D4.4 – Dynamic DSS for an Intelligent coach 2019.08.31

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

DELIVERABLE ID:	**WP4/D4.4**
DELIVERABLE TITLE:	Plans and recommendations based on profiles
RESPONSIBLE PARTNER:	Eurecat
AUTHOR (S):	Silvia Orte, Paula Subías (Eurecat), Filippo Palumbo, Michele Girolami (CNR-ISTI), Paolo Baronti (CNR-ISTI), Martin Sykora (LU-CIM)
CONTRIBUTORS:	
NATURE	Other
DISSEMINATION LEVEL:	PU
FILE:	DEL4.4_v1.2_short
REVISION:	V1.2
DUE DATE OF DELIVERABLE:	2019.08.31
ACTUAL SUBMISSION DATE:	
CALL	European Union's Horizon 2020 Grant agreement: No 769643
TOPIC	SC1-PM-15-2017 Personalized coaching for well-being and care of people as they age

Document History

Revision	Date	Modification	Author
0.0	2019.05.29	Creation of TOC for D4.4	EURECAT
0.1	2019.06.26	First draft of D4.4	EURECAT
0.2-0.4	2019.09.09	Content added to D4.4	CNR, LU-CIM
0.5	2019.09.16	Content added to D4.4	EURECAT
1.0	2019.09.16	Version ready to be submitted	EURECAT
1.1	2019.09.20	Comments from TUD added	EURECAT
1.2	2019.09.20	Comments from Ropardo added	EURECAT

Approvals

Date	Name	Organization	Role
19/09/2019	Isabelle Kniestedt	TUD	Reviewer
19/09/2019	Gabriela Candea	Ropardo	Reviewer
20/09/2019	Silvia Orte	EURECAT	WP leader
20/09/2019	Giuseppe Andreoni	POLIMI	Scientific coordinator

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

Short Abstract

This document represents the deliverable D4.4 (“Dynamic DSS for an Intelligent Coach”). It shows the advances in the implementation of the DSS concerning the results already shown in D4.2, where the basics of the DSS, its workflows and algorithms of the first prototype were explained.

In Chapter 1, we introduce the architecture of the Decision Support System (DSS) and its modules. In Chapter 2, we present the DSS as a system encompassing multiple workflows, which are explained in detail in Chapter 3. Chapter 4 contains a brief explanation of the modules that form the DSS. Last but not least, Chapter 5 addresses the code generated.

In order to preserve the Intellectual Property generated in these tasks, we have split this document in two parts: a public and a private document. We encourage the reader to ask for the private version of D4.4. to get the complete understanding of the technical development carried out in WP4; this public version only outlines what has been realised in terms of workflows, but does not go in depth into algorithms and actual implementation of the different modules.

Key Words

Decision Support System, REST API, coaching event, recommendation, personalization, coaching.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 769643

Table of Content

1. Introduction 5
1.1. Relation with other work packages 6
2. The DSS as a whole 7
3. The DSS main workflows 7
3.1. Workflow I: Creating the 'DSS user profile' 8
3.2. Workflow II: Pathway selection 8
3.3. Workflow III: Physical activity preferences 11
3.4. Workflow IV: Weekly plan (recommendation of structured coaching events) 11
3.5. Workflow V: Recommendation of non-structured coaching events 12
3.6. Workflow VI: Sleep monitoring algorithm 12
3.7. Workflow VII: Nutritional indicators extraction 13
3.8. Workflow VIII: Tagging system 14
3.9. REST APIs from other modules 15
4. DSS modules 16
5. Code repositories 17

Table of Figures

Figure 1. General architecture of the DSS in NESTORE (update from D4.2.) 5
Figure 2. Graphical representation of the relationships among WP4 – T4.4, the activities of other NESTORE work packages, and the activities of other WP4 tasks 6
Figure 3. DSS Workflows and the involved reasoning modules 7
Figure 4. Data collected from the BCG sensors sent to the NESTORE cloud. 13
Figure 5. Graphic representation of the tagging system module. 15
Figure 6. The DSS modules 16

List of Tables

Table 1. DSS User profile parameters 9

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

1. Introduction

In this deliverable, we explain the final implementation of the Decision Support System (DSS). It builds on the main concepts, functionalities and use-cases already described in deliverable 4.2 ("First prototype of the DSS"), and the personalization processes explained in D4.3. Therefore, in this document we describe the final workflows and modules developed in the last period in depth, but do not repeat the content that is already present in the other deliverables. In Figure 1, we summarize the DSS functionalities and the document where each of the blocks are explained.

