CONNECARE

WP4 – SELF-MANAGEMENT AND MONITORING

D4.2: BASIC MONITORING TOOLS

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

Project No. 689802

Start date of project: 01-04-2016

Duration: 42 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: 30/09/2017

Document Information

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

Deliverable	**Number**	4.2	**Title**	Basic Monitoring Tools
Work Package	**Number**	4	**Title**	Self-management and monitoring

Date of delivery	**Contractual**	MONTH 18	**Actual**	MONTH 18
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 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.

Table of contents

Executive Summary

In order to provide basic monitoring, in task 4.2 “Basic monitoring tools” EURECAT defined the services for gathering, collecting, fusing, and processing data coming from the patient through a wristband as well as manually inserted by mean of questionnaires and scales. The core services that have been defined and developed are: physical activity and sleeping (by EURECAT) and questionnaires (by IPHEALTH).

The physical activity service is aimed at monitoring walking activity through a wristband. The number of steps per day and the level of activity (low, medium, and high) as well as the allowed sedentary activity are prescribed by clinicians through the SACM. The adherence is calculated accordingly and alerts are sent in case the patient is not performing the prescribed activity or is too far to reach the daily goal. Motivational messages and recommendations will be sent by the Recommender System. An important issue that arose in defining and developing this service concerns the wristbands to be used in the clinical studies scheduled to start at M20 (November 2017). First of all, specific requirements on the number of devices are needed. As part of the 2nd PDSA cycle this issue will be finalized before its end (M18, September 2017). From a technological perspective, Fitbit charge HR and wristbands from Withings/Nokia have been tested and integrated in the SMS.

The sleeping service is aimed at monitoring sleeping and resting activity through the activity tracker (the same used to monitor physical activity). Through it, the SMS automatically recognizes when the patient is sleeping or is awake. Recommendations and suggestions to better rest are automatically sent to the patient, if needed.

The questionnaires service is aimed at allowing patients to fill self-checked questionnaires. The definition of the questionnaires was part of the co-design approach (WP2) and they are documented in the deliverable D2.4 “*Case studies description and the associated co-design process*”. Through the SACM clinicians prescribe one or more questionnaires together with the frequency to be filled (e.g., once, every week, every Wednesday, every month). Through the SMS the patient (or her/his caregiver) answers the questionnaire and receives a feedback according to the obtained result. Once filled, questionnaires are sent back to the clinicians that may access them through the SACM to verify the trend and the behaviour. Recommendation in form of questionnaires is also under investigation as part of the Recommender System.

Apart those core services, auxiliary services have been developed –or are under development– by IPHEALTH: messaging, agenda, and advices. The former is aimed at giving the possibility to the patient to communicate to the clinical staff sending text and/or images. The agenda service is devoted to send medical appointments to patients and put them in their calendar. The latter is aimed at giving advices, tutorials and education materials to the patient (in form of document or link to external resources).

Overall, the work summarized in this document is based on the work made by EURECAT and IPHEALTH in WP4 (T4.2). According to the CONNECARE fundamentals, that work has been done in collaboration with all the partners, especially clinical ones and decisions came from the co-design approach followed in WP2. Moreover, the work presented in this deliverable is strictly related with the overall work made in WP4 thus including the back-end and front-end of the SMS as well as the overall list of requirements. Therefore, the following deliverables are highly recommended to be read:

Number	**Title**	**Description**
D2.4	Case studies description and the associated co-design	The document provides a complete view of case study definitions as a product of the co-design process completed so far. It provides full details on the 1st PDSA cycle from the clinical perspective, summarizing the objectives and results of all held meetings and activities, as well as all the feedback provided to technical partners. Moreover, the current document includes detailed site-specific case studies definitions and associated workflows. Finally, full details on functional and non-functional requirements of the CONNECARE Smart Adaptive Case Management (SACM) platform And Self-Management System (SMS) are provided.
D4.1	First self-management system	This document describes the first version of the self-management systems (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 system and the requirements requested by the patients and professionals.

1. The Model

Activities to be monitored have been selected during the co-design approach according to requirements gathered during the 1st PDSA cycle in each site and for each Case Study (CS). The full list of requirements has been presented in D2.4 “Case studies description and the associated co-design” and then summarized in D4.1 “First self-management system”. Requirements have been selected according to their priority, i.e., according the commonalities in the sites, the more required the first implemented. Thus, as for basic monitoring the following services have been selected and implemented (as highlighted in Figure 1):

- Physical activity;
- Questionnaires;
- Sleeping;
- Messaging;
- Advices.

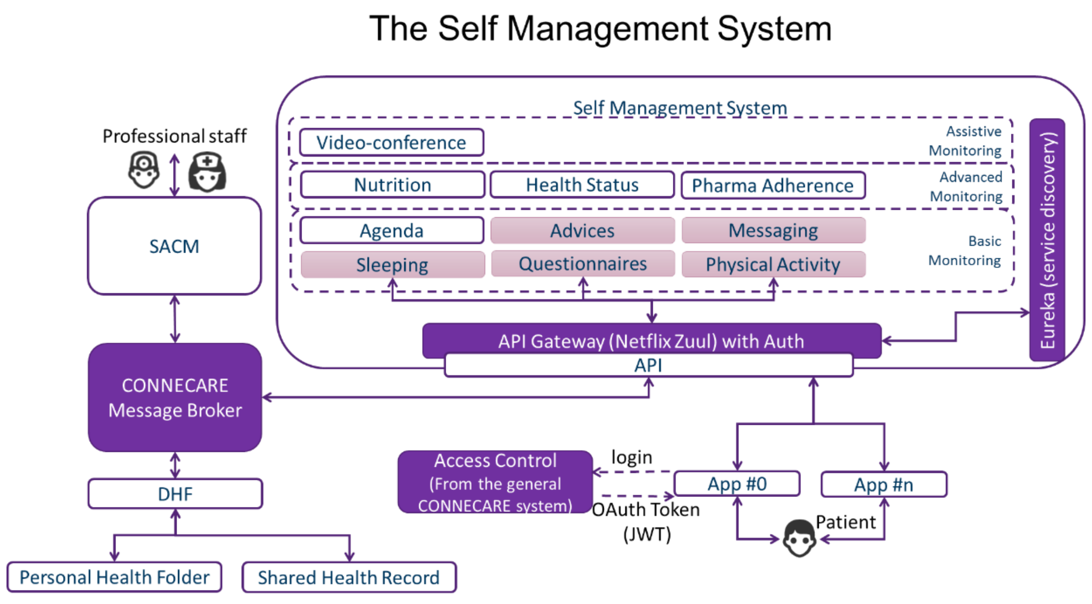


Figure 1 - The micro-services in the Study Release of SMS. Highlighted in light pink, those for Basic Monitoring.

Let us note that also the service “Agenda” has been selected and it is currently under development. Nevertheless, in this deliverable, we decided to report only the services that are part of the first release of the SMS. In fact, “Agenda” will be integrated as soon as it is ready and it will be part of the second release of the system.

As stated in D4.1 “*First Self-Management System*”, the CONNECARE SMS will be both autonomous and collaborative: the patient may use the SMS to monitor and access to her/his data and information (*autonomous behavior*) and professionals interact with it through the SACM, to allow participation by clinicians and to provide feedback to them (*collavorative behaviour*). Thus, each microservice that has been designed and implemented to support basic monitoring relies on a closed-loop approach in which the professional, during the Workplan definition step, prescribes a task (e.g., to fill a questionnaire or to perform some phyisical activity) to the patient that receives it through an alert in the SMS app and acts accordingly. Data gathered by the interaction of the patient with the SMS (automatically through the devices or manually through direct input) are sent in 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 hands, 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. Figure 2 summarizes the overall process underlying all the implemented core microservices. It is worth noting that this approach is not used in case of messaging, agenda, and advices.

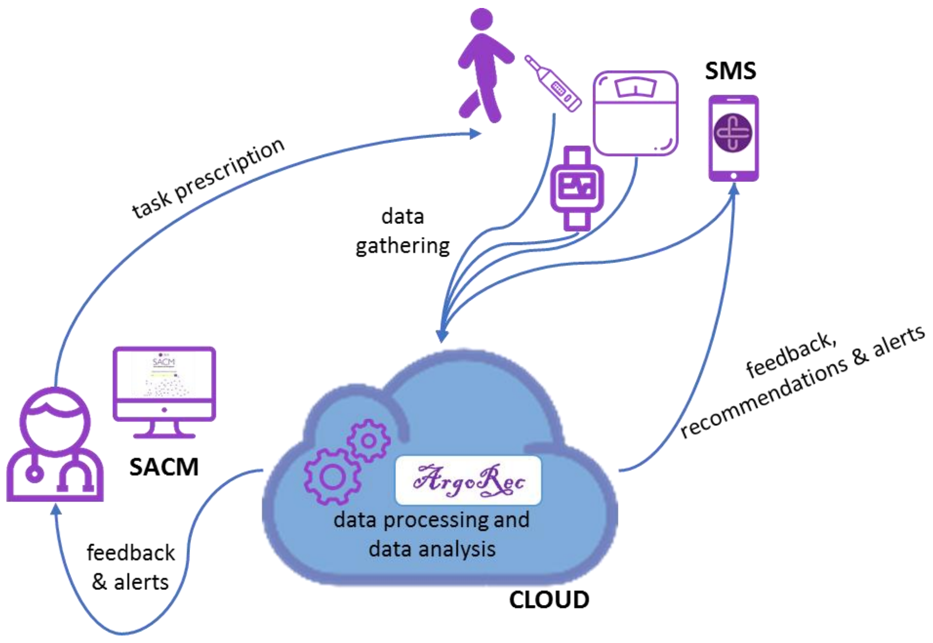

task prescription
data gathering
SMS
feedback, recommendations & alerts
SACM
ArgoRec
data processing and data analysis
CLOUD
feedback & alerts

Figure 2 - The closed-loop approach at a glance.

