CONNECARE

WP4 – SELF-MANAGEMENT AND MONITORING

D4.3: ADVANCED MONITORING TOOLS

H2020-EU.3.1: Personalised Connected Care for Complex Chronic Patients

Project No. 689802

Start date of project: 01-04-2016

Duration: 45 months

Project funded by the European Commission, call H2020 – PHC - 2015	
☑ **PU**	**Public**
☐ PP	Restricted to other programme participants (including the Commission Services)
☐ RE	Restricted to a group specified by the consortium (including the Commission Services)
☐ CO	Confidential, only for members of the consortium (including the Commission Services)

Revision: 01

Date: 30/06/2019

Document Information

Project Number	689802	**Acronym**	CONNECARE
Full title	Personalised Connected Care for Complex Chronic Patients		
Project URL	http://www.CONNECARE.eu		
Project officer	Carola Carstens		

Deliverable	**Number**	4.3	**Title**	Advanced Monitoring Tools
Work Package	**Number**	4	**Title**	Self-management and monitoring

Date of delivery	**Contractual**	MONTH 39	**Actual**	MONTH 39
Nature	Prototype □ Report ☑ Dissemination □ Other □			
Dissemination Level	Public ☑ Consortium □			

Responsible Author	Eloisa Vargiu	**Email**	eloisa.vargiu@eurecat.org
Partner	EURECAT	**Phone**	

Abstract	This deliverable illustrates the tools (services) that have been investigated and developed to be part of the SMS in order to perform monitoring of advanced activities. The underlying model, common for the basic tools, has been already introduced in D4.2. Each implemented service has been then described at a high-level, whereas in the annexes technical details of each are given.

This deliverable reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains. (Art. 29.5)

CONNECARE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 689802 (Art. 29.4)

Table of contents

EXECUTIVE SUMMARY 6

1. THE FINAL μ-SERVICES-BASED ARCHITECTURE 9

2. THE ADVANCED MONITORING SERVICES 11

2.1 HEALTH STATUS 11

2.1.1 Requirements 11

2.1.2 Implementation 11

2.2 DRUG PRESCRIPTION 13

2.2.1 Requirements 14

2.2.2 Implementation 15

3. THE UPDATED BASIC MONITORING SERVICES 17

3.1 PHYSICAL ACTIVITIES 17

3.1.1 Updated Requirements 17

3.1.2 Implementation 18

3.1.3 Improvements 19

3.2 SLEEPING 20

3.3 QUESTIONNAIRES 24

3.4 SIMPLE TASKS 26

4. AUXILIARY SERVICES 28

4.1 MESSAGING 28

4.1.1 Requirements 28

4.1.2 Implementation 28

4.2 ADVICES 31

4.2.1 Automatic advices 33

4.2.2 Personalized advices 34

4.2.3 Educational material 35

4.2.4 Health educational videos 35

4.3 ALERTS 36

4.4 NOTIFICATIONS ... 36
4.5 THIRD PARTY ... 38
5. CONCLUSIONS ... 39
6. ANNEXES ... 40
6.1 HEALTH STATUS ... 40
6.1.1 Data Model ... 40
6.1.2 API Definition ... 44
6.2 DRUG PRESCRIPTIONS ... 51
6.2.1 Data Model ... 51
6.2.2 API Definition ... 51
6.3 PHYSICAL ACTIVITY MONITORING ... 56
6.3.1 Data Model ... 56
6.3.2 API definition ... 62
6.4 SLEEPING ... 71
6.4.1 Data Model ... 71
6.4.2 API Definition ... 80
6.5 QUESTIONNAIRES ... 96
6.5.1 Data Model ... 96
6.5.2 API definition ... 99
6.6 SIMPLE TASKS ... 117
6.6.1 Data Model ... 117
6.6.2 API Definition ... 117
6.7 MESSAGING ... 124
6.7.1 Data Model ... 124
6.7.2 API Definition ... 127
6.8 ADVICES ... 141
6.8.1 Data Model ... 141
6.8.2 API Definition ... 141
6.9 ALERTS ... 149
6.9.1 Data Model ... 149

6.9.2 API Definition 149

6.10 NOTIFICATIONS 153

6.10.1 Data Model 153

6.10.2 API Definition 153

6.11 THIRD PARTY 156

6.11.1 Data Model 156

6.11.2 API Definition 157

Executive Summary

This deliverable focuses on the advanced monitoring tools that have been defined, implemented, and deployed in CONNECARE.

According to the requirements and the work done during the 1st period, the following services have been defined, developed, and are now integrated in the current version of the SMS: health-status monitoring, drug prescriptions (previously referred as pharmacological adherence). The health status monitoring service has been improved to allow input of the measurements through the Nokia/Withings and also manual data input, in case the patient is using a device without WI-FI/Bluetooth connection (required by IRBLL for some kind of measurements, as the temperature, and by ASSUTA for issues regarding the Israel market in which Nokia/Withings devices cannot be sold). Moreover, this service has been also improved thanks to the reminders and push notifications now in the SMS app. The service of drugs prescription has been integrated in the SMS app and enables pharmacological adherence follow-up. The list of drugs of each site are available in their original language and clinicians may prescribe them according to the patients' needs.

Due to the nature of CONNECARE and its iterative prototyping approach, the basic monitoring services (reported in D4.2) have been continuously updated. The physical activity service and the sleeping service were updated to take into account the decision of clinicians in IRBLL and UMCG not to use the Nokia/Withings wristbands. At the moment the Fitbit wristbands are supported, and the integration of the LifeVit wristband, required by IDIBAPS, has just started. Moreover, due to the termination of IPHEALTH, the sleeping service, the questionnaire service, the messages service, and the advices service have been totally redefined and implemented from scratch. As for the questionnaire service, all the sites listed the required questionnaires for each case study. Firstly, those questionnaires common to all sites have been implemented (i.e., SF-12 and EQ-5D) and then, all the required "site-dependent" questionnaires have been added. In addition to the questionnaire service, the auxiliary services for messaging and advices have been totally re-implemented due to the termination of IPHEALTH. Those services are now integrated in the current version of the SMS.

A totally new service has been added, as required by ASSUTA, it is called "simple tasks". In fact, apart from rehabilitation in hospitals and specialized centres, clinicians agree that daily life activities performed at home radically improve patients' recovery. With this aim, clinicians may prescribe simple tasks to be performed during the day (e.g., dancing, cooking, and reading). Moreover, depending on the disease and the kind of patient (e.g., in case of elderly people) the professional may ask to perform healthy activities such as drinking or eating a fruit. Through the SMS, the patient can accept or reject the request and report when the activity has been performed.

Two auxiliary services have been also implemented and are available in the SMS app: notifications and alerts. Notifications are generated by any of the services and transmitted to the patient (e.g., a new prescription has been generated). Through this service, the corresponding message is sent to the app for its visualization. Moreover, complex notifications that come from interactions with the patient's monitoring services may also be sent. Monitored data are continuously analysed at runtime. As soon as a patient's monitoring service finds an anomaly, it interacts with the alerts service and a suitable alert message is sent to both the patient (to be aware of the issue) and the clinical staff (to be informed and act accordingly). In particular, anomalies are triggered any time data gathered from the patient (e.g., through the medical devices or a questionnaire) exceeds a given threshold defined at prescription time (e.g., a critical heart rate value).

Finally, in the 1st period, IPHEALTH started working on the integration of their nutritional monitoring app (Vitalinq). Due to IPHEALTH termination and according to the revised requirements by clinicians, the consortium agreed on not implementing this service.

The following deliverables are highly recommended to be read:

Number	Title	Description
D4.1	First self-management system	This document describes the first version of the self-management system (SMS) as a study release to be used during the clinical studies by the patients. The document presents the architecture, development phases and deployment of the system, and the requirements requested by the patients and professionals.
D4.2	Basic monitoring tools	This deliverable illustrates the tools (services) that have been investigated and developed to be part of the SMS in order to perform monitoring of basic activities. The underlying model, that will be common also for the advanced and assistive tools, has been first introduced in order to give the big picture of how monitoring is performed in CONNECARE. Each implemented service has been then described at a high-level, whereas in the annexes technical details of each are given.
D6.1	Study release feasibility for the three clinical studies	The CONNECARE document D6.1 covers the operational aspects required to: i) Initiate the implementation studies at site level; ii) Do a proper follow-up of their progress until the final release of the system at the end of the second co-design period; iii) Perform assessment of the five main dimensions of the project (1. Service workflows design & cost-effectiveness; 2. Technological developments; 3. Health risk assessment & service selection; 4. Innovative assessment aspects; and 5. Transferability analysis & service adoption); and, iv) Prepare the elements required for accomplishment of Tasks 7.4 and 7.5

		(Recommendations of final services and proposals for scale-up integrated care) which constitute the core activity of the third co-design period, from M36 to M42.

1. The Final µ-services-based Architecture

As described in deliverable D4.1 “First Self-Management System”, the back-end of the SMS is based on a µ-services architecture. The corresponding SMS app, installed in the patient's smartphone, allows the patient to monitor her/his activity, receive recommendations to improve the treatment and to be encouraged in following it, access to her/his data and information, as well as communicate to the medical staff in charge of the case. According to the closed-loop approach described in D4.2 “Basic Monitoring Tools”, it is the medical staff who prescribes any kind of task to be performed by the patient. Data gathered through interaction of the patient with the system (automatically through the devices or manually through direct input) are sent to the cloud where are processed and analysed in order to give the corresponding information to both patient and professionals. In fact, on the one hand, an activity performed by the patient may require automatic recommendations to be sent to her/him. On the other hand, anomalies or low-adherence may require generation of specific alerts to be sent to the patient for her/his empowerment as well as to the professionals for better follow-up. In so doing, keeping informed the professionals about the treatment follow-up gives the possibility to change something in case of worsening of the health status of the patient, preventing and avoiding re-hospitalization. Moreover, in case high improvements are registered, informing the professionals could help in better follow-up reorganizing the care plan, if needed.

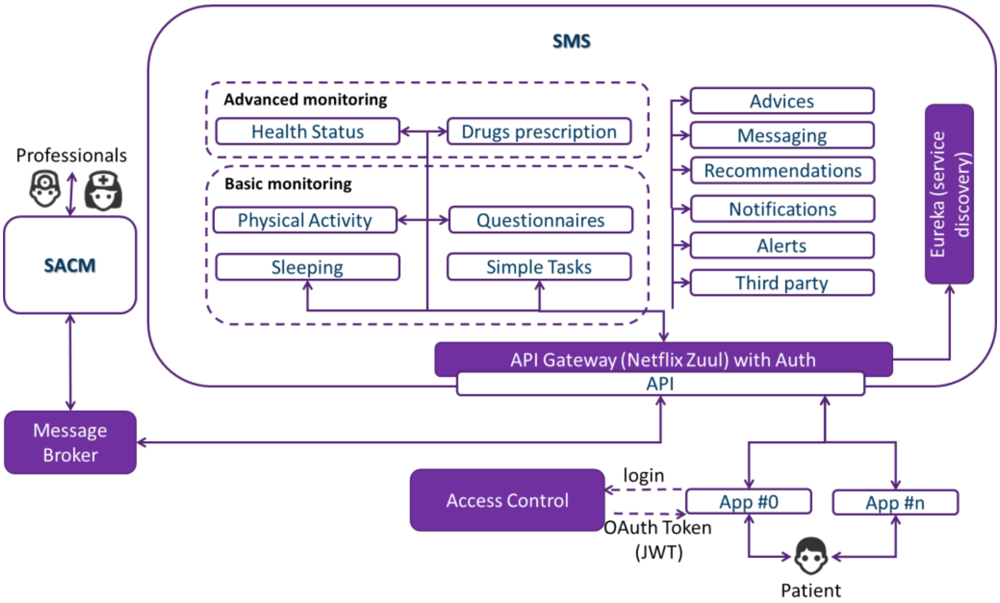


Figure 1 - The µ-service architecture

Figure 1 sketches the final architecture, composed of a set of µ-services each of them in charge of performing a specific task. Complex tasks may be achieved by the collaboration, interaction, and coordination of one or more µ-services. When changes are needed in a µ-service, they do not affect the rest of the system. In this way, after the termination of IPHEALTH, the services that were in charge of them have been developed from scratch by EURECAT without affecting the overall system. On the other hand, when a new functionality is required, a new µ-service (or set of µ-services collaborating together) is defined and simply added to the system. In so doing, the proposed solution is open to changes and improvements.

For locating the µ-services, with the purpose of load balancing and fail-over of middle-tier servers, Netflix-Eureka service discovery is adopted[1]. Eureka is a REST based µ-service for locating µ-services with the purpose of load balancing and failover of middle-tier servers (the Eureka Server). Eureka also comes with a client component (the Eureka Client), which makes interactions with the µ-services much easier. The client also has a built-in load balancer that does basic round-robin load balancing.

As a front door for all requests from the app installed in the patient's smartphone and the µ-services, we rely on Netflix Zuul[2]. Zuul is the front door for all requests from devices and web sites to the backend of the Netflix streaming application. As an edge µ-service application, Zuul is built to enable dynamic routing, monitoring, resiliency, and security. Zuul uses a range of different types of filters that enables to quickly and nimbly apply functionality to the edge µ-service. These filters help performing the following functions: authentication and security, identifying authentication requirements for each resource and rejecting requests that do not satisfy them; insights and monitoring, tracking meaningful data and statistics at the edge in order to give an accurate view of production servers state; dynamic routing, dynamically routing requests to different backend clusters as needed; stress testing, gradually increasing traffic to a cluster in order to gauge performance; load shedding, allocating capacity for each type of request and dropping requests that go over the limit; and static response handling, building some responses directly at the edge instead of forwarding them to an internal cluster.

Management of roles, rights, authentication control, and login is implemented by relying on Spring Cloud Security[3]. It offers a set of primitives for building secure applications and services with minimum fuss; as well as a declarative model which can be heavily configured externally (or centrally), lending itself to the implementation of large systems of co-operating, remote components, usually with a central identity management service.

Right now, 12 µ-services have been defined and developed (as shown in Figure 1), 7 of them specifically to perform patient's monitoring (5 for basic- and 2 for advanced-monitoring) and 5 are for giving support and providing further functionalities.

[1] https://github.com/Netflix/eureka

[2] https://github.com/Netflix/zuul/wiki

[3] https://cloud.spring.io/spring-cloud-security/

2. The Advanced Monitoring Services

According to the requirements and the work done during the 1st period, the following services have been defined, developed, and are now integrated in the current version of the SMS: health-status monitoring and drug prescriptions (previously referred as pharmacological adherence).

2.1 Health Status

Through suitable medical devices, health status of the patient is monitored. In particular, the professional prescribes when and which frequency to make the measurement with. Moreover, at prescription time, s/he may put thresholds to monitor anomalies. In so doing, both the professional and the patient receive alerts through the Alerts µ-service, in case a given threshold has been violated.

2.1.1 Requirements

First, for each site and each case study the requirements in terms of measurements to be taken have been collected. Requirements are summarized in Table 1.

Table 1 - Required measurement in each site and in each case. BCN = Barcelona (IDIBAPS), LL = Lleida (IRBLL), IL = Israel (ASSUTA), and NL = The Netherland (UMCG).

Measurement	CS1				CS2				CS3
	BCN	LL	IL	NL	BCN	LL	IL	NL	BCN
Temperature	X	--	X	--	X	X	X	X	X
Blood pressure	X	X	X	--	X	X	X	X	X
Heart rate	X	X	X	--	X	X	X	X	X
Weight	X	X	X	--	X	--	X	--	X
Blood oxygen saturation	X	--	X	--	X	--	X	--	X

2.1.2 Implementation

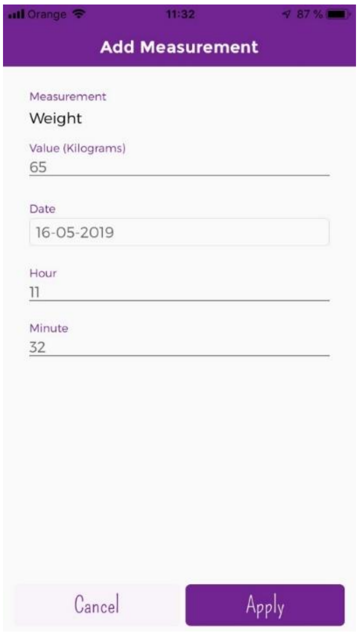
Starting from the requirements listed above, the medical devices listed in Figure 2 and Figure 3 have been considered and tested. Devices from Nokia/Withings have been then selected because of the availability of the API and the established contact with their technical department. Unfortunately, due to mark CE restrictions, Nokia/Withings is not able to sell devices in Israel. Thus, in ASSUTA, for both CS1 and CS2, manual devices (i.e., do not integrated with the SMS) have been used. Accordingly, the SMS app gives the possibility to manually input any measurement. The manual functionality has been adopted also in Lleida for the temperature, the blood oxygen saturation, and, in some cases, for the weight. In fact, the patients in Lleida regarded the Nokia/Withings thermometers too complicated to use and they asked permission to use the classical ones. On the other hand, regarding the blood oxygen saturation, due to the difficulty on finding a devices with an open-API to be integrated in the SMS, and after agreement with the professionals in Lleida, we decided to use a classical device and put the measurement manually (see an example in Figure 4).

Features	Devices			
	Withings Thermo	Withings Blood Pressure monitor	Withings Body+	Withings Body Cardio
Temperature	X	--	--	--
Blood pressure	--	X	--	--
Hearth rate	--	X	--	X
Weight	--	--	X	X
Blood oxygen saturation	--	--	--	--
Blood glucose level	--	--	--	--
ECG	--	--	--	--

Figure 2 - Tested devices and corresponding features of the brand Nokia/Withings.

Features	Devices			
	iHealth Air	iHealth Clear	iHealth Lite	iHealth Smart
Temperature	--	--	--	--
Blood pressure	--	X	--	--
Hearth rate	--	--	--	--
Weight	--	--	X	--
Blood oxygen saturation	X	--	--	--
Blood glucose level	--	--	--	X
ECG	--	--	--	--

Figure 3 - Tested devices and corresponding features of the brand iHealth.



Add Measurement

Measurement
Weight

Value (Kilograms)
65

Date
16-05-2019

Hour
11

Minute
32

Figure 4 - Example of measurement added manually

The list of devices adopted in each site is reported in Table 2.

Table 2 - Devices adopted in each site for health status monitoring

Site	Devices	Type
Barcelona (IDIBAPS)	--	--
Lleida (IRBLL)	Blood pressure monitor	Digital (Nokia/Withings)
	Thermometer	Classic
	Scale	Digital (Nokia/Withings)
	Oxygen saturation	Classic
Israel (ASSUTA)	Blood pressure	Classic
	Thermometer	Classic
	Scale	Classic
Groningen (UMCG)	Blood pressure monitor	Digital (Nokia/Withings)
	Thermometer	Digital (Nokia/Withings)
	Scale	Digital (Nokia/Withings)

2.2 Drug Prescription

Chronic patients are used to take several drugs during the day. Forgetting to take any may provoke serious problems. Thus, the professional prescribes all the drugs to be taken and the patient will receive a reminder any time s/he has to take one. In case of low adherence, a message is sent to both patient and professional through the Alerts µ-service.

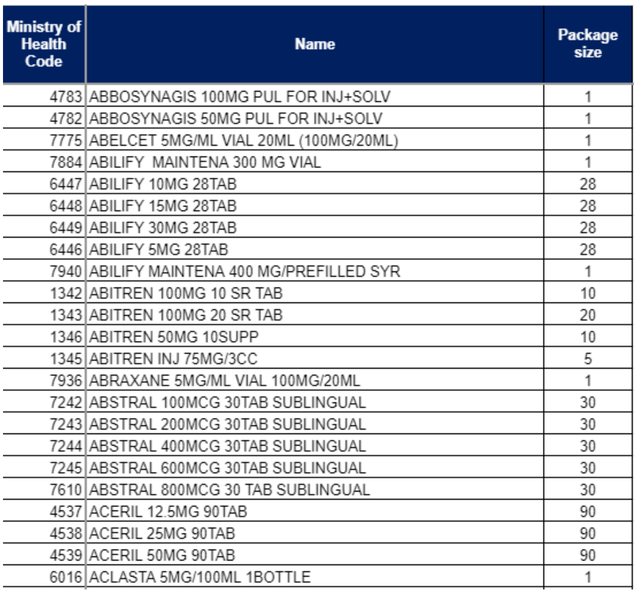
2.2.1 Requirements

Each site identified its own list of drugs to be prescribed.

Barcelona / Lleida. The Spain Agency of Medications and Health Products, AEMPS (Agencia Española de Medicamentos y Productos Sanitarios), provides a public list of drugs and medications (http://listadomedicamentos.aemps.gob.es/prescripcion.zip). The list is given in form of xml files in which every drug/medication is described as in the example below (active principle):

```
<principiosactivos>
        <nroprincipioactivo>350</nroprincipioactivo>
        <codigoprincipioactivo>12A</codigoprincipioactivo>
        <principioactivo>PARACETAMOL</principioactivo>
</principiosactivos>
```

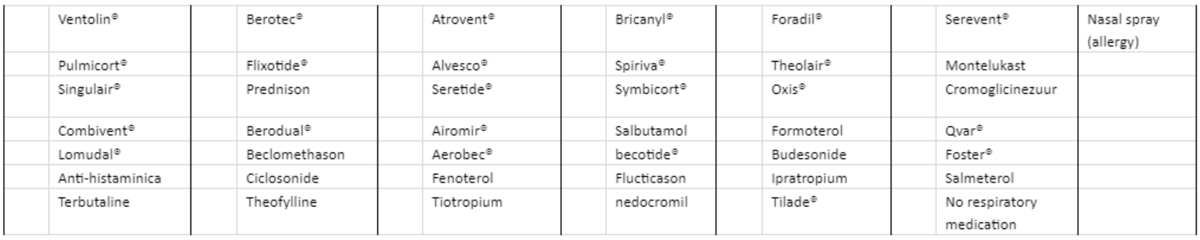
Israel. ASSUTA selected a set of drugs/medications from the list given by the Ministry of Health in Israel (http://www.health.gov.il/Subjects/Finance/DrugPrice/Pages/default.aspx, in Hebrew). Figure 5 shows an extract of the xls file provided by ASSUTA with the list of selected drugs and medications.



Ministry of Health Code	Name	Package size
4783	ABBOSYNAGIS 100MG PUL FOR INJ+SOLV	1
4782	ABBOSYNAGIS 50MG PUL FOR INJ+SOLV	1
7775	ABELCET 5MG/ML VIAL 20ML (100MG/20ML)	1
7884	ABILIFY MAINTENA 300 MG VIAL	1
6447	ABILIFY 10MG 28TAB	28
6448	ABILIFY 15MG 28TAB	28
6449	ABILIFY 30MG 28TAB	28
6446	ABILIFY 5MG 28TAB	28
7940	ABILIFY MAINTENA 400 MG/PREFILLED SYR	1
1342	ABITREN 100MG 10 SR TAB	10
1343	ABITREN 100MG 20 SR TAB	20
1346	ABITREN 50MG 10SUPP	10
1345	ABITREN INJ 75MG/3CC	5
7936	ABRAXANE 5MG/ML VIAL 100MG/20ML	1
7242	ABSTRAL 100MCG 30TAB SUBLINGUAL	30
7243	ABSTRAL 200MCG 30TAB SUBLINGUAL	30
7244	ABSTRAL 400MCG 30TAB SUBLINGUAL	30
7245	ABSTRAL 600MCG 30TAB SUBLINGUAL	30
7610	ABSTRAL 800MCG 30 TAB SUBLINGUAL	30
4537	ACERIL 12.5MG 90TAB	90
4538	ACERIL 25MG 90TAB	90
4539	ACERIL 50MG 90TAB	90
6016	ACLASTA 5MG/100ML 1BOTTLE	1

Figure 5 - Extract of the list of drugs/medications given by ASSUTA

Groningen. UMCG selected a list of drugs and medications from Farmacotherapeutisch Kompas (https://www.farmacotherapeutischkompas.nl/). An extract of that list is given in Figure 6.

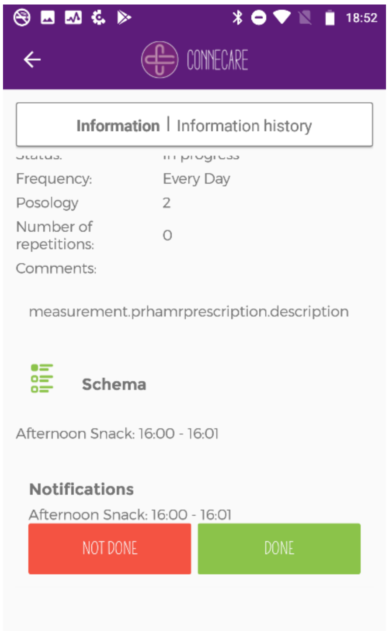


	Ventolin®		Berotec®		Atrovent®		Bricanyl®		Foradil®		Serevent®	Nasal spray (allergy)
	Pulmicort®		Flixotide®		Alvesco®		Spiriva®		Theolair®		Montelukast	
	Singulair®		Prednison		Seretide®		Symbicort®		Oxis®		Cromoglicinezuur	
	Combivent®		Berodual®		Airomir®		Salbutamol		Formoterol		Qvar®	
	Lomudal®		Beclomethason		Aerobec®		becotide®		Budesonide		Foster®	
	Anti-histaminica		Ciclosonide		Fenoterol		Flucticason		Ipratropium		Salmeterol	
	Terbutaline		Theofylline		Tiotropium		nedocromil		Tilade®		No respiratory medication	

Figure 6 - Extract of the list of drugs/medications given in Groningen

2.2.2 Implementation

The µ-service for Drugs prescription has been implemented taking into account the lists of drugs and medication provided in each site. Anytime a professional prescribes a drug/medication, the patient receives the corresponding notification in the self-management system. S/he may accept/reject the prescription. Besides starting and ending time, the prescription contains the information regarding the frequency to be take the drug/medication (daily, weekly, or monthly), the posology (e.g., 2 pills), as well as the moment of the day in which it has to be taken: breakfast, lunch, afternoon snack, dinner, before go to sleep, or anytime. It is worth noting, that each site defined each range of hours according to the habits of the corresponding Country. Each time a patient takes a drug/medication s/he has to enter the app and confirm (see Figure 7).



Status: In progress
Frequency: Every Day
Posology 2
Number of repetitions: 0
Comments:

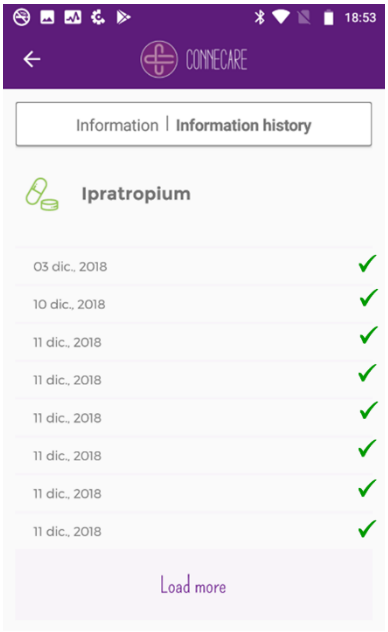
measurement.prhamrprescription.description

Afternoon Snack: 16:00 - 16:01

Afternoon Snack: 16:00 - 16:01

Figure 7 - Done/not done button to report if the drug/medication has been taken or not

Additionally, the service allows to gather all the past information regarding the adherence of taking each drug/medication (see Figure 8).



Ipratropium

03 dic., 2018	✓
10 dic., 2018	✓
11 dic., 2018	✓
11 dic., 2018	✓
11 dic., 2018	✓
11 dic., 2018	✓
11 dic., 2018	✓
11 dic., 2018	✓

Figure 8 - Example of historical information, 100% adherence

3. The Updated Basic Monitoring Services

The basic monitoring services have been presented in deliverable D4.2 "*Basic monitoring tools*". It is worth noting that now an update must be provided for two main reasons: 1) the termination partner IPHEALTH developed some of the services and, thus, they have been implemented from scratch again; and, 2) due to the evolutionary approach of the CONNECARE development and the real implementation studies, some requirements changed and the SMS has been adapted, accordingly.

As reported in D4.2, the core services developed by EURECAT were physical activity and sleeping. On the other hands, the questionnaires service was developed by IPHEALTH. Moreover, apart the core services, auxiliary services have been developed –or were under development– by IPHEALTH: messaging, agenda, and advices.

