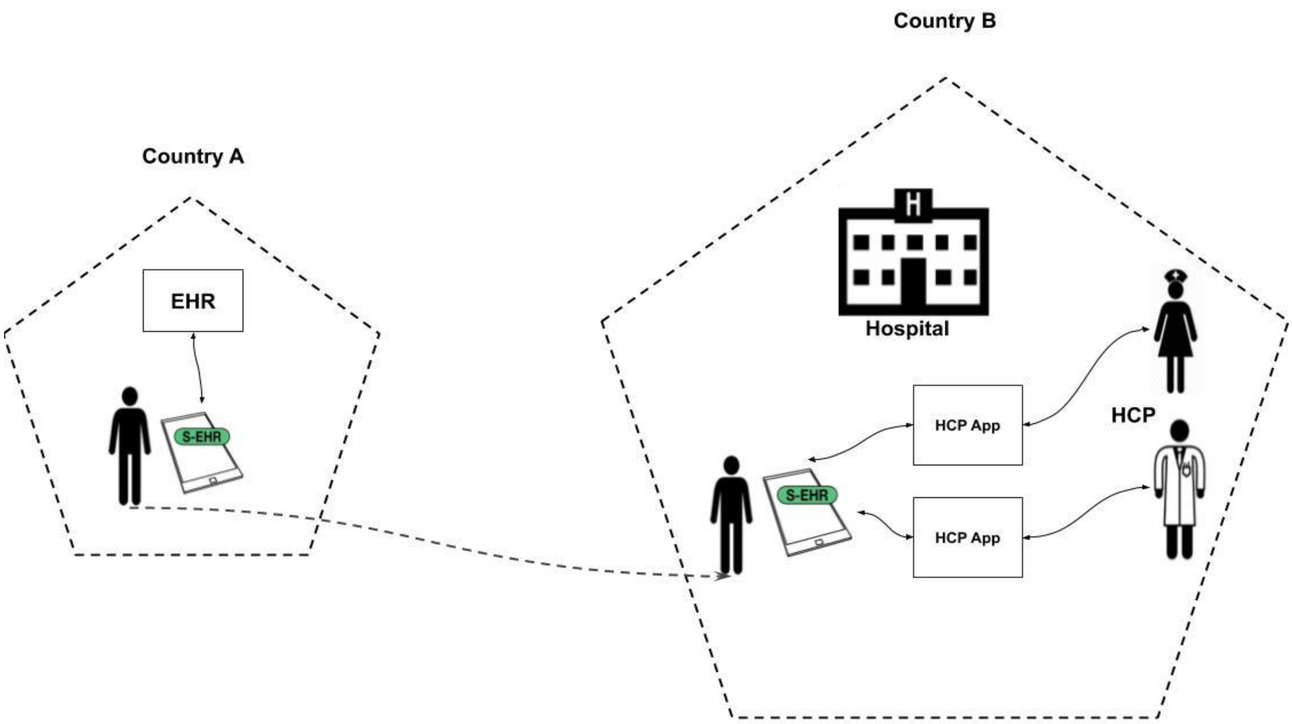

Figure 7: Diagram of the communication between S-EHR App and HCP App

Technically speaking, the communication between HCP App and S-EHR Mobile app is done through the D2D protocol having Bluetooth as backbone communication protocol as is described in [D4.1], some information being reproduced here.

From business point of view, FHIR is used for exchange health information using an interoperability profile that will be defined in [D2.7].

The purpose of the D2D protocol is to design a series of patterns for exchanging messages and healthcare related data between a healthcare practitioner and a citizen, without the usage of internet connection. This protocol is based on short-range wireless technologies and in particular Bluetooth, for secure exchange of health records between a smart device and a health information system. The smart device will use the developed S-EHR application, while the health information system will use the HCP application that will be used by the citizen and the healthcare practitioner accordingly.

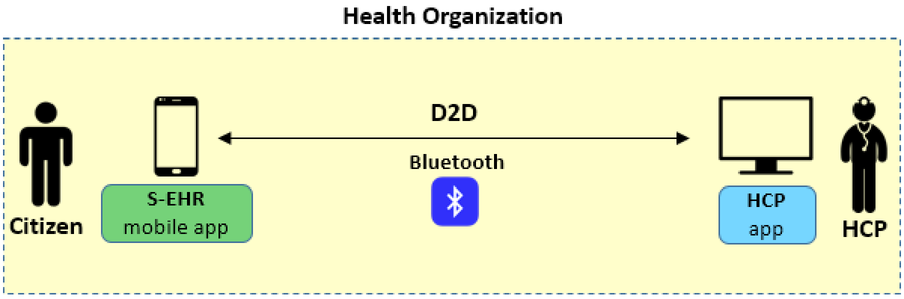

Figure 8: Diagram of the D2D communication

Bluetooth technology is most commonly associated with exchanging data between two bluetooth enabled devices in short distance (±10 meters), through which a bluetooth enabled device as soon as it listens to the initialization advertisement message of a different bluetooth enabled device, it connects to it, being thus able to exchange and display information between them, without needing any other technologies or types of connection (e.g. internet connection). Adopting a similar paradigm, the proposed D2D protocol will facilitate the information exchange between patients (i.e. through smartphones) and healthcare practitioners (i.e. through a desktop computer including a bluetooth adapter), without the usage of central cloud services or any other parties.

The D2D protocol will consist of two different libraries (MD2D and TD2D) that will be designed from the side of S-EHR app and the HCP app. These libraries will be included in both applications accordingly, and will be used by the two main actors of the D2D protocol, the citizens and the HCPs. These two actors will be the only involved ones in the overall interaction, for exchanging the consent of accessing each one's personal data, the healthcare related data, and the evaluation data accordingly.

In order for these two libraries to communicate and interact with each other, two different interfaces will be designed. The first interface (MD2DI) will be responsible for interconnecting and exporting messages and request from the MD2D towards the TD2D, while the second interface (TD2DI) will be responsible for interconnecting and exporting messages and request from the TD2D towards the MD2D.

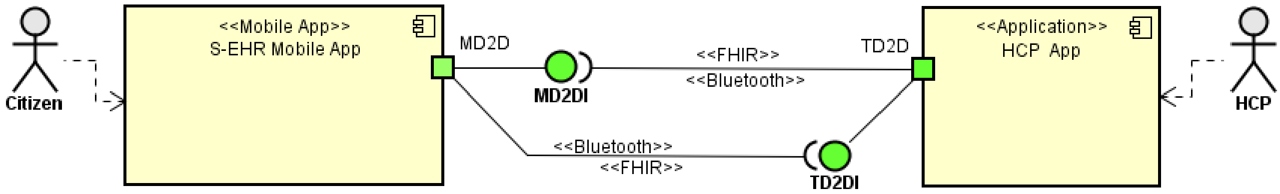

Figure 9: D2D communication interfaces

User perspective

The most relevant and intuitive insight concerning the communication between HCP app and S-EHR app is the visual / graphical illustration of different features of HCP app accessible to healthcare providers.