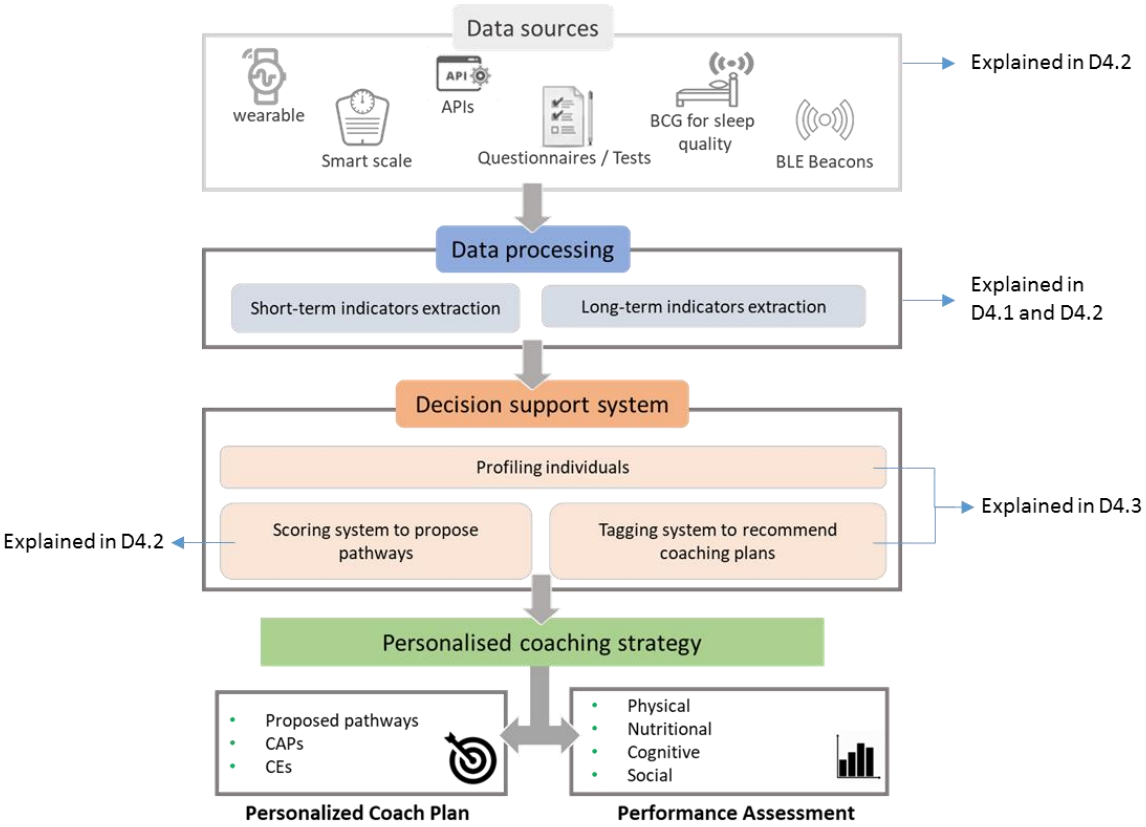

Figure 1. General architecture of the DSS in NESTORE (update from D4.2.)

In D4.2. we explained the "Coaching timeline" with all the data and types of recommendations that the user will get during all the NESTORE phases. Following, we recall some of the main concepts of the Coaching timeline:

- ***Coaching event***: a coaching event (CE) is an activity defined by the domain experts which belongs to a pathway. CEs are conceived as small steps that help to achieve the goal of a pathway. CEs can be of two kinds: structured and non-structured.

- ***Structured CEs***: this kind of CE is characterized by its thorough definition, its planning, and studied user response. The physical training workouts are clearly structured CEs since the intensity to achieve is defined in 10-minute-bouts, they are scheduled in the calendar, and the user response is measured by the Borg and TQR metrics. Other examples of structured CEs are cognitive games.

- ***Non-structured CEs***: this type of CE covers all the CEs that are not structured CEs. They are activities that do not follow any structure and are not planned in users' calendars. They are recommended by the DSS and sent as a suggestion to users when appropriate.

In parallel, the different indicators explained in D4.1. are extracted and analysed to be able to assess the performance of the user in the different domains of NESTORE: nutrition, physical activity, cognition and social. NESTORE DSS also

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

aims to analyse the trends and patterns of users in their emotional, social, and sleep dimensions. This analysis provides a further level of understanding of the global status of the final users. Specifically, we foresee two benefits from our analysis: (i) To validate the effectiveness of the recommended Coaching Events, e.g. to understand whether or not specific suggestions to the users enable a deeper level of interaction among them; (ii) to detect situations of interest in order to provide new inputs to the e-Coach for defining new goals and stimuli for the final users.