3.1 Physical Activities

3.1.1 Updated Requirements

As described in deliverable D4.2, we decided to use the following devices for the implementation studies:

- Fitbit charge HR – Barcelona, Groningen, Israel
- Withings/Nokia PulseOX – CS1 Lleida, in case monitoring oxygen saturation is needed
- Withings/Nokia GO – CS1 Lleida, in case there is no need to monitor oxygen saturation
- Withings/Nokia GO – CS2 Lleida

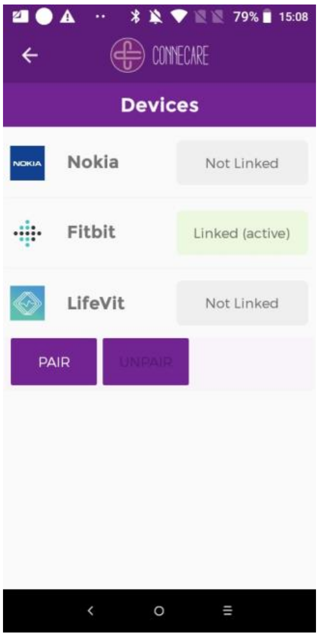
During the feasibility test (see D6.1 "Study release feasibility for the three clinical studies") and at the very beginning of the studies, IRBLL had usability problems with the PulseOX wristband and decided to stop using it. Similarly, problems in synchronizing the GO wristband with several smartphones and some reliability problems were also experienced. Thus, they decided to stop using them and started using Fitbit Flex 2.

In case of Israel, considering the provided features and the cost of the devices, they decided, from the very beginning, to use Fitbit Alta HR.

In Barcelona, during the feasibility test, they considered that for the sake of usability it is better not to have to install an app for the wristband together with the CONNECARE app. In fact, both Fitbit and Withings/Nokia wristbands require to have installed their app to be able to synchronize the devices with Bluetooth. Thus, they selected the LifeVit wristband with the new requirement to directly connect it to the CONNECARE app.

3.1.2 Implementation

Besides the integration of Withings/Nokia and Fitbit devices, the LifeVit has been integrated in the CONNECARE app (see Figure 9).

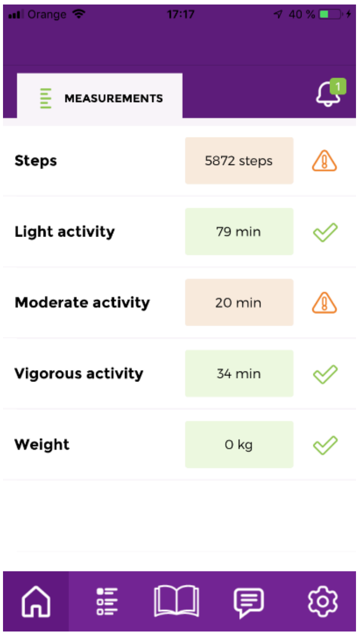


Nokia	Not Linked
Fitbit	Linked (active)
LifeVit	Not Linked

Figure 9 - Devices allowed in the CONNECARE SMS app

The corresponding Physical Activity µ-service has the same functionality for all the allowed devices. In so doing, all the patients may access to their data on number of steps and, according to the prescription they received[4], minutes of activities (high, moderate, and light), as well the minutes without activities. Figure 10 and Figure 11 show an example of data shown in the CONNECARE app.

[4] The actual prescription depends on the site. For example, in Lleida they did not monitor minutes without activities.

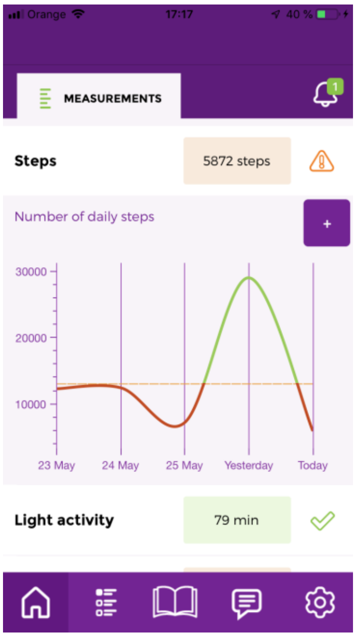


Orange 17:17 40 %

MEASUREMENTS

Steps	5872 steps	⚠
Light activity	79 min	✓
Moderate activity	20 min	⚠
Vigorous activity	34 min	✓
Weight	0 kg	✓

Figure 10 - Example of prescribed physical activities and the corresponding collected data



Orange 17:17 40 %

MEASUREMENTS

Steps	5872 steps	⚠

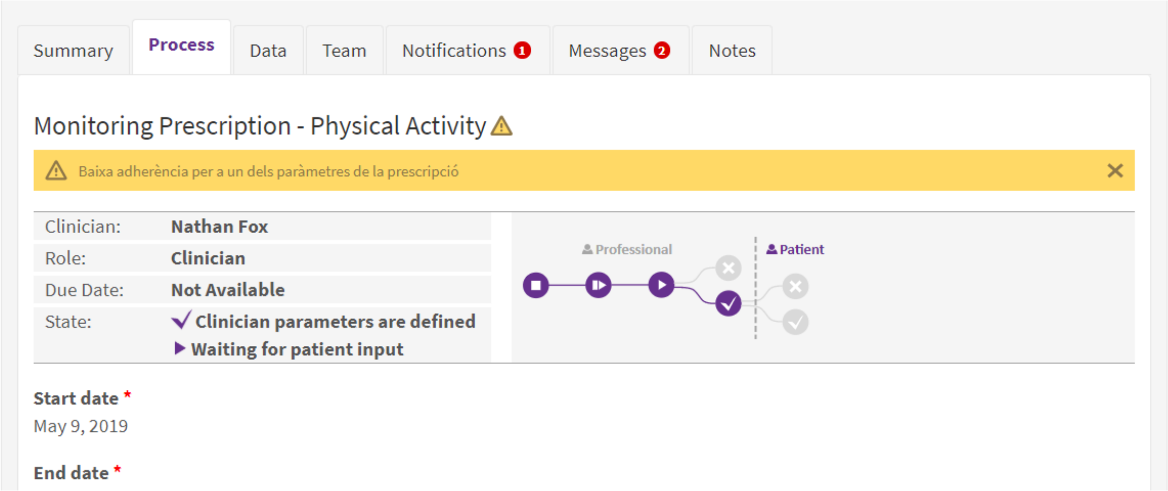
Number of daily steps

Light activity	79 min	✓

Figure 11 - Report of the performed steps in the last 5 days

3.1.3 Improvements

The Physical Activity µ-service has been improved in order to provide alerts to professionals reporting the status of the physical activity through the SACM (see Figure 12 and Figure 13) and providing push notifications and system notifications to the patients in order to automatically receive reminders to the action they have to perform.



Monitoring Prescription - Physical Activity ⚠

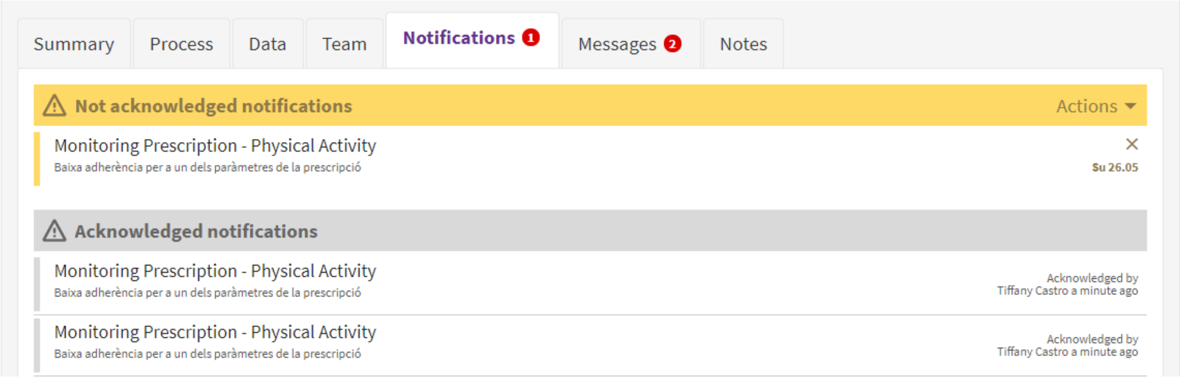
⚠ Baixa adherència per a un dels paràmetres de la prescripció ✕

Clinician:	Nathan Fox
Role:	Clinician
Due Date:	Not Available
State:	✓ Clinician parameters are defined ▶ Waiting for patient input

Start date *
May 9, 2019

End date *

Figure 12 - Example of received notification in the monitoring prescription page (SACM)



⚠ Not acknowledged notifications

Monitoring Prescription - Physical Activity
Baixa adherència per a un dels paràmetres de la prescripció ✕ Su 26.05

⚠ Acknowledged notifications

Monitoring Prescription - Physical Activity
Baixa adherència per a un dels paràmetres de la prescripció
Acknowledged by Tiffany Castro a minute ago

Monitoring Prescription - Physical Activity
Baixa adherència per a un dels paràmetres de la prescripció
Acknowledged by Tiffany Castro a minute ago

Figure 13 - Example of received and not acknowledged notification in the notification page (SACM)

3.2 Sleeping

This service has been totally re-implemented from scratch after the IPHEALTH termination. According to the new requirements, sleeping data are gathered from the wristband, in case the patient wears it during sleeping time. In particular, in ASSUTA and CS2 in Groningen there was agreement on asking patients to wear the wristband during night (i.e., 24/7) whereas in Lleida they preferred not to, and patients are free to wear or not the wristband.

Up until the end of May 2019, sleep data from 132 patients have been gathered. Table 3 describes these patients by gender, age group, site and case study. As shown, the number of patients by gender, site and case study is balanced. Regarding the age group, almost half of these patients are between 65 and 75 years old.

Table 3 - Number of patients with sleep data by gender, age group, site and case study.

Gender	N (%)	Age group	N (%)	Site	N (%)	Case study	N (%)
Female	60 (45.45)	55-65	27 (20.54)	Assuta	43 (32.57)	CS1	65 (49.24)
Male	71 (53.79)	65-75	59 (44.6)	IRBLL	44 (33.33)	CS2	67 (50.57)
Unknown	1 (0.76)	75-85	34 (25.76)	UMCG	45 (34.09)		
		85-95	12 (9.09)				

Figure 14 shows the distribution of the sleep data in time. Each row represents a patient and the colour represents the site. As shown, the data present a lot of discontinuities. One third of the patients has been wearing the device sporadically (bottom of the Figure), including patients from Assuta and UMCG even thought they were asked to wear the wristband during night.

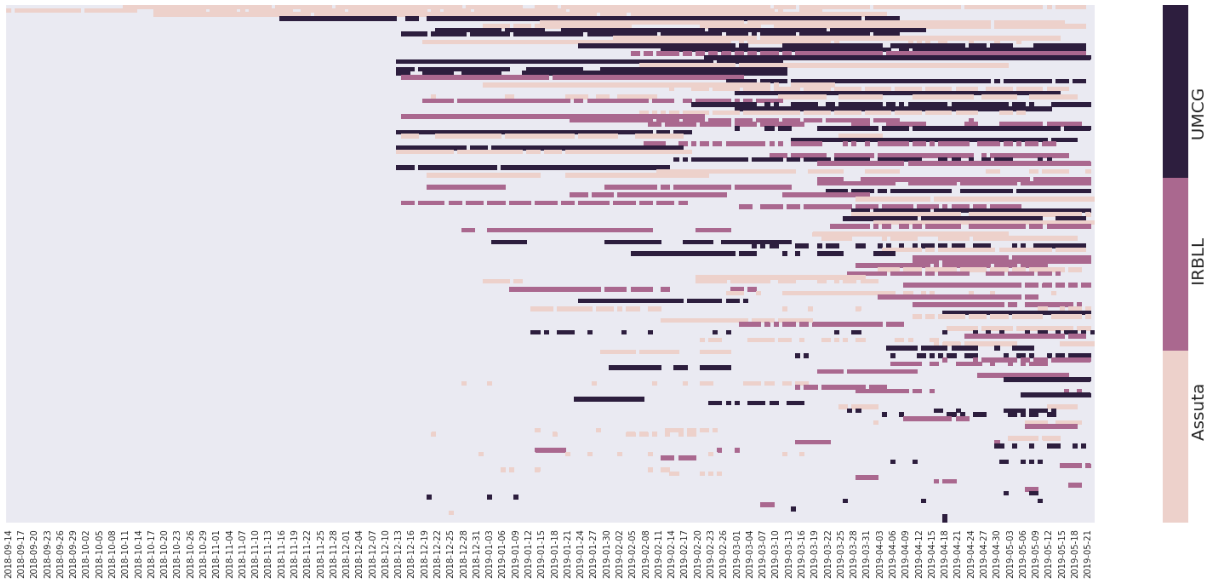


Figure 14 - Distribution of the sleep data in time per patient (row) , colours indicate the site

As a preliminary study the sleep data has been compared by gender, age group, site and case study. The analysed variables are: sleep time per night, arousals during night sleep (i.e., minutes that the user has been awake during the night sleep) and sleep time during the day (e.g. naps). Due to the discontinuities presented in the data the first results has been obtained by aggregating these variables for each patient. The results have been visualized using box plots in the Figures 15, 16, 17 and 18. Figure 15 shows that the values for male are a little bit higher than for woman, but the difference is not significant. Figure 16 shows that there are small differences between the sleep variables when groping the patients by age. Older patients tend to sleep more, either during day and during night, but they have less arousals during night. Moreover, between the older patients their sleep time during night is quite variable, but they are constant regarding the arousals. From the other three groups their behaviours are quite different and any clear pattern is shown, e.g. patient in group age 65-75 tend to sleep less during night than the other groups but more during day. Recall that the number of patients in each of this groups was not balance which might affect to the difference described. Figure 17 shows that the patients from the UMCG have more variability when comparing the time that they sleep at night, although the mean is similar to the other sites. Regarding the other variables, IRBLL patient has less arousals than the patients from the other sties but they sleep more during day. Figure 18 shows that there are no big differences when comparing between case study, however the patient from case study 1 has values smaller than those from the other case study.

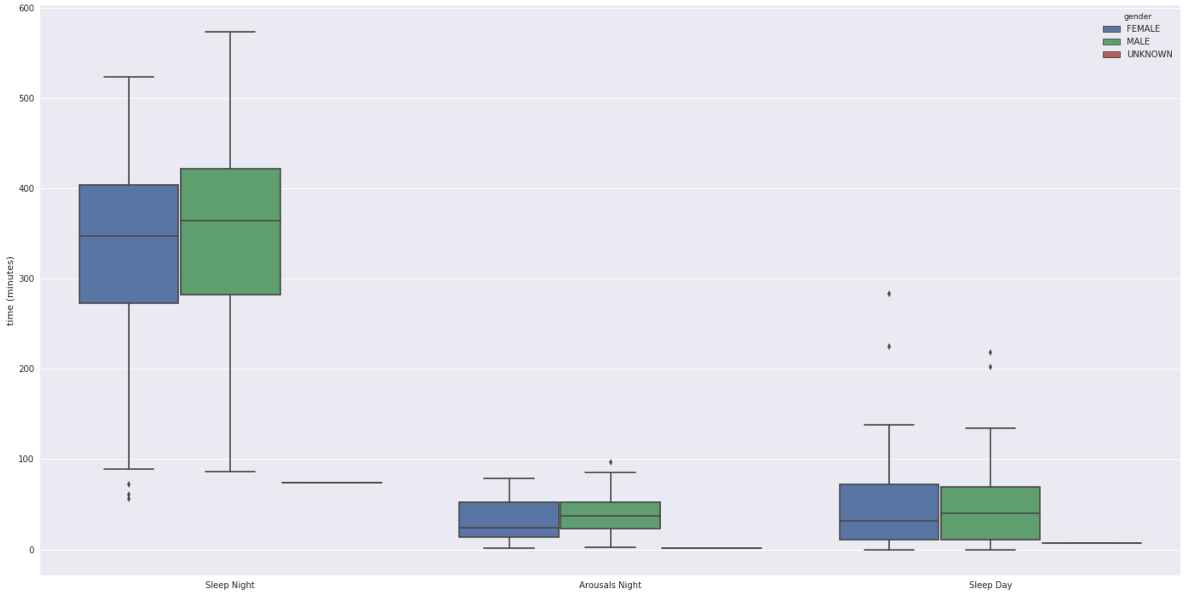


Figure 15 - Boxplot showing the difference of the sleep variables by gender.

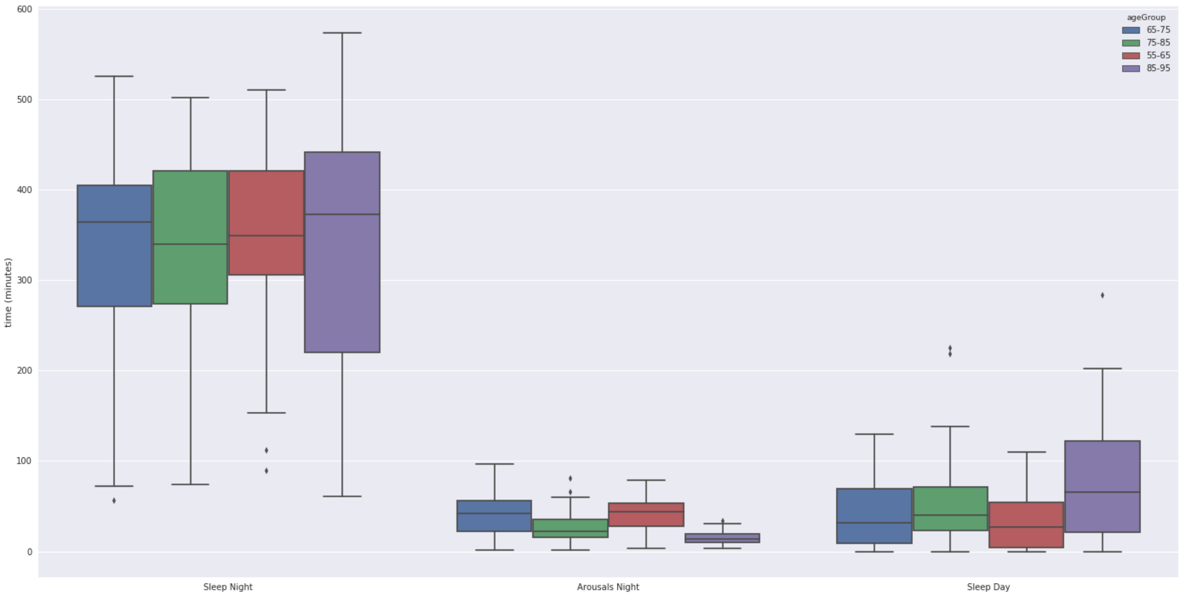


Figure 16 - Boxplot showing the difference of the sleep variables by age group.

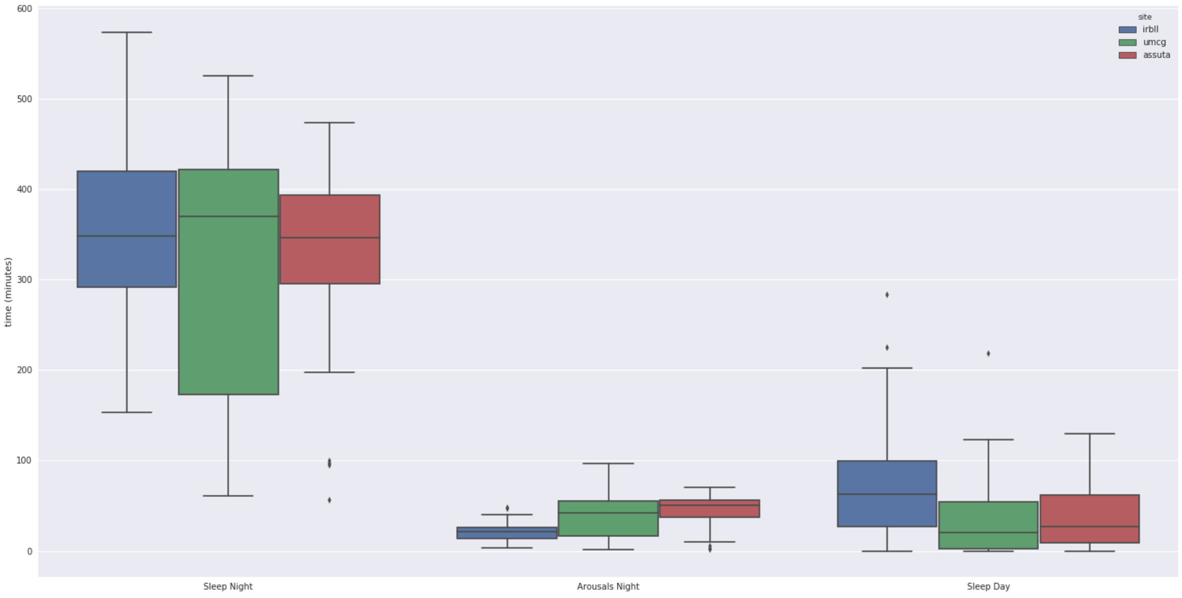


Figure 17 - Boxplot showing the difference of the sleep variables by site.

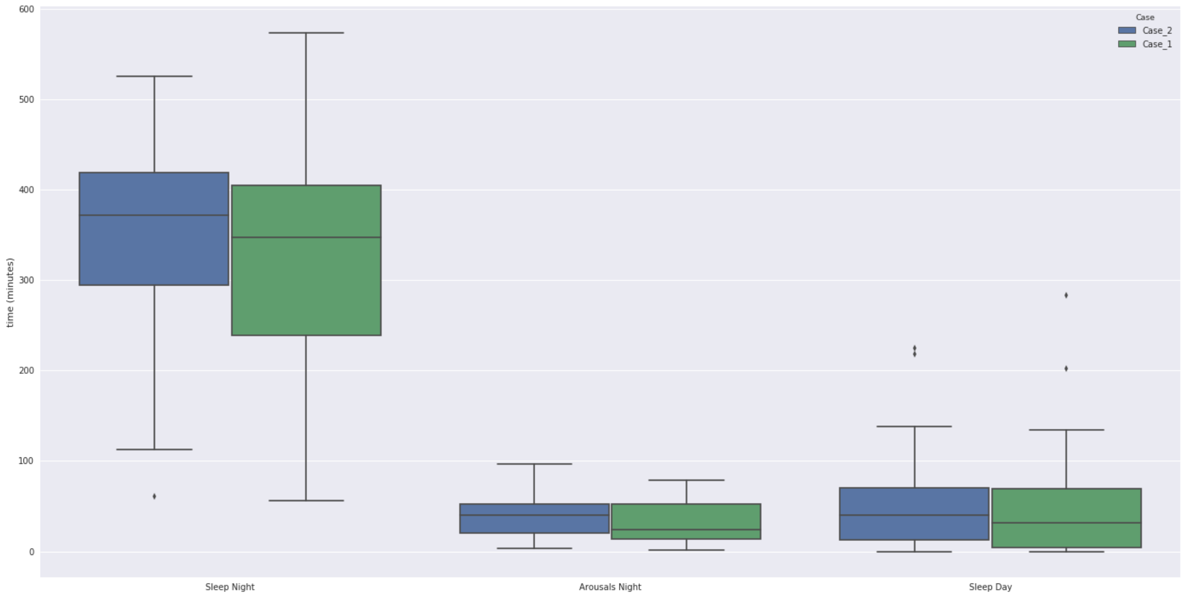


Figure 18 - Boxplot showing the difference of the sleep variables by case study.

Data collected by this service are also analysed by the quality of life assessment system, as will be described in D4.5 “QoL assessment system” that will be submitted on October 2019.

3.3 Questionnaires

This service has been totally re-implemented from scratch after the IPHEALTH termination.

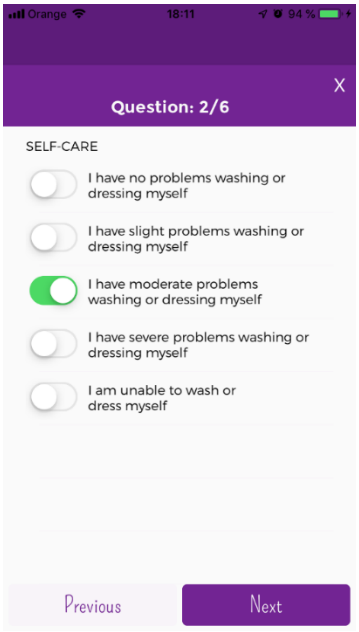
This µ-service allows assignment of one or more questionnaires to a patient; setting-up of the questionnaire(s) to be answered, together with the frequency that questionnaires will be requested to the patient's under the medical surveillance provisions; sending back questionnaire answers to the clinician; and checking the list of prescribed questionnaires and their answers. The patient answers each prescribed questionnaire through the self-management system receiving a suitable message with the corresponding result, and the professionals are automatically informed about the filled questionnaire.

First of all, the 4 sites agreed on 2 questionnaires to be adopted in each site and in each case study:

- EQ5D[5] (see Figure 19);
- SF-12[6].

[5] https://euroqol.org/wp-content/uploads/2016/09/EQ-5D-5L_UserGuide_2015.pdf

[6] https://www.hss.edu/physician-files/huang/SF12-RCH.pdf



Question: 2/6

SELF-CARE

○ I have no problems washing or dressing myself

○ I have slight problems washing or dressing myself

● I have moderate problems washing or dressing myself

○ I have severe problems washing or dressing myself

○ I am unable to wash or dress myself

Figure 19 - Example of items of the EQ-5D questionnaire

Once those questionnaires were implemented and their functionality tested, each site selected, for each case study, the questionnaires they wanted to prescribe to the patients:

- Visual Analogic Scale (VAS) for pain (Lleida, Groningen);
- Post-surgical (Lleida, Groningen), defined by the CONNECARE consortium;
- SUS (Barcelona, Groningen)[7];
- YALE (Barcelona)[8];
- HADS (Barcelona)[9];
- WOMAC (Lleida)[10];
- S-LANSS (Lleida)[11];
- COPD Status (Lleida), defined by the CONNECARE consortium;
- HF Status (Lleida), defined by the CONNECARE consortium;
- Leave (ASSUTA), defined by the CONNECARE consortium;
- Eat (ASSUTA), defined by the CONNECARE consortium;)

[7] https://www.measuringux.com/SUS.pdf

[8] https://yalehealth.yale.edu/sites/default/files/New_Patient_Questionnaire_July_31_2014.pdf

[9] http://www.scalesandmeasures.net/files/files/HADS.pdf

[10] http://www.performanceptpc.com/paperwork/womac.pdf

[11] https://bpac.org.nz/BPJ/2016/May/docs/s-lanss.pdf

- Feel (ASSUTA), defined by the CONNECARE consortium;
- Drink (ASSUTA), defined by the CONNECARE consortium;
- NPS (Groningen)[12];
- CCQ (Groningen)[13];
- ACQ (Groningen)[14];
- CARAT (Groningen)[15].

Once answered, the results of each questionnaire are sent to the SACM for consultation by the professionals.

3.4 Simple Tasks

Apart from rehabilitation in hospitals and specialized centres, clinicians agree that daily life activities performed at home radically improve patients' recovery. With this aim, clinicians may prescribe simple tasks to be performed during the day (e.g., dancing, cooking, reading). Moreover, depending on the disease and the kind of patient (e.g., in case of elderly people) the professional may ask to perform healthy activities such as drinking or eating a fruit. Through the self-management system, the patient can accept or reject the request and inform when the activity has been performed.

According to the gathered requirements, this service has been developed for the ASSUTA case studies. This service was not reported in the D4.2 deliverable because this requirement was gathered in a second phase of the project.

The following tasks may be prescribed to be performed by the patient at home: straightening and bending the knee while lying on the back; keeping the leg away from the body while lying on the back; lifting the pelvis lying on the back with bent legs; lifting the leg upright while lying on a side; bending the knee and bringing it to the stomach while lying on the back; bending the knee while sitting; bending the knees standing with the back to the wall; straightening and bending the seated knee; getting up from the chair and sitting back; walking outside at moderate speed; up and down stairs; strengthening the thigh; strengthening the elbow, buttocks, and fingers; training session for physiotherapy; drinking water; eating a fruit; playing cognitive games; visiting a nurse at the clinic.

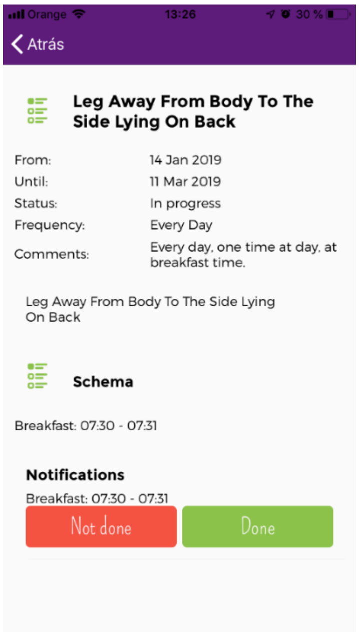
[12] https://www.netpromoter.com/know/

[13] https://thorax.bmj.com/content/69/9/793

[14] http://media.mycme.com/documents/171/w16-11_asthma_q_42563.pdf

[15] https://www.new.caratnetwork.org/

Each time a patient is required to perform a task, s/he receives a reminder and has to input if the task has been done or not (see Figure 20). The professional may check the performed activities through the SACM.