Methodological approach

This section presents relevant aspects concerning the proposed methodology (Agile) and the principles and techniques of User Centred Design (UCD) applied in this stage of designing the HCP app solution.
HCP app design uses a combination of Agile methodology and UCD (User Centred Design), specific for software development.
Agile and UCD are iterative approaches which perfectly suit the features and objectives of the InteropEHRate project. In this particular stage of UI design, the Agile methodology is applied as an iterative and incremental process, in order to draft the software requirements specification of HCP app.

Wide-range stakeholder studies were conducted as a basis for software development and enabled to identify and range user needs, requirements and specifications to be further used in developing the UI.
Focus groups, personal and group interviews and surveys were performed to test and refine our ideas and concepts by exploring the needs of healthcare providers.
This approach depicted in deliverable D2.1 helped us identify technical constraints and gain an in-depth understanding of our final users.
First requirements were raised from the comprehensive scenarios for (i) Device to Device HR exchange, (ii) Remote to Device HR exchange and (iii) Research protocol. The scenarios enabled the identification of specific features of user interfaces designed for HCP App.

Considering the specificity of our target group, we started to draft and optimise the design of the user interface from the perspective of healthcare providers, based on valuable feedback gathered from the end users within iterative co-design / co-creation sessions.
The UCD approach supposes the involvement of end-users in this stage of design in an iterative manner, in order to adapt the solution continuously, based on their valuable feedback. Using this iterative approach, the application will be built incrementally by adding new features and functionality over several iterations.
Applying the UCD methods during the implementation of the InteropEHRate project, we aim to obtain an increased usability, accessibility, user satisfaction and comfort.

In this stage of design, the software developers and graphic designers use good-quality and context-adapted UI templates/patterns in order to enhance the usability of the solution.
The following section summarises the progress and results of the first development iterations.

Two co-design sessions were already organized in March 2019 and June 2019. Within these meetings with the healthcare partners involved in the project were presented examples of mock-ups for HCP App and valuable feedback was gathered from the healthcare providers.

Hereinafter are presented some suggestive screens reflecting the *iterative approach* of GUI design.