The Basic Monitoring tools revolve around the following main abstractions:

- *Prescription*: any kind of prescription made by a clinician to a given patient to monitor, e.g., perform physical activities and fill a questionnaire.
- *Adherence*: the adherence of the patient to the clinician's prescriptions, both regarding individual prescriptions (*adherence level*) and their history based on a given time window (*adherence profile*).
- *Fulfilment*: the fulfilment of a prescription achieved by the patient, necessary to measure the patient's adherence –either automatically (e.g., through the activity tracker) or manually (e.g., by the patient inputting some relevant data).

- *Feedback*: the message to inform the clinician regarding patient's adherence and fulfilment. According to the setting of a prescription, different kinds of feedback may be received and will be part of the summary available to the clinician.
- *Recommendation*: the message to dispatch to the patient for *engagement*, *reward*, or *warning*, depending on her/his adherence, or the one to be sent to the clinician for continuous follow-up (in this case, it is called *feedback*). According to the corresponding adherence, recommendations may have a punctuation from 1 ("very bad") to 5 ("very good"), thus messages sent accordingly: an alert for low punctuation (e.g., "You've to be more active. Go out and take a walk!") or a reward for a high one (e.g., "Wonderful! Walk 100 steps more and you'll reach the goal!").
- *Alerts*: the message to dispatch to patient and clinicians in case some fulfilment has not been reach. Clinicians may set suitable thresholds during the prescription creation asking to be informed in case they are passed (e.g., the patient sleeps less hours than the corresponding threshold).
- *Strategy*: the criteria guiding decision making about *how* to compute the adherence, and *which* recommendation/feedback/alert to send, *when*.
- *Monitoring engine*: the components responsible of gathering data from the patient and dispatching alerts, based on patients' adherence regarding the thresholds given at prescription time.
- *Recommendation engine*: the component responsible of generating and dispatching recommendations and feedbacks, based on patients' adherence regarding their fulfilment of prescriptions, and on a dynamically configurable strategy.

2. Physical Activity Monitoring

2.1 Requirements

As reported in D2.4 “Case studies description and the associated co-design”, physical activity monitoring has been required by all sites in every Case Study (CS) but CS1 in Barcelona. Table 1 summarises the features required to be monitored in each site for each CS.

Table 1. Features to be monitored.

Measurement	CS1					CS2				CS3
	BCN	LL	IL	NL1	NL2	BCN	LL	IL	NL	BCN
Steps	--	X	X	X	X	X	X	X	X	X
Distance (km)	--	--	X	X	X	--	--	X	X	--
Calories	--	--	X	X	X	--	--	X	X	--
Seconds of activity by intensity	--	--	X	X	X	--	--	X	X	--

Accordingly, the activity trackers described in Table 2 have been tested and the API from Fitbit[1] and from Withings[2] have been integrated.

Table 2 - Tested activity trackers.

Activity Tracker	Description	Features							
		Steps	Distance	Calories	Elevation	Activity intensity	HR	SPO2	Sleeping
Fitbit Charge HR	The Fitbit Charge HR is textured and has a screen that can display caller ID information from a connected smartphone through the Fitbit app. The Charge automatically tracks users' steps, sleep, flights of stairs (using an altimeter) and an approximation of distance travelled. It tracks steps using a 3 axis accelerometer by tracking forward movement along with upward	X	X	X	--	X	X	--	X

[1] https://dev.fitbit.com/docs/

[2] https://developer.health.nokia.com/api

	movements. The Charge HR moreover contains a heart-rate monitor.								
PulseOx	This activity tracker by Withings contains a pedometer, heart rate monitor and blood oxygen reader. It can connect to other Withings devices such as the Smart Body Analyzer smart scales and blood pressure monitor. The user's data is pulled from those devices and into the companion app. Information like the user's weight is then used to increase the accuracy of the calorie counter. When the device is not at hand, activity can still be tracked through the companion app itself. The device does reflexive measurement, so users do not need to clip their finger for the SPO2 measurement.	X	X	X	X	X	X	X	X
Go	Withings Go is an activity tracker that can be clipped or hung on belts, or worn on the wrist with a silicone strap. It uses a replaceable battery that lasts eight months, is waterproof for swimming, and has an E Ink screen for always-on activity progress status (or, at a touch, the analog time). Goal progress will be displayed on the tracker using a prominent circular countdown. The E Ink display also serves as a touch-sensitive button, letting users switch between activity goals and the watch function.	X	X	X	X	X	--	--	X
Activité Steel	Withings Activité by Withings is an activity tracking that has no buttons; instead, everything is controlled from the phone app. It can track the user's sleep, swimming, walking and running automatically. Sleep and activity are displayed on the app as graphs.	X	X	X	X	X	X	--	X

Prices are the following[3]:

- Fitbit Charge HR: 80 €
- PulseOx: 58.30 €
- Withings Go: 20 €
- Activité Steel: 80 €

[3] Prices are those proposed by Withings/Nokia with which we are in contact with and from the market for the Fitbit.

Apart cases in which also oxygen saturation measurement is needed (e.g., CS1 in Lleida) and, then, PulseOx is required[4], Withing Go seems to be the best option. Anyway, all the described devices are currently supported in the Study Release of the SMS through the Physical Activity service.

2.2 The Service

The Physical Activity monitoring service allows patients to connect their activity tracker to monitor steps, distance, calories, and activity intensity. Monitoring of physical activity may be directly linked to a prescription by a clinician. In that case, it is the clinician who decides how many steps the patient has to do every day and the corresponding minutes of each activity level (e.g., 10000 steps at day, 30 minutes of high activity, 45 minutes of moderate activity, 150 minutes of low activity, and a maximum of 750 minutes of inactivity). Given a prescription, the system automatically calculates the adherence day by day and sends recommendations to the patient and feedback to the clinician as well as alerts, if needed. Alternatively, in case no prescription is given, this service automatically calculates the average physical activity of the patient, in terms of step and level of activity, and uses that as baseline to then calculate the adherence. Also in this case, recommendations to the patient and feedback to the clinician, as well as alerts, are sent back.

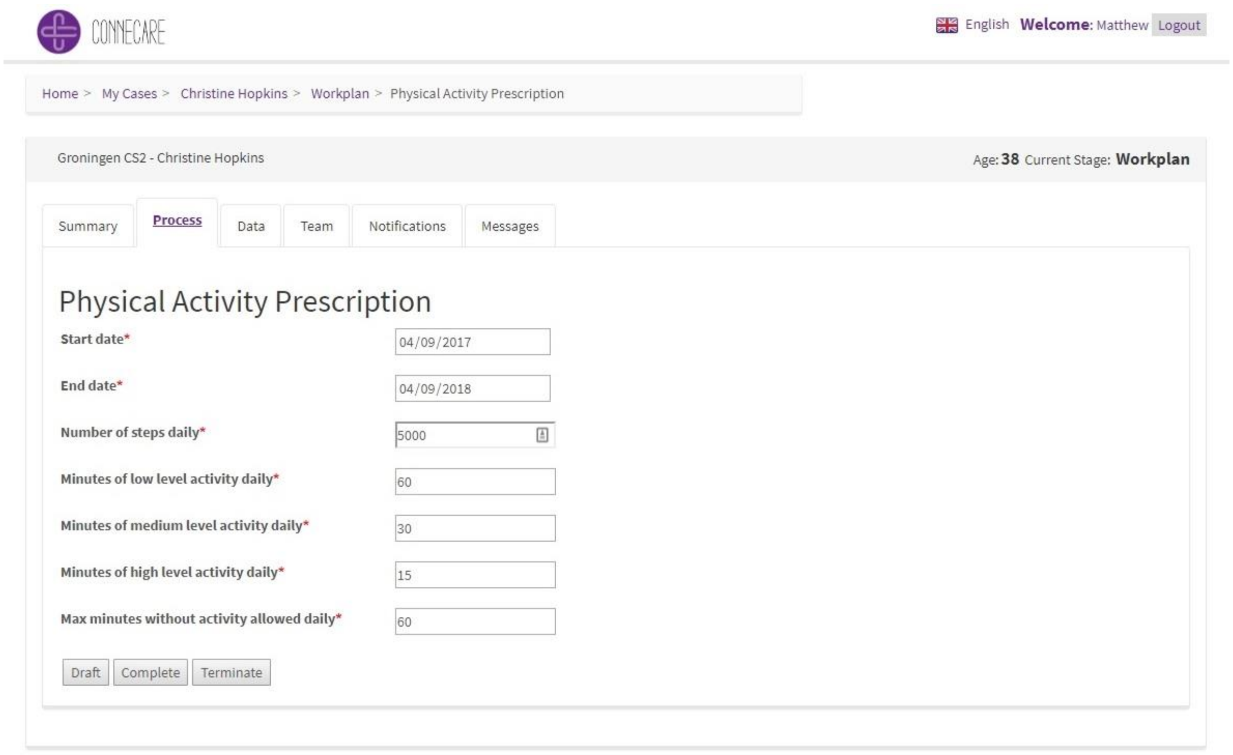
The key functionalities of this service are:

- Prescription of a physical activity plan.
- Generation of reports of patient's activity and the fulfilment of the prescription.
- Generation of reports regarding the activity levels.
- Dispatching of alerts generated by the system the live time of a prescription.