Leg Away From Body To The Side Lying On Back

From:	14 Jan 2019
Until:	11 Mar 2019
Status:	In progress
Frequency:	Every Day
Comments:	Every day, one time at day, at breakfast time.

Leg Away From Body To The Side Lying On Back

Schema

Breakfast: 07:30 - 07:31

Notifications

Breakfast: 07:30 - 07:31

Figure 20 - Example of simple task.

4. Auxiliary Services

In the Study Release presented and documented in D4.1 “First Self-Management System” and D4.2 “Basic monitoring tools”, IPHEALTH was in charge of implementing the required auxiliary services starting from their experience with the Vitalinq app. Due to their termination, EURECAT rebuilt from scratch all those services: Messaging, Advices, Alerts, and Notifications. Moreover, EURECAT updated and improved the Third Party service, also considering that the Vitalinq app and its backend are not to be integrated, anymore.

4.1 Messaging

This service offers a bidirectional communication tool that allows patients to interact with members of the medical staff in charge of the case, a given clinician, or even with other patients or communities of patients. Similarly to the well-known Whatsapp app, patients may send/receive text, images, videos, links, and documents.

4.1.1 Requirements

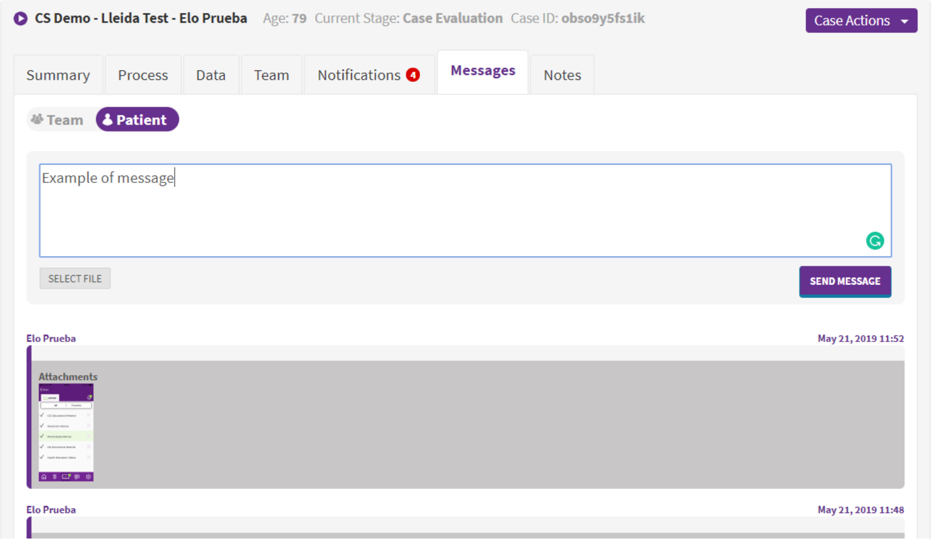
All the sites agreed on the need to have a direct communication channel to connect the professional team with the patient. Thus, a chat has been developed that allows for asynchronous communication between all the professionals in charge of the case and the associated patient.

Through the chat, all the actors may send/receive pictures and videos already stored in their gallery. Moreover, patients may also access to the camera of their smartphone, take pictures and videos and send them to the professional team.

4.1.2 Implementation

The chat has been implemented according to the following requirements:

- Any professional in charge of a case may write a message to the corresponding patient through the SACM (see Figure 21);
- The patient may include images or videos;
- The patient receives a notification anytime a message is received (see Figure 22);
- The icon in the SMS app is updated with the number of unread messages (see Figure 23);
- The patient accesses to the chat through the SMS to read/write a message;
- The patient may send a text message, an image, a video and may take a new picture or video (see Figure 24);
- In any moment the patient may access to the list of professional to which is in communication (see Figure 25).



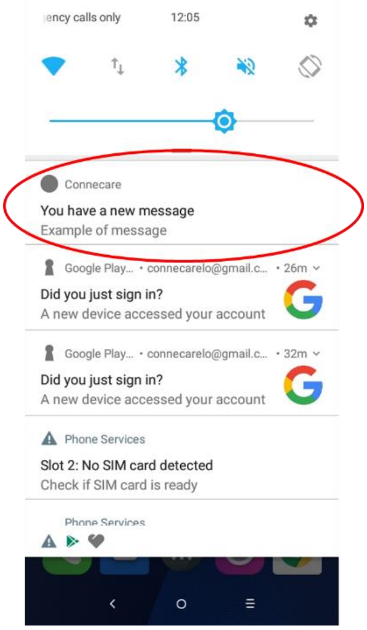
CS Demo - Lleida Test - Elo Prueba Age: 79 Current Stage: Case Evaluation Case ID: obso9y5fs1ik

Example of message

Elo Prueba May 21, 2019 11:52

Elo Prueba May 21, 2019 11:48

Figure 21 - Example of message through the SACM



Connecare
You have a new message
Example of message

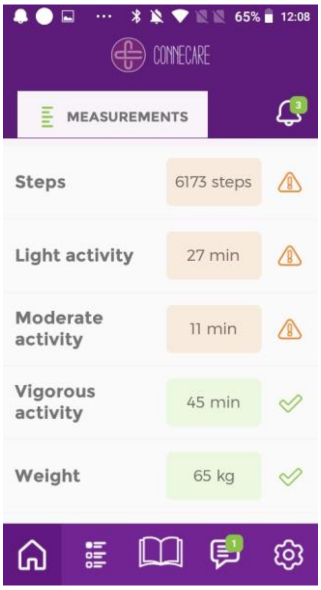
Google Play... • connecarelo@gmail.c... • 26m
Did you just sign in?
A new device accessed your account

Google Play... • connecarelo@gmail.c... • 32m
Did you just sign in?
A new device accessed your account

Phone Services
Slot 2: No SIM card detected
Check if SIM card is ready

Phone Services

Figure 22 - Example of notification of a message



Steps	6173 steps	⚠
Light activity	27 min	⚠
Moderate activity	11 min	⚠
Vigorous activity	45 min	✓
Weight	65 kg	✓

Figure 23 - The patient has an unread message

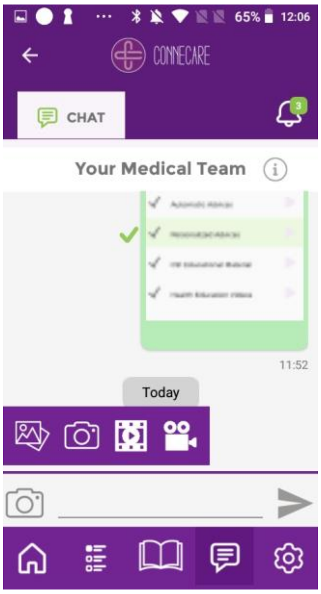
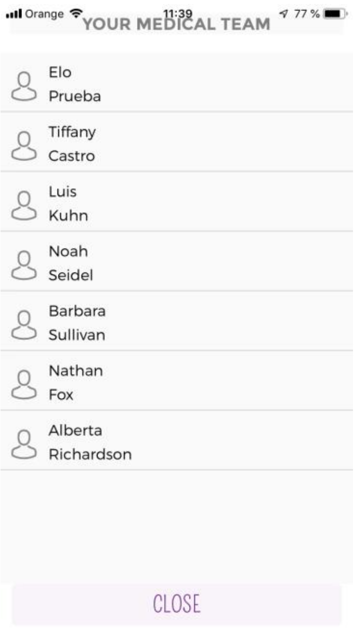


Figure 24 - List of actions the patient may do



Elo Prueba
Tiffany Castro
Luis Kuhn
Noah Seidel
Barbara Sullivan
Nathan Fox
Alberta Richardson

Figure 25 - List of users of the chat: the patient (the first one) and the team of professionals

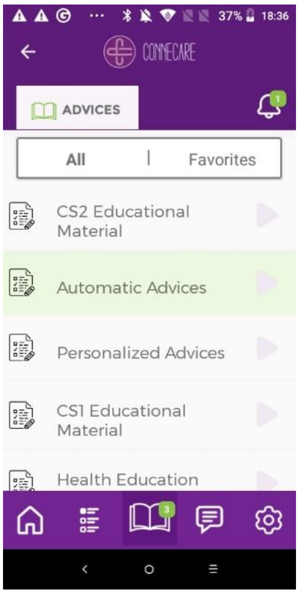
4.2 Advices

Education is really important to improve follow-up of a therapy and to empower the patient. Thus, the professional can select advices and training material in form of text, images, or videos and the patient is asked to see it. The patient may create her/his own list of bookmarks to have a direct access to relevant information any time is needed.

Three categories of advices have been defined:

- Automatic advices, advices generated by the system according to the collected data;
- Personalized advices, advices written by the professional for a specific patient;
- Educational material, advices specific of a given case study in a specific site and common to all the patients of the corresponding case and site;
- Health educational videos, videos specific for a given site and common to all the patients to that site.

All the advices are shown in the app divided according to those categories, as shown in Figure 26. Each advice may have an expiration date, because it could be an advice specific to a particular moment of the workplan or the therapy. Once an advice is received, the icon in the SMS app is updated with the number of unread advices (see Figure 27) and the professional receives a notification once the advice has been seen for the first time (see Figure 28). Finally, the patient may decide to save advices into a favourite list to access to them easily (see Figure 29).



CS2 Educational Material

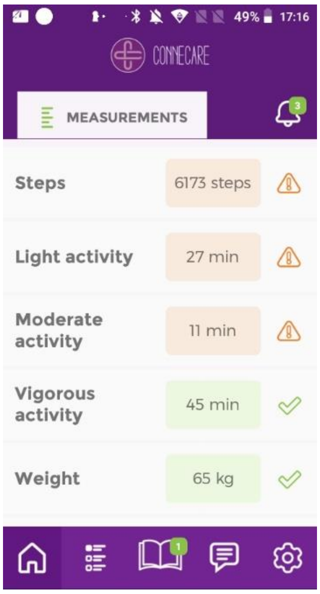
Automatic Advices

Personalized Advices

CS1 Educational Material

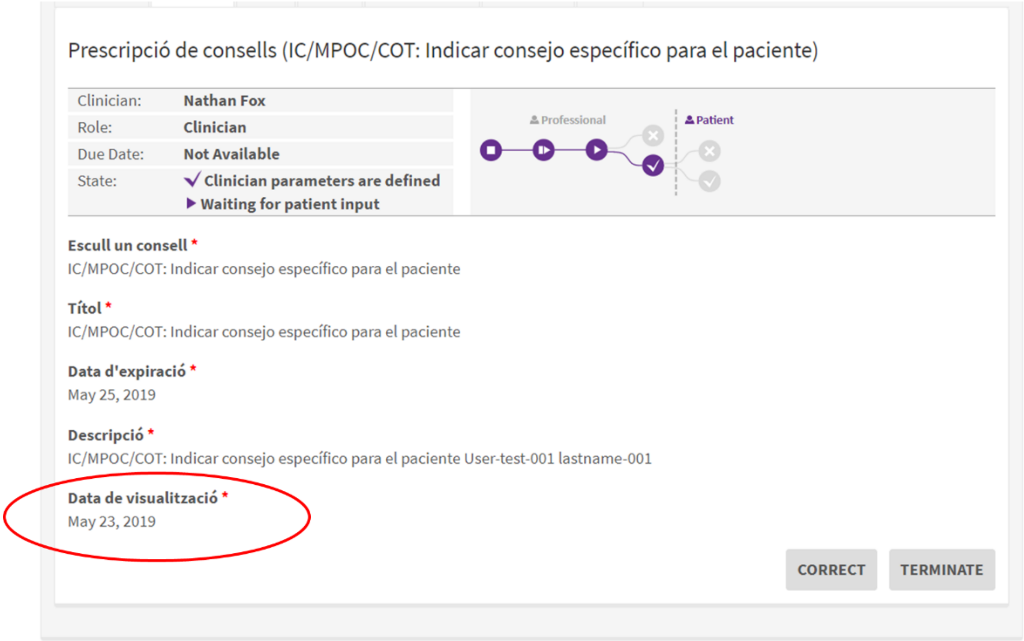
Health Education

Figure 26 - List of advices



Steps	6173 steps	⚠
Light activity	27 min	⚠
Moderate activity	11 min	⚠
Vigorous activity	45 min	✓
Weight	65 kg	✓

Figure 27 - There is an unread advice



Prescripció de consells (IC/MPOC/COT: Indicar consejo específico para el paciente)

Clinician:	Nathan Fox
Role:	Clinician
Due Date:	Not Available
State:	✓ Clinician parameters are defined ▶ Waiting for patient input

Escull un consell *
IC/MPOC/COT: Indicar consejo específico para el paciente

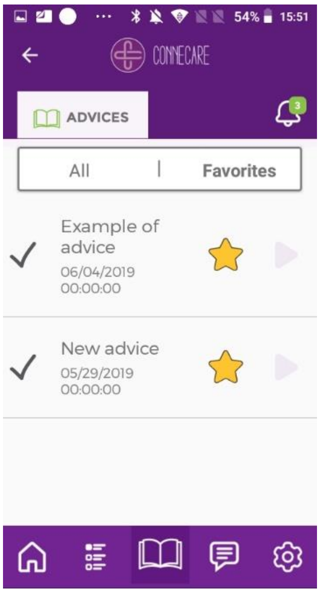
Títol *
IC/MPOC/COT: Indicar consejo específico para el paciente

Data d'expiració *
May 25, 2019

Descripció *
IC/MPOC/COT: Indicar consejo específico para el paciente User-test-001 lastname-001

Data de visualització *
May 23, 2019

Figure 28 - Confirmation of visualization of an advice



Example of advice
06/04/2019
00:00:00

New advice
05/29/2019
00:00:00

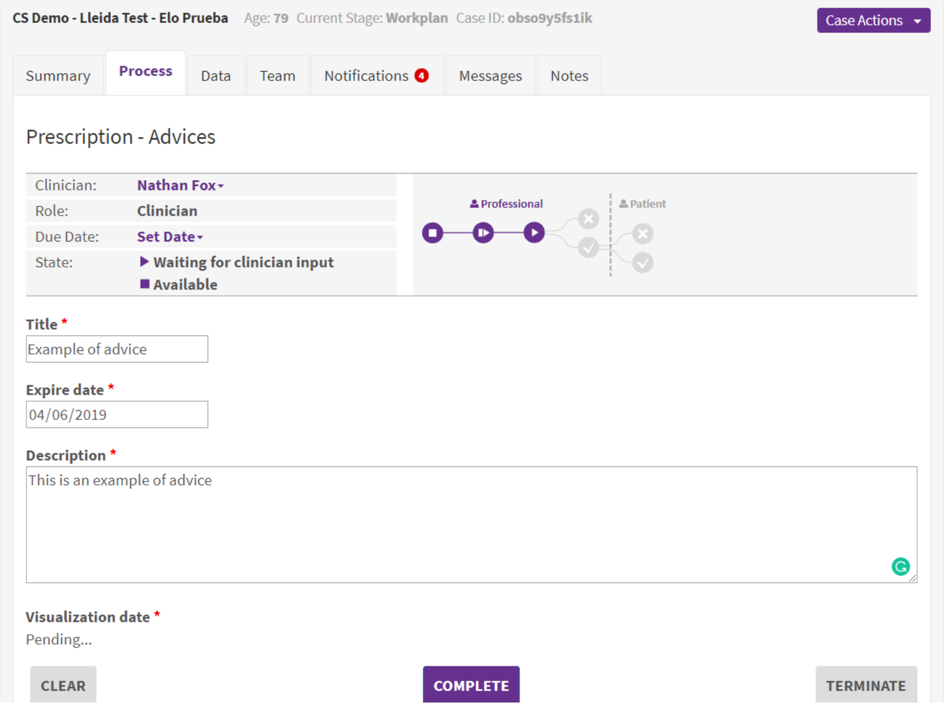
Figure 29 - List of advices saves as favourite

4.2.1 Automatic advices

All data gathered by the devices worn by the patient are automatically analysed to give an intelligent support. In fact, the Recommendation µ-service provides suitable recommendations to patients. In particular, this µ-service is designed and developed to provide support to patients in each of the required tasks. Both a distributed and central solution is adopted. On the one hand, each patient's monitoring µ-service will contain its own recommender system. On the other hand, thanks to the interaction and collaboration of all the patient's monitoring µ-service, advanced recommendations could be given aimed at giving support at a broader level. Currently, a specific recommender system for physical activity support has been defined and designed. The description of the recommender system is out of the scope of this work, it will be part of the deliverable D4.6 "*Recommender system for self-management*" that will be submitted on October 2019.

4.2.2 Personalized advices

Anytime a professional gives a specific advice to a patient, s/he can "prescribe" it through the SACM and send it to the SMS. An example is given in Figure 30 and Figure 31.



CS Demo - Lleida Test - Elo Prueba Age: 79 Current Stage: Workplan Case ID: obso9y5fs1ik

Prescription - Advices

Clinician:	Nathan Fox
Role:	Clinician
Due Date:	Set Date
State:	▶ Waiting for clinician input ■ Available

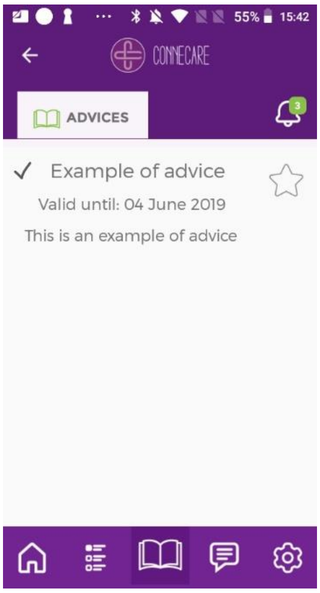
Title *
Example of advice

Expire date *
04/06/2019

Description *
This is an example of advice

Visualization date *
Pending...

Figure 30 - Example of prescription of an advice (SACM)



✓ Example of advice

Valid until: 04 June 2019

This is an example of advice

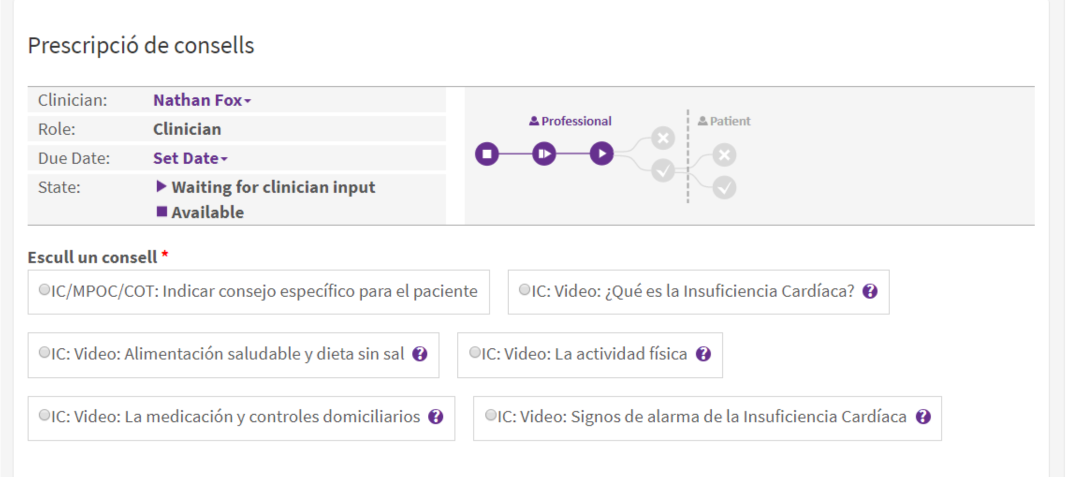
Figure 31 - Example of advice (SMS)

4.2.3 Educational material

A professional or the team of professionals may decide to send educational material in form of documents, images, videos, or external links to all the patients belonging to a given case.

4.2.4 Health educational videos

As a peculiar case of the “personalized advice” category, Lleida asked for having a further category, namely “health educational videos” for sending/prescribing patients specific videos with information related to their disease and/or comorbidities. Figure 32 shows the list of available videos to be prescribed (SACM) and Figure 33 shows one of them visualized in the SMS.



Prescripció de consells

Clinician:	Nathan Fox
Role:	Clinician
Due Date:	Set Date
State:	▶ Waiting for clinician input ■ Available

Escull un consell *

○ IC/MPOC/COT: Indicar consejo específico para el paciente

○ IC: Video: ¿Qué es la Insuficiencia Cardíaca?

○ IC: Video: Alimentación saludable y dieta sin sal

○ IC: Video: La actividad física

○ IC: Video: La medicación y controles domiciliarios

○ IC: Video: Signos de alarma de la Insuficiencia Cardíaca

Figure 32 - List of educational videos available for prescription in Lleida

Figure 33 - Screenshot of an educational video played in the SMS

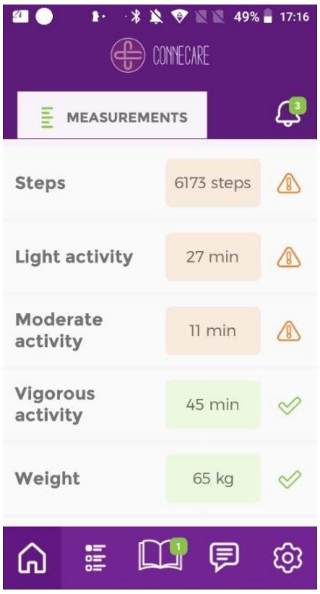
4.3 Alerts

Monitored data are continuously analysed at runtime. As soon as a patient's monitoring µ-service finds an anomaly, it interacts with the Alerts µ-service and a suitable alert message is sent to both the patient (to be aware of the issue) and the clinical staff (to be informed and act accordingly). In particular, anomalies are triggered any time data gathered from the patient (e.g., through the medical devices or a questionnaire) exceeds a given threshold defined at prescription time (e.g., a critical heart rate value).

4.4 Notifications

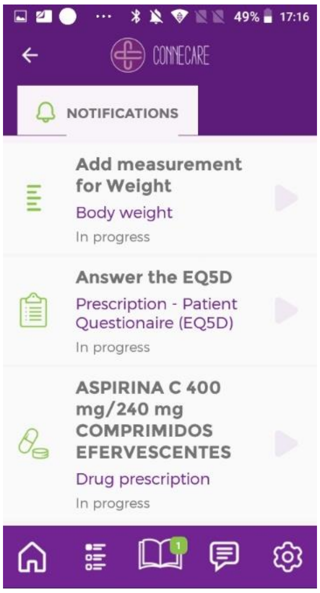
This µ-service receives the notifications generated by any of the other µ-services whose recipient is the patient (e.g., a new prescription has been generated). Through the Notifications µ-service, the corresponding message is sent to the app for its visualization. Moreover, complex notifications coming from interaction with the patient's monitoring µ-service may also be sent. In fact, a reminder is sent before any required task (e.g., 15 minutes before taking the blood pressure) and a confirmation given after any task performed with suitable information (e.g., the value of the blood pressure).

Notifications are displayed in the app in form of a bell, updated with the number of tasks pending to be performed (see Figure 34 and Figure 35). Moreover, system notifications are sent and shown in the smartphone when the CONNECARE app is in background (see Figure 36).



Steps	6173 steps	⚠
Light activity	27 min	⚠
Moderate activity	11 min	⚠
Vigorous activity	45 min	✓
Weight	65 kg	✓

Figure 34 - The bell shows that there are 3 notifications to be addressed

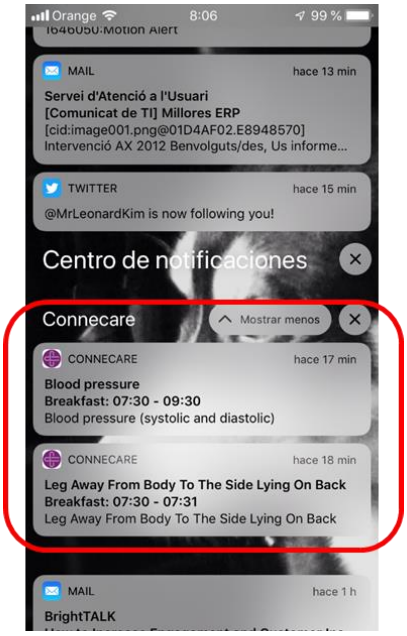


Add measurement for Weight
Body weight
In progress

Answer the EQ5D
Prescription - Patient Questionaire (EQ5D)
In progress

ASPIRINA C 400 mg/240 mg COMPRIMIDOS EFERVESCENTES
Drug prescription
In progress

Figure 35 - List of notifications to be addressed



1646050:Motion Alert

Servei d'Atenció a l'Usuari
[Comunicat de TI] Millores ERP
[cid:image001.png@01D4AF02.E8948570]
Intervenció AX 2012 Benvolguts/des, Us informe...

@MrLeonardKim is now following you!

Blood pressure
Breakfast: 07:30 - 09:30
Blood pressure (systolic and diastolic)

Leg Away From Body To The Side Lying On Back
Breakfast: 07:30 - 07:31
Leg Away From Body To The Side Lying On Back

BrightTALK

Figure 36 – Example of system notifications from CONNECARE

4.5 Third Party

In order to provide patients' monitoring, also integration with third party elements, such as wearable and medical devices, has been implemented. To handle these devices, the system has a dedicated µ-service to implement the connections and calls to the selected external µ-services. This µ-service has two main functionalities: managing the connections to external providers, and standardization of the data model. By using standard communication and authorization protocols, such as OAuth[16], data shared by the patient from these third parties can be gathered. The µ-service saves all the connection information and hide the details to the µ-services that need information from them. Regarding the standardization of the data model used inside the system, each third party uses its own data model to obtain the information from them but this µ-service transforms the model to a common one used by all the connected third parties. In this way, the µ-services only have to ask for the information to the third party connector and wait for a well-known message with the data of the patient without to worry about the communication process and the different data model used. In so doing, the third party µ-service works as an abstraction layer.

[16] https://oauth.net/

5. Conclusions

In CONNECARE, three levels of monitoring are proposed and suitable services defined and developed accordingly. In this deliverable, first of all the final overall architecture based on µ-services is presented. Then, the 2 services to perform advanced monitoring that have been developed have been presented: health status and drug prescription. Moreover, due to the iterative approach based on co-design that is adopted in CONNECARE and due to the IPHEALTH termination partner, the basic monitoring services have been redesigned and re-implemented or updated. All those services have been also presented in this deliverable: physical activity, sleeping, and questionnaires. Furthermore, the auxiliary services that were part of the Study Release and, thus, presented in the deliverable D4.2 “Basic monitoring tools” have been improved or re-designed and implemented in the case they were previously designed and implemented by IPHEALTH: messaging, advices, alerts, and notifications. Thus, this deliverable advances and improves deliverable 4.2 “Basic monitoring tools”. Finally, the third party service that has been implemented to allow the integration of external devices is described.