First iteration GUI - March 2019

Figure 10: Healthcare provider screen S-EHR access request

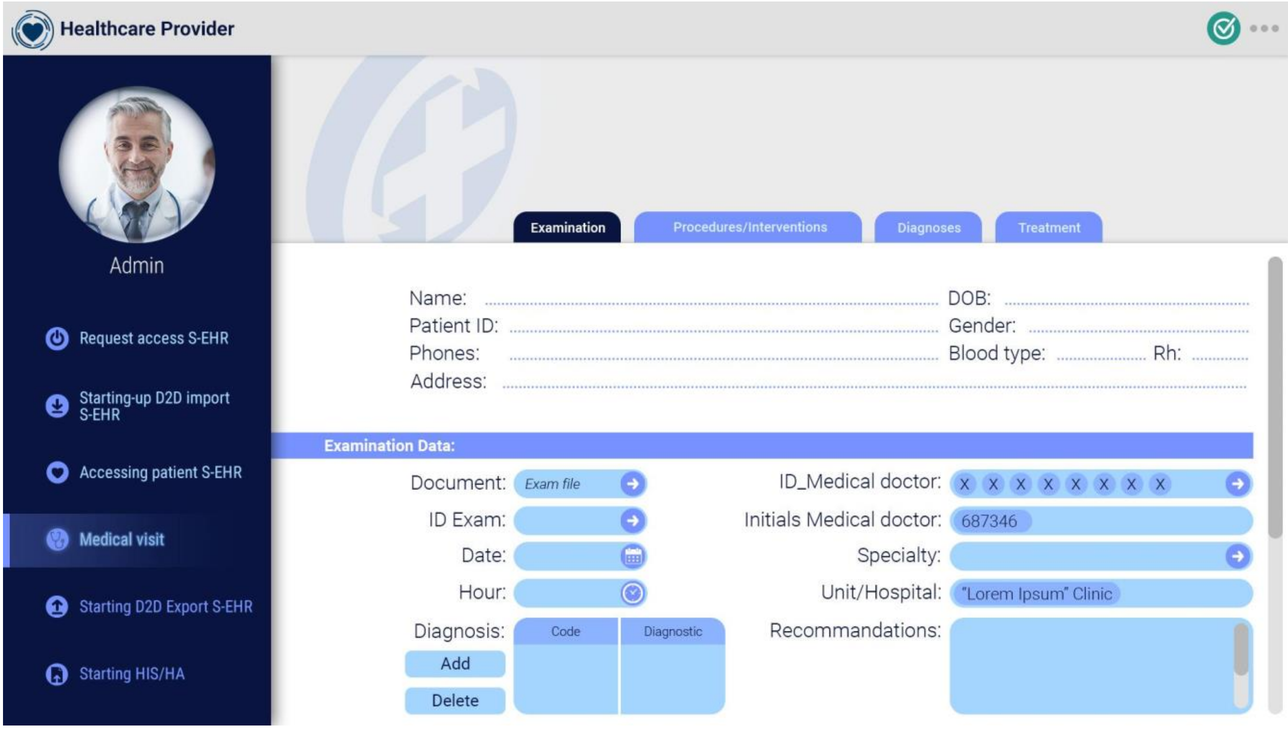

Figure 11: Healthcare provider screen accessing S-EHR data

Second iteration GUI - June 2019

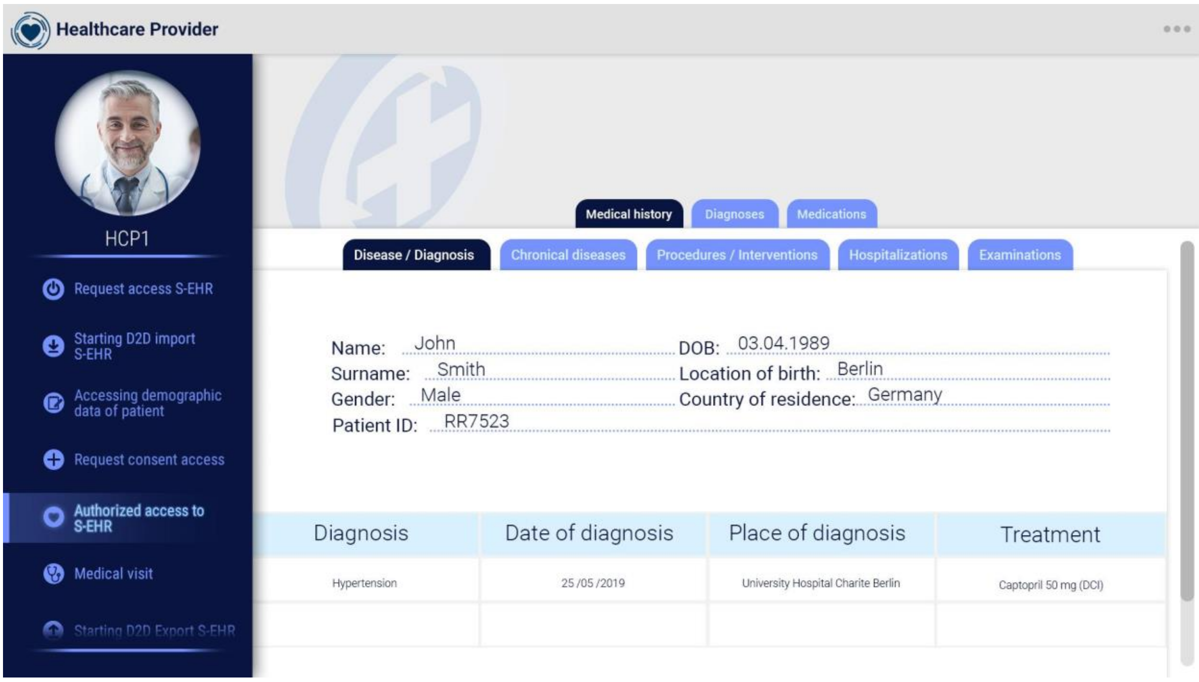

Figure 12: Healthcare provider second iteration screen for accessing S-EHR data

Third iteration GUI - June 2019

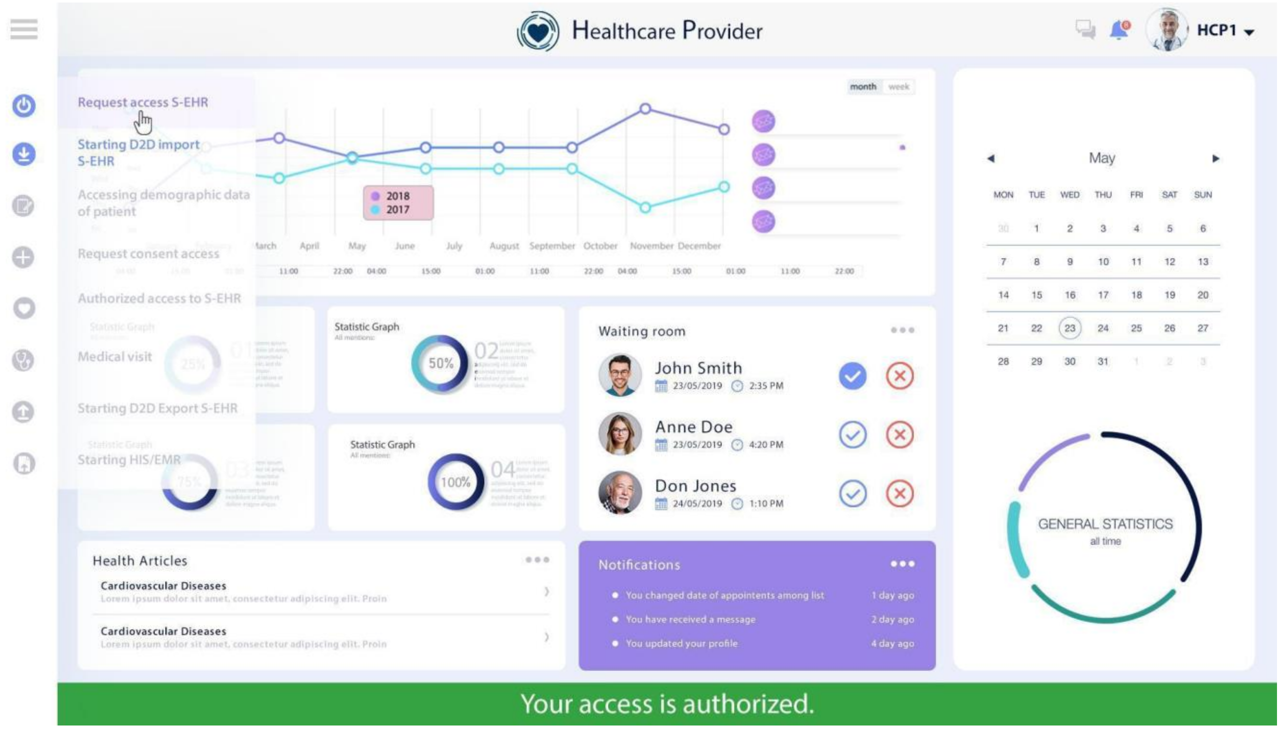

Figure 13: Healthcare provider third iteration screen accessing S-EHR data

Visual approach

Specific mock-ups for user interfaces are designed to transfer the written requirements into a visual representation.

The simplified sketches of the UI focus on functional elements and are used for drafting structure and functionality of HCP App. The deliverable [D6.1] contains the corresponding sketches of the S-EHR Mobile App.

The initial draft of the mock-ups was reviewed by the Consortium members and identified a range of usability features.

Valued feedback from end-users was an essential step to determine the core features of the application.

As presented above, the UI design supposes an iterative approach; the visual representation of HCP app will be drafted incrementally by adding new features and functionality over several iterations.

Hereinafter, are presented a series of mock-ups which reflect the outcome of two co-design iterations where the healthcare partners of the project were involved.

The proposed graphical illustration meets the users' requirements as they were detailed in **Scenario S1 – Device to device HR exchange** in deliverable D2.1 (the first 16 steps) and follows the logical flow of information of this scenario.

The following examples of user interfaces (screens) were associated with the relevant steps of the Scenario S1 to visualize the end users perspective about how to operate in HCP app. The UI corresponding to S-EHR app (patient perspective) are not the subject of this presentation. ***HCP1 - Doctor*** and ***HCP2 - Nurse*** use HCP app to access and operate the patient data (healthcare records) from S-EHR app.