1.1. Relation with other work packages

This deliverable focuses on the results of task 4.4., in which the development of the DSS has been carried out. The data used by the DSS comes from two different places: the short-term and long-term indicators extracted in T4.1. and T4.2. on the one hand and the user profiling results coming from T4.3 on the other.
A number of Rest API and communication systems (i.e. MQTT) coming from other modules of the system and integrated through the activities performed under WP6, are also used in the DSS.

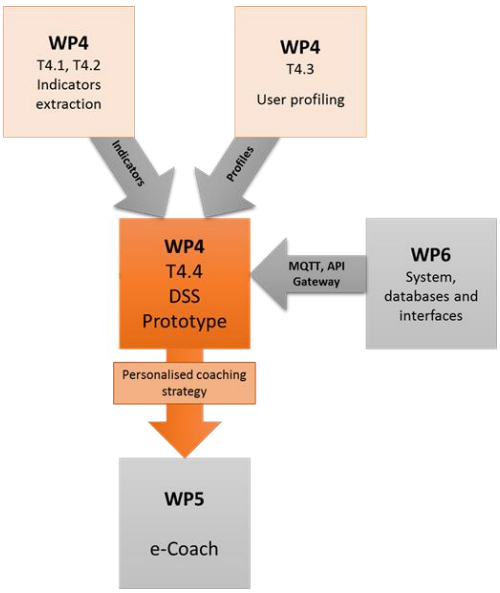

Figure 2. Graphical representation of the relationships among WP4 – T4.4, the activities of other NESTORE work packages, and the activities of other WP4 tasks

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

2. The DSS as a whole

The DSS is a *Software as a Service* (SaaS) platform, built with the main objective of providing information in terms of a) Coaching Plans, and b) performance in the various indicators described in D4.1.
Various systems inside NESTORE interrogate the different DSS APIs to get the necessary information about proposed pathways, recommended coaching events, and status of the user in the different domains. The main system that interacts with the DSS is the e-Coach, the functionalities of which are explained in the different deliverables of WP5.

Figure 3 depicts the main workflows implemented in the DSS and the modules that run behind them. In section 3 and section 4, we describe each of the workflows and each of the modules that conform the DSS.

We define *workflow* as the tasks performed in series or in parallel by two or more NESTORE subsystems working to reach a common goal. In Figure 3, there is the connection between the main workflows. For example, once *Workflow I: Create user profile*, is triggered, *Workflow II: Pathway selection*, can be triggered too, but not before. In the figure, we also mark which are the reasoning modules involved in the different workflows.