The user roles involved in the process are, of course, the patient and the clinician, depending on the CS and the study different kind of clinicians may interact (e.g., physiotherapist)[5].

[4] Oxygen saturation monitoring is part of the Health Status monitoring services that is part of the Study Release of the SMS. The corresponding services will be described in D4.3 "Advanced Monitoring Tools" at M36 (March 2019).

[5] The list of user groups for CS is given in D2.4 "Case Studies Description and the Associated Co-Design".



English **Welcome:** Matthew Logout

Home > My Cases > Christine Hopkins > Workplan > Physical Activity Prescription

Groningen CS2 - Christine Hopkins

Age: **38** Current Stage: **Workplan**

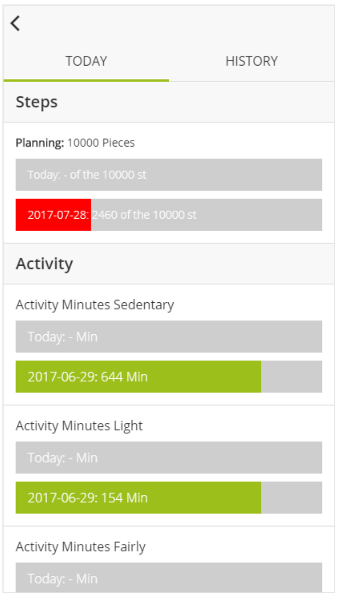
Physical Activity Prescription

Start date*	04/09/2017
End date*	04/09/2018
Number of steps daily*	5000
Minutes of low level activity daily*	60
Minutes of medium level activity daily*	30
Minutes of high level activity daily*	15
Max minutes without activity allowed daily*	60

Figure 3 - Physical activity prescription (SACM).

The data model and the API description of this service are given in the Annex 8.1, in the following of this Section we illustrate how the service works through some screenshots.

Through the SACM the clinician prescribes the minimum number of steps that the patient has to perform per day (Figure 3). The patient receives the prescription in her smartphone through the SMS and may access to it to see the details (Figure 4). In any moment, the patient may access to the SMS to check her activity such as steps and minutes of activity (Figure 5). Through the SACM the clinician may access to that information (Figure 7). The SMS every time the professional sends a new prescription shows an alert into the notification section, also this section allows the SMS app to remind the patient to execute the prescribed activity (Figure 6).



Steps

Planning: 10000 Pieces

Today: - of the 10000 st

2017-07-28: 2460 of the 10000 st

Activity

Activity Minutes Sedentary

Today: - Min

2017-06-29: 644 Min

Activity Minutes Light

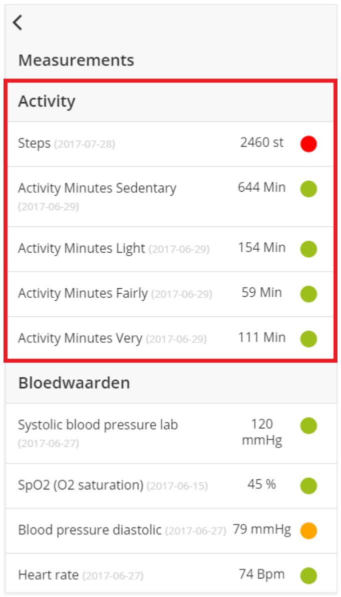
Today: - Min

2017-06-29: 154 Min

Activity Minutes Fairly

Today: - Min

Figure 4 - Physical activity prescription (SMS).


Measurements

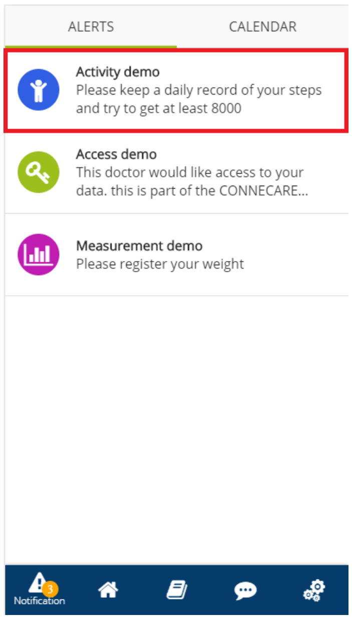
Activity

Steps (2017-07-28)	2460 st	🔴
Activity Minutes Sedentary (2017-06-29)	644 Min	🟢
Activity Minutes Light (2017-06-29)	154 Min	🟢
Activity Minutes Fairly (2017-06-29)	59 Min	🟢
Activity Minutes Very (2017-06-29)	111 Min	🟢

Bloedwaarden

Systolic blood pressure lab (2017-06-27)	120 mmHg	🟢
SpO2 (O2 saturation) (2017-06-15)	45 %	🟢
Blood pressure diastolic (2017-06-27)	79 mmHg	🟠
Heart rate (2017-06-27)	74 Bpm	🟢

Figure 5 - Physical activity data of today (SMS).

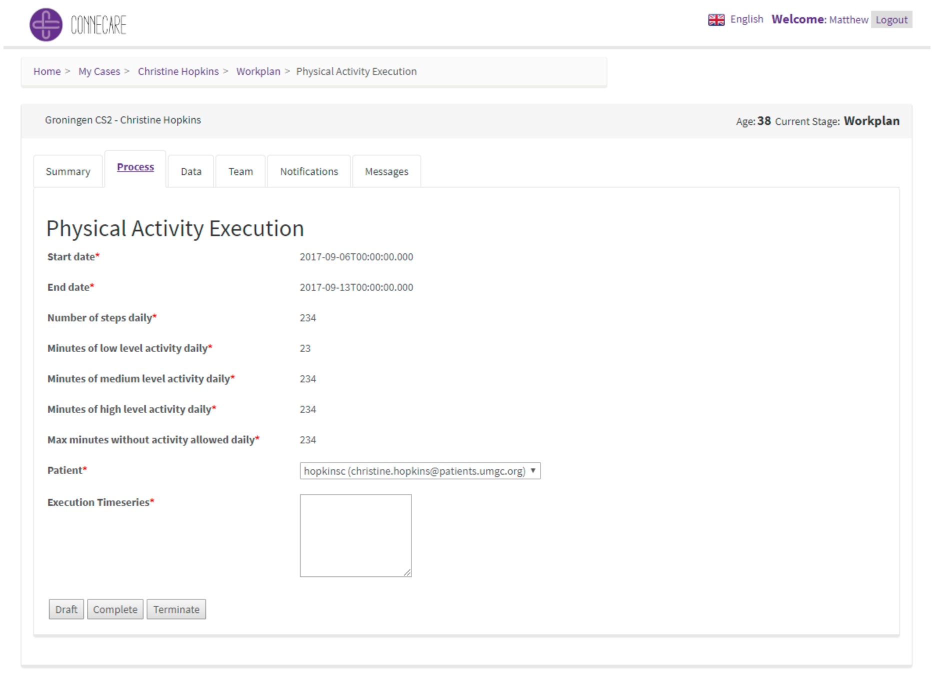


Activity demo
Please keep a daily record of your steps and try to get at least 8000

Access demo
This doctor would like access to your data. this is part of the CONNECARE...

Measurement demo
Please register your weight

Figure 6 - An alert to follow the physical activity prescription (SMS).



Home > My Cases > Christine Hopkins > Workplan > Physical Activity Execution

Groningen CS2 - Christine Hopkins

Age: 38 Current Stage: Workplan

Physical Activity Execution

Start date*	2017-09-06T00:00:00.000
End date*	2017-09-13T00:00:00.000
Number of steps daily*	234
Minutes of low level activity daily*	23
Minutes of medium level activity daily*	234
Minutes of high level activity daily*	234
Max minutes without activity allowed daily*	234
Patient*	hopkinsc (christine.hopkins@patients.umgc.org)
Execution Timeseries*	

Figure 7 - Physical activity monitoring (SACM).

2.3 The Recommender System

With the goal of providing patient empowerment through the use of the SMS with particular reference to the physical activity monitoring, a first recommender system has been defined and designed[6] [1].

The recommender system, namely *ArgoRec*, will be presented in the following by describing its model and inner functioning.