The options of HCP1 and HCP2 when accessing the Main Menu (Dashboard) of HCP app are the following:

- Request access S-EHR
- Starting D2D import S-EHR
- Accessing demographic data of patient
- Request consent access (request *the consent of patient* to access his health history from S-EHR app)
- Authorized access to S-EHR (*access health data / health history of patient*)
- Medical visit
- Starting D2D export S-EHR
- Starting HIS / EMR (*connection with existing health applications - Hospital Information System / Electronic Medical Record*).

1) The HCP1 asks the patient if he/she owns a S-EHR. As the patient answers yes, the HCP1 asks him/her to approach his/her Smart Device to the HCP1 terminal for the identification by means of the D2D protocol.

Figure 14: HCP1 screen S-EHR access request pending

2) As soon as the connection is successfully completed, the patient may see on the screen of his/her Smartphone the data describing the identity of the Health Organization (name, address, etc.) of the HCPs.

3) The patient recognizes that the description corresponds to organization where he/she is in that moment, so he/she approves the connection to share his/her identifying data with the HCP1.

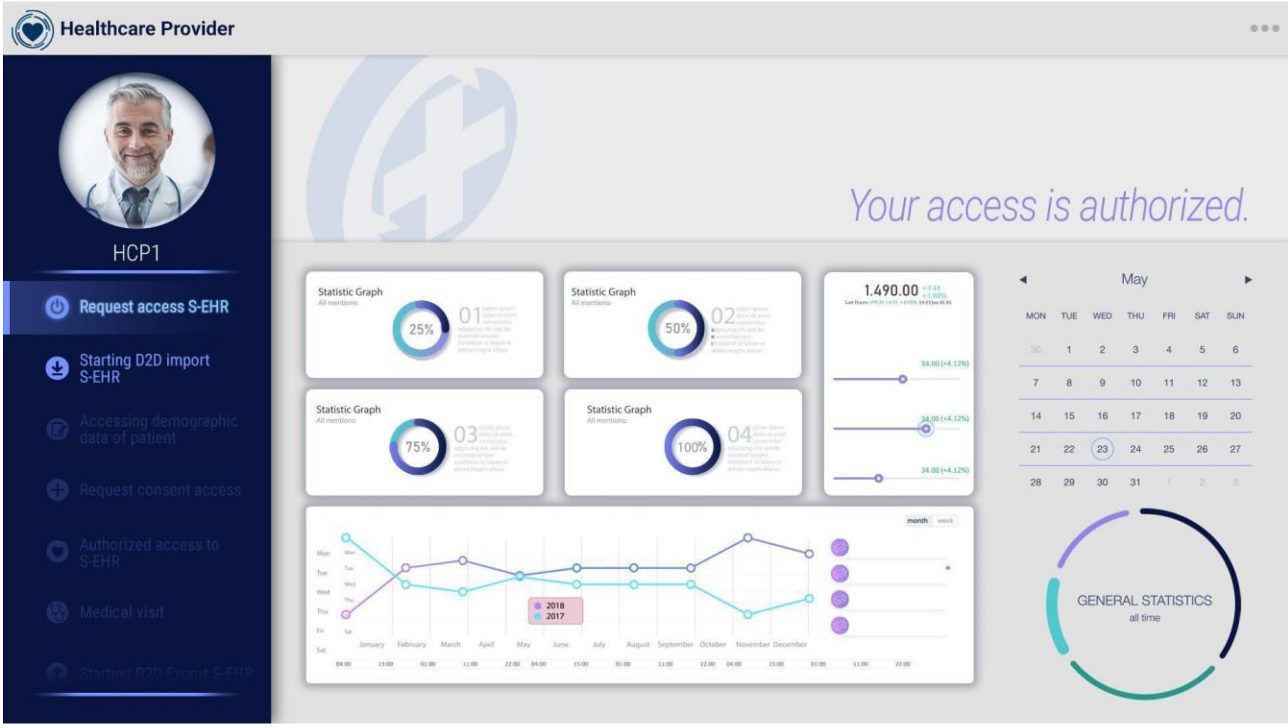

Figure 15: HCP1 screen - S-EHR access authorized

Figure 16: HCP1 screen - S-EHR data exchange

Figure 17: HCP1 screen - S-EHR connection completed

4) As soon as the connection has been approved by the patient, the HCP1 may see on the screen of his/her HCP app the name, surname, date of birth, location of birth, gender, country of residence (corresponding to the identity document) and social security number (or equivalent identifying data).