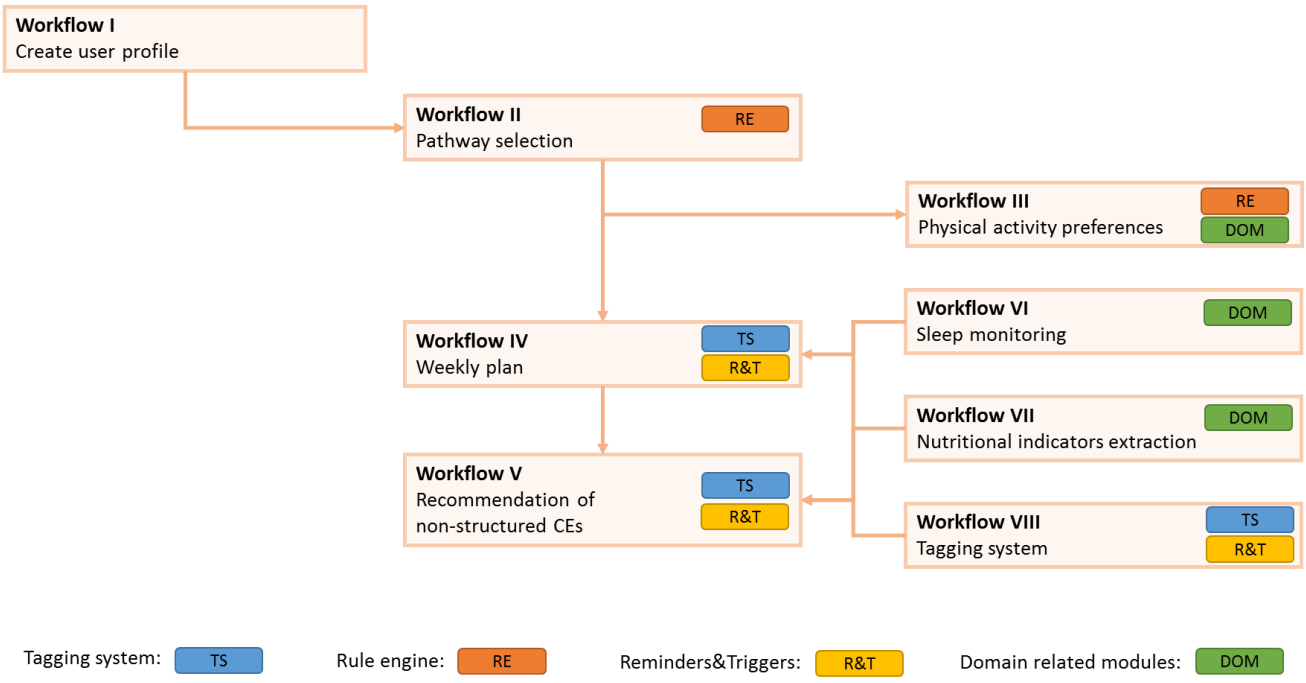

Figure 3. DSS Workflows and the involved reasoning modules

3. The DSS main workflows

Several workflows have been created to a) integrate the recommendations of *Personalized Coaching Plans* from the DSS into the e-Coach, and b) to provide the necessary information about the *User Performance* and status in the different domains. In this section we list the main ones following the information covered in Figure 3.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

3.1. Workflow I: Creating the 'DSS user profile'

The DSS is initiated when the user does their first log-in into the e-Coach mobile application. In that precise moment, the e-Coach has to POST the necessary token using this endpoint:

1. First sign-in [POST]: Get the proposed pathway for a given user
    - *URL:* /dss/signed-in
    - Parameters:
        - iam_token
2. Get the baseline data [GET]: Get the baseline data for a given user
    - URL: profile.nestore-coach.eu/api/user_options/profile/complete
    - Parameters:
        - developer_access_token
        - developer_user_id

When this workflow is triggered, the DSS creates a specific entry for this user in the DSS internal database. This entry will contain all the necessary information to follow the user in the different stages of NESTORE.

3.2. Workflow II: Pathway selection

As explained in D5.6., after some days gathering baseline information from the user, he/she will have to select a pathway. This workflow will be triggered by the e-Coach. One pathway can contain sub-pathways from the different NESTORE domains.

1. Pathway selection alarm [MQTT]:
    - *Topic:* 'dss/'+userId_token+'/pathway-selection'
    - Message: 'ready'

2. Get all the possible pathways [GET]: Get the possible pathways that apply to the user
    - *URL:* dss/possiblepathways/<developer_access_token>
    - *Output*: A list of the possible pathways per domain. Example:

    [{"domain": "PHY", "id_paths": ["PHY_RET_AF", "PHY_RET_ST"]}, {"domain": "NUT", "id_paths": ["NUT_INC_MM", "NUT_ACH_HD"]}, {"domain": "SOC", "id_paths": ["SOC_IMP_SI", "SOC_IMP_SA"]}, {"domain": "COG", "id_paths": ["COG_IMP_ME", "COG_IMP_TS", "COG_IMP_MS"]}]

3. Select a pathway [POST]: Post the pathway selected by the user
    - *URL:* /dss/currentpathway
    - Parameters:
        - *user_id:* unique identifier of the user
        - *currentpathway:* a list containing pathway-name and domain-name of the selected pathway. Example:

    "currentpathway":[{"pathway-name": "SOC_IMP_SI", "domain-name": "SOC" }, {"pathway-name": "PHY_RET_AF", "domain-name": "PHY"}, {"pathway-name": "COG_IMP_ME", "domain-name": "COG"}, {"pathway-name": "NUT_DEC_BW", "domain-name": "NUT"}]}

As a result of workflows I and II, the DSS user profile is completed following the structure provided in Table 1.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