2.3.1 The Model

In *ArgoRec*, recommendations and feedback are interpreted as *arguments*, whose *claims* (i.e.; the fact that the patient is doing well or not) are supported by *premises* constituted by the patient's adherence. The strength of *support* relations is *dynamically* computed (and adjusted), and depends on the time window that the adherence of the patient refers to: recent activity events (that is, fulfilment to more recent prescriptions) are stronger premises with respect to more ancient events. Accordingly, *attack* relations between arguments are possible because the recommendation engine may be tempted to generate conflicting recommendations based on different *time windows*, i.e., focusing on the adherence level (memoryless) versus the adherence profile (historical). In this case, argumentation helps *ArgoRec* to generate the most correct recommendation (or feedback), by exploiting argumentation-based reasoning to select the stronger claim –that is, the one supported by the strongest premises. Figure 1 depicts an example of argumentation graph in which recommendation "keep going" is the strongest argument, thus gets generated and dispatched. Essentially, despite comparison of latest fulfilment event ($fulfilment_{i,t}$) with previous one ($\text{fulfilment}_{i,t-1}$) suggests to warn the patient about the need for improvement (recommendation "must improve") –since her/his adherence is worsening–, the fact that there is still time left to complete prescription ($prescription_i$) steers arguments strength in favour of recommendation "keep going", to further motivate the patient.

[6] This work is part of the task 4.6 "Recommender system for self-management" led by UNIMORE. The final recommender system in the CONNECARE project is expected at M40 and will be described in D4.6 "Recommender System for Self Management". For the sake of completeness, we present here its first release as part of the Study Release at M18 (September 2017).

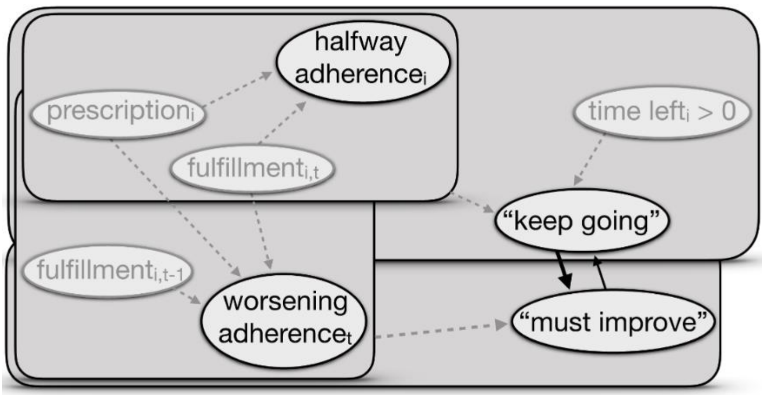


Figure 1 - Example of argumentation graph exploited by ArgoRec.

Besides correctness, this way ArgoRec can, on the one hand, provide to patients more convincing recommendation messages, by motivating and *explaining* the reasons behind them (the *why*) and, on the other hand, provide to clinicians *insights* on the decision making process leading to that precise feedback (the *how*).

Both can be achieved by navigating the argumentation (sub)graph whose claim is the recommendation or feedback itself to, for instance, generate explanation sentences through *Natural Language Processing* (NLP) techniques and *argumentation mining*.

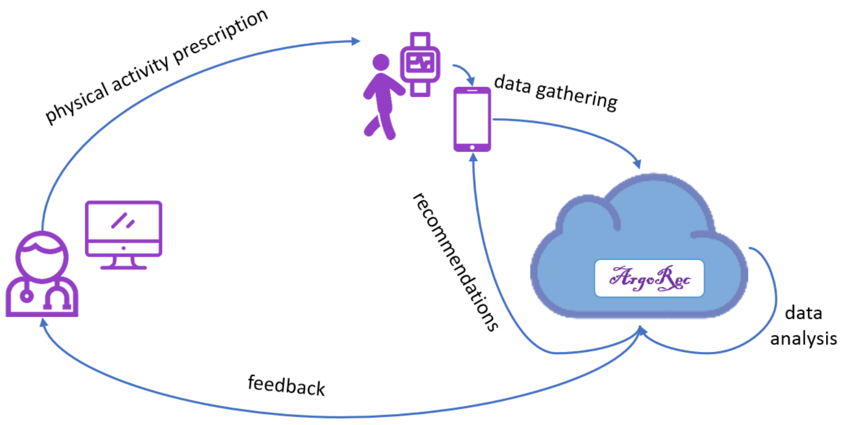


Figure 2 - Flow of data regarding the prescription of a physical activity.

To deliver its functionalities, *ArgoRec* works as follows (see also Figure 2**¡Error! No se encuentra el origen de la referencia.**). Whenever an activity fulfilment event is received: (i) it is checked against the corresponding prescription to compute adherence level of the patient and update her/his adherence profile, depending on the configured strategy (i.e.; defining how to weight older versus newer events); (ii) new arguments are generated accordingly and added to *ArgoRec* argumentation graph (i.e., an "halfway" fulfilment may support a "keep going" recommendation); and (iii) weights of relations are updated depending on the newly-added arguments (i.e.; new premises for a claim increasing support strength) and *ArgoRec* own strategy (i.e.; decreasing strength of arguments as time flows). Finally, periodically and depending on the configured policies, *ArgoRec* generates recommendations and feedback based on the strongest argument(s) in the graph –i.e.; navigating the graph to generate sentences through NLP.

2.3.2 Challenges

Despite argumentation being an active field of research for so long, most of the fundamental results achieved are theoretical.

Being interested in applying argumentation in a recommender system to empower complex chronic patients, we move from a theoretical perspective to the real-world. It is worth noting that the main challenge is moving from a technical perspective (such as finding the best logic frameworks) to an organisational and social change in case management for both patients and clinicians. In fact, on the one hand, patients have to learn how to interact with suitable devices (e.g., wristband and smartphone or wireless medical devices) and they have to be confident about the recommendations they receive. On the other hand, clinicians have to receive the right information (*grey-box* approach) to trust the recommendations automatically generated. What may happen is that, if not correctly motivated, patients stop to use the SMS and clinicians interrupt prescription of activities through the SMS or checking of the received feedback due to the lack of trust and transparency of decision making.

2.3.3 Next Steps

Being clinical studies not started yet, the recommender system has been preliminary tested with data from healthy-volunteers. Data collected during the clinical studies from M20 (November 2017) to M36 (March 2019) will be used to improve the system and to test it in the CONNECARE scenario.

In line with the co-design approach and the PDSA cycle feedback will be continuing kept and use to iteratively improve and update the system. The final recommender system is expected at M40 (July 2019).

3. Monitoring through Questionnaires

3.1 Requirements

During the co-design process of CONNECARE leaded by the clinical partners, the questionnaires to be answered by the patient in each case study and in each site have been selected from off-the-shelf (standard) ones as well as specifically defined for the project. Questionnaire have been mentioned and reported in Annexes of D2.4 "*Case studies description and the associated co-design*" (Annex 6.2 and its subsections 6.2.1 –Barcelona–, 62.2 –Lleida–, 6.2.3 –Groningen–, and 6.2.4 –Israel). For the sake of completeness and to clarify the role of this microservice in the overall scenario, let us list them here:

- Standard questionnaires
    - Hospital Anxiety and Depression Scale[7]
    - The Western Ontario and McMaster Universities Osteoarthritis Index[8]
    - SF-12[9]
    - S-LANSS
    - Barthel index[10]
    - EQ5D[11]
    - TiC-P[12]
- Defined in CONNECARE
    - Self-care auto-test (by IRBLL)
    - Verbal Numerical Rating Scale during hospitalization (by IRBLL)
    - Autocheck Health Status (by IRBLL)
    - Asthma control (by UMCG)
    - COPD health status (by UMCG)
    - Illness perception questionnaire (by UMCG)
    - Self-assessment (by UMCG)

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

[8] http://www.performanceptc.com/paperwork/womac.pdf

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

[10] http://www.massgeneral.org/stopstroke/assets/PDFs/barthel_index.pdf

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

[12] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3694473/

3.2 The Service

The Questionnaires service allows clinicians to monitor health status, quality of life, mood, and pain, as well as rehabilitation status by asking patients to fill selected questionnaires. Thus, the clinician prescribes one or more questionnaires to be filled by the patient who receives a notification in the SMS with the request to filling it together with the frequency (only once, every week, etc.).

The key functionalities of this service are:

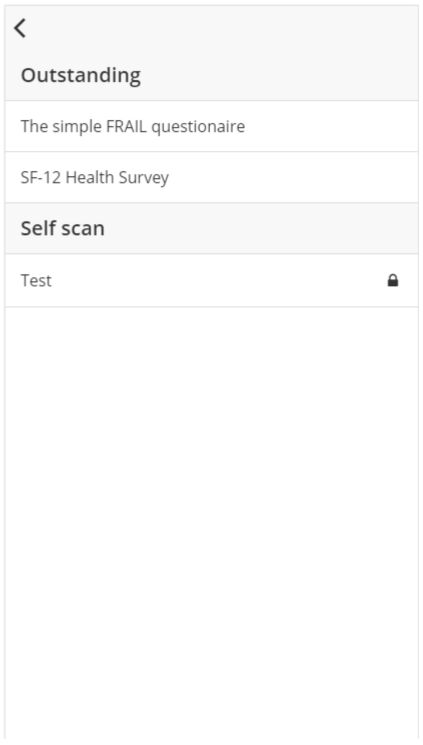
- Assignment of one or more questionnaires to a patient.
- Nullifying a previous assignment of one or more questionnaires.
- Set-up of the questionnaire(s) to be answered, together with the frequency that questionnaires will be requested to the patient's assigned under the medical surveillance provisions.
- Send back questionnaire answers to the clinician.
- Check the list of prescribed questionnaires and their answers.