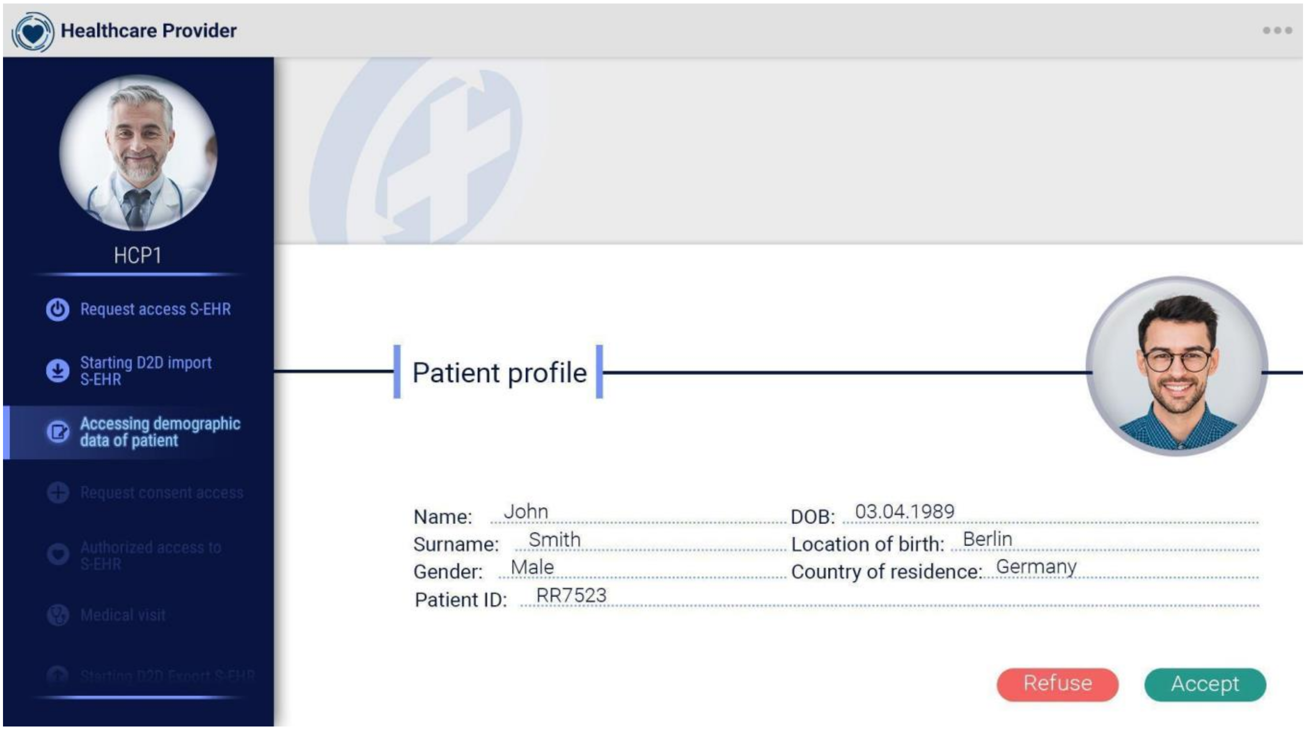

Figure 18: HCP1 screen - patient profile from S-EHR

5) The HCP1 asks the citizen for his/her identity document and compares it with the information shown on the HCP App.

6) As the data are correct, the HCP1 confirms, using the HCP app, the identity of the patient (button "Accept"). If data are not corresponding, Scenario stops here (button "Refuse").

7) HCP1 contextually (i.e. implicitly) asks the citizen for temporary (limited to this encounter) consent for the healthcare organization of the HCPs to:

- download data from S-EHR app
- upload the updated/acquired data back to S-EHR app
- store, for the amount of time required and allowed from the national law, the downloaded data on the systems controlled by the authorized healthcare organization.

Figure 19: HCP1 screen - Request consent access to S-EHR (pending)

8) The admission data are stored by the HCP app for future traceability.

9) Using his/her phone, the patient sees on the S-EHR the description of the healthcare organization that just identified him/her.

10) He/she sees on screen the request for consent for the admitting organization to download data from the S-EHR app and upload the updated/acquired data back to the S-EHR app.

11) By means of the S-EHR the patient gives his/her consent, implicitly giving the default view/transmission permissions he/she may have previously configured on the S-EHR (see the assumptions under 5.1). Every other HCP scoped by the Healthcare Organization and involved in patient care/treatment are authorized to access S-EHR.

12) The consent is transmitted to the HCP App and recorded by it for future traceability.

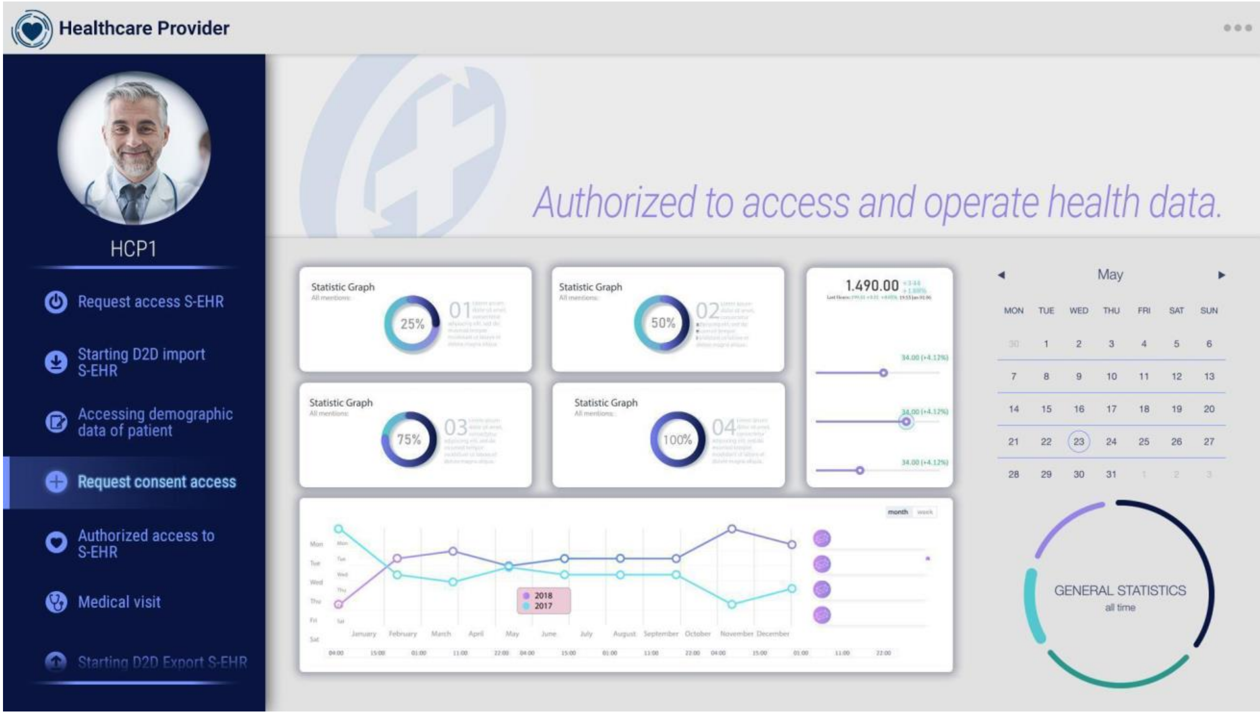

Figure 20: HCP1 screen - Authorized request to access S-EHR data

13) A preconfigured (by the HCP on the HCP App) dataset of patient’s data are transferred from the patient’s S-EHR app to the HCP App in a few seconds (5 to 10), up to a couple of minutes if the amount of requested data is relevant (10-20 Mb). Admission is now completed, patient move on to consultation.
From this on, patient interacts with HCP2.