Table 1. DSS User profile parameters

Name of the parameter	Type of value	Description	Example
_id	Object Id	Identifier of the DSS user profile	5d1b38eb3bc0571bb84ae899
Tokens. userid	String	Identifier to access Zivacare data	9caeb69a-810b-451f-aa27-83cfd1e5e993
Tokens. email	String	E-mail of the user	testuser206@gmail.com
Tokens. logMeal token	String	Token to access LogMeal API	731ccfa35745110aabb32edb162be5446f0c79bd
Sex	String	User's gender	Female
Birthday	String	User's birthday date	1953-01-13
Baseline.Dog	Boolean	Boolean that indicates whether user has a dog	True
Baseline.Stairs	Boolean	Boolean that indicates whether user has a dog	True
Baseline.Bike	Boolean	Boolean that indicates whether user has a bike	False
Baseline.mod6mwt	Integer	Value of the 6MWT at baseline	444
Baseline.mod30scrt	Integer	Value of the 60SCRT at baseline	15
Baseline.Body height	Double	Body height at baseline	1.68
Baseline.Body weight	Double	Body weight at baseline	85.69
Baseline.Body fat	Double	Body fat at baseline	0.29
Baseline.Calcium	Double	Average calcium intake at baseline	978
Baseline.Fat	Double	Fat at baseline	453.2
Baseline.Muscle mass	Double	Muscle mass at baseline	0.68
Baseline.Vitamin B12	Double	Average vitamin B12 intake at baseline	31
Baseline.Protein intake	Double	Average protein intake at baseline	0.94
Baseline.Vitamin D	Double	Average vitamin D intake at baseline	70
Baseline.BMI	Double	BMI at baseline	30.2
Baseline. Carbohydrate	Double	Average carbohydrate intake at baseline	432.69
Baseline – Waist circumference	Double	Waist circumference at baseline	69
Baseline – Activity preferences	List of String	List of activity preferences indicated at baseline	["Theatre","Cinema","Play","Cook","Handicraft"]
Baseline – Refused foods	List of String	List of refused foods indicated at baseline	["Lactose"]
Baseline – CFQ	Integer	Result of CFQ at baseline	5
Baseline – N-back	Double	Result of n-back test at baseline	0.62
Baseline – Numerical updating	Double	Result of numerical updating test at baseline	0.59
Baseline – LSN scale	Double	Result of LSN scale at baseline	11.53
Baseline - De Jong Gierveld Loneliness Scale	Integer	Result of De Jong Gierveld Loneliness Scale at baseline	4
Baseline - Daily loneliness	Integer	Result of daily loneliness test at baseline	1
Baseline – Living area	String	Living area	Urban
Tags	List of String	List of user tags inferred from the user profile	["bike","read","music","write","run", "ride a bicycle", "sweetened or carbonated beverages"]

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

Pathways	List of Object	List of chosen pathways	[{"pathway-name":"PHY_RET_AF", "domain-name":"PHY"}, {"pathway-name":"NUT_ACH_HD", "domain-name":"NUT"}, {"pathway-name":"SOC_IMP_SA", "domain-name":"SOC"}, {"pathway-name":"COG_IMP_ME", "domain-name":"COG"}]
Preferences – intensity	Char	Value chosen as the preferred intensity	V
Preferences – number sessions	Integer	Number of sessions chosen as the preferred intensity	3
Working memory - Time	Integer	Integer that indicates which time of the day is (categorized as integers)	1
Working memory – pilot day	Integer	Number of days since pilot started	2
Working memory – stage	String	Phase of the user journey within NESTORE	Intervention
Daily routines	Object	Hours that describe the users' daily routines in UTC format (the date is omitted)	"awakening":"1970-01-01T07:00:00", "breakfast":"1970-01-01T07:30:00", "morning":"1970-01-01T08:00:00", "lunch":"1970-01-01T13:00:00", "afternoon":"1970-01-01T16:00:00", "snack":"1970-01-01T17:30:00", "evening":"1970-01-01T20:00:00", "dinner":"1970-01-01T20:30:00", "night":"1970-01-01T22:00:00"

The e-Coach can extract the information of the DSS user profile at their convenience by using the following endpoint:

DSS user profile [GET]: Get the user information provided by the DSS

- *URL:* /dss/realUserInfo/<developer_userId>
- Output: a json containing all DSS user profile parameters. Example:

{"_id": {"$oid": "5d31aaf747d7e9318ed4bb3d"},"baseline": {"Activity preferences": ["Cook", "Music", "Handicraft", "Run"], "BMI": 28.47, "Bike": false, "Body fat": 0.38, "Body height": 1.56, "Body weight": 69.69, "CFQ": 0, "Calcium intake": 1019, "Carbohydrate intake": 408.66, "Daily loniless": 0, "De Jong Gierveld Loneliness Scale": 1, "Dog": false, "Facilities nearby": [[]], "Fat intake": 500.82, "LSN scale": 7.18, "Living area": ["Urban"], "Muscle mass": 0.53, "N-back test": 0.68, "Numerical updating": 0.46, "Refused foods": ["Nothing"], "Stairs": true, "Vitamin B12 intake": 26, "Waist circumference": 71, "mod30scrt": 11, "mod6mwt": 391}, "birthday": "1945-07-16", "enabled": true, "id_test": 11, "name": "testuser203", "real_user": true, "sex": "Female", "tokens": {"developer_userid": "9eed3a28-044f-4bf3-be7a-5ff2da12d890", "logMeal": "none"}, "working_memory": {"pathways": [{"pathway-name": "PHY_RET_AF", "domain-name": "PHY"}, {"pathway-name": "NUT_ACH_HD", "domain-name": "NUT"},