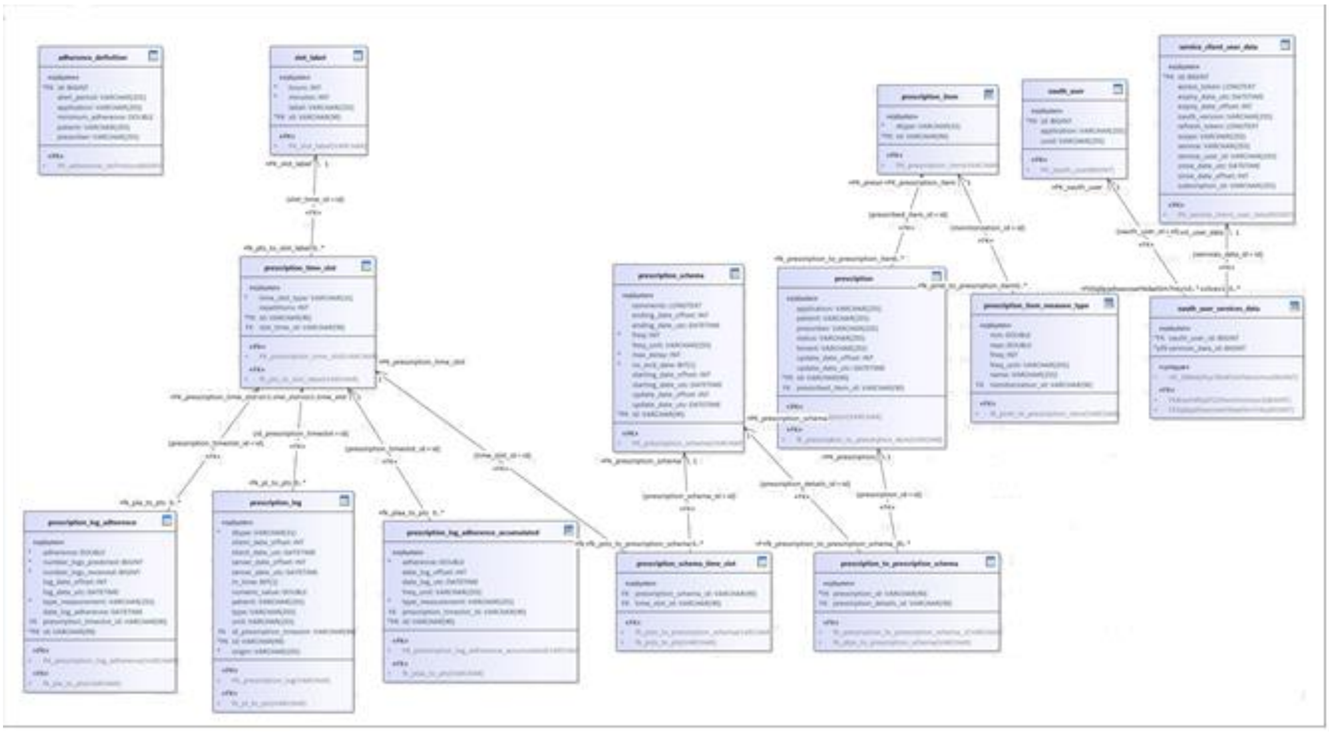
All those systems are part of the Final Release of the CONNECARE SMS and have been iteratively added to the versions provided in the Android and iOS markets to the patients involved in the implementation studies.

It is worth noting that, besides the services presented in this deliverable, to have the whole vision of the functionalities provided by the CONNECARE SMS, the reader should also refer to deliverables D4.5 “QoL assessment system” and D4.6 “Recommender system for self-management” that will be submitted on October 2019 (M43).

6. Annexes

6.1 Health Status

6.1.1 Data Model



The following table summarizes the database parameters that are mandatory in the health status service.

Parent	Attribute	Nullable	UI Visibility	Description
-	Prescriber	NO	NO	Uuid of the clinician who made the prescription
-	Patient	NO	NO	Uuid of the patient
-	Application	NO	NO	Uuid of the application
-	Tenant	NO	NO	Uuid of the tenant

-	prescribedItem:	NO	NO	Prescribed ítem contains a list of measurementType:
prescribedItem	measurementType:			
measurementType	Name		YES	Has to be one of the following (ignoring upper/lower case) MeasurementTypes: • Weight • Height • FatFreeMass • FatRatio • FatMassWeight • DiastolicBloodPressure • SystolicBloodPressure • HeartPulse • Temperature • SP02 • BodyTemperature • SkinTemperature • MuscleMass • Hydration • BoneMass • PulseWaveVelocity • Sleep If Sleep is chosen the prescriptionDetails must have a label of BeforeSleep (see below prescriptionDetails > timeSlot > slotTime > label)
measurementType	alertDefinition			
alertDefinition	Min		Yes	The minimum value of a measure to throw an alert
alertDefinition	Max		Yes	The maximum value of a measure to throw an alert.
alertDefinition	alertFrequency		Yes	The frequency number of FreqUnit when the system checks if the measure has been taken of is less or more than the minimum or

				maximum stablished to throw an alert. This frequency has to be consistent with the PrescriptionDetails frequency.
alertDefinition	alertFrequencyUnit		Yes	Has to be one of the following (ignoring upper/lower case) FreqUnit: • Days • Weeks • Months This frequency unit has to be consistent with the prescriptionDetails frequency unit.
-	prescriptionDetails	NO	Yes	PrescriptionDetails contains a list of PrescriptionSchema
prescriptionDetails	startDate	NO	Yes	The start date has to be in the present/future: the lowest value that startDate may have is today at 00:00:00 UTC startDate has to be before the endDate of the prescription
prescriptionDetails	endDate	NO	Yes	The endDate has to be after the startDate.
prescriptionDetails	freq	NO	Yes	The frequency number of FreqUnit when the measures must be taken. This frequency has to be consistent with the prescribedItem alert frequency

prescriptionDetails	freqUnit		Yes	Has to be one of the following (ignoring upper/lower case) FreqUnit: • Days • Weeks • Months This frequencyUnit has to be consistent with the prescribedItem alert frequencyUnit.
prescriptionDetails	maxDelay	YES	Yes	The max delay time for taking a measure in minutes
prescriptionDetails	noEndDate	YES	Yes	
prescriptionDetails	timeSlot			
timeslot	slotTime	NO	Yes	
slotTime	Label	NO	Yes	Has to be one of the following (ignoring upper/lower case) TimeSlotMeal: • Breakfast • Lunch • AfternoonSnack • Dinner • BeforeSleep • SpecificHour If Sleep measure type is specified in prescription Item, it has to have a Label Before Sleep on all schemas of the prescription.
slotTime	Hour	YES	Yes	It can be omitted only if the label is different from specific hour.

				If it is present, it has to be an Integer representing the hour from 0 to 23.
slotTime	Minute	YES	Yes	It can be omitted only if the label is different from specific hour. If it is present it has to be an Integer representing the minutes from 0 to 59

6.1.2 API Definition

POST /v1/prescription/user/{user}/save

Prescribes the monitorization of the given user.

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Request

Atribute	Optional	Description
PrescriptionDto	application (string, *optional*), id (string, *optional*), patient (string, *optional*), prescribedItem (MonitorizationDto, *optional*), prescriber (string, *optional*), prescriptionDetails(Array[PrescriptionSchemaDto], *optional*), status (string, *optional*),	

	tenant (string, *optional*), treatmentDocumentation(DocumentDto, *optional*), updateDate (LocaleDate, *optional*)
MonitorizationDto	measurementType(Array[MeasurementTypeDto], optional)
PrescriptionSchemaDto	comments (string, *optional*), endDate (LocaleDate, *optional*), freq (integer, *optional*), freqUnit (string, *optional*), id (string, *optional*), maxDelay (integer, *optional*), noEndDate (boolean, *optional*), startDate (LocaleDate, *optional*), timeSlot (Array[PrescriptionTimeSlotDto], *optional*), timeStamp (LocaleDate, *optional*)
DocumentDto	name (string, optional)
MeasurementTypeDto	alertDefinition (MeasurementAlertDto, *optional*), name (string, *optional*)
PrescriptionTimeSlotDto	id (string, *optional*), repetitions (integer, *optional*), slotTime (SlotLabelDto, *optional*)
MeasurementAlertDto	alertFrequency (integer, *optional*), alertFrequencyUnit (string, *optional*), freq (integer, *optional*),

	freqUnit (string, *optional*), max (number, *optional*), min (number, *optional*)
SlotLabelDto	hour (string, *optional*), id (string, *optional*), label (string, *optional*), minute (string, *optional*)

Response (Status 200)

```
{
  "application": "string",
  "id": "string",
  "patient": "string",
  "prescribedItem": {
    "measurementType": [
      {
        "alertDefinition": {
          "alertFrequency": 0,
          "alertFrequencyUnit": "string",
          "freq": 0,
          "freqUnit": "string",
          "max": 0,
          "min": 0
        },
        "name": "string"
      }
    ]
  },
  "prescriber": "string",
  "prescriptionDetails": [
    {
      "comments": "string",
      "endDate": "2018-02-17T08:00:00.000+0000",
      "freq": 0,
      "freqUnit": "string",
      "id": "string",
      "maxDelay": 0,
      "noEndDate": true,
      "startDate": "2018-01-17T08:00:00.000+0000",
      "timeSlot": [
        {
          "id": "string",
          "repetitions": 0,
          "slotTime": {
            "hour": "string",
            "id": "string",
            "label": "string",
            "minute": "string"
          }
        }
      ],
    }
  ],
  "status": "string",
  "tenant": "string",
  "treatmentDocumentation": {
    "name": "string"
  },
  "updateDate": "2018-02-17T08:00:00.000+0000"
}
```

GET /v1/prescription/user/{user}/retrieve

Retrieve the monitorization of the given user

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response

```
[
  {
    "application": "string",
    "id": "string",
    "patient": "string",
    "prescribedItem": {
      "measurementType": [
        {
          "alertDefinition": {
            "alertFrequency": 0,
            "alertFrequencyUnit": "string",
            "freq": 0,
            "freqUnit": "string",
            "max": 0,
            "min": 0
          },
          "name": "string"
        }
      ]
    },
    "prescriber": "string",
    "prescriptionDetails": [
      {
```

```
    "comments": "string",
    "endDate": ""
    "freq": 0,
    "freqUnit": "string",
    "id": "string",
    "maxDelay": 0,
    "noEndDate": true,
    "startDate": ""
    "timeSlot": [
     {
      "id": "string",
      "repetitions": 0,
      "slotTime": {
       "hour": "string",
       "id": "string",
       "label": "string",
       "minute": "string"
      }
     }
    ],

   }
  ],
  "status": "string",
  "tenant": "string",
  "treatmentDocumentation": {
   "name": "string"
  },
  "updateDate": ""
 }
]
```

GET /v1/prescription/{prescription}/retrieve

Retrieve a prescription by its id

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response

```
{
 "application": "string",
 "id": "string",
 "patient": "string",
 "prescribedItem": {
  "measurementType": [
   {
        "alertDefinition": {
          "alertFrequency": 0,
          "alertFrequencyUnit": "string",
          "freq": 0,
          "freqUnit": "string",
          "max": 0,
          "min": 0
        },
        "name": "string"
      }
    ]
  },
  "prescriber": "string",
  "prescriptionDetails": [
    {
      "comments": "string",
      "endDate": "2018-01-17T08:00:00.000+0000"
      "freq": 0,
      "freqUnit": "string",
      "id": "string",
      "maxDelay": 0,
      "noEndDate": true,
      "startDate": "2018-01-17T08:00:00.000+0000"
      "timeSlot": [
        {
          "id": "string",
          "repetitions": 0,
          "slotTime": {
            "hour": "string",
            "id": "string",
            "label": "string",
            "minute": "string"
          }
        }
      ]
    }
  ],
  "status": "string",
  "tenant": "string",
  "treatmentDocumentation": {
    "name": "string"
  },
  "updateDate": ""
}
```

6.1.2.1 Error

This class contains the information for a request which doesn’t generate a specific content. For instance, correct PUT requests generate this kind of answers or any other request if they generate an error.

Attribute	Optional	Description
errorCode Int	**N**	Internal code of the error / success.
userMessage String	**N**	User friendly message.
internalMessage integer	**N**	Internal error message.

JSON representation:

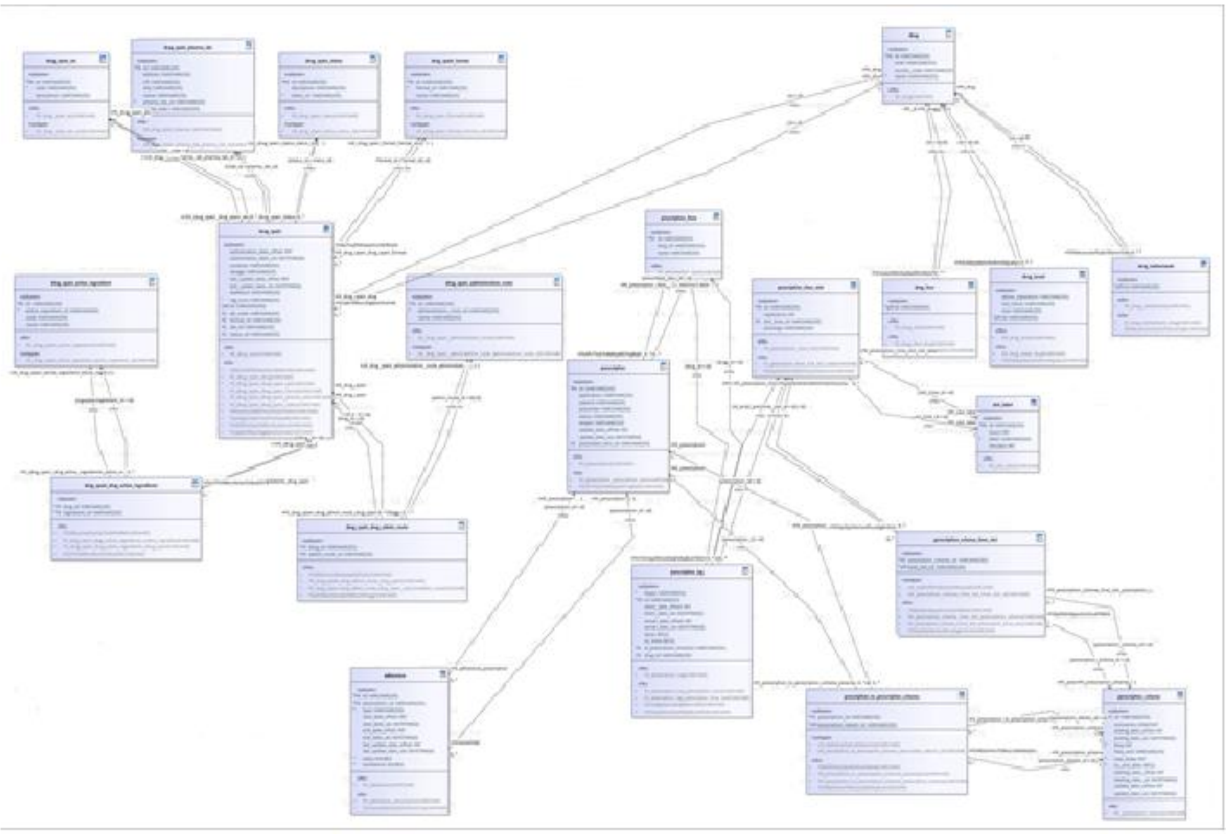
```
{
  "errorCode": int,
  "userMessage": "String",
  "internalMessage": "String"
}
```

Example:

```
{
  "errorCode": 4003,
  "userMessage": "No prescription found",
  "internalMessage": "No prescription found"
}
```

6.2 Drug Prescriptions

6.2.1 Data Model



6.2.2 API Definition

POST /v1/api/user/prescription

Retrieve Drug Prescriptions

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response code: 200 OK

Response body:

```
{
    "message": "Ok",
    "status": 0,
    "prescriptions": [
        {
            "schemas": [
                {
                    "takingMoments": [
                        {
                            "id": "2c92808566c44b960166c45b61ee000a",
                            "defaultTime": "09:30",
                            "takingMomentModifierId": null,
                            "takingMomentModifierLabel": null,
                            "takingMomentTypeId": null,
                            "takingMomentTypeLabel": "Breakfast",
                            "repetitions": null
                        }
                    ],
                    "id": "2c92808566c44b960166c45b61ee0009",
                    "endDateTime": "2018-10-30T09:29:00.001+0100",
                    "startDateTime": "2018-10-30T09:29:00.001+0100",
                    "maxDelayAllowedToAnswer": 120,
                    "repetitionFreq": {
                        "freqLabelId": "Days",
                        "repetitionValue": 0
                    },
                    "posology":"1 pill"
                    "schemaComments": "Drug Prescription TEST"
                }
            ],
            "name": "measurement.pharma.name",
            "id": "2c92808566c44b960166c45b61ee0007",
            "type": 6,
            "description": "measurement.prhamrprescription.description",
            "specificName": null,
            "linkedItem": "",
            "targetNorm": null,
            "targetValue": null,
            "targetIsMax": false,
            "thresholdMax": null,
            "thresholdMin": null,
            "repeating": 0,
            "endDateTime": "2018-10-31T22:59:00.001+0100",
            "startDateTime": "2018-10-30T09:29:00.001+0100",
            "createDateTime": "2018-10-30T08:43:35.149+0000",
            "measureObjectId": "23bd6661-7921-11e8-a179-0242ac120002",
            "professionalProfileId": "2c9480845bee03e7015bfcad28990010",
            "status": 0
        }
    ]
}
```

POST /v1/api/user/prescription/saveprescriptionstatus

Endpoint to Accept or decline the prescription:

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Request:

```
{
            "Id": "2c9380896253a336016268aeee060031",
            "Status": 1
}
```

Response code: 200 OK

Response body:

```
{
    "Status": 0,
    "Message": "OK"
}
```

POST /v1/api/user/prescription/saveprescriptionstatus

Endpoint to save tasks (Done/Not Done):

Headers

Parameter	**Position**	**Description**
Authorization String	Header	The Bearer access token for SMS

Request:

```
{
  "TaskId": "2c93809462b9da060162c42e78ee0070",
  "TakingMomentsLog": [
    {
      "MeasurementDate": "2018-04-27T10:00:00.0000000+02:00",
      "TakingMomentValue": true,
      "TakingMomentOnTime": false,
      "TakingMomentDelayMin": 30,
      "TakingMomentId": "2c93809462b9da060162c42e78ef0073"
    }

  ]
}
```

Response code: 200 OK

Response body:

```
{
    "Status": 0,
    "Message": "OK"
}
```

GET /pharmaprescription/v1/logs/{{user-uuid}}/retrieve?prescription-uuid={{prescription-uuid}}

Endpoint to list taking moments results

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response code: 200 OK

Response body:

```
{
    "content": [
        {
            "done": true,
            "drug": {
                "name": "GLUCOSALINO HIPERTONICO TECSOLPAR SOLUCION PARA PERFUSION",
                "uuid": "2c928095667759240166775f3ca416fe"
            },
            "prescriptionTimeSlotId": "2c92808566c44b960166c45b61ee000a",
            "onTime": false,
            "clientTimeStamp": "2018-10-31T11:49:00.000+0200"
        },
        {
            "done": true,
            "drug": {
                "name": "GLUCOSALINO HIPERTONICO TECSOLPAR SOLUCION PARA PERFUSION",
                "uuid": "2c928095667759240166775f3ca416fe"
            },
            "prescriptionTimeSlotId": "2c92808566c44b960166c45b61ee000a",
            "onTime": false,
            "clientTimeStamp": "2018-10-31T11:49:00.000+0200"
        }
    ],
    "last": true,
    "totalElements": 4,
    "totalPages": 1,
    "size": 20,
    "number": 0,
    "sort": null,
    "numberOfElements": 4,
    "first": true
}
```

6.2.2.1 Error

This class contains the information for a request which doesn’t generate a specific content. For instance, correct PUT requests generate this kind of answers or any other request if they generate an error.

Attribute	Optional	Description
errorCode Int	**N**	Internal code of the error / success.
userMessage String	**N**	User friendly message.
message integer	**N**	Internal error message.

Error codes

- 404 Not found
- 500 Internal Server Error
- 503 Service Unavailable
- 400 Bad Request, 401 Unauthorized

JSON representation:

```
{
  “errors”:[
    {
      "errorCode": int,
      "userMessage": "String",
      " message ": "String"
    }
  ]
}
```

Example:

```
{
  "errors":[
    {
      "errorCode":int,
      "userMessage":"string",
      "message":"string"
    }
  ]
}
```

Generic error body:

```
{
  "errors":[
    {
      "errorCode":int,
      "userMessage":"string",
      "message":"string"
    }
  ]
}
```

6.3 Physical Activity Monitoring

6.3.1 Data Model

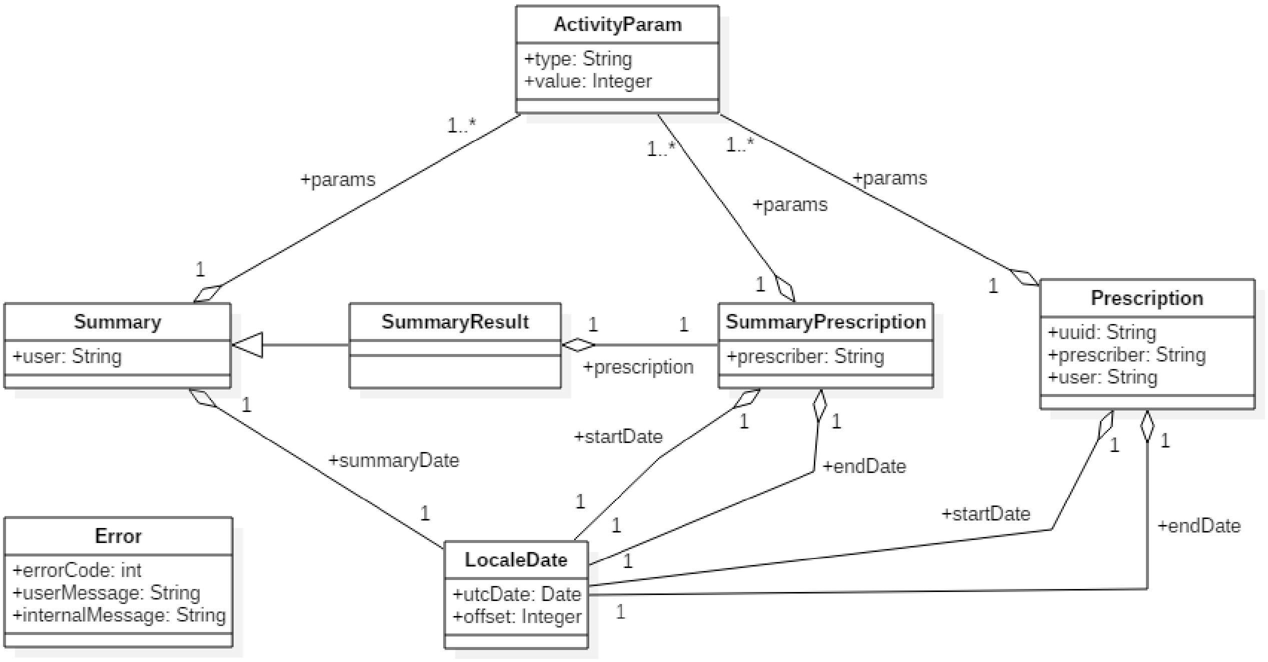


Figure 37 - Data model diagram for the Physical Activity service.

6.3.1.1 Preliminary Considerations

All the user’s id are string in order to offer flexibility to use any kind of id (characters, numbers, etc.).

6.3.1.2 ActivityParam

This class models the data monitored by the service. Due to the differences between the different wearable available in the market, each one offers different parameters to control and this class helps to refers all of them in a generic way.

Atribute	Optional	Description
Type String	**N**	Name of the activity parameter.
Value Integer	**N**	Value of the activity parameter.

JSON representation:

```
{
 "type": "string",
 "value": int
}
```

Example:

```
{
 "type": "steps",
 "value": 10000
}
```

6.3.1.3 *Prescription*

This class models the prescription of physical activity related to a user and prescribed by a professional. This class is used for saving, querying, and modifying the prescription.

Attribute	Optional	Description
uuid String	**Y**	The id of the prescription within the system (in uuid format). This attribute is used only during the querying process and is omitted during the saving or modifying process.
startDate String	**N**	First day of the prescription in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.
endDate String	**N**	Last day of the prescription in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.
prescriber String	**N**	The prescriber's id of the physical activity plan.
user String	**N**	The patient's id assigned to this prescription.
params array	**N**	Array with the prescription of parameters motorized from the user wearable. The parameters are an array of ActivityParam objects with two elements each one: • type (string): name or type of the parameter • value (integer): value of the parameter

JSON representation:

```
{
  "uuid": "string",
  "startDate": "string",
  "endDate": "string",
  "prescriber": "string",
  "user": "string",
  "params": [
    {
      "type": "string",
      "value": int
    }
  ]
}
```

Example:

```
{
  "uuid": "4028b8815dd0e28f015dd0e362050000",
  "startDate": "2017-07-01T00:00:00+0200",
  "endDate": "2017-07-31T00:00:00+0200",
  "prescriber": "2c9480845bee03e7015bfcad28990010",
  "user": "2c9480845bee03e7015bfcad7d0e0011",
  "params": [
    {
      "type": "steps",
      "value": 10000
    },
    {
      "type": "low",
      "value": 30
    }
  ]
}
```

6.3.1.4 SummaryPrescription

The minimum amount of information of a prescription is modelled with this class (used by SummaryResults model). The main difference of this class with the Prescription class consists in the omission of the fields uuid and user.

Attribute	Optional	Description
startDate String	**N**	First day of the prescription in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.
endDate String	**N**	Last day of the prescription in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.
prescriber String	**N**	The prescriber's id of the physical activity plan.
Params array	**N**	Array with the prescription of parameters motorized from the user wearable. The parameters are an array of ActivityParam objects which contains the next two elements: • type (string): name or type of the parameter • value (integer): value of the parameter

JSON representation:

```
{
  "startDate": "string",
  "endDate": "string",
  "prescriber": "string",
  "params": [
    {
      "type": "string",
      "value": int
    }
  ]
}
```

Example:

```
{
  "startDate": "2017-07-01T00:00:00+0200",
  "endDate": "2017-07-31T00:00:00+0200",
  "prescriber": "2c9480845bee03e7015bfcad28990010",
  "params": [
    {
      "type": "steps",
      "value": 10000
    },
    {
      "type": "low",
      "value": 30
    }
  ]
}
```

6.3.1.5 Summary

This class models the results of physical activity retrieved from the service of the given wearable device and stored within the CONNECARE system. This class is used for saving summary only.

Attribute	Optional	Description
summaryDate String	**N**	The date of obtaining the summary in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.
User String	**N**	The patient's id assigned to this prescription.
Params array	**N**	Array with the results of parameters motorized from the user wearable. The parameters are an array of ActivityParam objects which contains the next two elements: • type (string): name or type of the parameter • value (integer): value of the parameter

JSON representation:

```
{
  "summaryDate": "string",
  "user": "string",
  "params": [
    {
      "type": "string",
      "value": int
    }  ]
}
```

Example:

```
{
  "summaryDate": "2017-07-01T00:00:00+0200",
  "user": "2c9480845bee03e7015bfcad7d0e0011",
  "params": [
    {
      "type": "steps",
      "value": 8675
    },
    {
      "type": "high",
      "value": 25
    }
  ]
}
```

6.3.1.6 SummaryResult

This class extended from the Summary class models the results of physical activity stored within the CONNECARE system customized for querying summary/s only. It includes also the prescription related to the summary represented by SummaryPrescription object (see Section 3.4).

Attribute	Optional	Description
summaryDate String	**N**	The date of obtaining the summary in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.
user String	**N**	The patient's id assigned to this prescription.
params array	**N**	Array with the results of parameters motorized from the user wearable. The parameters are an array of ActivityParam objects which contains the next two elements: • type (string): name or type of the parameter • value (integer): value of the parameter
prescription SummaryPrescription	**N**	The prescription object related to the given summary (see Section 3.4).

JSON representation:

```
{
  "summaryDate": "string",
  "user": "string",
  "params": [
    {
      "type": "string",
      "value": int
    }
  ],
  "prescription": SummaryPrescription
}
```

Example:

```
{
  "summaryDate": "2017-07-01T00:00:00+0200",
  "user": "2c9480845bee03e7015bfcad7d0e0011",
  "params": [
    {
      "type": "steps",
      "value": 8675
    },
    {
      "type": "low",
      "value": 25
    }
  ],
  "prescription": {
    "prescriber": "2c9480845bee03e7015bfcad28990010",
    "params": [
      {
        "type": "steps",
        "value": 10000
      },
      {
        "type": "low",
        "value": 30
      }
    ]
  }
}
```

6.3.1.7 Error

This class contains the information for a request which doesn't generate a specific content. For instance, correct PUT requests generate this kind of answers or any other request if they generate an error.

Attribute	Optional	Description
errorCode Int	**N**	Internal code of the error / success.
userMessage String	**N**	User friendly message.
message integer	**N**	Internal error message.

JSON representation:

```
{
  “errors”:[
    {
      "errorCode": int,
      "userMessage": "String",
      " message ": "String"
    }
  ]
}
```

Example:

```
{
  "errors":[
    {
      "errorCode":int,
      "userMessage":"string",
      "message":"string"
    }
  ]
}
```

6.3.2 API definition

6.3.2.1 *Physical Activity Prescription*

EndPoint

POST - /physicalactivity/v1/prescription/save

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Not needed

Body

The endpoint waits for a **Prescription** object (see Section 3.3).

Responses

Success

Response code:

- 201 – Created (the operation was successfully done)

Response body:

- The endpoint returns a **Prescription** object. For example:

```
{
  "uuid": "4028b8815dd0e28f015dd0e362050000",
  "prescriber": "2c9480845bee03e7015bfcad28990010",
  "user": "2c9480845bee03e7015bfcad7d0e0011",
  "startDate": "2017-08-11T00:00:00.000+0200",
  "endDate": "2017-08-16T23:59:59.000+0200",
  "params": [
    {
      "type": "steps",
      "value": 8888
    }
    ]
}
```

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 – Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.3.2.2 Current Prescription Retrieval

EndPoint

GET - /physicalactivity/v1/prescription/user/{user-uuid}/retrieve

Authorization

Parameter	**Position**	**Description**
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	**Position**	**Description**
user-uuid String	Path	The uuid of the patient objective of the prescription.