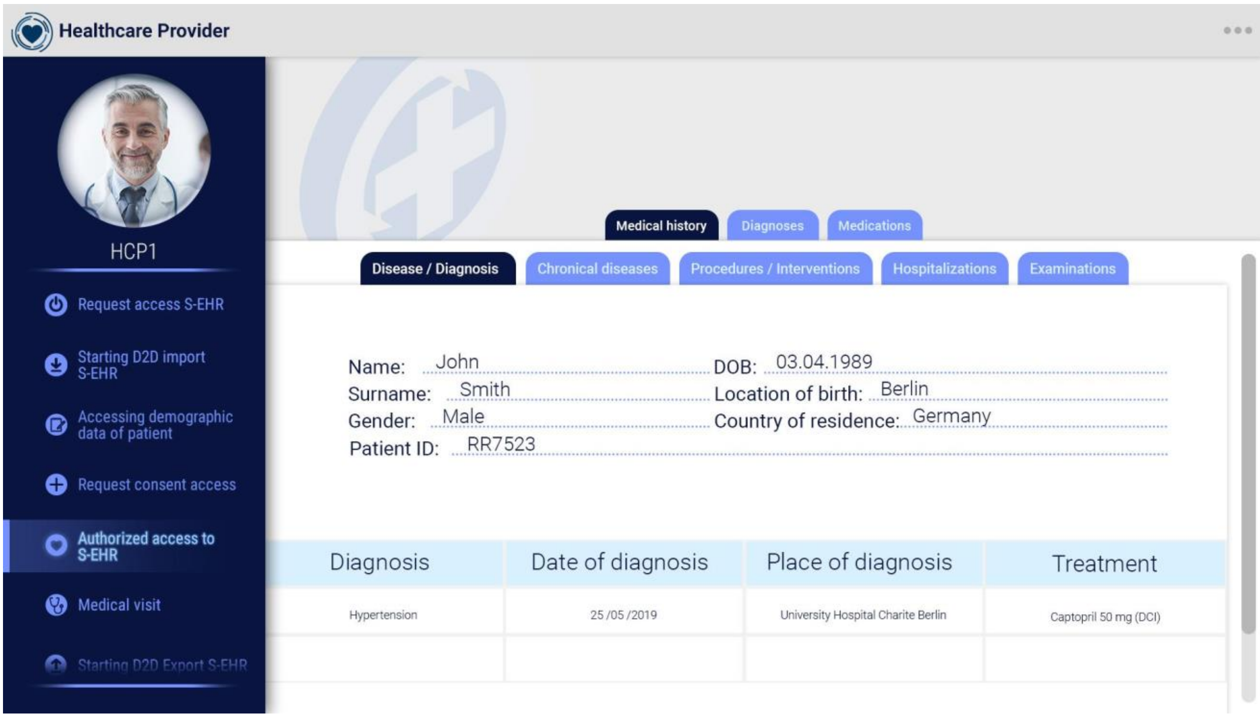

Figure 21: HCP 1 screen - Patient Data

14) Downloaded patient's data may be visualized, using the HCP App, by the HCP2, which is currently authorized by the healthcare organization to treat the data of that patient (i.e. involved in patient's treatment process).

Figure 22: HCP2 screen - S-EHR access request

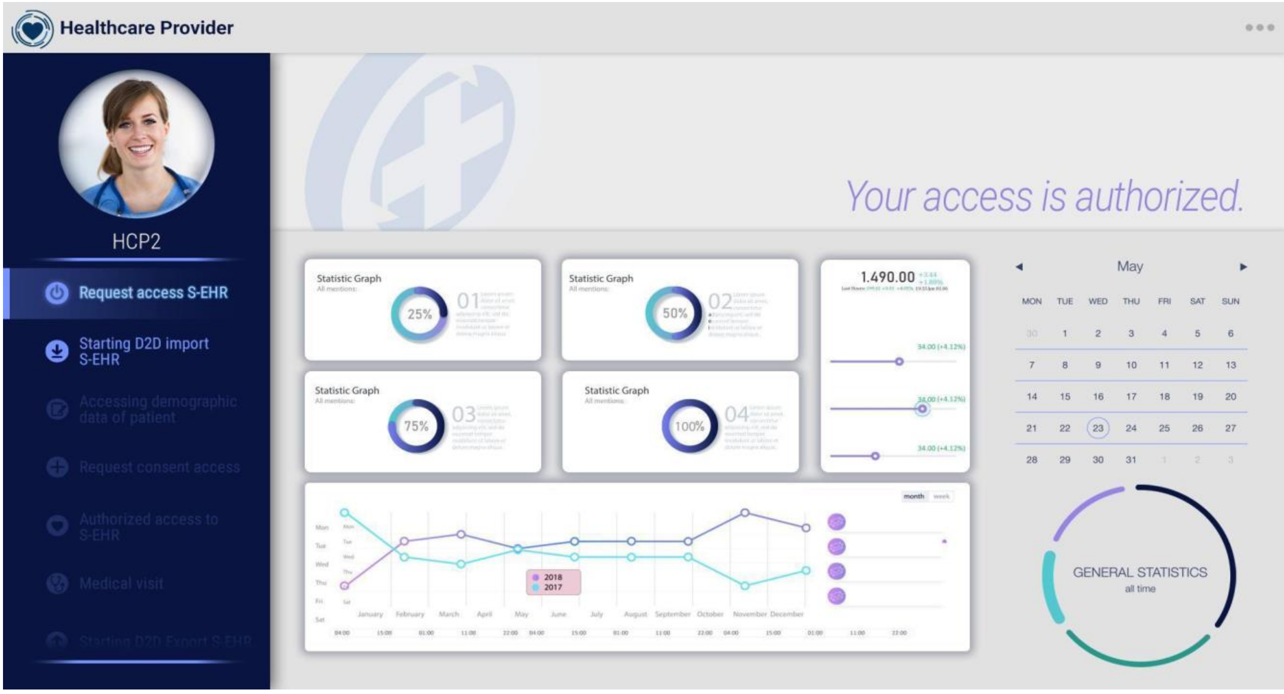

Figure 23: HCP2 screen - Authorized request to access S-EHR data

Figure 24: HCP2 screen - S-EHR data exchange

Figure 25: HCP2 screen - S-EHR access connection completed

15) Downloaded patient's data are translated into HCPs natural language. HCPs natural language is the one officially related to the Healthcare provider.