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

{"pathway-name": "SOC_IMP_SA", "domain-name": "SOC"}, {"pathway-name": "COG_IMP_ME", "domain-name": "COG"}]}}

3.3. Workflow III: Physical activity preferences

Once the pathway has been selected, the rule-based reasoning system is prompted, so that it groups the user in the "retain" or "improve" group. Afterwards, depending on the group that he/she belongs to, the user is asked for some preferences related to physical activity. The e-Coach can get the possibilities from the following endpoint:

1. Get the possible structured aerobic fitness sessions [GET]: Get the intensities and number of sessions that the specified user can choose
    - *URL:* dss/possiblepathways/<access_token>
    - *Output*: A list of the possible sessions calculated according to the group that the user belongs to. Example: [{"intensity": "vigorous", "sessions": 3}, {"intensity": "moderate", "sessions": 5}, {"intensity": "light", "sessions": 7}]

3.4. Workflow IV: Weekly plan (recommendation of structured coaching events)

The user receives the weekly planning through a MQTT message generated by the DSS. The decision to generate this message comes from the "Reminders & Triggers" module, which periodically evaluates if the contextual information matches the status and stage of the user.
When the e-Coach receives the MQTT message, the user is able to schedule the structured activity through the e-Coach interface.
Note that this workflow applies only to structured activities, which are: aerobic fitness sessions, cognitive games, cooking and doing groceries.

1. New weekly plan alarm [MQTT]:
    - *Topic:* 'dss/'+userId_token+'/weekly-plan'
    - Message: 'ready'

2. Get weekly plan [GET]: Get the weekly plan for the user announced in the mqtt message.
    - *URL:* dss/weekly-plan/<developer_access_token>
    - *Output*: A list of the structured activities planned for the current week. Example: [{"_id": {"$oid": "5d5d4443bb81922b24b4a55d"}, "ce_id": "ce_phy_1015"}]

3. Get unavailability [GET]: Get the timeslots for when the user is not available.
    - *URL*: /dss/unavailable/<ceentryid>
    - *Output*: A list of the timeslots when the user is not available during the next week. Example: ["2019-08-23T15:00:00+0000","2019-08-24T10:00:00+0000","2019-08-24T11:00:00+0000","2019-08-24T12:00:00+0000","2019-08-26T19:00:00+0000", "2019-08-27T10:00:00+0000"]

4. Push scheduled time [POST]: Post the time when the user selects when to perform the structured activity.
    - URL: /dss/scheduled-time
    - Parameters:
        - ce-entry-id: identifier of the entry
        - time: the timestamp when the user will start the activity

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

3.5. Workflow V: Recommendation of non-structured coaching events

When the DSS knows there is a non-structured CE to be done by a user id in the next hours, this workflow starts.

1. New next-ce alarm [MQTT]:
   - *Topic:* 'dss/'+userId_token+'/next-ce'
   - Message: 'ready'

2. Get next CE [GET]: Get the next coaching event recommended for the user.
   - *URL:* dss/nextce/<developer_userId>
   - Output: The next coaching event recommended. Example: {"ce-id": "ce_phy_0006", "ce-entry-id": "5d6ce20c8d16932140913340"}

3. Post CE performed [POST]: When user indicates it, coach POST indicating that a CE has been done.
   - URL: /dss/ce-unstructured-performed
   - Parameters:
     - ce-entry-id: identifier of the entry

4. Review of CE alarm [MQTT]:
   - *Topic:* 'dss/'+userId_token+'/review-ces'
   - Message: JSON containing a list of the CEs that need to be reviewed

5. Coach POST the review of each CE proposed during the day

3.6. Workflow VI: Sleep monitoring algorithm

The raw data collected from the BCG sensor and listed in Figure 4, are used as input for the sleep library developed by the Murata company. Then, the following workflow in NESTORE DSS is triggered to acquire the necessary information:

1. Initialization and sensor calibration (once)
2. Raw data is acquired at 1Hz and uploaded into the NESTORE cloud in batches of 60 entries (once each minute)
3. Raw data are processed once a day at noon
4. Extracting sleep stages
5. Pre-processing of the raw data: detection of bed occupied
6. Process the bed occupied time slot by invoking the Murata library for the sleep stage identification
7. Extracting sleep variables
8. Evaluation of the sleep variables according to the formulas described in D4.1.
9. Updating user's Zivacare profile