Body

Not applicable.

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **Prescription** object.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 – Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.3.2.3 Prescription Retrieval for a Given Date

EndPoint

GET - /physicalactivity/v1/prescription/user/{user-uuid}/date/{date}/retrieve

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
user-uuid String	Path	The uuid of the patient objective of the prescription.
date String	Path	The date for which the consultation is made if there is any prescription in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.

Body

Not applicable.

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **Prescription** object.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 – Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.3.2.4 List of Prescriptions Retrieval in a Date Interval

EndPoint

GET - /physicalactivity/v1/prescription/user/{user-uuid}/startDate/{startDate}/endDate/{endDate}/retrieve

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
user-uuid String	Path	The uuid of the patient objective of the prescription.
startDate String	Path	The start date of the period for which the consultation is made if there is any prescription in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.
endDate String	Path	The end date of the period for which the consultation is made if there is any prescription in format yyyy-MM-dd'T'HH:mm:ssZ. The time will not be taken into account, only the date.

Body

Not applicable.

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **Prescription** object.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 – Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.3.2.5 Current Prescription Deleting

EndPoint

PUT - /physicalactivity/v1/prescription/user/{user-uuid}/cancel

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
user-uuid String	Path	The uuid of the patient objective of the prescription.

Body

Not applicable.

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the next message: “The active prescription has been cancelled”

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 400 – Bad request

- 401 – Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.3.2.6 Prescription Updating

EndPoint

PUT - /physicalactivity/v1/prescription/{prescription-uuid}/update

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
prescription-uuid String	Path	The uuid of the target prescription for the modification

Body

The endpoint waits for a **Prescription**.

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Response body:

- The endpoint returns a **Prescription** object (see Section 3.3). For example:

```
{
  "uuid": "4028b8815dd0e28f015dd0e362050000",
  "prescriber": "2c9480845bee03e7015bfcad28990010",
  "user": "2c9480845bee03e7015bfcad7d0e0011",
  "startDate": "2017-08-11T00:00:00.000+0200",
  "endDate": "2017-08-16T23:59:59.000+0200",
  "params": [
    {
      "type": "steps",
      "value": 8888
    }
  ]
}
```

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 – Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.3.2.7 User's Summary Saving

EndPoint

POST - /physicalactivity/v1/summary/save

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
username String	Header	The username or user id of the person/system who does the consult.

Body

The endpoint waits for a **Summary** object.

Responses

Success

- 201 – Created (the operation was successfully done)

Body message:

- The endpoint returns the next message: "Summary Saved"

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 – Unauthorized

- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.3.2.8 User's Summary Retrieval for a Given Date

EndPoint

GET - /physicalactivity/v1/summary/user/{user-uuid}/date/{date}/retrieve?filters={filters}

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
user-uuid String	Path	The uuid of the patient objective of the prescription.
date String	Path	Date of the day to retrieve
filters Array	Query	Array with the filters to apply (optional)

Body

Not applicable.

Responses

Success

Code error:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **SummaryResult** object.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 – Unauthorized
- 403 – Forbidden
- 404 – Not found

- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.3.2.9 User's Daily Summaries Retrieval in a Date Interval

EndPoint

GET - /physicalactivity/v1/summary/user/{user-uuid}/startDate/{startDate}/endDate/{endDate}/list?filters={filters}

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
user-uuid String	Path	Patient objective of the prescription.
startDate String	Path	Date of the first day to retrieve
endDate String	Path	Date of the last day to retrieve
filters Array	QueryString	Array with the filters to apply (optional)

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array of **SummaryResults** object.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 – Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See Section 6.3.1.7 for more details

6.4 Sleeping

6.4.1 Data Model

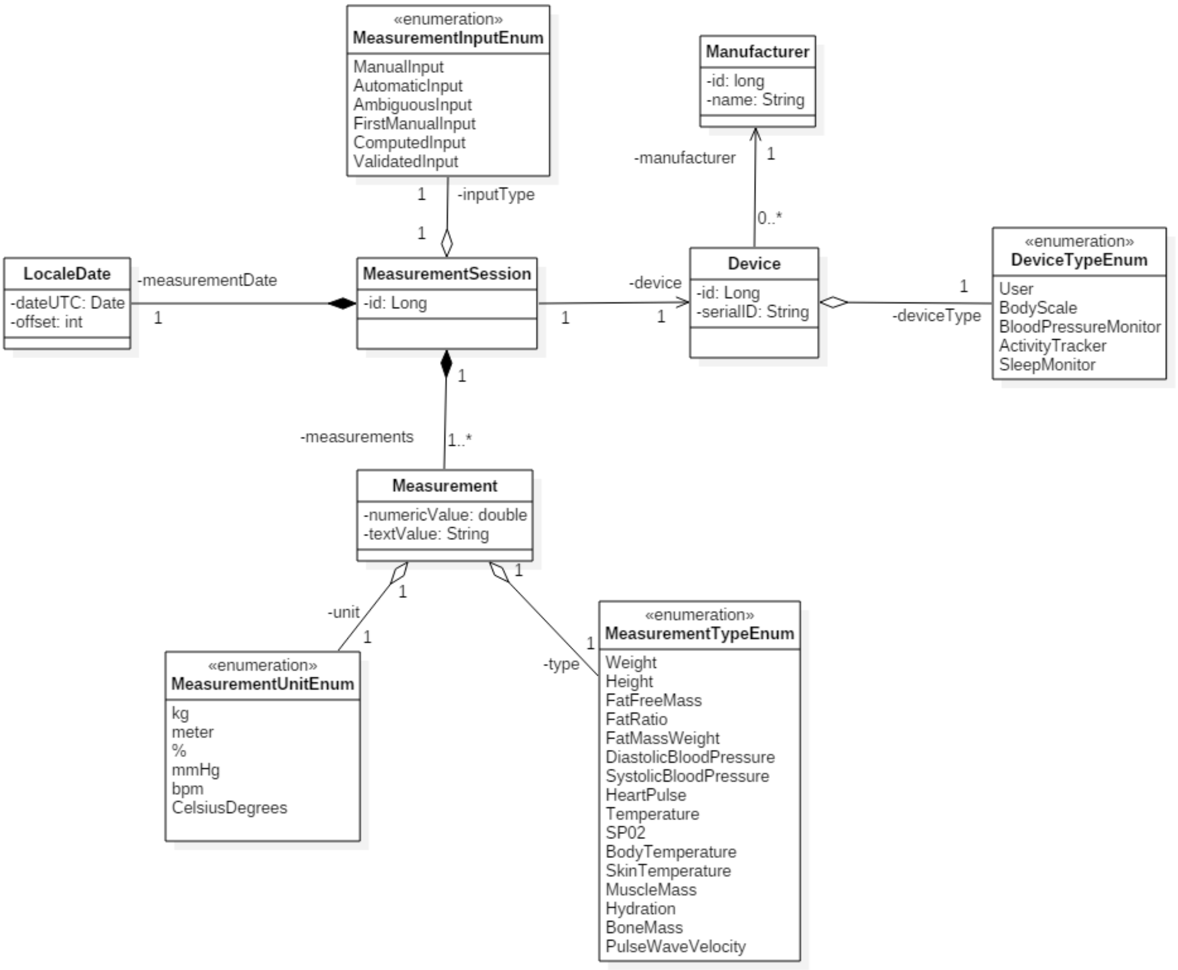


Figure 38 - Monitoring Data model diagram.

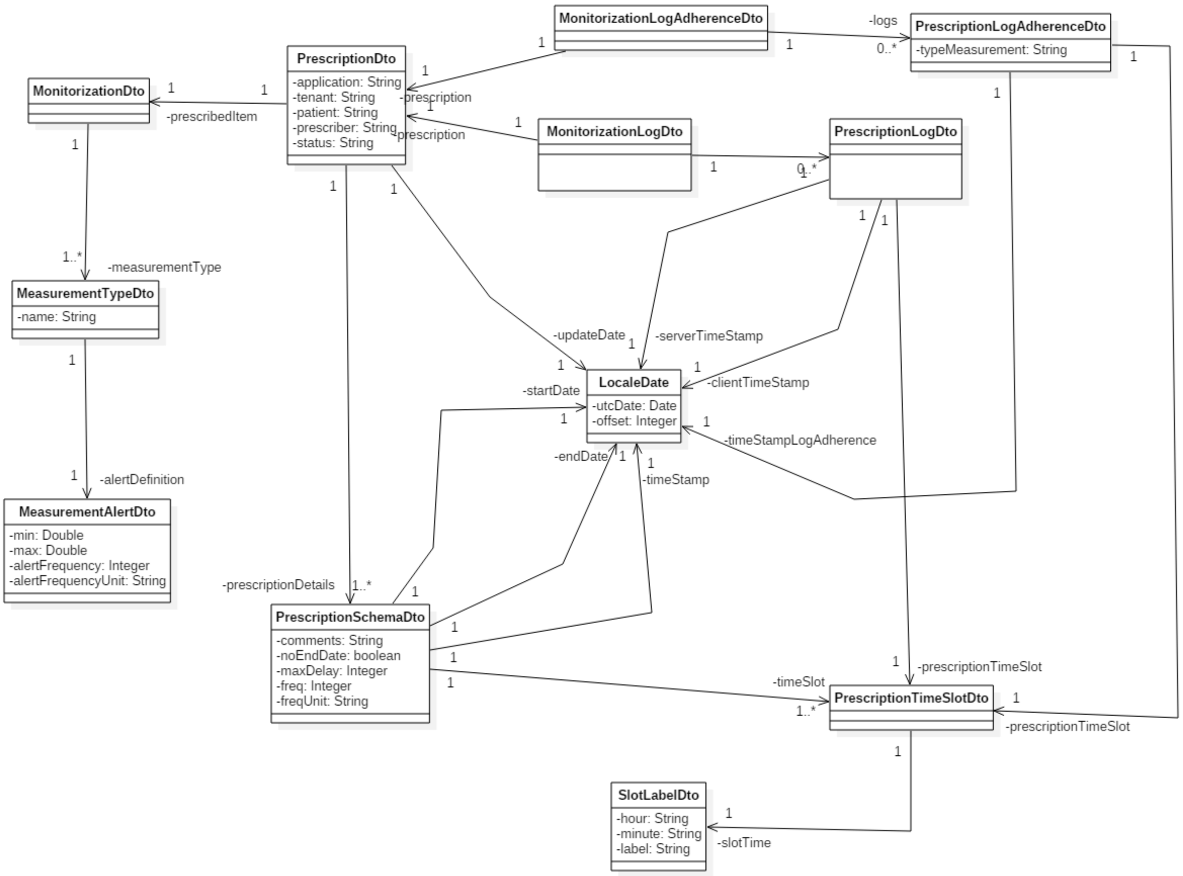


Figure 39. Prescription model diagram

6.4.1.1 Preliminary considerations

All the user's id are Universal Unique Ids. The representation as string is to allow the possibility to offer flexibility to use any kind of id (characters, numbers, etc.).

6.4.1.2 PrescriptionDto

The object of type **PrescriptionDto**

Atribute	Optional	Description
Id Long		Prescription Identifier
application String	**N**	Uuid of the application
tenant String	**N**	Uuid of the tenant of the application
patient String	**N**	Uuid of the patient objective of the prescription
prescriber String	**N**	Uuid of the prescriber of the medical device plan

status String		Enum that represents the different status that a prescription can have: • Prescribed • InTreatment • Finished • Deleted • Canceled
updateDate <LocalDate>		The date of the last update of the prescription in UTC
prescribedItem <MonitorizationDto>	**N**	The MonitorizationDto object of a prescription *See section 6.4.1.33*
prescriptionDetails ArrayList<PrescriptionSchemaDto>	**N**	List of the prescriptionSchemaDto objects that a prescription can have *See section 6.4.1.6*

Model Schema:

```
{
  "id": Long,
  "application": "string",
  "tenant": "string",
  "patient": "string",
  "prescriber": "string",
  "status": "string",
  "udateDate": "string",
  "treatmentDocumentation": <DocumentDto>,
  "prescribedItem": <MonitorizationDto>,
  "prescriptionDetails": ArrayList<PrescriptionSchemaDto>
}
```

6.4.1.3 MonitorizationDto

The object of type **MonitorizationDto**

Atribute	**Optional**	**Description**
measurementType ArrayList<MeasurementTypeDto>		List of the measurement types of the prescription *See section 6.4.1.4*

Model Schema:

```
{
  "measurementType": ArrayList<MeasurementTypeDto>
}
```

6.4.1.4 MeasurementTypeDto

The object of type **MeasurementTypeDto**

Atribute	**Optional**	**Description**

name MeasurementTypeEnum		Enum that represents the type of the measure to be taken: • Sleep • SleepWakeUpDuration • LightSleepDuration • DeepSleepDuration • SleepWakeUpCount • DurantionToSleep • SleepHour • SleepWakeUpHour
alertDefinition MeasurementAlertDto		Alert’s definition for the measurement type indicated *See section 6.4.1.5*

Model Schema:

```
{
    “name” : “String”,
    “alertDefintion” : <MeasurementAlertDto>
}
```

6.4.1.5 MeasurementAlertDto

The object of type **MeasurementAlertDto**

Atribute	**Optional**	**Description**
min double		Minimum value of a measure to throw an alert
max double		Maximum value of a measure to throw an alert
alertFrequency Integer		Frequency that represents when the alert must be thrown
alertFrequencyUnit FreqUnitEnum		Enum that represents the frequency of the alert: • Days • Weeks • Months

Model Schema:

```
{
    “min” : double,
    “max” : double,
    “alertFrequency” : Integer,
    “alertFrequencyUnit” : “String”
}
```

6.4.1.6 PrescriptionSchemaDto

The object of type **PrescriptionSchemaDto**

Atribute	Optional	Description
Id Long		Prescription Schema Identifier
comments String		Commentaries of the prescription schema
noEndDate Boolean		Boolean that represents if the prescription schema has an end date definition or not
maxDelay Integer		Represent the maximum delay allowed when taking a measure (in minutes)
freq Integer		Represent the frequency when the measures have to be taken
freqUnit FreqUnitEnum	**N**	Enum that represents the different frequencies allowed for a schema: • Days • Weeks • Months
startingDate <LocaleDate>	**N**	The date when the schema will start being active in UTC
endingDate <LocaleDate>		The date when the schema will stop being active in UTC
timeStamp <LocaleDate>		The datetime of the last update in UTC
timeSlot ArrayList<PrescriptionTimeSlotDto>	**N**	List of the PrescriptionTimeSlotDto objects *See section 6.4.1.7*

Model Schema:

```
{
    “id” : Long,
    “comments” : “String”,
    “noEndDate” : Boolean,
    “maxDelay” : Integer,
    “freq” : Integer,
    “freqUnit” : “String”,
    “startingDate” : <LocaleDate>,
    “endingDate” : <LocaleDate>,
    “timeStamp” : <LocaleDate>,
    “timeslot” : ArrayList<PrescriptionTimeSlotDto>
}
```

6.4.1.7 PrescriptionTimeSlotDto

The object of type **PrescriptionTimeSlotDto**

Atribute	Optional	Description
slotTime <SlotLabelDto>	**N**	The SlotLabelDto object *See section 6.4.1.8*

Model Schema:

```
{
    "slotTime" : <SlotLabelDto>
}
```

6.4.1.8 SlotLabelDto

The object of type **SlotLabelDto**

Atribute	Optional	Description
hours Integer		Hours when the measure has to be taken
minutes Integer		Hours when the measure has to be taken
label TimeSlotMealEnum	**N**	Enum that represents the different time slots of the prescription: • Breakfast • Lunch • AfternoonSnack • Dinner • BeforeSleep • SpecficHour

Model Schema:

```
{
    "hours" : Integer,
    "minutes" : Integer,
    "label" : "String"
}
```

6.4.1.9 MonitorizationLogDto

The object of type **MonitorizationLogDto**

Atribute	Optional	Description
prescription <PrescriptionDto>		The data of the description of the prescription *See section 6.4.1.2*
logs ArrayList<PrescriptionLogDto>		List of the prescriptions logs of the prescription *See section 6.4.1.11*

Model Schema:

```
{
  "prescription": <PrescriptionDto>,
  "logs": ArrayList<PrescriptionLogDto>
}
```

6.4.1.10MonitorizationLogAdherenceDto

The object of type **MonitorizationAdherenceLogDto**

Atribute	Optional	Description
prescription <Prescription>		The data of the description of the prescription *See section 6.4.1.2*
logs ArrayList<PrescriptionLogAdherenceDto>		List of the prescriptions logs of the prescription *See section 6.4.1.12*

Model Schema:

```
{
  "prescription": <PrescriptionDto>,
  "logs": ArrayList<PrescriptionLogAdherenceDto>
}
```

6.4.1.11PrescriptionLogDto

The object of type **MonitorizationAdherenceLogDto**

Atribute	Optional	Description
patient String		Uuid of the patient objective of the prescription.
type MeasurementTypeEnum		Enum that represents the type of the measure to be taken: • Sleep • SleepWakeUpDuration • LightSleepDuration • DeepSleepDuration • SleepWakeUpCount • DurantionToSleep • SleepHour • SleepWakeUpHour
numericValue Double		The numeric value of the measure taken.
unit MeasurementUnitEnum		Enum that represents the unit of the measure to be taken: • kg • meter • % • mmHg • bpm • CelsiusDegrees

		• Seconds • Times • timestamp
inTime Boolean		Boolean that represents if the measure has been taken in time.

Model Schema:

```
{
  "patient" : "String",
  "type" : "String",
  "numericValue" : Double,
  "unit" : "String",
  “inTime” : Boolean
}
```

6.4.1.12PrescriptionLogAdherenceDto

The object of type **MonitorizationAdherenceLogDto**

Atribute	**Optional**	**Description**
prescriptionTimeSlot <PrescriptionTimeSlotDto>		The data of the description of the prescription *See section 6.4.1.7*
timeStampLogAdherence <LocaleDate>	**N**	The datetime of the last update in UTC
typeMeasurement MeasurementTypeEnum		Enum that represents the type of the measure to be taken: • Sleep • SleepWakeUpDuration • LightSleepDuration • DeepSleepDuration • SleepWakeUpCount • DurantionToSleep • SleepHour • SleepWakeUpHour
numberLogsReceived Double		Double that represents the number of logs received.
numberLogsPredicted Double		Double that represents the number of logs expected.
Adherence Double		Double that represents the adherence of the patient in a prescription.

Model Schema:

```
{
  "prescripitonTimeSlot": <PrescriptionTimeSlotDto>.id,
  "timeStampLogAdherence": "String",
  "typeMeasurement" : "String",
  "numberLogsReceived" : Double,
  "numberLogsPredicted" : Double,
  "Adherence" : Double
}
```

6.4.1.13AdherenceDefinitionDto

Atribute	Description
alertPeriod String	Enum that represents the frequency of the alert: • Days • Weeks • Months
application String	Uuid of the application.
patient String	Uuid of the patient.
minimumAdherence Double	Value that represent the minimum adherence of the patient for the patient to throw an alert.
prescriber String	Uuid of the prescriber.

Model Schema:

```
{
    "alertPeriod": "String",
    "application": "String",
    "minimumAdherence": Double,
    "patient": "String",
    "prescriber": "String"
}
```

6.4.1.14Error

This class contains the information for a request which doesn’t generate a specific content. For instance, correct PUT requests generate this kind of answers or any other request if they generate an error.

Attribute	Description
errorCode Int	Internal code of the error / success.
userMessage String	User friendly message.
internalMessage integer	Internal error message.

Model Schema:

```
{
  "errorCode": int,
  "userMessage": "String",
  "internalMessage": "String"
}
```

6.4.2 API Definition

6.4.2.1 Patient Monitoring Prescription

EndPoint

POST - /patientmonitoring/v1/prescription/user/{user}/save

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
application String	Header	The application from which the request is performed. In case of CONNECARE it will be SMS.
tenant String	Header	UUID of the tenant for the given user.
uuid String	Header	UUID of the service's current user.
user String	Path	Patient objective of the prescription.

Body

The endpoint waits for a **PrescriptionDto** object (see section 6.4.1.2).

Responses

Success

Code error:

- 200 – OK (The operation was successfully done).

Body message:

- The endpoint returns a **PrescriptionDto** object (see section 6.4.1.2).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.14 for more details

6.4.2.2 Prescription Retrieval by User

EndPoint

GET - /patientmonitoring/v1/user/{user}/retrieve

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service's current user.
user **String**	Path	Patient objective of the prescription.
filter **String**	Query	Filter to choose to retrieve between all prescriptions or only the active ones. It can have two values: All or InTreatment

Body

Not applicable.

Responses

Success

Error code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array of **PrescriptionDto** objects (see section 6.4.1.2).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.146.3.1.7 for more details

6.4.2.3 Cancel an Active Prescription

EndPoint

PUT - /patientmonitoring/v1/prescription/{prescriptionUuid}/cancel

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service’s current user.
prescriptionUuid String	Path	UUID of the active prescription.

Body

Not applicable.

Responses

Success

Code error:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **PrescriptionDto** object (see section 6.4.1.2).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.146.3.1.7 for more details

6.4.2.4 Delete a Non-Active Prescription

EndPoint

DELETE - /patientmonitoring/v1/prescription/{prescriptionUuid}/delete

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service's current user.
prescriptionUuid String	Path	UUID of the non-active prescription.

Body

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array of **PrescriptionDto** object (see section 6.4.1.2).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.14 for more details

6.4.2.5 Update Non-Active Prescription

EndPoint

PUT - /patientmonitoring/v1/prescription/{prescriptionUuid}/update

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service's current user.
prescriptionUuid String	Path	UUID of the non-active prescription.

Body

The endpoint waits for a **PrescriptionDto** object (see section 6.4.1.2).

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **PrescriptionDto** object (see section 6.4.1.2).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.14 for more details

6.4.2.6 Update Active Prescription

EndPoint

PUT - /patientmonitoring/v1/prescription/{prescripitionUuid}/schema/{schemaUuid}/update

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service's current user.
prescriptionUuid String	Path	Uuid of the active prescription.
schemaUuid String	Path	Uuid of the prescription schema.
startDate String	Path	Date of the first day to retrieve.
endDate String	Path	Date of the last day to retrieve.
measurementType String	Path	The type of measurement to be listed. The allowed values are the contained in the MeasurementTypeEnum described in the section 3.

Body

The endpoint waits for a **PrescriptionSchemaDto** object (see section 6.4.1.66.4.1.2).

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **PrescriptionDto** object (see section 6.4.1.2).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 **-** Unauthorized

- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.146.3.1.7 for more details

6.4.2.7 Logs Retrieval by User

EndPoint

GET - /patientmonitoring/v1/prescription/log/user/{user}/status/report

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service's current user.
user String	Path	Patient objective of the prescription.

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array of **MonitorizationLogDto** objects (see section 6.4.1.9).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.146.3.1.7 for more details

6.4.2.8 Logs Retrieval by User and Dates

EndPoint

GET - /patientmonitoring/v1/prescription/log/user/{user}/start/{startDate}/end/{endDate}/status/report

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service's current user.
user String	Path	Patient objective of the prescription.
startDate String	Path	Date of the first day to retrieve
endDate String	Path	Date of the last day to retrieve

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array of **MonitorizationLogDto** objects (see section 6.4.1.9).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.146.3.1.7 for more details

6.4.2.9 Adherence Prescription

EndPoint

POST - /patientmonitoring/v1/adherence/definition/save

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service's current user.
tenant String	Header	UUID of the tenant for the given user.

Body

The endpoint waits for an **AdherenceDefinitionDto** object (see section 6.4.1.13).

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an **AdherenceDefinitionDto** object (see section 6.4.1.13).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.146.3.1.7 for more details

6.4.2.10Adherence Definition Retrieval by User

EndPoint

GET - /patientmonitoring/v1/adherence/user/{user}/application/{application}/retrieve

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service's current user.
tenant String	Header	UUID of the tenant for the given user.
user String	Path	Patient objective of the prescription.
application String	Path	The application from which the request is performed. In case of CONNECARE it will be SMS.

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array of **AdherenceDefinitionDto** objects (see section 6.4.1.13).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

- See section 6.4.1.146.3.1.7 for more details

6.4.2.11Adherence Definition Retrieval by User and Prescriber

EndPoint

GET -
/patientmonitoring/v1/adherence/prescriber/{prescriber}/user/{user}/application/{application}/retrieve

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service’s current user.
tenant String	Header	UUID of the tenant for the given user.
prescriber String	Path	UUID of the prescriber.
user String	Path	Patient objective of the prescription.
application String	Path	The application from which the request is performed. In case of CONNECARE it will be SMS.

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array of **AdherenceDefinitionDto** objects (see section 6.4.1.13).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

See section 6.4.1.146.3.1.7 for more details

6.4.2.12Calculate Adherence Daily

EndPoint

POST - /patientmonitoring/v1/prescription/calculate/adherence/daily

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Not applicable.

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the message “Prescription log adherence has been calculated correctly (Daily)”

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

See section 6.4.1.146.3.1.7 for more details

6.4.2.13Calculate Adherence Weekly

EndPoint

POST - /patientmonitoring/v1/prescription/calculate/adherence/weekly

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Not applicable.

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the message "Prescription log accumulated adherence has been calculated correctly (Weekly)"

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

See section 6.4.1.146.3.1.7 for more details

6.4.2.14Calculate Adherence Monthly

EndPoint

POST - /patientmonitoring/v1/prescription/calculate/adherence/monthly

Authorization

Parameter	**Position**	**Description**
Authorization String	Header	The Bearer access token for SMS

Parameters

Not applicable.

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the message “Prescription log accumulated adherence has been calculated correctly (Monthly)”.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

See section 6.4.1.146.3.1.7 for more details

6.4.2.15Update Adherence Prescription

EndPoint

PUT - /patientmonitoring/v1/adherence/user/{user}/application/{application}/update

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service’s current user.
tenant String	Header	UUID of the tenant for the given user.
user String	Path	Patient objective of the prescription.
application String	Path	The application from which the request is performed. In case of CONNECARE it will be SMS.

Body

The endpoint waits for an **AdherenceDefinitionDto** object (see section6.4.1.13 6.4.1.13).

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an **AdherenceDefinitionDto** object (see section 6.4.1.13).

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

See section 6.4.1.146.3.1.7 for more details

6.4.2.16Delete Adherence Prescription

EndPoint

DELETE - /patientmonitoring/v1/adherence/user/{user}/application/{application}/delete

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
uuid String	Header	UUID of the service’s current user.
tenant String	Header	UUID of the tenant for the given user.
user String	Path	Patient objective of the prescription.
application String	Path	The application from which the request is performed. In case of CONNECARE it will be SMS.

Body

Not applicable.