The user roles involved in the process are, of course, the patient and the clinician. Depending on the specific questionnaire, the CS and the site, different roles of clinicians may prescribe a questionnaire[13].

The data model and the API description of this service are given in the Annex 8.2 , in the following of this Section we illustrate how the service works through some screenshots.

Through the SACM the clinician prescribes the questionnaire to be filled together with a deadline and a frequency of answering. The patient receives the request in her smartphone through the SMS and may access to it to see the questionnaire and fill it (Figure 8). Through the SACM the clinician may access to that given answers or receive an alert in case the patient did not fill the questionnaire by the fixed deadline (Figure 9).

[13] The list of user groups for CS is given in D2.4 “Case Studies Description and the Associated Co-Design”.



Outstanding

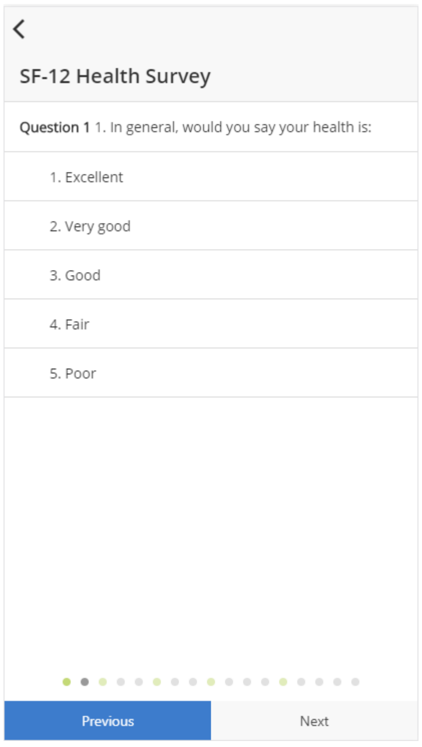
The simple FRAIL questionaire

SF-12 Health Survey

Self scan

Test

Figure 8 - List of pending questionnaires for the patient (SMS).



SF-12 Health Survey

Question 1 1. In general, would you say your health is:

1. Excellent
2. Very good
3. Good
4. Fair
5. Poor

Figure 9 - Example of question to be answered by the patient (SMS).

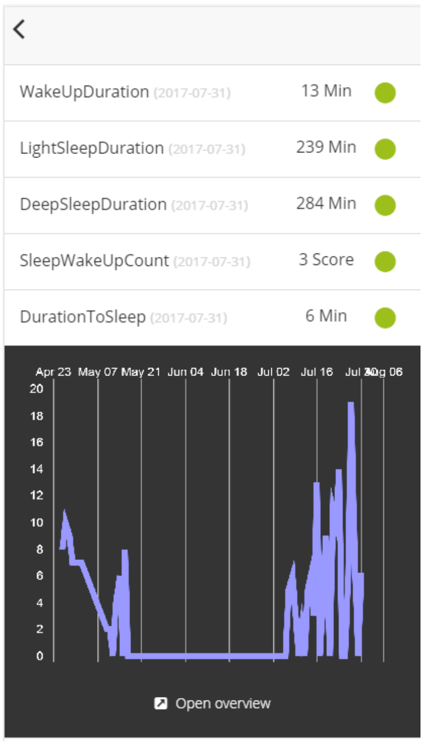
4. Sleeping Monitoring

4.1 Requirements

In Israel and Groningen for both CS1 and CS2, clinicians asked to monitor also sleeping activity with particular reference to timeslots of awake, light sleep and deep sleep. As sketched in Table 2, all the tested activity trackers allow this kind of monitoring. Thus, also this feature is gathered from the Fitbit and the Withings API.

4.2 The Service

In case of monitoring the sleeping activity, not a real prescription could be made. Nevertheless, the clinician may set suitable thresholds to be alerted in case the patient is not sleeping enough (minimum number of hours) and/or is sleeping too much (maximum number of hours). Alerts are then sent to the clinician in case of overpassing the thresholds. Moreover, the patient and also the clinician may check the sleeping activity through the SMS and SACM, respectively. Figure 3 shows an example of how the data are shown to the patient through the SMS.



WakeUpDuration (2017-07-31)	13 Min	●
LightSleepDuration (2017-07-31)	239 Min	●
DeepSleepDuration (2017-07-31)	284 Min	●
SleepWakeUpCount (2017-07-31)	3 Score	●
DurationToSleep (2017-07-31)	6 Min	●

Figure 3 - Summary of the sleep information and detailed information about the evolution of the last days in a chart (SMS).

Details on data modelling and API definition are given in the Annex 8.3.

5. Further Services to Support Monitoring

Besides specific services to monitoring basic activities such as walking or sleeping, main features that have been required as general features of the SMS are: messaging, agenda, and advices. Thus, those services have been defined and developed and are currently part of the Study Release of the SMS.

5.1 Messaging

This service is aimed at putting in contact the patient with the clinical staff in charge of following her/his case, through text messages and images and video sharing. From the one hand, the SMS provides a chat message box (see **¡Error! No se encuentra el origen de la referencia.**) to allow the patient to start or to follow a conversation. From the other hand, professionals in the SACM interact with a forum-style message application in which all the professionals in charge of the case may read/write, send/receive images and videos (see Figure 11).

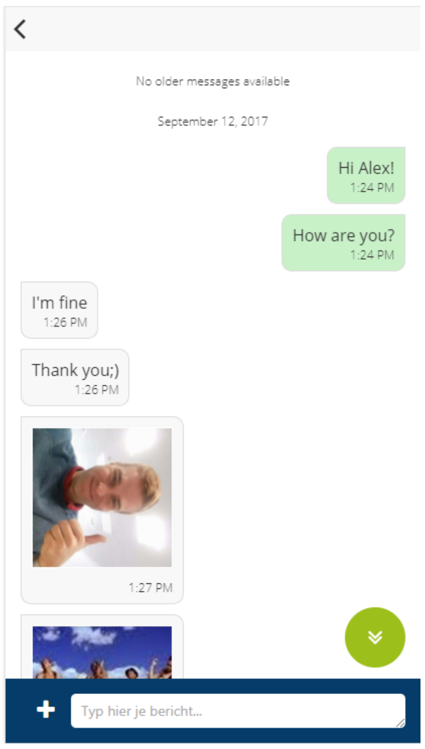
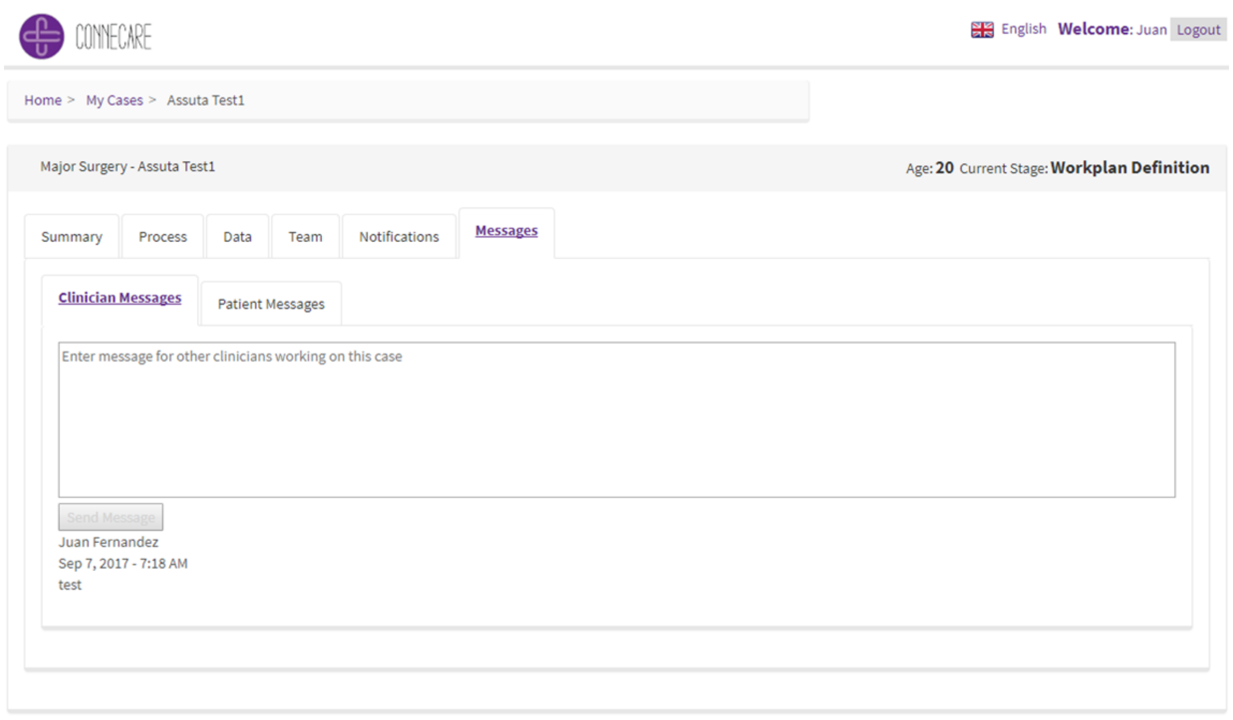

No older messages available
September 12, 2017
Hi Alex!
1:24 PM
How are you?
1:24 PM
I'm fine
1:26 PM
Thank you;)
1:26 PM
1:27 PM

Figure 10 - Example of conversation between the clinician and the patient sending images and text (SMS)..