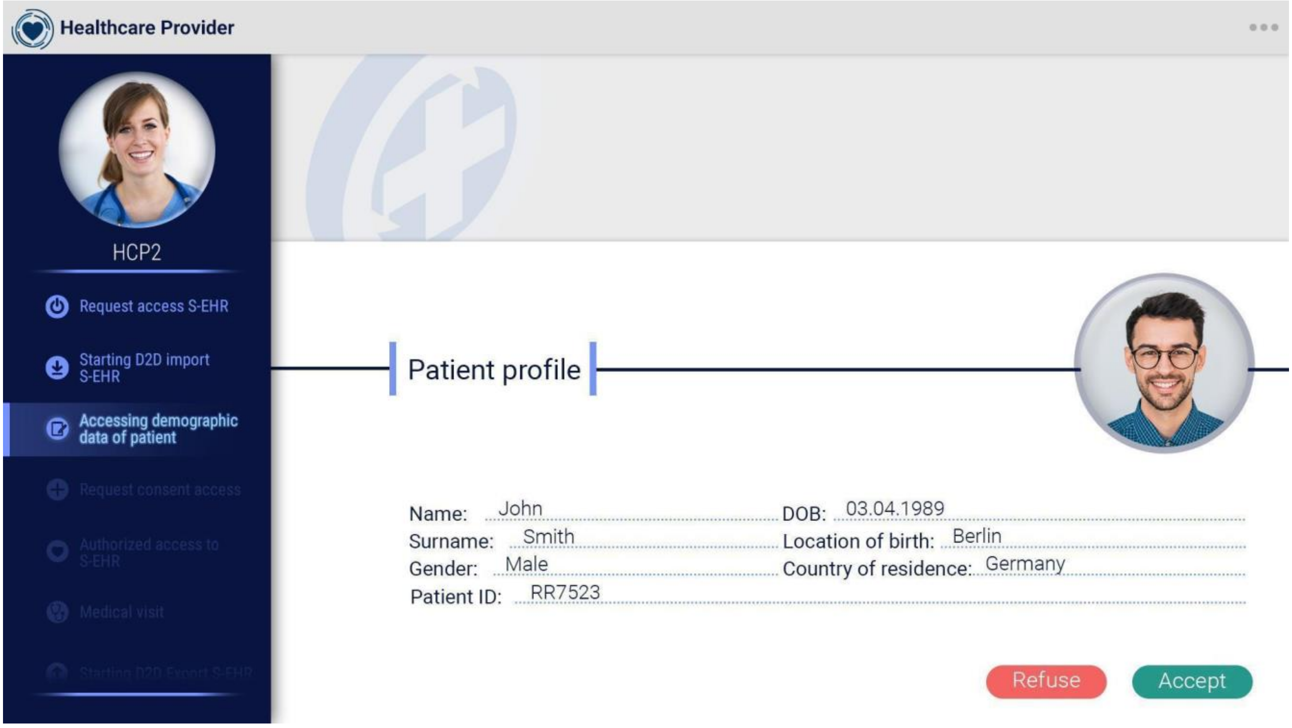

Figure 26: HCP2 screen - S-EHR patient profile data

Figure 27: HCP2 screen - Request consent access to S-EHR (pending)

Figure 28: HCP2 screen - S-EHR access authorized

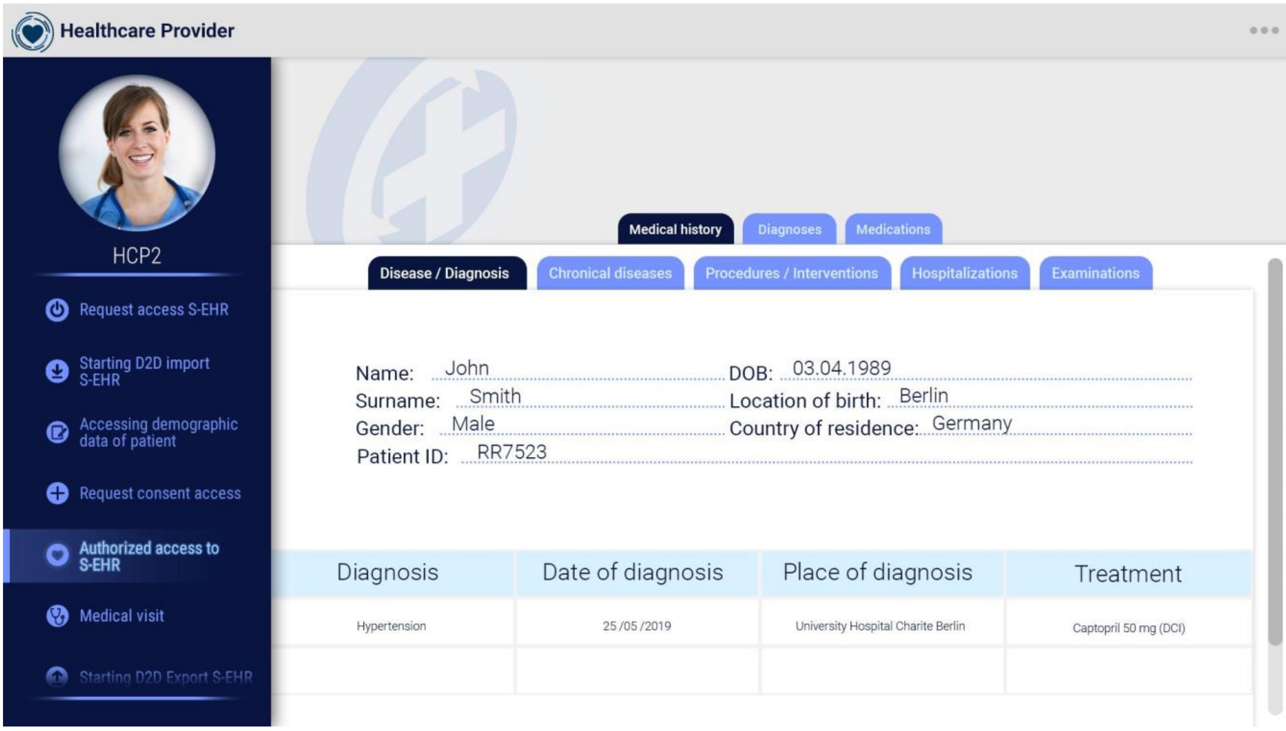

Figure 29: HCP2 screen - S-EHR patient data

16) During the Medical visit (consultation) HCP2 measures vital signs, body weight, BP, pulse, respiratory rate, SPO2, Temp, AVPU and alertness. Data are entered in HCP App (menu Procedures / Interventions).