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the message “Deleted the AdherenceDefinition”.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

- 401 - Unauthorized
- 403 – Forbidden
- 404 – Not found
- 409 – Conflict

Body message:

See section 6.4.1.146.3.1.7 for more details

6.5 Questionnaires

6.5.1 Data Model

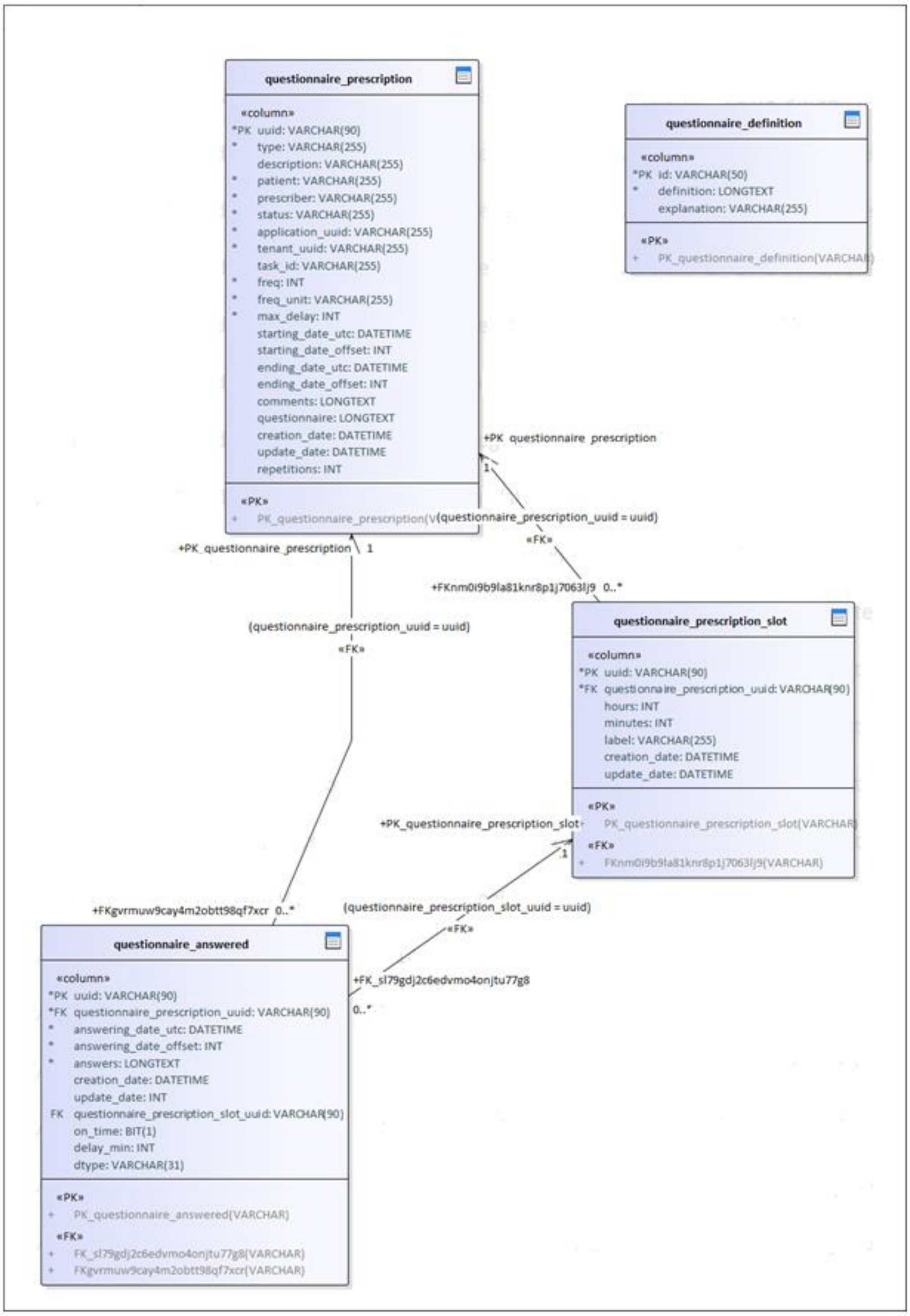

questionnaire_prescription
«column»
*PK uuid: VARCHAR(90)
* type: VARCHAR(255)
description: VARCHAR(255)
* patient: VARCHAR(255)
* prescriber: VARCHAR(255)
* status: VARCHAR(255)
* application_uuid: VARCHAR(255)
* tenant_uuid: VARCHAR(255)
task_id: VARCHAR(255)
* freq: INT
* freq_unit: VARCHAR(255)
* max_delay: INT
starting_date_utc: DATETIME
starting_date_offset: INT
ending_date_utc: DATETIME
ending_date_offset: INT
comments: LONGTEXT
questionnaire: LONGTEXT
creation_date: DATETIME
update_date: DATETIME
repetitions: INT
«PK»
+ PK_questionnaire_prescription(V
questionnaire_definition
«column»
*PK id: VARCHAR(50)
* definition: LONGTEXT
explanation: VARCHAR(255)
«PK»
+ PK_questionnaire_definition(VARCHAR)
+PK questionnaire prescription
1
(questionnaire_prescription_uuid = uuid)
«FK»
+FKnm0i9b9la81knr8p1j7063lj9 0..*
+PK_questionnaire_prescription 1
(questionnaire_prescription_uuid = uuid)
«FK»
questionnaire_prescription_slot
«column»
*PK uuid: VARCHAR(90)
*FK questionnaire_prescription_uuid: VARCHAR(90)
hours: INT
minutes: INT
label: VARCHAR(255)
creation_date: DATETIME
update_date: DATETIME
«PK»
+ PK_questionnaire_prescription_slot(VARCHAR)
«FK»
+ FKnm0i9b9la81knr8p1j7063lj9(VARCHAR)
+PK_questionnaire_prescription_slot
1
+FKgvrmuw9cay4m2obtt98qf7xcr 0..*
(questionnaire_prescription_slot_uuid = uuid)
«FK»
+FK_sl79gdj2c6edvmo4onjtu77g8
0..*
questionnaire_answered
«column»
*PK uuid: VARCHAR(90)
*FK questionnaire_prescription_uuid: VARCHAR(90)
* answering_date_utc: DATETIME
* answering_date_offset: INT
* answers: LONGTEXT
creation_date: DATETIME
update_date: INT
FK questionnaire_prescription_slot_uuid: VARCHAR(90)
on_time: BIT(1)
delay_min: INT
dtype: VARCHAR(31)
«PK»
+ PK_questionnaire_answered(VARCHAR)
«FK»
+ FK_sl79gdj2c6edvmo4onjtu77g8(VARCHAR)
+ FKgvrmuw9cay4m2obtt98qf7xcr(VARCHAR)

Figure 40 - Data model diagram for the Questionnaire service.

6.5.1.1 Preliminary considerations

- All the user's id are string in order to offer flexibility to use any kind of id (characters, numbers, etc.)
- A Postman collection file is available.

6.5.1.2 Error

This class contains the information for a request which doesn't generate a specific content. For instance, correct POST requests generate this kind of answers or any other request if they generate an error.

Attribute	Optional	Description
errorCode Int	**N**	Internal code of the error / success.
userMessage String	**N**	User friendly message.
internalMessage integer	**N**	Internal error message.

Model Schema:

```
{
  "errorCode": Integer,
  "userMessage": "String",
  "internalMessage": "String"
}
```

6.5.1.3 QuestionGroupSummary

This object contains the summary of the question group available in the system to which the user has access.

Atribute	Description
Id String	Alphanumeric id of the question group
Name String	The name of the question group
SortOrder Int	Number that indicates the order of the given question group within the list
AdviceGroupId String	Alphanumeric id of the advice group
IsOpen Boolean	Indicate if this question group is open to be answered

6.5.1.4 QuestionGroup

This object models the question group which contains the questionnaire information, questions and possible answers.

Atribute	Description
Name String	The name of the questionnaire
Instruction String	Contain the description of the questionnaire
Questions List<Questions>	The list of Question objects
Status Integer	The code which indicates the status of the question
Message String	Some optional message within the question

6.5.1.5 Question

This object contains the question model.

Atribute	Description
Id String	Alphanumeric id of the question
Name String	The question
Type Int	The type of question. Available types of question are: 0 – Unknown 1 – NoQuestion: No question 2 – OpenQuestion: Open question 3 – Number: Number field 4 – Date: Date 5 – YesNoSlide: Yes / No slide 6 – SingleRadio: Single selection checkbox 7 – SingleDropDown: Single selection drop down 8 – SingleAutoComplete: Single auto completion selection 9 – SingleSlide: Slide 1 value 10 – RangeSlide: Slide 2 values (range) 11 – MultipleCheckbox: Multiple selection checkbox 12 – MultipleAutoComplete: Multiple auto completion selection
SortOrder Int	Number that indicates the order of the given question within the list
FollowUpQuestionGroupId String	Alphanumeric id of the followup question group
Answers List<Answer>	The list with the possible answers

6.5.1.6 Answer

This object contains the answer model.

Atribute	Description
Id String	Alphanumeric id of the answer
Name String	The answer description

SortOrder Int	Number that indicates the order of the given answer within the list
FollowUpQuestionGroupId String	Alphanumeric id of the followup question group

6.5.1.7 UserAnswerGroup

This object contains the list with the user answers to a given question group.

Atribute	Description
BaseQuestionGroupId String	Alphanumeric id of the target question group
Answers List<UserAnswer>	The list with the user answers to the given questionnaire

6.5.1.8 UserAnswer

This object models the user answer.

Atribute	Description
Id String	Alphanumeric id of the user answer
AnswerText String	The the which the answer includes
LeadingAnswerId String	Alphanumeric id of the leading answer

6.5.2 API definition

6.5.2.1 Get Available Question Groups

EndPoint

POST – /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/questionnaire/availablequestiongroups.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Description

Retrieves a list of available question groups from the service for the given user.

Parameters

Parameter	Position	Description
application String	Query	The application from which the request is performed.

		In case of CONNECARE it will be 2c9480845bee03e7015bfc0266d00000.
user-uuid String	Query	The uuid of the user for which the question group is being consulted.

Body

Empty body

Responses

Success

HTTP response:

- 200 – OK (the operation was successfully done)

Body message:

```
{
    "QuestionGroups": [
        {
            "Id": "8c540b13-4258-4cbe-8798-79ae4601b59d",
            "Name": "The simple FRAIL questionaire",
            "SortOrder": 0,
            "AdviceGroupId": "645c28de-63f7-4503-bfaa-b8ca942dbfa8",
            "IsOpen": true
        },
        {
            "Id": "025506c4-8476-4c19-99dd-e4eae45da4e4",
            "Name": "Test",
            "SortOrder": 0,
            "AdviceGroupId": null,
            "IsOpen": false
        }
    ],
    "Status": 0,
    "Message": ""
}
```

The answer contains a list of objects of type **QuestionGroupSummary**.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

400 – BAD REQUEST

404 – NOT FOUND

500 – SERVER ERROR

Body message:

- See Section 6.5.1.2 for more details

6.5.2.2 Question Group Triggering

EndPoint

POST – /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/questionnaire/triggerquestiongroup.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Description

The question group (questionnaire) with the given Id will be linked to the user.

Parameters

Parameter	Position	Description
application String	Query	The application from which the request is performed. In case of CONNECARE it will be 2c9480845bee03e7015bfc0266d00000.
user-uuid String	Query	The uuid of the user for which the question group is being consulted.

Body

```
{
  "Id": "8c540b13-4258-4cbe-8798-79ae4601b59d",
  "AvailableFromDate": "2017-07-19T12:55:56.4801769+02:00"
}
```

Responses

Success

HTTP response:

- 200 – OK (the operation was successfully done)

Body message:

```
{
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

400 – BAD REQUEST

404 – NOT FOUND

500 – SERVER ERROR

Body message:

- See Section 6.5.1.2 for more details

6.5.2.3 Getting the Information of a Specific Questionnaire

EndPoint

POST – /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/questionnaire/questiongroup.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Description

Get the detailed information of the given question group (questionnaire).

Parameters

Parameter	Position	Description
application String	Query	The application from which the request is performed. In case of CONNECARE it will be 2c9480845bee03e7015bfc0266d00000.
user-uuid String	Query	The uuid of the user for which the questions are being consulted.

Body

```
{
  "Id": "8c540b13-4258-4cbe-8798-79ae4601b59d",
  "LoadQuestions": true,
  "LoadAnswers": true
}
```

Responses

Success

HTTP response:

- 200 – OK (the operation was successfully done)

Body message:

```
{
    "Name": "The simple FRAIL questionaire",
    "Instruction": " ",
    "Questions": [
        {
            "Id": "e8c3be45-7374-4565-a3df-48baf69c41c2",
            "Name": "How much of the time during the past 4 weeks did you
feel tired? ",
            "Type": 6,
            "SortOrder": 0,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "e33ebab9-f5e9-4a63-b668-b1060157288b",
                    "Name": "1) All of the time",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "b75316a9-90ee-41f5-aaff-1972dec95a17",
                    "Name": "2) Most of the time",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "2a424051-c951-4233-bbc2-29a3487cab7b",
                    "Name": "3) Some of the time",
                    "SortOrder": 2,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "54acc3ff-5ecf-4aaf-b183-af79d4e57adf",
                    "Name": "4) A little of the time",
                    "SortOrder": 3,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "b994e593-b910-42ba-9d0d-3f625f87a6d3",
                    "Name": "5) None of the time",
                    "SortOrder": 4,
                    "FollowUpQuestionGroupId": null
                }
            ]
        },
        {
            "Id": "cf828533-57a2-485e-b1db-4ac83da5a67d",
            "Name": "By yourself and not using aids, do you have any
difficulty walking up 10 steps without resting? ",
            "Type": 6,
            "SortOrder": 1,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "5f5090a2-a45c-4591-ae92-aafc8a1d2854",
                    "Name": "1) Yes ",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "9fd1974f-fe95-4518-ad25-809a8d57ef0a",
                    "Name": "2) No ",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                }
            ]
        },
        {
            "Id": "2416c1a7-e58b-4e66-8814-f544c7dbbf97",
            "Name": "By yourself and not using aids, do you have any 
difficulty walking several hundred yards? ",
            "Type": 6,
            "SortOrder": 2,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "eba99bb6-afb1-4aba-8440-32bc1635028b",
                    "Name": "1) Yes ",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "ca5d65ba-175b-40cb-864f-ec19b4a9ec99",
                    "Name": "2) No ",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                }
            ]
        },
        {
            "Id": "fd82a226-dc80-48f5-9adf-1580dffe912c",
            "Name": "Did a doctor ever tell you that you have: ",
            "Type": 11,
            "SortOrder": 3,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "e0d8598b-8c0e-4401-9dca-4e37fe05681f",
                    "Name": "hypertension",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "955c02bd-86b4-415e-b9a2-7ffd6884748f",
                    "Name": "diabetes",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "d565558d-2e65-490b-860a-0e17fef1e829",
                    "Name": "cancer (other than a minor skin cancer) ",
                    "SortOrder": 2,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "33e68603-06b3-46f0-9103-ccbac2e6a109",
                    "Name": "chronic lung disease ",
                    "SortOrder": 3,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "c9b8ca7a-1ef2-4b79-a2c8-d55d3de1fb9c",
                    "Name": "heart attack",
                    "SortOrder": 4,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "7242f6d9-c015-4b58-98ef-357bb9a1926b",
                    "Name": "congestive heart failure ",
                    "SortOrder": 5,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "0628814c-aea0-41a8-b285-81db468ed8e4",
                    "Name": "angina",
                    "SortOrder": 6,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "6791c992-4e07-4384-90fc-24cef470b878",
                    "Name": "asthma ",
                    "SortOrder": 7,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "51ce283d-5c34-4c3d-accb-8bd6e409b043",
                    "Name": "arthritis ",
                    "SortOrder": 8,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "abcfbc16-4de8-4d51-8ecc-412ed44a1e24",
                    "Name": "stroke ",
                    "SortOrder": 9,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "d6a4cfa4-cb56-403e-9530-9898d3308b06",
                    "Name": "kidney disease",
                    "SortOrder": 10,
                    "FollowUpQuestionGroupId": null
                }
            ]
        },
        {
            "Id": "69141a80-6a18-4ae2-b799-4d12e66a97dd",
            "Name": "Have you lost more than 5% of your weight in the past 6
months",
            "Type": 5,
            "SortOrder": 4,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "8f0f6b68-5620-4d71-9253-6c4010100b1f",
                    "Name": "Yes",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "f4035c01-62e4-4865-86a5-597a3cf50cef",
                    "Name": "No",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                }
            ]
        }
    ],
    "Status": 0,
    "Message": ""
}
```

The answer contains an object of type **QuestionGroup** *which contains* Question and Answer objects.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

400 – BAD REQUEST

404 – NOT FOUND

500 – SERVER ERROR

Body message:

- See Section 6.5.1.2 for more details

6.5.2.4 Getting Questions of a Specific Questionnaire

EndPoint

POST – /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/questionnaire/questionsforgroup.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Description

Retrieves a list of questions for a given question group (questionnaire).

Parameters

Parameter	Position	Description

application String	Query	The application from which the request is performed. In case of CONNECARE it will be 2c9480845bee03e7015bfc0266d00000.
user-uuid String	Query	The uuid of the user for which the questions are being consulted.

Body

```
{
  "Id": "8c540b13-4258-4cbe-8798-79ae4601b59d",
  "LoadAnswers": true
}
```

Responses

Success

HTTP response:

- 200 – OK (the operation was successfully done)

Body message:

```
{
    "Questions": [
        {
            "Id": "e8c3be45-7374-4565-a3df-48baf69c41c2",
            "Name": "How much of the time during the past 4 weeks did you
feel tired? ",
            "Type": 6,
            "SortOrder": 0,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "e33ebab9-f5e9-4a63-b668-b1060157288b",
                    "Name": "1) All of the time",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "b75316a9-90ee-41f5-aaff-1972dec95a17",
                    "Name": "2) Most of the time",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "2a424051-c951-4233-bbc2-29a3487cab7b",
                    "Name": "3) Some of the time",
                    "SortOrder": 2,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "54acc3ff-5ecf-4aaf-b183-af79d4e57adf",
                    "Name": "4) A little of the time",
                    "SortOrder": 3,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "b994e593-b910-42ba-9d0d-3f625f87a6d3",
                    "Name": "5) None of the time",
                    "SortOrder": 4,
                    "FollowUpQuestionGroupId": null
                }
            ]
        },
        {
            "Id": "cf828533-57a2-485e-b1db-4ac83da5a67d",
            "Name": "By yourself and not using aids, do you have any
difficulty walking up 10 steps without resting? ",
            "Type": 6,
            "SortOrder": 1,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "5f5090a2-a45c-4591-ae92-aafc8a1d2854",
                    "Name": "1) Yes ",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "9fd1974f-fe95-4518-ad25-809a8d57ef0a",
                    "Name": "2) No ",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                }
            ]
        },
        {
            "Id": "2416c1a7-e58b-4e66-8814-f544c7dbbf97",
            "Name": "By yourself and not using aids, do you have any
difficulty walking several hundred yards? ",
            "Type": 6,
            "SortOrder": 2,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "eba99bb6-afb1-4aba-8440-32bc1635028b",
                    "Name": "1) Yes ",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "ca5d65ba-175b-40cb-864f-ec19b4a9ec99",
                    "Name": "2) No ",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                }
            ]
        },
        {
            "Id": "fd82a226-dc80-48f5-9adf-1580dffe912c",
            "Name": "Did a doctor ever tell you that you have: ",
            "Type": 11,
            "SortOrder": 3,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "e0d8598b-8c0e-4401-9dca-4e37fe05681f",
                    "Name": "hypertension",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "955c02bd-86b4-415e-b9a2-7ffd6884748f",
                    "Name": "diabetes",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "d565558d-2e65-490b-860a-0e17fef1e829",
                    "Name": "cancer (other than a minor skin cancer) ",
                    "SortOrder": 2,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "33e68603-06b3-46f0-9103-ccbac2e6a109",
                    "Name": "chronic lung disease ",
                    "SortOrder": 3,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "c9b8ca7a-1ef2-4b79-a2c8-d55d3de1fb9c",
                    "Name": "heart attack",
                    "SortOrder": 4,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "7242f6d9-c015-4b58-98ef-357bb9a1926b",
                    "Name": "congestive heart failure ",
                    "SortOrder": 5,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "0628814c-aea0-41a8-b285-81db468ed8e4",
                    "Name": "angina",
                    "SortOrder": 6,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "6791c992-4e07-4384-90fc-24cef470b878",
                    "Name": "asthma ",
                    "SortOrder": 7,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "51ce283d-5c34-4c3d-accb-8bd6e409b043",
                    "Name": "arthritis ",
                    "SortOrder": 8,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "abcfbc16-4de8-4d51-8ecc-412ed44a1e24",
                    "Name": "stroke ",
                    "SortOrder": 9,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "d6a4cfa4-cb56-403e-9530-9898d3308b06",
                    "Name": "kidney disease",
                    "SortOrder": 10,
                    "FollowUpQuestionGroupId": null
                }
            ]
        },
        {
            "Id": "69141a80-6a18-4ae2-b799-4d12e66a97dd",
            "Name": "Have you lost more than 5% of your weight in the past 6 
months",
            "Type": 5,
            "SortOrder": 4,
            "FollowUpQuestionGroupId": null,
            "Answers": [
                {
                    "Id": "8f0f6b68-5620-4d71-9253-6c4010100b1f",
                    "Name": "Yes",
                    "SortOrder": 0,
                    "FollowUpQuestionGroupId": null
                },
                {
                    "Id": "f4035c01-62e4-4865-86a5-597a3cf50cef",
                    "Name": "No",
                    "SortOrder": 1,
                    "FollowUpQuestionGroupId": null
                }
            ]
        }
    ],
    "Status": 0,
    "Message": ""
}
```

The answer contains a list of objects of type **Question** *which also contain a list of objects of type **Answer***.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

400 – BAD REQUEST

404 – NOT FOUND

500 – SERVER ERROR

Body message:

- See Section 6.5.1.2 for more details

6.5.2.5 Get the available answer options for a given question

EndPoint

POST – /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/questionnaire/answersforquestion.json

Authorization

Parameter	**Position**	**Description**
Authorization String	Header	The Bearer access token for SMS

Description

Get the available answer options for a question.

Parameters

Parameter	**Position**	**Description**
application String	Query	The application from which the request is performed. In case of CONNECARE it will be 2c9480845bee03e7015bfc0266d00000.
user-uuid String	Query	The uuid of the user for which the answer options are being consulted.

Body

```
{
  "Id": "e8c3be45-7374-4565-a3df-48baf69c41c2"
}
```

Responses

Success

HTTP response:

- 200 – OK (the operation was successfully done)

Body message:

```
{
    "Answers": List<Answer>,
    "Status": 0,
    "Message": ""
}
```

The answer body contains a list of objects of type **Answer**.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

400 – BAD REQUEST

404 – NOT FOUND

500 – SERVER ERROR

Body message:

- See Section 6.5.1.2 for more details

6.5.2.6 Get the Available Answer Options for a Question with the Help of a Search Text

EndPoint

POST – /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/questionnaire/answersforquestionwithsearch.json

Authorization

Parameter	**Position**	**Description**
Authorization String	Header	The Bearer access token for SMS

Description

Get the available answer options for a question with the help of a search text (any string is accepted). The answers of a given question will be filtered by the [SearchText] value.

- Id: the id of the question within the questionnaire
- SearchText: any string

Parameters

Parameter	**Position**	**Description**
application String	Query	The application from which the request is performed. In case of CONNECARE it will be 2c9480845bee03e7015bfc0266d00000.
user-uuid String	Query	The uuid of the user for which the answer options are being consulted.

Body

```
{
  "Id": "e8c3be45-7374-4565-a3df-48baf69c41c2",
  "SearchText": "All"
}
```

Responses

Success

HTTP response:

- 200 – OK (the operation was successfully done)

Body message:

```
{
    "Answers": [
        {
            "Id": "e33ebab9-f5e9-4a63-b668-b1060157288b",
            "Name": "1) All of the time",
            "SortOrder": 0,
            "FollowUpQuestionGroupId": null
        }
    ],
    "Status": 0,
    "Message": ""
}
```

The answer body contains a list of objects of type **Answer**.

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

400 – BAD REQUEST

404 – NOT FOUND

500 – SERVER ERROR

Body message:

- See Section 6.5.1.2 for more details

6.5.2.7 Saving Answers for a Given Question within a Question Group

EndPoint

POST – /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/questionnaire/saveanswerforquestion.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Description

Save answers for a given questions within a questionnaire. If multiple answers are selected, they must be saved at once and not multiple calls.

Parameters

Parameter	Position	Description
application String	Query	The application from which the request is performed. In case of CONNECARE it will be 2c9480845bee03e7015bfc0266d00000.
user-uuid String	Query	The uuid of the user for which the answers are being saved.

Body

- BaseQuestionGroupId: question id
- Id: answer id
- AnswerText: optional in case the answer requires any text
- LeadingAnswerId: first answer id (optional in case of multiple answers)

```
{
  "BaseQuestionGroupId": "e8c3be45-7374-4565-a3df-48baf69c41c2",
  "Answers": [
    {
      "Id": "e33ebab9-f5e9-4a63-b668-b1060157288b",
      "AnswerText": "1) All of the time",
      "LeadingAnswerId": "e33ebab9-f5e9-4a63-b668-b1060157288b"
    }
  ]
}
```

The request body contains an object of type **UserAnswerGroup** which contains a list of objects of type **UserAnswer**.

Responses

Success

HTTP response:

- 200 – OK (the operation was successfully done)

Body message:

```
{
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

400 – BAD REQUEST

404 – NOT FOUND

500 – SERVER ERROR

Body message:

- See Section 6.5.1.2 for more details

6.5.2.8 Complete the Question Group

EndPoint

POST – /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/questionnaire/completebasequestiongroup.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Description

- The question group (questionnaire) should be set completed after all questions are answered.
- After calling this endpoint the parameter [IsOpen] of the question group with the given [BaseQuestionGroupId] is set to false (means questionnaire is completed).
- The state of the questionnaires can be consulted by endpoint from section 4.1.
- To change the [IsOpen] parameter value of the questionnaire the endpoint from section 4.2 should be used (the [IsOpen] parameter of the given questionnaire will be set to true again).

Parameters

Parameter	Position	Description
application String	Query	The application from which the request is performed. In case of CONNECARE it will be 2c9480845bee03e7015bfc0266d00000.
user-uuid String	Query	The uuid of the user for which the question group is being completed.