CONNECARE
English Welcome: Juan Logout
Home > My Cases > Assuta Test1
Major Surgery - Assuta Test1
Age: 20 Current Stage: Workplan Definition
Summary
Process
Data
Team
Notifications
Messages
Clinician Messages
Patient Messages
Enter message for other clinicians working on this case
Send Message

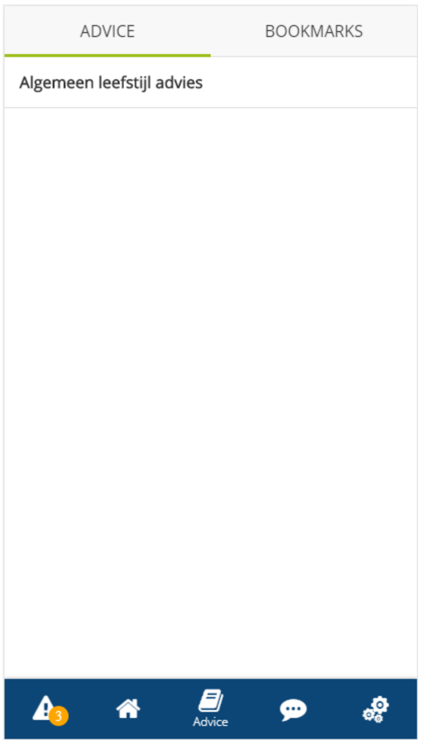
Juan Fernandez
Sep 7, 2017 - 7:18 AM
test

Figure 11 - Professional User Interface to send messages and read them (SACM).

Details on data modelling and API definition are given in Annex 8.4.

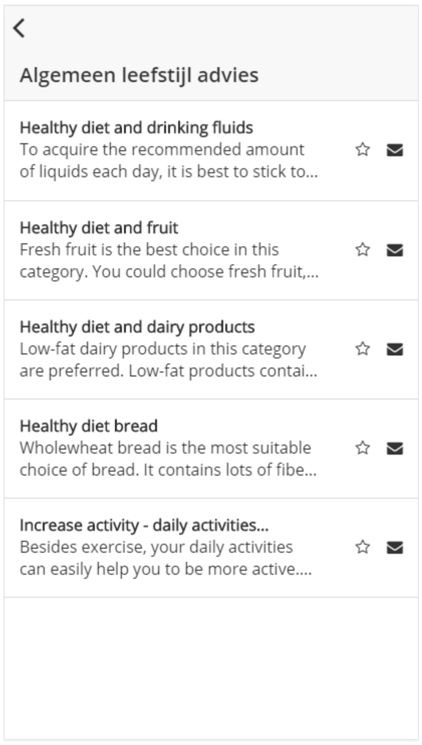
5.2 Advices

This service is aimed at giving assistance to the patient in terms of training material as links to suitable resources, videos, tutorials. In fact, depending on the case study and on the specific site, clinicians will select suitable training and educational material to be suggested to patients to better be aware of their disease and improve follow-up of the corresponding care plan (see Figure 12 and Figure 13). That material could be in form of documents or links to external video.


ADVICE BOOKMARKS

Algemeen leefstijl advies

Figure 12 - Example of advices grouped (SMS).


Algemeen leefstijl advies

Healthy diet and drinking fluids
To acquire the recommended amount of liquids each day, it is best to stick to...

Healthy diet and fruit
Fresh fruit is the best choice in this category. You could choose fresh fruit,...

Healthy diet and dairy products
Low-fat dairy products in this category are preferred. Low-fat products contai...

Healthy diet bread
Wholewheat bread is the most suitable choice of bread. It contains lots of fibe...

Increase activity - daily activities...
Besides exercise, your daily activities can easily help you to be more active....

Figure 13 - Example of advices for the patient (SMS).

Details on data modelling and API definition are given in Annex 8.5.

6. Conclusions and Next Steps

In CONNECARE, three levels of monitoring are proposed and suitable services defined and developed accordingly. In this deliverable the 5 services to perform basic monitoring that have been developed have been presented: physical activity, questionnaires, sleeping, messaging, and advices. All those systems are part of the Study Release and will be used during the clinical studies that will start at M20 (November 2017). Data gathered during this monitoring activity are used to give an immediate feedback to both patients and professionals, as well as to send alerts in case anomalies or not fulfilment are registered. Moreover, those data will be also used by the recommender system to improve the current model and to define and developed the quality of life assessment system.

Results presented in this document belongs to the work of Task 4.2 “Basic monitoring” that ended at M18 (September 2017) and part of Task 4.6 “Recommender system for self-management”. Nevertheless, due to the iterative approach based on co-design that is adopted in CONNECARE, further work on these services would be done in the future to improve the current version of the services as well as to better align them to with the feedback that will be received during and after the clinical studied that will start at M20 (November 2017) and will end at M36 (March 2019). The corresponding updates of the services will be provided in the D4.3 “Advanced monitoring tools” (M36 – March 2019) and/or in the D4.4 “Assistive monitoring tools” (M40 – July 2019). Finally, the Final Release of the recommender system will be presented in D4.6 “Recommender System for self-management” at M40 (July 2019).

7. References

[1] Fernández, J.M., Mamei, M., Mariani, S., Miralles, F., Steblin, A., Vargiu, E., & Zambonelli, F. Towards Argumentation-based Recommendations for Personalised Patient Empowerment. International Workshop on Health Recommender Systems, co-located with ACM RecSys 2017. August 31st, 2017, Como, Italy.

8. Annexes

8.1 Physical Activity Monitoring

8.1.1 Data Model

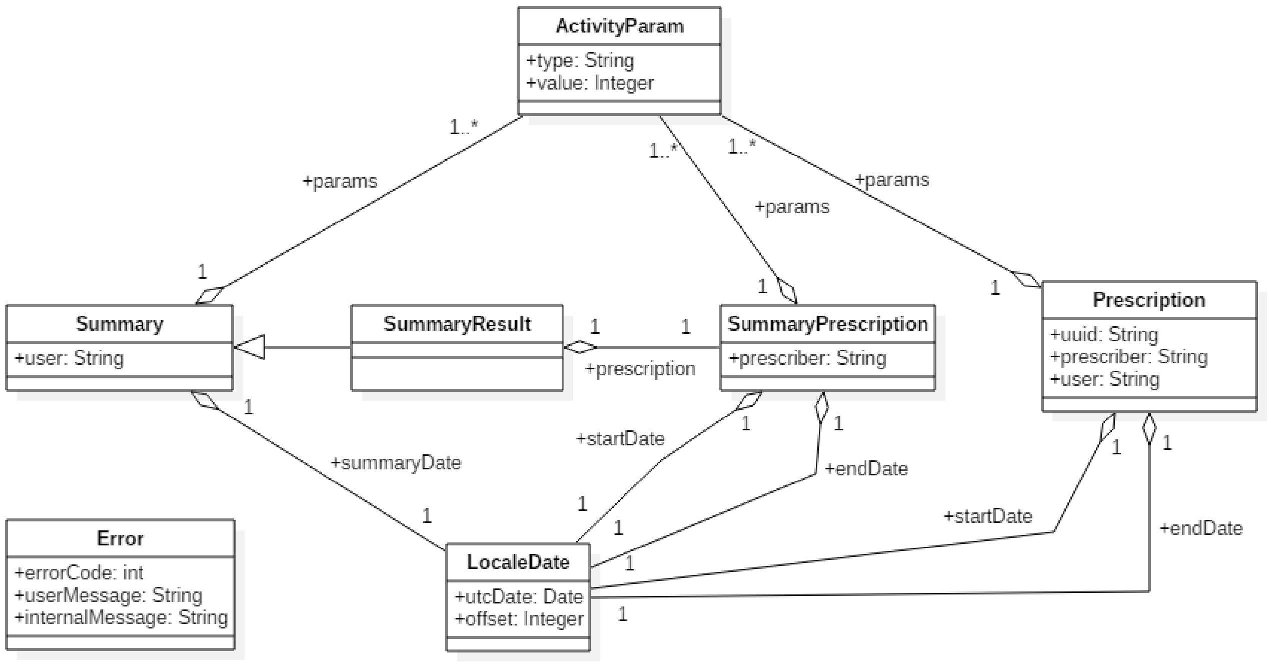


Figure 14 - Data model diagram for the Physical Activity service.