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

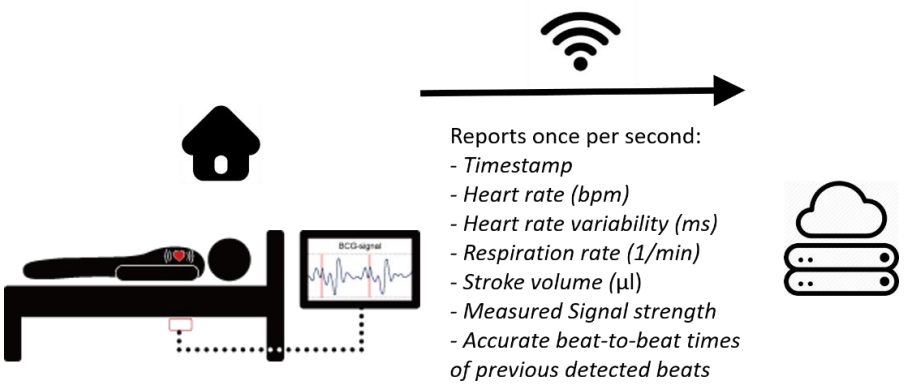

Figure 4. Data collected from the BCG sensors sent to the NESTORE cloud.

3.7. Workflow VII: Nutritional indicators extraction

The DSS also acts as a middleware between the e-Coach and the LogMeal API (see D3.3.). Some endpoints that permit to post the photos of the meals and to get information about the macronutrients and recipes are also available through the DSS.

1. Upload photo [POST]: Upload a photo to get food recognition
   - URL: /dss/upload_photo
   - *Output*: the recognized type of food (dish, ingredient or drink) and a list of recognized food containing the most probable one and other alternative dishes.

2. Modify type [POST]: Modify the type of food
   - URL: /dss/modify_type
   - *Output*: a list of recognized food given the indicated type.

3. Confirm dish [POST]: Confirm the dish/ingredient/drink recognized by the system as the one ingested by the user.
   - URL: /dss/confirm_dish
   - *Output1*: An educational message to be shown to the user. It depends on the food group of the dish.
   - *Output2*: The calculation of nutrients of the selected dish.

4. Add refused foods [POST]: Aggregate more types of food to the list of food preferences of a user
   - URL: /dss/refusedFoodsList
   - *Output*: OK/NOK in case there is any format error

5. Get refused foods [GET]: Get the list of current food preferences of a user
   - *URL*: /dss/getRefusedFoodsList
   - *Output*: The current list of refused foods

6. Get all classes [GET]: Get all the available ingredients, drinks and dishes that the system is able to recognize
   - *URL*: /dss/get_allClasses
   - *Output*: A list from LogMeal API with the classes in JSON format and contextualized depending on the language of the user.

7. Get summary of a dish [GET]: Get summary of the nutrients of a dish
   - *URL*: /dss/get_summaryDish
   - *Output*: type of meal, nutrients and timestamp of a given dish

8. History of dishes per month, week or day [GET]

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

- *URL*: /dss/summary/<day,week,month>/<userId>
- *Output*: List of dishes uploaded during the specified period

9. Get summary of a period[GET]: Summary of the nutritional information
   - *URL*: /dss/history/<start_period>/<end_period>/<userid>
   - *Output*: Summary of nutrients given a period

10. Get thresholds [GET]: Get the personalised thresholds for every nutrient given the cut-offs provided by nutritionists.
    - *URL*: /dss/thresholds
    - *Output*: Thresholds per nutrient for a given user

11. Get recipe [GET]: Get the recipe of a dish.
    - *URL*: /dss/get_recipe/<dish>
    - *Output:* A list of the ingredients and grams per ingredient that conform the recipe of the dish.

12. Get food group [GET]: Get the food group of a dish.
    - *URL:* /dss/foodGroup/<dish>
    - *Output:* the name of the food group that the dish belongs to.

3.8. Workflow VIII: Tagging system

The tagging system is in charge of sending recommendations of non-structured CEs to users; it is triggered once a week by the DSS. The internal workflow is as follows:

1. Get info from the requested user
   1.1. Get activated pathways and calculate the number of non-structured CEs that should be recommended.
   1.2. Get user tags.
2. Get all the CEs defined for a pathway.
3. Apply the constraint-based algorithm to filter out all the CEs that do not apply to the user.
4. Use the filtered list of CEs as input to get the most appropriate CEs to that specific user.
   4.1. Run the content-based algorithm.
   4.2. Run the collaborative filtering algorithm.
   4.3. Run the log filtering algorithm.
5. Mix the output of the three aforesaid algorithms to obtain a weighted list of CEs and return the CEs with a higher weight (as many as indicated in step 1.1.).
6. Write the chosen CEs in the dss.coaching_nonstructured collection.