Body

```
{
  "BaseQuestionGroupId": "8c540b13-4258-4cbe-8798-79ae4601b59d"
}
```

Responses

Success

HTTP response:

- 200 – OK (the operation was successfully done)

Body message:

```
{
    "AdviceGroupId": "645c28de-63f7-4503-bfaa-b8ca942dbfa8",
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom error code must be provided in the appropriate error message along with the corresponding messages. The Http codes to use are as follows:

400 – BAD REQUEST

404 – NOT FOUND

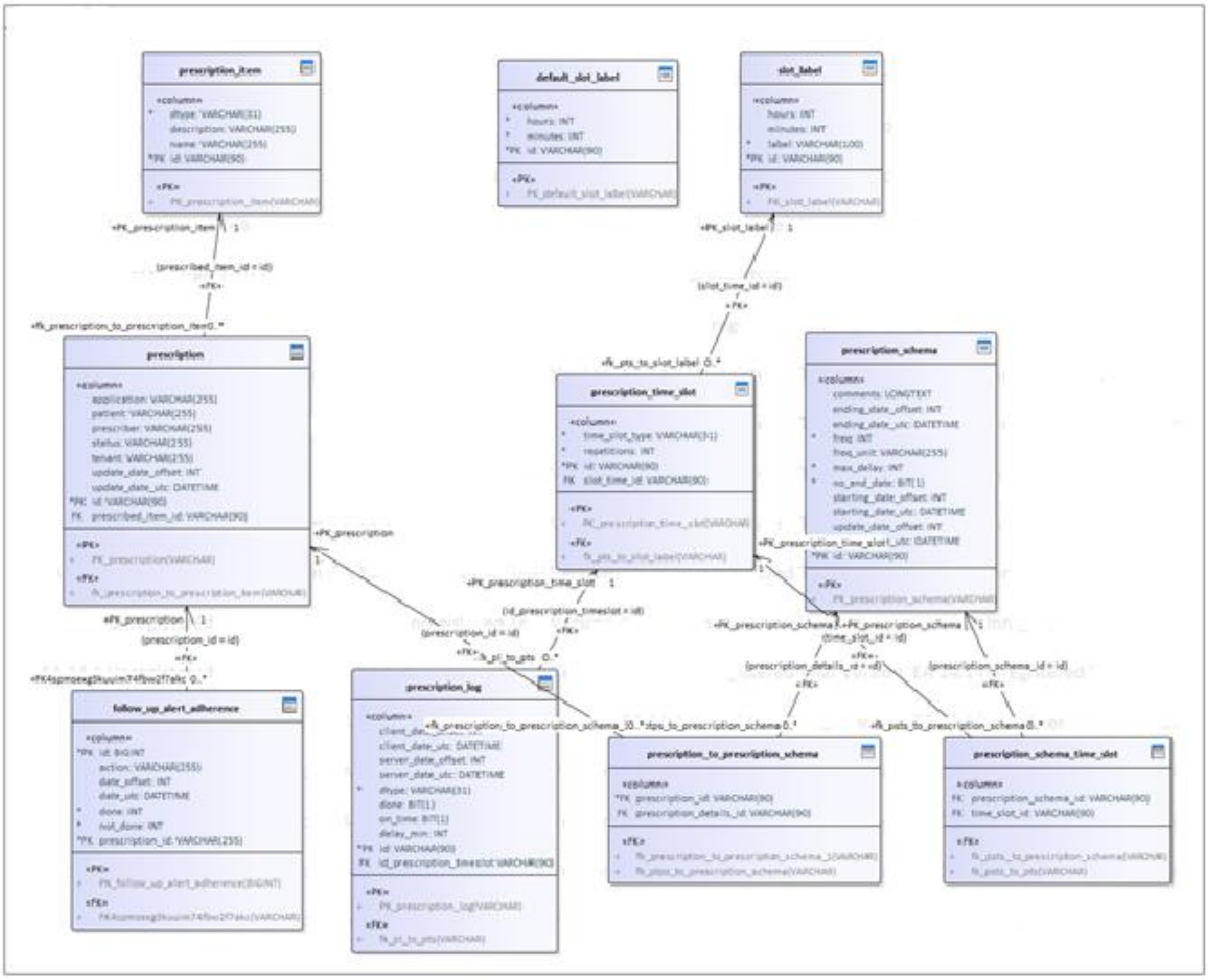
500 – SERVER ERROR

Body message:

- See Section 6.5.1.2 for more details

6.6 Simple Tasks

6.6.1 Data Model


prescription_item
«column»
* dtype: VARCHAR(31)
description: VARCHAR(255)
name: VARCHAR(255)
*PK id: VARCHAR(90)
«PK»
+ PK_prescription_item(VARCHAR)
default_slot_label
«column»
* hours: INT
* minutes: INT
*PK id: VARCHAR(90)
«PK»
+ PK_default_slot_label(VARCHAR)
slot_label
«column»
hours: INT
minutes: INT
* label: VARCHAR(100)
*PK id: VARCHAR(90)
«PK»
+ PK_slot_label(VARCHAR)
+PK_prescription_item 1
(prescribed_item_id = id)
«FK»
+fk_prescription_to_prescription_item0..*
+PK_slot_label 1
(slot_time_id = id)
«FK»
+fk_pts_to_slot_label 0..*
prescription
«column»
application: VARCHAR(255)
patient: VARCHAR(255)
prescriber: VARCHAR(255)
status: VARCHAR(255)
tenant: VARCHAR(255)
update_date_offset: INT
update_date_utc: DATETIME
*PK id: VARCHAR(90)
FK prescribed_item_id: VARCHAR(90)
«PK»
+ PK_prescription(VARCHAR)
«FK»
+ fk_prescription_to_prescription_item(VARCHAR)
+PK_prescription 1
prescription_time_slot
«column»
* time_slot_type: VARCHAR(31)
* repetitions: INT
*PK id: VARCHAR(90)
FK slot_time_id: VARCHAR(90)
«PK»
+ PK_prescription_time_slot(VARCHAR)
«FK»
+ fk_pts_to_slot_label(VARCHAR)
prescription_schema
«column»
comments: LONGTEXT
ending_date_offset: INT
ending_date_utc: DATETIME
* freq: INT
freq_unit: VARCHAR(255)
* max_delay: INT
* no_end_date: BIT(1)
starting_date_offset: INT
starting_date_utc: DATETIME
update_date_offset: INT
update_date_utc: DATETIME
*PK id: VARCHAR(90)
«PK»
+ PK_prescription_schema(VARCHAR)
+PK_prescription_time_slot 1
+PK_prescription_time_slot 1
(id_prescription_timeslot = id)
«FK»
(prescription_id = id)
«FK»
+fk_pl_to_pts 0..*
+PK_prescription 1
(prescription_id = id)
«FK»
+FK4spmoeeg9kuuim74fbw2f7afkc 0..*
+PK_prescription_schema 1
+PK_prescription_schema 1
(time_slot_id = id)
«FK»
(prescription_details_id = id)
«FK»
(prescription_schema_id = id)
«FK»
+fk_prescription_to_prescription_schema_10..*
+fpts_to_prescription_schema 0..*
+fk_psts_to_prescription_schema 0..*
follow_up_alert_adherence
«column»
*PK id: BIGINT
action: VARCHAR(255)
date_offset: INT
date_utc: DATETIME
* done: INT
* not_done: INT
*FK prescription_id: VARCHAR(255)
«PK»
+ PK_follow_up_alert_adherence(BIGINT)
«FK»
+ FK4spmoeeg9kuuim74fbw2f7afkc(VARCHAR)
prescription_log
«column»
client_date_offset: INT
client_date_utc: DATETIME
server_date_offset: INT
server_date_utc: DATETIME
* dtype: VARCHAR(31)
done: BIT(1)
on_time: BIT(1)
delay_min: INT
*PK id: VARCHAR(90)
FK id_prescription_timeslot: VARCHAR(90)
«PK»
+ PK_prescription_log(VARCHAR)
«FK»
+ fk_pl_to_pts(VARCHAR)
prescription_to_prescription_schema
«column»
*FK prescription_id: VARCHAR(90)
FK prescription_details_id: VARCHAR(90)
«FK»
+ fk_prescription_to_prescription_schema_1(VARCHAR)
+ fk_ptps_to_prescription_schema(VARCHAR)
prescription_schema_time_slot
«column»
FK prescription_schema_id: VARCHAR(90)
FK time_slot_id: VARCHAR(90)
«FK»
+ fk_psts_to_prescription_schema(VARCHAR)
+ fk_psts_to_pts(VARCHAR)

6.6.2 API Definition

POST /v1/prescription/user/{user}/save

Prescribes the monitorization of the given user.

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Request

Atribute	Optional	Description
PrescriptionDto	application (string, *optional*), id (string, *optional*), patient (string, *optional*), prescribedItem (MonitorizationDto, *optional*), prescriber (string, *optional*), prescriptionDetails(Array[PrescriptionSchemaDto], *optional*), status (string, *optional*), tenant (string, *optional*), treatmentDocumentation(DocumentDto, *optional*), updateDate (LocaleDate, *optional*)	
MonitorizationDto	measurementType(Array[MeasurementTypeDto], optional)	
PrescriptionSchemaDto	comments (string, *optional*), endDate (LocaleDate, *optional*), freq (integer, *optional*), freqUnit (string, *optional*), id (string, *optional*), maxDelay (integer, *optional*), noEndDate (boolean, *optional*), startDate (LocaleDate, *optional*), timeSlot (Array[PrescriptionTimeSlotDto], *optional*),	

	timeStamp (LocaleDate, *optional*)
DocumentDto	name (string, optional)
MeasurementTypeDto	alertDefinition (MeasurementAlertDto, *optional*), name (string, *optional*)
PrescriptionTimeSlotDto	id (string, *optional*), repetitions (integer, *optional*), slotTime (SlotLabelDto, *optional*)
MeasurementAlertDto	alertFrequency (integer, *optional*), alertFrequencyUnit (string, *optional*), freq (integer, *optional*), freqUnit (string, *optional*), max (number, *optional*), min (number, *optional*)
SlotLabelDto	hour (string, *optional*), id (string, *optional*), label (string, *optional*), minute (string, *optional*)

Response (Status 200)

```
{
  "application": "string",
  "id": "string",
  "patient": "string",
  "prescribedItem": {
    "measurementType": [
      {
        "alertDefinition": {
          "alertFrequency": 0,
          "alertFrequencyUnit": "string",
          "freq": 0,
          "freqUnit": "string",
          "max": 0,
          "min": 0
        },
        "name": "string"
      }
    ]
  },
  "prescriber": "string",
  "prescriptionDetails": [
    {
      "comments": "string",
      "endDate": "2018-02-17T08:00:00.000+0000",
      "freq": 0,
      "freqUnit": "string",
      "id": "string",
      "maxDelay": 0,
      "noEndDate": true,
      "startDate": "2018-01-17T08:00:00.000+0000",
      "timeSlot": [
        {
          "id": "string",
          "repetitions": 0,
          "slotTime": {
            "hour": "string",
            "id": "string",
            "label": "string",
            "minute": "string"
          }
        }
      ],
    }
  ],
  "status": "string",
  "tenant": "string",
  "treatmentDocumentation": {
    "name": "string"
  },
  "updateDate": "2018-02-17T08:00:00.000+0000"
}
```

GET /v1/generictask/user/{targetUserUuid}

Get the list of simple tasks prescriptions for the user

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response

```
[
  {
    "application": "string",
    "id": "string",
    "patient": "string",
    "prescribedItem": {
      "measurementType": [
        {
          "alertDefinition": {
            "alertFrequency": 0,
            "alertFrequencyUnit": "string",
            "freq": 0,
            "freqUnit": "string",
            "max": 0,
            "min": 0
          },
          "name": "string"
        }
      ]
    },
    "prescriber": "string",
    "prescriptionDetails": [
      {
        "comments": "string",
        "endDate": ""
        "freq": 0,
        "freqUnit": "string",
        "id": "string",
        "maxDelay": 0,
        "noEndDate": true,
        "startDate": ""
        "timeSlot": [
          {
            "id": "string",
            "repetitions": 0,
            "slotTime": {
              "hour": "string",
              "id": "string",
              "label": "string",
              "minute": "string"
            }
          }
        ],

      }
    ],
    "status": "string",
    "tenant": "string",
    "treatmentDocumentation": {
      "name": "string"
    },
    "updateDate": ""
  }
]
```

GET /v1/generictask/{uuid}

Retrieve a prescription by its id

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response

```
{
  "application": "string",
  "id": "string",
  "patient": "string",
  "prescribedItem": {
    "measurementType": [
      {
        "alertDefinition": {
          "alertFrequency": 0,
          "alertFrequencyUnit": "string",
          "freq": 0,
          "freqUnit": "string",
          "max": 0,
          "min": 0
        },
        "name": "string"
      }
    ]
  },
  "prescriber": "string",
  "prescriptionDetails": [
    {
      "comments": "string",
      "endDate": "2018-01-17T08:00:00.000+0000"
      "freq": 0,
      "freqUnit": "string",
      "id": "string",
      "maxDelay": 0,
      "noEndDate": true,
      "startDate": "2018-01-17T08:00:00.000+0000"
      "timeSlot": [
        {
          "id": "string",
          "repetitions": 0,
          "slotTime": {
            "hour": "string",
            "id": "string",
            "label": "string",
            "minute": "string"
          }
        }
      ]
    }
  ],
  "status": "string",
  "tenant": "string",
  "treatmentDocumentation": {
    "name": "string"
  },
  "updateDate": ""
}
```

6.6.2.1 Error

This class contains the information for a request which doesn't generate a specific content. For instance, correct PUT requests generate this kind of answers or any other request if they generate an error.

Attribute	Optional	Description
errorCode Int	**N**	Internal code of the error / success.
userMessage String	**N**	User friendly message.
internalMessage integer	**N**	Internal error message.

JSON representation:

```
{
  "errorCode": int,
  "userMessage": "String",
  "internalMessage": "String"
}
```

Example:

```
{
  "errorCode": 4003,
  "userMessage": "No prescription found",
  "internalMessage": "No prescription found"
}
```

6.7 Messaging

6.7.1 Data Model

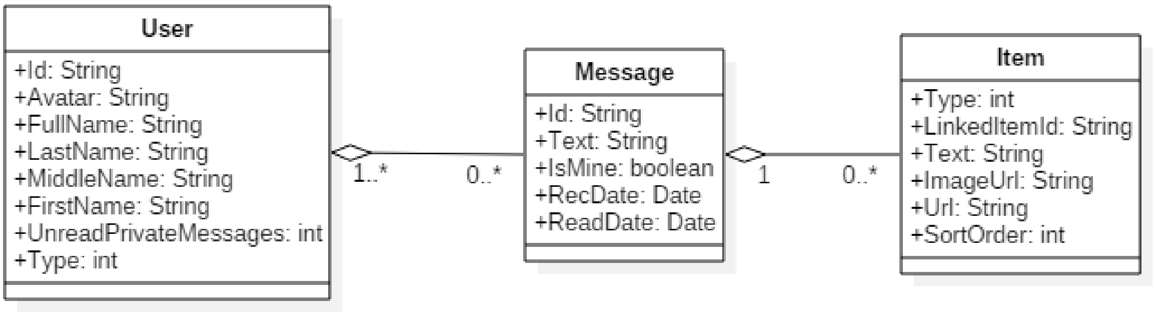


Figure 41 - Data model diagram.

6.7.1.1 User

This class models the user data within the VitalinQ platform in order to be used from the Messaging and Community services.

Attribute	Optional	Description
Id **String**	**N**	Id of the user.
Avatar **String**	**Y**	URL to avatar image.
FullName **String**	**Y**	Full name of the user.
LastName **String**	**Y**	Last name of the user.
MiddleName **String**	**N**	Middle name of the user.
FirstName **String**	**N**	First name of the user.
UnreadPrivateMessages **Integer**	**N**	Number of unread private messages.
Type **Integer**	**N**	Type of user (1: Friends, 2: Acquaintances, 3: Advisors).

JSON representation:

```
{
  "Id": "string",
  "Avatar": "string",
  "FullName": "string",
  "LastName": "string",
  "MiddleName": "string",
  "FirstName": "string",
  "UnreadPrivateMessages": int,
  "Type": int
}
```

Example:

```
{
  "Id": "d9Y7856Z9Yf029X39a99656970IRd16067gF",
  "Avatar": "https://example.com/Public/ImageProfile?imageRef=2264",
  "FullName": "Russel Matthew",
  "LastName": "Matthew ",
  "MiddleName": "",
  "FirstName": "Russel",
  "UnreadPrivateMessages": 2,
  "Type": 1
}
```

6.7.1.2 Message

This class models the private messages that are sent and consulted by users.

Attribute	Optional	Description
Id **String**	**Y**	The id of the message.
Text **String**	**N**	Text of the message.
IsMine **Boolean**	**N**	Indicator whether the message is from the user consulting it or not.
RecDate **Date**	**N**	The date the message was sent. The format is “yyyy-MM-dd'T'HH:mm:ss.SSS”
ReadDate **Date**	**N**	The date the message was read. The format is “yyyy-MM-dd'T'HH:mm:ss.SSS”. Null in case if the message is unread.
Items **Array**	**N**	Array of Items which represents the attached elements as an image or video from YouTube (see Section 3.4).

JSON representation:

```
{
  "Id": "string",
  "Text": "string",
  "IsMine": boolean,
  "RecDate": "date",
  "ReadDate": "date",
  "Items": [
    {
      "Type": int,
      "LinkedItemId": "string",
      "Text": "string",
      "ImageUrl": "string",
      "Url": "string",
      "SortOrder": int
    }
  ]
}
```

Example:

```
{
  "Id": "28dd6b15-f7b5-43mc-g12d-77a67b18a9e9",
  "Text": "Hello!",
  "IsMine": false,
  "RecDate": "2017-09-10T09:57:15.183",
  "ReadDate": "2017-09-10T10:21:22.113",
  "Items": []
}
```

6.7.1.3 Item

This element represents the attached content within the message such as an image or video from YouTube.

Atribute	Optional	Description
Type **Int**	Y	Type of attached element (1: Advice, 2: Youtube video, 3: Pol, 4: Appointment, 5: Recipe, 6: Plan, 7: Activity program, 8: Badge, 9: Image).
LinkedItemId **String**	N	The id of the linked item if exists.
Text **String**	N	Message text.
ImageUrl **String**	N	URL of the image.
Url **String**	N	URL of the element (e.g. Youtube video).
SortOrder **Int**	N	Classification order.

JSON representation:

```
{
  "Type": int,
  "LinkedItemId": "string",
  "Text": "string",
  "ImageUrl": "string",
  "Url": "string",
  "SortOrder": int
}
```

Example:

```
{
  "Type": 2,
  "LinkedItemId": null,
  "Text": "Red Hot Chili Peppers - Californication [Official Music Video]",
  "ImageUrl": "https://i.ytimg.com/vi/YlUKcNNmywk/default.jpg",
  "Url": "https://www.youtube.com/embed/YlUKcNNmywk",
  "SortOrder": 0
}
```

6.7.2 API Definition

6.7.2.1 Send private message

This endpoint allows to send private messages to another contact. A message can contain both the text as an image or a Youtube video. Before starting a conversation with a user it is necessary to add him/her to the contact list and he/she must accept the invitation (see Sections 5.4, 5.1, 5.5 and 5.3). It is only possible to send messages to the users that appear in the contact list of the logged in user (see Section 5.4).

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/message/saveprivatemessage.json

Authorization

Parameter	Position	Description
Authorization **String**	Header	The Bearer access token for SMS.

Parameters

Parameter	Position	Description
application String	Path	The uuid of the application.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for a custom extract of the **Message** object (see Section 3.3). The identifier of the user to whom the message is sent must be defined. The Items field is optional.

Body example:

```
{
  "Id": "ooo3485S9150B9X518971a49026Ue-f3eh-2",
  "Text": "Hello!"
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Response body:

- The endpoint returns another extract of the **Message** object (see Section 3.3) adding the information about the request (Status and Message fields). For example:

```
{
  "Id": "8753r9b9-5g66-35h9-585f-vv1245g6589w",
  "RecDate": "2017-09-15T10:09:14.307",
  "Status": 0,
  "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok
- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error
- 5 - UnefficientData
- 6 - Warning
- 7 - NoRightsBySettings

6.7.2.2 Get private messages

This endpoint allows to retrieve the messages from the start of the conversation with another contact until the specified date.

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/message/privatemessages.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
application String	Path	The uuid of the application.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the contact id with whom the conversation has been kept and the date that determines until when the messages are consulted. For example:

```
{
  "ContactId": "d9Y7856Z9Yf029X39a99656980YRd06067gU",
  "BeforeRecDate": "2017-09-14T16:00:00+02:00"
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Response body:

- The endpoint returns an array of the **Message** objects (see Section 3.3) adding the information about the request (IsOlderAvailable, Status and Message fields). For example:

```
{
    "Messages": [
        {
            "Id": "8753r9b9-5g66-35h9-585f-vv1245g6589w",
            "Text": "",
            "IsMine": false,
            "RecDate": "2017-09-12T13:27:47.783",
            "ReadDate": "2017-09-13T10:21:22.613",
            "Items": [
                {
                    "Type": 9,
                    "LinkedItemId": null,
                    "Text": null,
                    "ImageUrl":
"https://example.com/Uploaded/Messages/568.jpg",
                    "Url": "https://example.com/Uploaded/Messages/568.jpg",
                    "SortOrder": 0
                }
            ]
        },
        {
            "Id": "8753r9b9-5g66-35h9-585f-vv1245g6589w",
            "Text": "I send you a picture",
            "IsMine": false,
            "RecDate": "2017-09-12T13:26:59.61",
            "ReadDate": "2017-09-13T10:21:23.113",
            "Items": []
        },
        {
            "Id": "8753r9b9-5g66-35h9-585f-vv1245g6589w ",
            "Text": "Hi Patient",
            "IsMine": false,
            "RecDate": "2017-09-12T13:26:49.807",
            "ReadDate": "2017-09-13T10:21:21.117",
            "Items": []
        },
        {
            "Id": "f269f390-4a17-55f3-b64d-642b449f8e09",
            "Text": "I have some questions?",
            "IsMine": true,
            "RecDate": "2017-09-12T13:24:23.553",
            "ReadDate": "2017-09-12T13:26:46.417",
            "Items": []
        },
        {
            "Id": "f269f390-4a17-55f3-b64d-642b449f8e09",
            "Text": "Hi Doctor!",
            "IsMine": true,
            "RecDate": "2017-09-12T13:24:17.977",
            "ReadDate": "2017-09-12T13:26:45.91",
            "Items": []
        }
    ],
    "IsOlderAvailable": false,
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok
- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error

- 5 - UnefficientData
- 6 - Warning
- 7 - NoRightsBySettings

6.7.2.3 Set message as read

This endpoint allows to set a received message as read. After marking a message as read the "ReadDate" field of the **Message** object will specify the date when it has been marked as read. To check the identifier of the message to be marked as read it is possible to use the endpoint described in Section 4.2

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/message/savemessageasread.json

Authorization

Parameter	Position	Description
Authorization **String**	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
application String	Path	The uuid of the application.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the identifier of the message. For example:

```
{
  "Id": "3089e9b8-4b99-48a8-969b-aa4280c1584d"
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the information about the request (Status and Message fields). For example:

```
{
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok
- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error
- 5 - UnefficientData
- 6 - Warning
- 7 - NoRightsBySettings

6.7.2.4 Search users

This endpoint allows to search users registered on the VitalinQ platform by different types of field such as email, gender, birthdate, first name, middle name or last name.

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/community/searchuser.json

Authorization

Parameter	Position	Description
Authorization **String**	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
application String	Path	The uuid of the application.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for one or a set of the following fields:

```
{
  "Email": "email@example.com",
  "Gender": true,
  "BirthDate": "2017-09-15T10:32:18.898352+02:00",
  "FirstName": "sample string 2",
  "MiddleName": "sample string 3",
  "LastName": "sample string 4"
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Response body:

- The endpoint returns an array (could be empty) of **User** objects (see Section 3.2) adding the information about the request (Status and Message fields). For example:

```
{
    "Users": [
        {
          "Id": "d9Y7856Z9Yf029X39a99656970IRd16067gF",
          "Avatar": "https://example.com/Public/ImageProfile?imageRef=2264",
          "FullName": "Russel Matthew",
          "LastName": "Matthew ",
          "MiddleName": "",
          "FirstName": "Russel"
        }
    ],
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok
- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error
- 5 - UnefficientData
- 6 – Warning
- 7 - NoRightsBySettings

6.7.2.5 Get available contact types

This endpoint retrieves the available types of contact on the VitalinQ platform which can be assigned to a newly added contact (used in the endpoint described in Section 5.5). At the moment there are three types of contact as Friends, Acquaintances and Advisors.

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/community/usertypes.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
application String	Path	The uuid of the application.
user-uuid String	Path	The uuid of the user on SMS.

Body

Not applicable.

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array with the available contact types adding the information about the request (Status and Message fields). For example:

```
{
    "Types": [
        {
            "Name": "Advisors",
            "Description": "",
            "Type": 3
        },
        {
            "Name": "Friends",
            "Description": "",
            "Type": 1
        },
        {
            "Name": "Acquaintances",
            "Description": "",
            "Type": 2
        }
    ],
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok
- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error
- 5 - UnefficientData
- 6 – Warning
- 7 - NoRightsBySettings

6.7.2.6 Get all open invitations

This endpoint shows the list of invitations that are received from other contacts in order to start a conversation. To accept the invitation it is necessary to add the user that appears in this list to the contact list through the endpoint described in Section 5.5 using their identifier.

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/community/openinvitations.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
application String	Path	The uuid of the application.

user-uuid String	Path	The uuid of the user on SMS.

Body

Not applicable.

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array (could be empty) of **User** objects (see Section 3.2) adding the information about the request (Status and Message fields). For example:

```
{
    "Users": [
        {
          "Id": "d9Y7856Z9Yf029X39a99656970IRd16067gF",
          "Avatar": "https://example.com/Public/ImageProfile?imageRef=2264",
          "FullName": "Russel Matthew",
          "LastName": "Matthew ",
          "MiddleName": "",
          "FirstName": "Russel",
          "Type": 1
        }
    ],
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok
- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error
- 5 - UnefficientData
- 6 – Warning
- 7 - NoRightsBySettings

6.7.2.7 Get my contacts

This endpoint shows the list of contacts of the logged user. It is possible to start a conversation with any of the users that appear in this list using their identifier (see Section 4.1).

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/community/contacts.json

Authorization

Parameter	Position	Description
Authorization **String**	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
application String	Path	The uuid of the application.
user-uuid String	Path	The uuid of the user on SMS.

Body

Not applicable.

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an array (could be empty) of **User** objects (see Section 3.2) adding the information about the request (Status and Message fields). For example:

```
{
    "Contacts": [
        {
          "Id": "d9Y7856Z9Yf029X39a99656970IRd16067gF",
          "Avatar": "https://example.com/Public/ImageProfile?imageRef=2264",
          "FullName": "Russel Matthew",
          "LastName": "Matthew ",
          "MiddleName": "",
          "FirstName": "Russel",
          "UnreadPrivateMessages": 0,
          "Type": 1
        }
    ],
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok
- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error
- 5 - UnefficientData
- 6 – Warning
- 7 - NoRightsBySettings

6.7.2.8 Add an user to my contacts

Through this endpoint it is possible to add a new contact to the list of the logged user. To search users and consult their identifiers it is possible to use the endpoint described in Section 5.1. Until the invited user does not accept the invitation, he/she will not appear in the invitee’s contact list and it is not possible to send messages to him/her.

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/community/saveusertocontacts.json

Authorization

Parameter	**Position**	**Description**
Authorization **String**	Header	The Bearer access token for SMS

Parameters

Parameter	**Position**	**Description**
application String	Path	The uuid of the application.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the identifier of the contact to be added and the type to be assigned. For example:

```
{
  "Id": "d9Y7856Z9Yf029X39a99656970IRd16067gF",
  "Type": 0
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the information about the request (Status and Message fields). For example:

```
{
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok
- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error
- 5 - UnefficientData
- 6 – Warning
- 7 - NoRightsBySettings

6.7.2.9 Remove contact

Through this endpoint it is possible to delete a contact from the contact list of the logged user. To consult the list of contacts and their identifiers it is possible to use the endpoint described in Section 5.4

EndPoint

POST - /xcare-service-vitalinq-connector/v1/proxypass/vitalinq/application/{application}/user/{user-uuid}/api/user/community/removecontact.json

Authorization

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Parameters

Parameter	Position	Description
application String	Path	The uuid of the application.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the identifier of the contact to be removed. For example:

```
{
  "Id": " d9Y7856Z9Yf029X39a99656970IRd16067gF"
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the information about the request (Status and Message fields). For example:

```
{
    "Status": 0,
    "Message": ""
}
```

Error

In case of error, a custom status code will be notified within the Status field. The Status codes to use are as follows:

- 0 - Ok

- 1 - NoRightsBySubscriptionType
- 2 - NoRightsByRelation
- 3 - NoRightsBySystem
- 4 - Error
- 5 - UnefficientData
- 6 – Warning
- 7 - NoRightsBySettings

6.8 Advices

6.8.1 Data Model

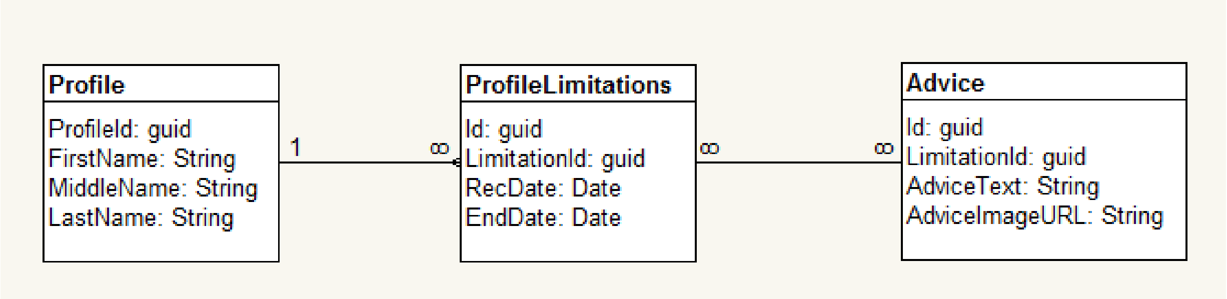


Figure 42 - Data model of the Advices service.

6.8.2 API Definition

6.8.2.1 Get available limitations

POST api/user/profile/limitationsoptions.* (json or xml)

Will retrieve all options and with indication if currently active

Parameters

Parameter	Position	Description
Group String	Path	The uuid limitation group.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the group to retrieve the limitations for:

```
{
  "GroupId": "9040199a-d775-4a1b-bc6d-93bed42a5a6e"
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns the information about the request (Status and Message fields). For example:

```
{
  "LimitationOptions": [
    {
      "Id": "9040199a-d775-4a1b-bc6d-93bed42a5a6e",
      "Name": "Limitation name 1",
      "Linked": true
    },
    {
      "Id": "9040199a-d775-4a1b-bc6d-93bed42a5a6e",
      "Name": "Limitation name 2",
      "Linked": true
    }
  ],
  "Status": 0,
  "Message": "sample string 1"
}
```

6.8.2.2 Set limitation

POST api/user/profile/savelimitations.* (json or xml)

Save multiple limitations to the Profile

Parameters

Limitations List	Path	The uuids of the limitations.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the limitations to set for the profile:

```
{
  "Limitations": [
    "da402d8b-9a7f-4150-8871-6cfa70397ccd",
    "29536f5a-417c-4093-9aca-3501180621ae"
  ],
  "ReplaceOld": true
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

6.8.2.3 Remove limitations

POST api/user/profile/deletelimitations.* (json or xml)

Remove multiple limitations from the Profile

Parameters

Parameter	Position	Description
Limitations List	Path	The uuids of the limitations.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the limitations to set for the profile:

```
{
  "Limitations": [
    "da402d8b-9a7f-4150-8871-6cfa70397ccd",
    "29536f5a-417c-4093-9aca-3501180621ae"
  ]
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

6.8.2.4 Get advice list

POST api/user/advice.* (json or xml)

Get advice items for given or default groups

Parameters

Parameter	Position	Description
Amount Int	Path	Number of items to retrieve
Limitations List	Path	The uuids of the groups.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the data:

```
{
  "Amount": 1,
  "Groups": [
    "81603a68-86e4-43cf-8872-1b9b25d19ff2",
    "79bc50dd-e30e-4d5f-96c2-564ac76ba2fc"
  ]
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Response information

Name	Description	Type
RecieveDate	the date/time the data had been collected	date
Amount	the amount of items that are used for selecting	integer

AdviceItems	a list of advices you have requested	Collection of AdviceModel+AdviceItem
Status		BaseModel+ResponseStatus
Message		string

Example:

```
{
  "RecieveDate": "2017-09-26T07:25:08.0861437Z",
  "Amount": 1,
  "AdviceItems": [
    {
      "Id": "e51ada09-0807-44dd-9916-1be344798db2",
      "Title": "Weight is good",
      "Description": "The measured values you entered indicate that your
current weight is healthy. A healthy diet and sufficient exercise reduce the
chance of obesity.",
      "Image": "https://mijn.domain.nl/images/Advice/a9a1daa4-d3c9-4642-
9990-4173e1206e7f.jpg",
      "IsFavourite": false,
      "ReadDate": null,
      "EvaluationScore": 0
    }
  ],
  "Status": 0,
  "Message": ""
}
```

6.8.2.5 *Get favourites*

POST api/user/advice/favourites.* (json or xml)

Get the favourited items for a deault or given set of groups

Parameters

Groups List	Path	The uuids of the groups.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the limitations to set for the profile:

```
{
  "Groups": [
    "9eb1a1b3-4767-4d8e-96bf-af0493f0b853",
    "f4902860-5588-4bb3-95e5-b880d470af20"
  ]
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Result:

```
{
  "AdviceItems": [
    {
      "Id": "4a455264-e120-4f5c-b6c7-ae8d4de900ad",
      "Title": "sample string 2",
      "Description": "sample string 3",
      "Image": "sample string 4",
      "IsFavourite": true,
      "ReadDate": "2017-09-26T14:19:46.6337549+02:00",
      "EvaluationScore": 6
    },
    {
      "Id": "4a455264-e120-4f5c-b6c7-ae8d4de900ad",
      "Title": "sample string 2",
      "Description": "sample string 3",
      "Image": "sample string 4",
      "IsFavourite": true,
      "ReadDate": "2017-09-26T14:19:46.6337549+02:00",
      "EvaluationScore": 6
    }
  ],
  "Status": 0,
  "Message": "sample string 1"
}
```

6.8.2.6 Get advice item

POST api/user/advice/adviceitem.* (json or xml)

Get all details for single advice item

Parameters

Parameter	Position	Description
AdviceId guid	Path	The uuid of the advice.
user-uuid String	Path	The uuid of the user on SMS.