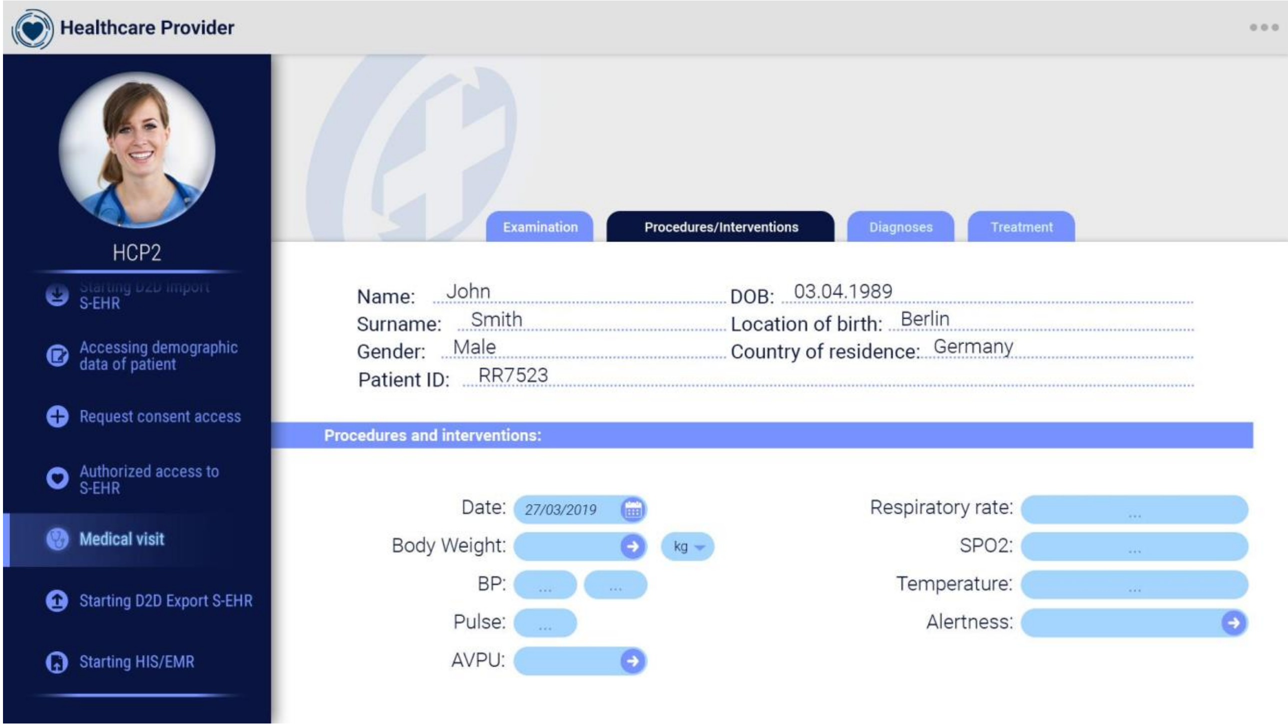

Figure 30: HCP2 screen - S-EHR medical visit data entry

4.2.3 Technological approach

Because a programming language with strong support for web development and that is able to produce an application that portable between different operating systems, Java has been chosen as the main programming language for the development of the HCP App.

Until recently, any web application (independent of the programming language) needed to be put on a dedicated server (and not on the local computer that it runs). The server is either a real computer, with independent hardware, or a virtual machine that runs on an even stronger computer, but that it has multiple usages. Although it is possible for the computer that the application resides to act as the server of the application, it is advisable against this approach as servers should run 24 hours per day, 7 days per week so that the web client (browsers) would be limited if the server is not always reachable.

Of course that was the traditional approach and since then, new technologies have emerged. Because of the size of the project and because it is believed that the new technologies approaches that have emerged are better, cloud application development approach has been chosen. The best implementation of the cloud application development approach is using microservices.

Using microservices allows construction of the application based on an architectural style that structures the application as a collection of services which are highly maintainable and testable, small, loosely coupled, reusable, independently deployable and disposable.

In Java there are two competing service development framework:

- Java EE (Enterprise Edition)
- Spring Framework (Including Boot and Cloud).

Although both Java EE and Spring work on the same core API (Servlets, JPA, JMS, BeanValidation etc), the main difference is what is connecting these components, Spring or AppServer, and how fast is a developer able to configure his application to run with microservices. Because of the faster and easier configuration of the and a more free way of the programming model, the HCP App will be based on Spring application framework.

Spring is an application framework and inversion of control container for the Java platform. The core of the Spring Framework can be used for building any Java application, so expansions for building web applications are a part of Spring. Spring is basically a structure for building reliable application with exceptionally decoupled frameworks. Spring is one of the best systems for building web applications. And to make the code even faster to write, Spring Boot is going to be the base of the HCP application. Spring Boot aims to shorten the code length and provide the developer with the easiest way to develop a web application. Spring Boot has provided an opinionated approach to developing microservices. In the development of the HCP App, certain modules of Spring/Boot Framework will be used.

Spring Boot JPA focuses on using JPA to store data in a relational database. Java Persistence API (JPA) is a Java application programming interface specification that describes the management of relational data in applications.

Thymeleaf is a Java-based library used to create a web application. It provides a good support for serving a XHTML/HTML5 in web applications.

Spring Web MVC framework models the web application according to the Model-View-Controller (MVC) architecture and provides components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic and UI logic), while providing a loose coupling between these elements.

Spring provides mostly the back end of the HCP application. For the front end part, a suite of web technologies will be used. This includes CSS, Bootstrap and jQuery.

Bootstrap is a CSS framework that is mostly known for responsiveness and fastness of developing front-end web application. It contains CSS and JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

jQuery is a fast, small, and feature-rich JavaScript library, designed to simplify HTML DOM tree traversal and manipulation, as well as event handling and CSS animation.

5 CONCLUSIONS AND NEXT STEPS

The current deliverable aims to present the preliminary software requirements specification and the design of the HCP App solution used by healthcare professionals for accessing and creating health data of foreign patients within the InteropEHRate project.

Taking into consideration the particular requirements of Task 5.1, the deliverable encompasses the preliminary architectural design elements and UI design of HCP App that will then be detailed and updated incrementally in the next deliverables – D5.4, D5.5 and D5.6.

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

- Architectural view
- Design / Technical view
- End User view.

The deliverable presents the preliminary technical characteristics / features of HCP App, while the details and iterative completions of the technical solution will be described in the next deliverables D5.2 and D5.4.

REFERENCES

- **[D2.1]** InteropEHRate Consortium, *D2.1: User requirements for cross-border HR integration – V1*, InteropEHRate project. www.interopehrate.eu/resources

- **[D2.4]** InteropEHRate Consortium, *D2.4: InteropEHRate Architecture - V1*, InteropEHRate project. www.interopehrate.eu/resources

- **[D4.1]** InteropEHRate Consortium. *D4.1: Specification of remote and D2D protocol and APIs for HR exchange – V1*, InteropEHRate project. www.interopehrate.eu/resources

- **[D2.7]** InteropEHRate consortium. *D2.7: FHIR profile for EHR interoperability - V1*. InteropEHRate project. www.interopehrate.eu/resources

- **[FHIR standard for health care data exchange]** published by HL7®. https://www.hl7.org/fhir/index.html

- **[Human-Centered Software Engineering – Integrating Usability in the Software Development Lifecycle]**, Ahmed Seffah, Jan Gulliksen, Michel C. Desmarais, 2005, Springer. https://books.google.ro/

- **[Bringing User Centered Design to the Agile Environment],** Antony Colfelt, 2010. http://boxesandarrows.com/bringing-user-centered-design-to-the-agile-environment/