8.1.1.1 Preliminary Considerations

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

8.1.1.2 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
}
```

8.1.1.3 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
    }
  ]
}
```

8.1.1.4 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
    }
  ]
}
```

8.1.1.5 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
    }
  ]
}
```

8.1.1.6 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
      }
    ]
  }
}
```

8.1.1.7 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.
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"
}
```

8.1.2 API definition

8.1.2.1 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 8.1.1.7 for more details

8.1.2.2 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 8.1.1.7 for more details

8.1.2.3 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 8.1.1.7 for more details

8.1.2.4 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 8.1.1.7 for more details

8.1.2.5 *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 8.1.1.7 for more details

8.1.2.6 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 8.1.1.7 for more details

8.1.2.7 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 8.1.1.7 for more details

8.1.2.8 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 8.1.1.7 for more details

8.1.2.9 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 8.1.1.7 for more details

8.2 Monitoring through Questionnaires

8.2.1 Data Model

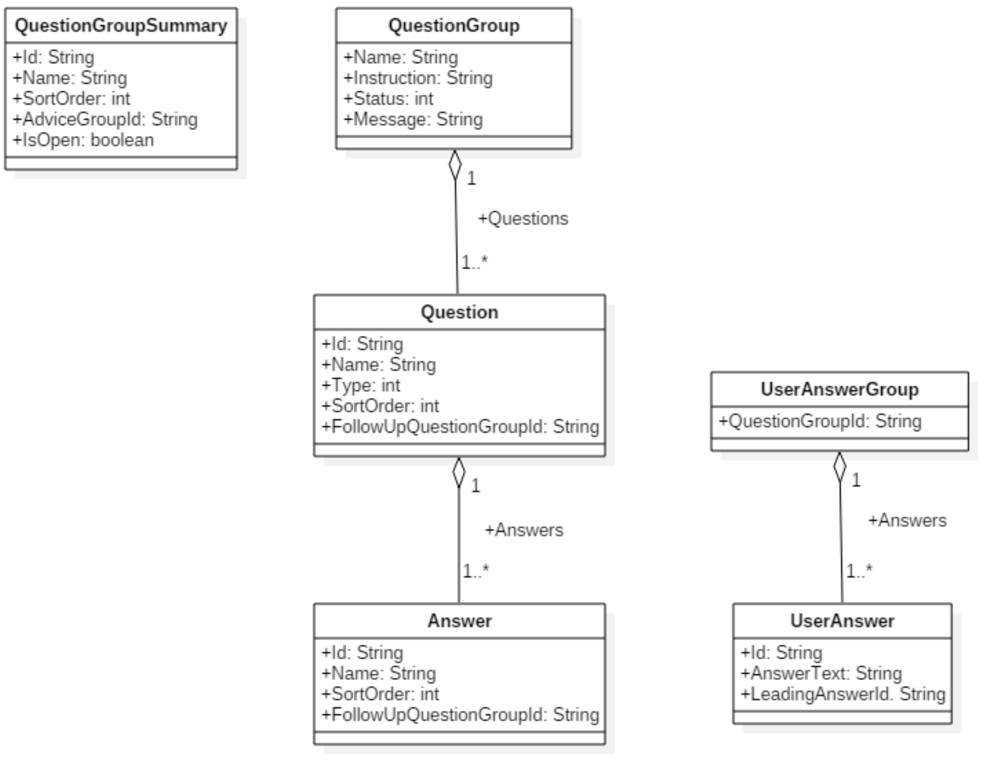


Figure 15 - Data model diagram for the Questionnaire service.

8.2.1.1 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.

8.2.1.2 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"
}
```

8.2.1.3 *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

8.2.1.4 *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

8.2.1.5 *Question*

This object contains the question model.

Atribute	**Description**
Id String	Alphanumeric id of the question
Name	The question

String	
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

8.2.1.6 *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

8.2.1.7 *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

8.2.1.8 *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

8.2.2 API definition

8.2.2.1 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 8.2.1.2 for more details

8.2.2.2 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	Query	The application from which the request is performed.

String		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 8.2.1.2 for more details

8.2.2.3 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 8.2.1.2 for more details

8.2.2.4 *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 8.2.1.2 for more details

8.2.2.5 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 8.2.1.2 for more details

8.2.2.6 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 8.2.1.2 for more details

8.2.2.7 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 8.2.1.2 for more details

8.2.2.8 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 8.2.1.2 for more details

8.3 Sleeping Monitoring

8.3.1 Data Model

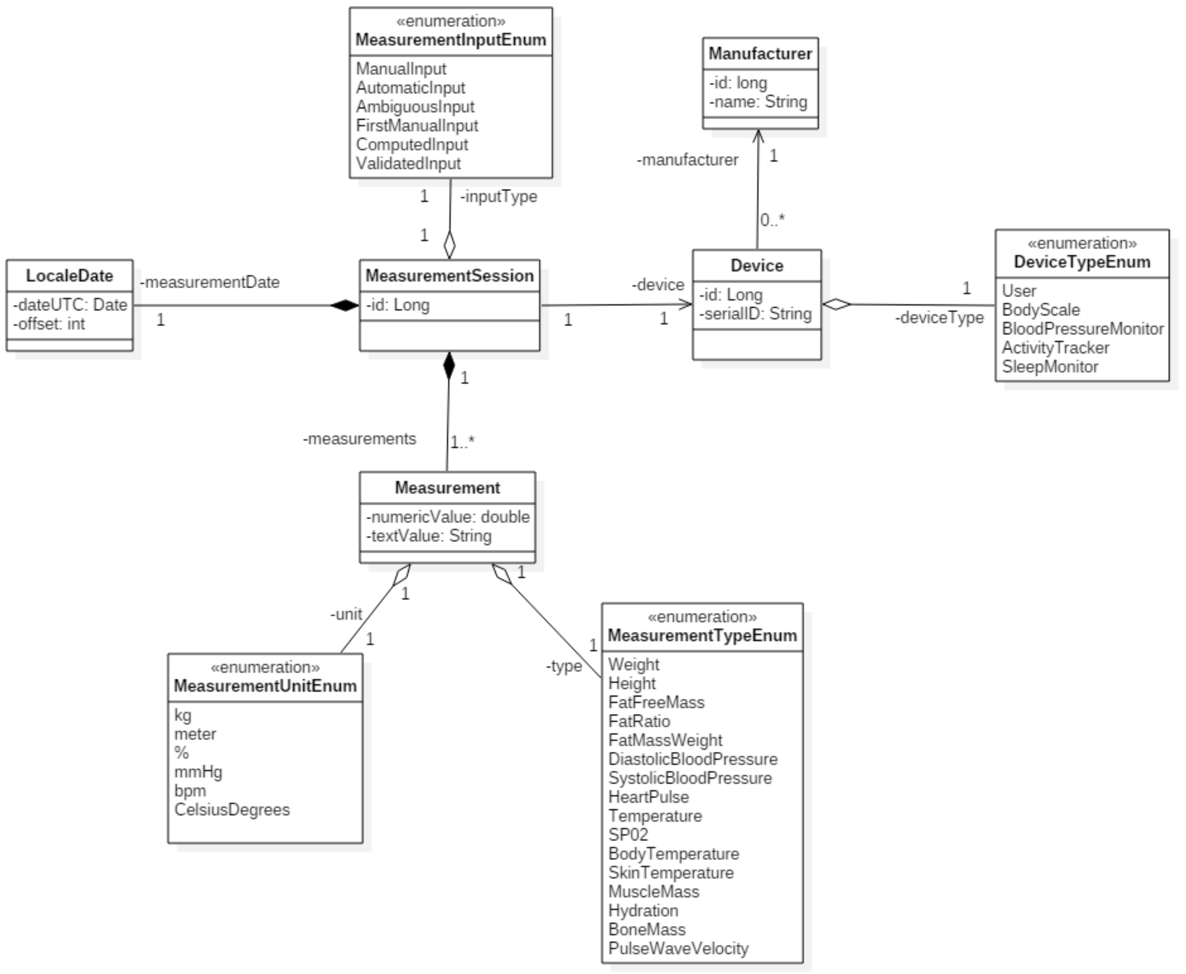


Figure 16 - Monitoring Data model diagram.

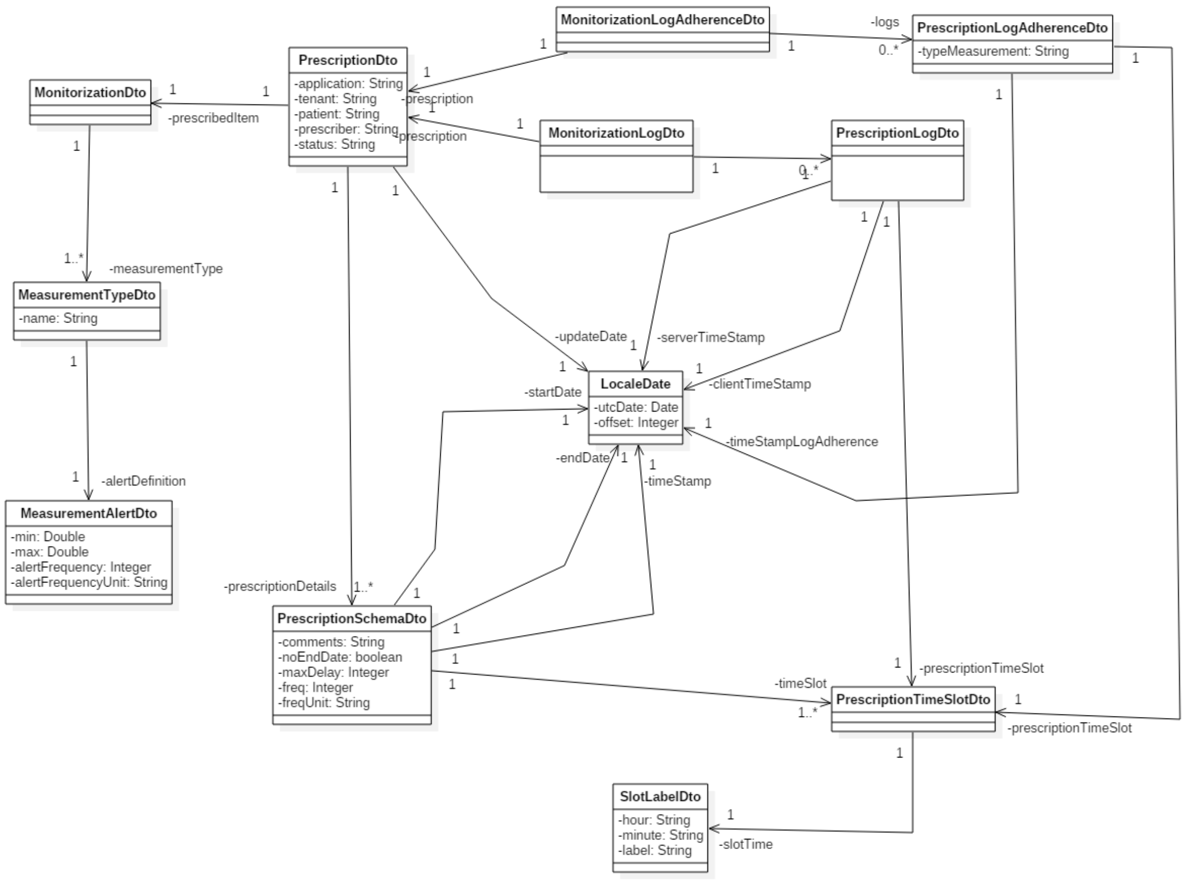


Figure 17. Prescription model diagram

8.3.1.1 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.).

8.3.1.2 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 8.3.1.33*
prescriptionDetails ArrayList<PrescriptionSchemaDto>	**N**	List of the prescriptionSchemaDto objects that a prescription can have *See section 8.3.1.6*

Model Schema:

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

8.3.1.3 MonitorizationDto

The object of type **MonitorizationDto**

Atribute	Optional	Description
measurementType ArrayList<MeasurementTypeDto>		List of the measurement types of the prescription *See section 8.3.1.4*

Model Schema:

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

8.3.1.4 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 8.3.1.5*

Model Schema:

```
{
    "name" : "String",
    "alertDefintion" : <MeasurementAlertDto>
}
```

8.3.1.5 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"
}
```

8.3.1.6 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<PrescriptionTime SlotDto>	**N**	List of the PrescriptionTimeSlotDto objects *See section 8.3.1.7*

Model Schema:

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

8.3.1.7 PrescriptionTimeSlotDto

The object of type **PrescriptionTimeSlotDto**

Atribute	Optional	Description
slotTime <SlotLabelDto>	**N**	The SlotLabelDto object *See section 8.3.1.8*

Model Schema:

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

8.3.1.8 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"
}
```

8.3.1.9 MonitorizationLogDto

The object of type **MonitorizationLogDto**

Atribute	Optional	Description
prescription <PrescriptionDto>		The data of the description of the prescription *See section 8.3.1.2*
logs ArrayList<PrescriptionLogDto>		List of the prescriptions logs of the prescription *See section 8.3.1.11*

Model Schema:

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

8.3.1.10MonitorizationLogAdherenceDto

The object of type **MonitorizationAdherenceLogDto**

Atribute	Optional	Description
prescription <Prescription>		The data of the description of the prescription *See section 8.3.1.2*
logs ArrayList<PrescriptionLogAdherenceDto>		List of the prescriptions logs of the prescription *See section 8.3.1.12*

Model Schema:

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

8.3.1.11PrescriptionLogDto

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
}
```

8.3.1.12PrescriptionLogAdherenceDto

The object of type **MonitorizationAdherenceLogDto**

Atribute	Optional	Description
prescriptionTimeSlot <PrescriptionTimeSlotDto>		The data of the description of the prescription *See section 8.3.1.7*
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
}
```

8.3.1.13AdherenceDefinitionDto

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"
}
```

8.3.1.14Error

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"
}
```

8.3.2 API Definition

8.3.2.1 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 8.3.1.2).

Responses

Success

Code error:

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

Body message:

- The endpoint returns a **PrescriptionDto** object (see section 8.3.1.2).

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 8.3.1.14 for more details

8.3.2.2 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 8.3.1.2).

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 8.3.1.148.1.1.7 for more details

8.3.2.3 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 8.3.1.2).

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 8.3.1.148.1.1.7 for more details

8.3.2.4 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 8.3.1.2).

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 8.3.1.14 for more details

8.3.2.5 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 8.3.1.2).

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **PrescriptionDto** object (see section 8.3.1.2).

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 8.3.1.14 for more details

8.3.2.6 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 8.3.1.68.3.1.2).

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns a **PrescriptionDto** object (see section 8.3.1.2).

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 8.3.1.148.1.1.7 for more details

8.3.2.7 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 8.3.1.9).

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 8.3.1.148.1.1.7 for more details

8.3.2.8 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 8.3.1.9).

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 8.3.1.148.1.1.7 for more details

8.3.2.9 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 8.3.1.13).

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an **AdherenceDefinitionDto** object (see section 8.3.1.13).

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 8.3.1.148.1.1.7 for more details

8.3.2.10Adherence 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 8.3.1.13).

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 8.3.1.148.1.1.7 for more details

8.3.2.11Adherence 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 8.3.1.13).

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 8.3.1.148.1.1.7 for more details

8.3.2.12Calculate 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 8.3.1.148.1.1.7 for more details

8.3.2.13Calculate 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 8.3.1.148.1.1.7 for more details

8.3.2.14 Calculate 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 8.3.1.148.1.1.7 for more details

8.3.2.15 Update 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 section8.3.1.13 8.3.1.13).

Responses

Success

- 200 – OK (the operation was successfully done)

Body message:

- The endpoint returns an **AdherenceDefinitionDto** object (see section 8.3.1.13).

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 8.3.1.148.1.1.7 for more details

8.3.2.16Delete 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 8.3.1.148.1.1.7 for more details

8.4 Messaging & community

8.4.1 Data Model

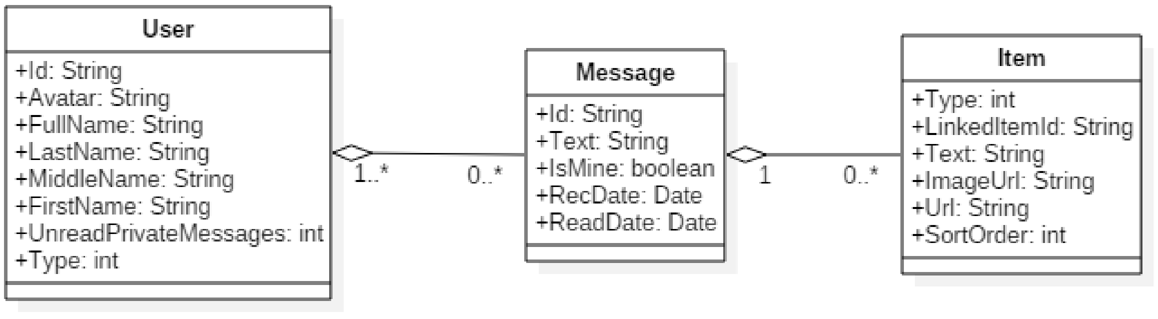


Figure 18 - Data model diagram.

8.4.1.1 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	**N**	Middle name of the user.

String		
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
}
```

8.4.1.2 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": []
}
```

8.4.1.3 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
}
```

8.4.2 API Definition

8.4.2.1 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

8.4.2.2 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

8.4.2.3 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

8.4.2.4 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

8.4.2.5 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

8.4.2.6 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

8.4.2.7 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

8.4.2.8 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

8.4.2.9 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

8.5 Advices

8.5.1 Data Model

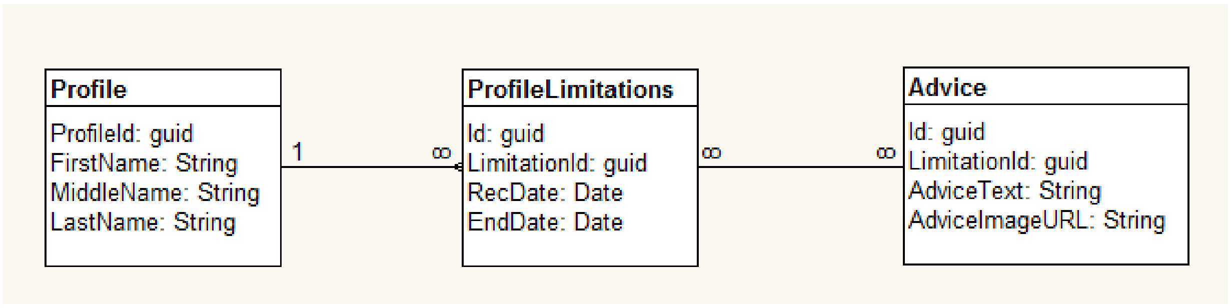


Figure 19 - Data model of the Advices service.

8.5.2 API Definition

8.5.2.1 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"
}
```

8.5.2.2 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)

8.5.2.3 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)

8.5.2.4 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": ""
}
```

8.5.2.5 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"
}
```

8.5.2.6 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"
}
```

8.5.2.7 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)