Body

The endpoint waits for the data:

```
{
  "Id": "da402d8b-9a7f-4150-8871-6cfa70397ccd"
}
```

Responses

Success

Response code:

- 200 – OK (the operation was successfully done)

Result:

```
{
  "Id": "aa2d19e7-bf03-4017-b506-5d8d6095d4f2",
  "Title": "sample string 2",
  "Description": "sample string 3",
  "Image": "sample string 4",
  "IsFavourite": true,
  "ReadDate": "2017-09-26T14:20:16.7120583+02:00",
  "EvaluationScore": 6,
  "Status": 0,
  "Message": "sample string 7"
}
```

6.8.2.7 Save advice item

POST api/user/advice/saveitem.* (json or xml)

Set some details for single advice item

Parameters

Name	Description	Type

Id	the id of the advice item you like to save	globally unique identifier
IsFavourite		boolean
ReadDate		date
EvaluationScore		integer

Body

The endpoint waits for the data:

```
{
  "Id": "6b55c914-072b-427f-8415-cf3e8f1f2c60",
  "IsFavourite": true,
  "ReadDate": "2017-09-26T14:20:20.7932697+02:00",
  "EvaluationScore": 3
}
```

Responses

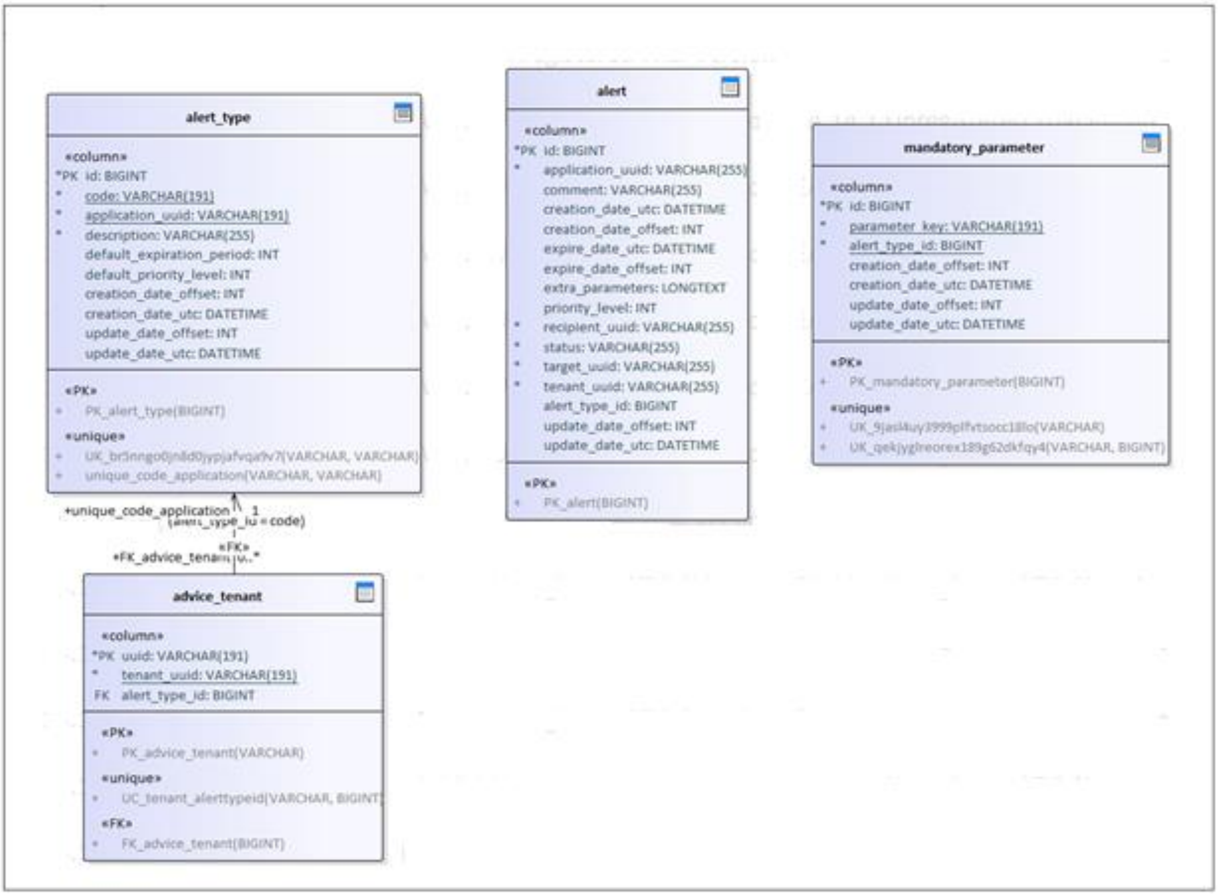
Success

Response code:

- 200 – OK (the operation was successfully done)

6.9 Alerts

6.9.1 Data Model


alert_type
«column»
*PK id: BIGINT
* code: VARCHAR(191)
* application_uuid: VARCHAR(191)
* description: VARCHAR(255)
default_expiration_period: INT
default_priority_level: INT
creation_date_offset: INT
creation_date_utc: DATETIME
update_date_offset: INT
update_date_utc: DATETIME
«PK»
+ PK_alert_type(BIGINT)
«unique»
+ UK_br3nngo0jn8d0jypjafvqa9v7(VARCHAR, VARCHAR)
+ unique_code_application(VARCHAR, VARCHAR)
+unique_code_application
1
(alert_type_id = code)
«FK»
+FK_advice_tenant
0..*
alert
«column»
*PK id: BIGINT
* application_uuid: VARCHAR(255)
comment: VARCHAR(255)
creation_date_utc: DATETIME
creation_date_offset: INT
expire_date_utc: DATETIME
expire_date_offset: INT
extra_parameters: LONGTEXT
priority_level: INT
* recipient_uuid: VARCHAR(255)
* status: VARCHAR(255)
* target_uuid: VARCHAR(255)
* tenant_uuid: VARCHAR(255)
alert_type_id: BIGINT
update_date_offset: INT
update_date_utc: DATETIME
«PK»
+ PK_alert(BIGINT)
mandatory_parameter
«column»
*PK id: BIGINT
* parameter_key: VARCHAR(191)
* alert_type_id: BIGINT
creation_date_offset: INT
creation_date_utc: DATETIME
update_date_offset: INT
update_date_utc: DATETIME
«PK»
+ PK_mandatory_parameter(BIGINT)
«unique»
+ UK_9jasl4uy3999plfvtsocc18lo(VARCHAR)
+ UK_qekjyglreorex189g62dkfqy4(VARCHAR, BIGINT)
advice_tenant
«column»
*PK uuid: VARCHAR(191)
* tenant_uuid: VARCHAR(191)
FK alert_type_id: BIGINT
«PK»
+ PK_advice_tenant(VARCHAR)
«unique»
+ UC_tenant_alerttypeid(VARCHAR, BIGINT)
«FK»
+ FK_advice_tenant(BIGINT)

6.9.2 API Definition

POST /v1/alert/save

Insert Alert

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Request

Atribute	Optional	Description
AlertListDto	alerts (Array[AlertDto], *optional*)	
AlertDto	alertType (string): Alert Type Code comments (string, *optional*), expirationPeriod (integer, *optional*): Expiration time: [hours]. Override Alert Type value extraParameters (inline_model, *optional*): Extra Parameters: Map of key-value's priorityLevel (integer, *optional*): Valid Range: [0,5]. Override Alert Type value recipient (string), target (string)	

Response (Status 200)

```
{
  "alerts": [
    {
      "alertType": "string",
      "comments": "string",
      "creationDate": "string",
      "expireDate": "string",
      "extraParameters": {},
      "id": 0,
      "priorityLevel": 0,
      "recipient": "string",
      "status": "string",
      "target": "string"
    }
  ]
}
```

GET /v1/alertId/{alertId}/alert/retrieve

Get Alert By Id.

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response (Status 200)

```
{
  "alerts": [
    {
      "alertType": "string",
      "comments": "string",
      "creationDate": "string",
      "expireDate": "string",
      "extraParameters": {},
      "id": 0,
      "priorityLevel": 0,
      "recipient": "string",
      "status": "string",
      "target": "string"
    }
  ]
}
```

GET /v1/target/{target}/alert/list

Get Alert By Target

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response (Status 200)

```
{
  "alerts": [
    {
      "alertType": "string",
      "comments": "string",
      "creationDate": "string",
      "expireDate": "string",
      "extraParameters": {},
      "id": 0,
      "priorityLevel": 0,
      "recipient": "string",
      "status": "string",
      "target": "string"
    }
  ]
}
```

6.9.2.1 Error

This class contains the information for a request which doesn't generate a specific content. For instance, correct PUT requests generate this kind of answers or any other request if they generate an error.

Attribute	Optional	Description
errorCode Int	**N**	Internal code of the error / success.
userMessage String	**N**	User friendly message.
message integer	**N**	Internal error message.

Error codes

- 404 Not found
- 500 Internal Server Error
- 503 Service Unavailable
- 400 Bad Request, 401 Unauthorized

JSON representation:

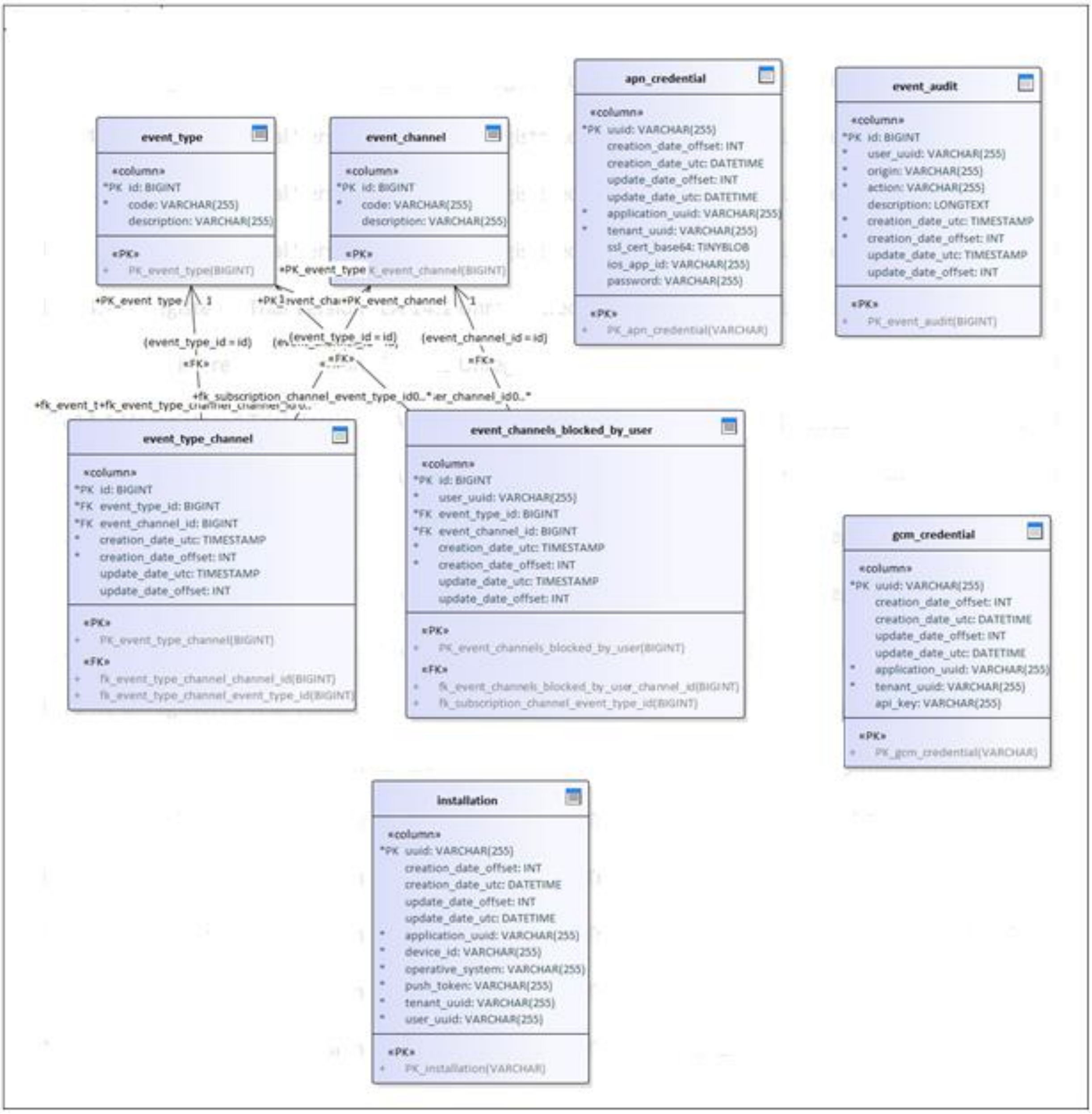
```
{
  “errors”:[
    {
      "errorCode": int,
      "userMessage": "String",
      " message ": "String"
    }
  ]
}
```

Example:

```
{
  "errors":[
    {
      "errorCode":int,
      "userMessage":"string",
      "message":"string"
    }
  ]
}
```

6.10 Notifications

6.10.1 Data Model


apn_credential
«column»
*PK uuid: VARCHAR(255)
creation_date_offset: INT
creation_date_utc: DATETIME
update_date_offset: INT
update_date_utc: DATETIME
* application_uuid: VARCHAR(255)
* tenant_uuid: VARCHAR(255)
ssl_cert_base64: TINYBLOB
ios_app_id: VARCHAR(255)
password: VARCHAR(255)
«PK»
+ PK_apn_credential(VARCHAR)
event_audit
«column»
*PK id: BIGINT
* user_uuid: VARCHAR(255)
* origin: VARCHAR(255)
* action: VARCHAR(255)
description: LONGTEXT
* creation_date_utc: TIMESTAMP
* creation_date_offset: INT
update_date_utc: TIMESTAMP
update_date_offset: INT
«PK»
+ PK_event_audit(BIGINT)
event_type
«column»
*PK id: BIGINT
* code: VARCHAR(255)
description: VARCHAR(255)
«PK»
+ PK_event_type(BIGINT)
event_channel
«column»
*PK id: BIGINT
* code: VARCHAR(255)
description: VARCHAR(255)
«PK»
+ PK_event_channel(BIGINT)
+PK_event_type
+PK_event_type 1
+PK_event_cha
+PK_event_channel 1
1
(event_type_id = id)
(event_type_id = id)
(event_channel_id = id)
«FK»
«FK»
«FK»
+fk_subscription_channel_event_type_id0..*
er_channel_id0..*
+fk_event_t+fk_event_type_channel_channel_id0..
event_type_channel
«column»
*PK id: BIGINT
*FK event_type_id: BIGINT
*FK event_channel_id: BIGINT
* creation_date_utc: TIMESTAMP
* creation_date_offset: INT
update_date_utc: TIMESTAMP
update_date_offset: INT
«PK»
+ PK_event_type_channel(BIGINT)
«FK»
+ fk_event_type_channel_channel_id(BIGINT)
+ fk_event_type_channel_event_type_id(BIGINT)
event_channels_blocked_by_user
«column»
*PK id: BIGINT
* user_uuid: VARCHAR(255)
*FK event_type_id: BIGINT
*FK event_channel_id: BIGINT
* creation_date_utc: TIMESTAMP
* creation_date_offset: INT
update_date_utc: TIMESTAMP
update_date_offset: INT
«PK»
+ PK_event_channels_blocked_by_user(BIGINT)
«FK»
+ fk_event_channels_blocked_by_user_channel_id(BIGINT)
+ fk_subscription_channel_event_type_id(BIGINT)
gcm_credential
«column»
*PK uuid: VARCHAR(255)
creation_date_offset: INT
creation_date_utc: DATETIME
update_date_offset: INT
update_date_utc: DATETIME
* application_uuid: VARCHAR(255)
* tenant_uuid: VARCHAR(255)
api_key: VARCHAR(255)
«PK»
+ PK_gcm_credential(VARCHAR)
installation
«column»
*PK uuid: VARCHAR(255)
creation_date_offset: INT
creation_date_utc: DATETIME
update_date_offset: INT
update_date_utc: DATETIME
* application_uuid: VARCHAR(255)
* device_id: VARCHAR(255)
* operative_system: VARCHAR(255)
* push_token: VARCHAR(255)
* tenant_uuid: VARCHAR(255)
* user_uuid: VARCHAR(255)
«PK»
+ PK_installation(VARCHAR)

6.10.2 API Definition

POST /v1/registration/save

Registration service. Register application push credentials

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Request

Atribute	Description
RegistrationDto	apikey (string, optional): Required if Operative System is Android , iOSAppId (string, optional): Required if Operative System is iOS , operativeSystem (string) = ['android', 'ios', 'all'], password (string, optional), sslcertBase64 (string, optional): Required if Operative System is iOS

Response (Status 200)

```
{
  "apikey": "string",
  "iOSAppId": "string",
  "operativeSystem": "android",
  "password": "string",
  "sslcertBase64": "string"
}
```

POST /v1/installation/save

Save user installation (push token)

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Request

Atribute	Description
Installation Dto	deviceId (string, optional),

operativeSystem (string, optional),

pushToken (string, optional)

Response (Status 200)

```
{
  "deviceId": "string",
  "operativeSystem": "string",
  "pushToken": "string"
}
```

6.10.2.1Error

This class contains the information for a request which doesn't generate a specific content. For instance, correct PUT requests generate this kind of answers or any other request if they generate an error.

Attribute	Optional	Description
errorCode Int	**N**	Internal code of the error / success.
userMessage String	**N**	User friendly message.
message integer	**N**	Internal error message.

Error codes

- 404 Not found
- 500 Internal Server Error
- 503 Service Unavailable
- 400 Bad Request, 401 Unauthorized

JSON representation:

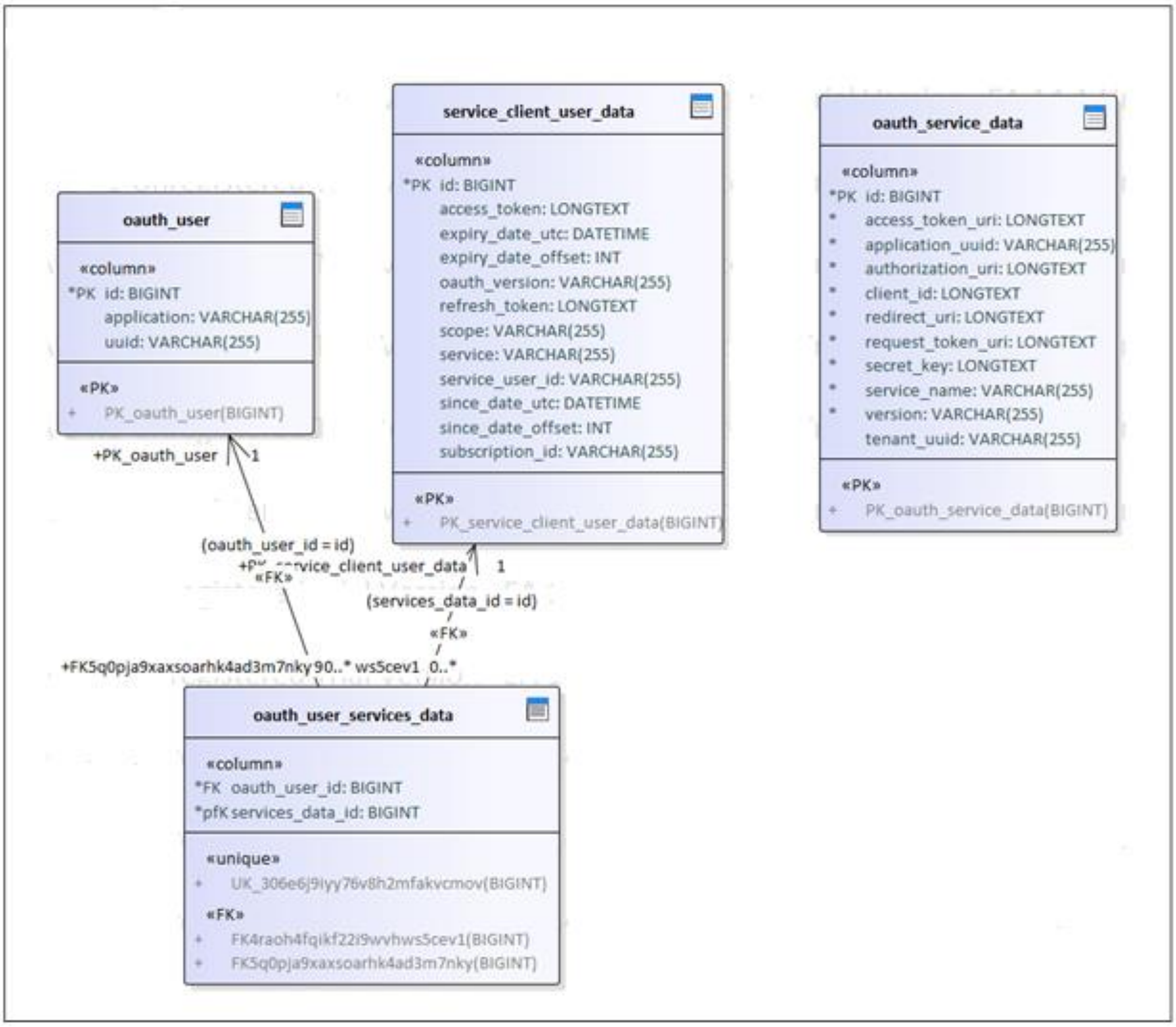
```
{
  "errors":[
    {
      "errorCode": int,
      "userMessage": "String",
      " message ": "String"
    }
  ]
}
```

Example:

```
{
  "errors":[
    {
      "errorCode":int,
      "userMessage":"string",
      "message":"string"
    }
  ]
}
```

6.11 Third Party

6.11.1 Data Model


service_client_user_data
«column»
*PK id: BIGINT
access_token: LONGTEXT
expiry_date_utc: DATETIME
expiry_date_offset: INT
oauth_version: VARCHAR(255)
refresh_token: LONGTEXT
scope: VARCHAR(255)
service: VARCHAR(255)
service_user_id: VARCHAR(255)
since_date_utc: DATETIME
since_date_offset: INT
subscription_id: VARCHAR(255)
«PK»
+ PK_service_client_user_data(BIGINT)
oauth_service_data
«column»
*PK id: BIGINT
* access_token_uri: LONGTEXT
* application_uuid: VARCHAR(255)
* authorization_uri: LONGTEXT
* client_id: LONGTEXT
* redirect_uri: LONGTEXT
* request_token_uri: LONGTEXT
* secret_key: LONGTEXT
* service_name: VARCHAR(255)
* version: VARCHAR(255)
tenant_uuid: VARCHAR(255)
«PK»
+ PK_oauth_service_data(BIGINT)
oauth_user
«column»
*PK id: BIGINT
application: VARCHAR(255)
uuid: VARCHAR(255)
«PK»
+ PK_oauth_user(BIGINT)
+PK_oauth_user
1
(oauth_user_id = id)
+PK_service_client_user_data
1
«FK»
(services_data_id = id)
«FK»
+FK5q0pja9xaxsoarhk4ad3m7nky 0..*
ws5cev1 0..*
oauth_user_services_data
«column»
*FK oauth_user_id: BIGINT
*pfK services_data_id: BIGINT
«unique»
+ UK_306e6j9iyy76v8h2mfakvcmov(BIGINT)
«FK»
+ FK4raoh4fqikf22i9wvhws5cev1(BIGINT)
+ FK5q0pja9xaxsoarhk4ad3m7nky(BIGINT)

6.11.2 API Definition

POST /v1/oauth/withings/measures/application/{application-uuid}/tenant/{tenant-uuid}/user/{user-uuid}/callback

Callback endpoint for all measures.

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Request

Atribute	Description
application-uuid	Appication identifier
tenant-uuid	Tenant identifier
user-uuid}	User identifier

GET /v1/oauth/withings/measures/measurement/user/{user-uuid}/retrieve

Get the body measures for a user

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response (Status 200)

```
[
  {
    "device": {
      "deviceType": "User",
      "id": 0,
      "manufacturer": {
        "id": 0,
        "name": "string"
      },
      "serialID": "string"
    },
  "id": 0,
  "inputType": "ManualInput",
  "measurementDate": "string",
  "measurements": [
   {
    "numericValue": 0,
    "textValue": "string",
    "type": "Weight",
    "unit": "kg"
   }
  ]
 }
]
```

GET /v1/oauth/withings/measures/measurement/user/{user-uuid}/start/{start}/end/{end}/retrieve

Get the body measures for a user between two dates

Headers

Parameter	Position	Description
Authorization String	Header	The Bearer access token for SMS

Response (Status 200)

```
[
 {
  "device": {
   "deviceType": "User",
   "id": 0,
   "manufacturer": {
    "id": 0,
    "name": "string"
   },
   "serialID": "string"
  },
  "id": 0,
  "inputType": "ManualInput",
  "measurementDate": "string",
  "measurements": [
   {
    "numericValue": 0,
    "textValue": "string",
    "type": "Weight",
    "unit": "kg"
   }
  ]
 }
]
```

GET /v1/oauth/withings/measures/measurement/{measurement}/user/{user-uuid}/retrieve

Get the measures by type for a user

Headers

Parameter	**Position**	**Description**
Authorization String	Header	The Bearer access token for SMS

Response (Status 200)

```
[
  {
    "device": {
      "deviceType": "User",
      "id": 0,
      "manufacturer": {
        "id": 0,
        "name": "string"
      },
      "serialID": "string"
    },
    "id": 0,
    "inputType": "ManualInput",
    "measurementDate": "string",
    "measurements": [
      {
        "numericValue": 0,
        "textValue": "string",
        "type": "Weight",
        "unit": "kg"
      }
    ]
  }
]
```

6.11.2.1Error

This class contains the information for a request which doesn’t generate a specific content. For instance, correct PUT requests generate this kind of answers or any other request if they generate an error.

Attribute	**Optional**	**Description**
errorCode Int	**N**	Internal code of the error / success.
userMessage String	**N**	User friendly message.
message	**N**	Internal error message.

integer

Error codes

- 404 Not found
- 500 Internal Server Error
- 503 Service Unavailable
- 400 Bad Request, 401 Unauthorized

JSON representation:

```
{
  “errors”:[
    {
      "errorCode": int,
      "userMessage": "String",
      " message ": "String"
    }
  ]
}
```

Example:

```
{
  "errors":[
    {
      "errorCode":int,
      "userMessage":"string",
      "message":"string"
    }
  ]
}
```