Now the coaching scheduler module is called:

7. Read the CEs from the dss.coaching_nonstructured collection and schedule them.
8. Once it is the time of the recommendation, the coaching scheduler sends an MQTT to the Coach to announce there is a non-structured activity to recommend.

Figure 5 shows a graphic representation of the just explained workflow.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

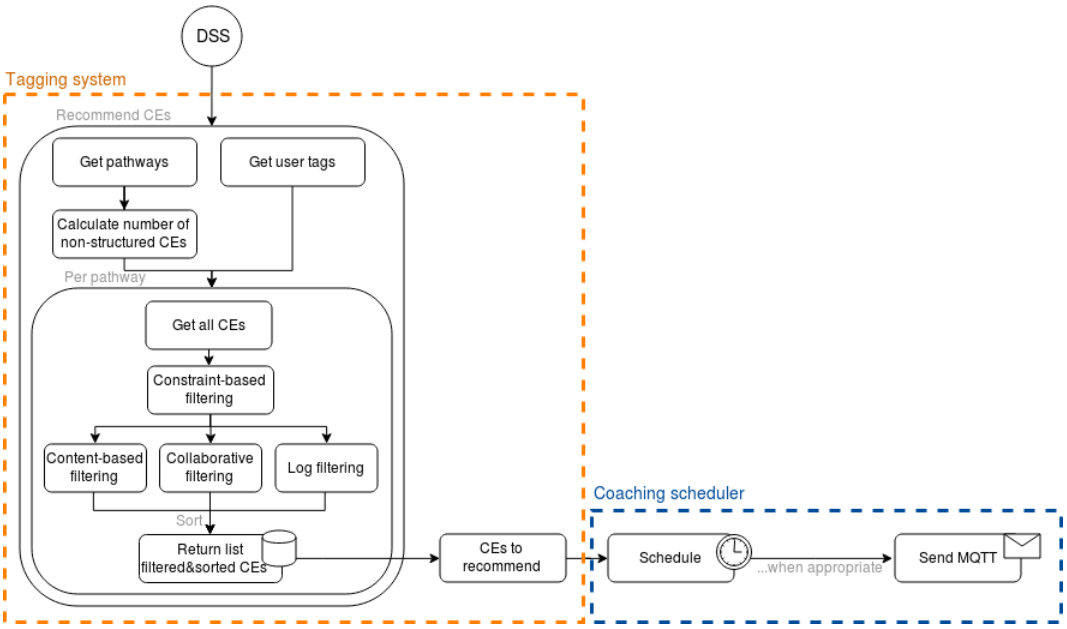

Figure 5. Graphic representation of the tagging system module.

3.9. REST APIs from other modules

The DSS connects to other databases of NESTORE located in the NESTORE cloud, such as Zivacare. All the connections the DSS does are presented below.

In order to schedule activities in users' calendar the DSS needs to know their availability.

- GET users' availability:
  - URL: https://dev.nestore-coach.eu/api/jsonws/social-portlet.calendar/get
  - Parameters:
    - startDate
    - endDate
  - Output: list of events from given startDate until given endDate. Each event is detailed with information about the attendees, the name of the event, the start and end time.

To respect the given or rejected consents, the DSS queries the user's statement.

- GET status of GDPR consent:
  - URL: https://dev.nestore-coach.eu/api/jsonws/gdpr-portlet.gdprlog/list
  - Output: list of all GDPR consents and the user response

To run all the workflows for all the users the DSS needs to know the list of all users.

- GET all users in database:
  - URL: https://developer.nestore-coach.eu/api/v3/users?offset=0&limit=100000
  - Parameters:
    - clientId
    - clientSecret
  - Output: list of all users

The DSS gets all information from the user profile to build its own version of the user profile.

- GET user profile:
  - URL: https://developer.nestore-coach.eu/api/v1/user_profile
  - Output: get user profile (name, birthday, gender, time zone, country, language)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

It is necessary to get a token per each user to query the Log Meal API.

- GET the log meal token: make a post to get data
  - URL: https://developer.nestore-coach.eu/api/v1/appserver/log_meal_token
  - Data:
    - iam_token
    - clientId

4. DSS modules

With the aim of bringing dynamism, flexibility and extensibility, the DSS has been designed and implemented as a modular system. Some of the modules provide the reasoning mechanisms. Other modules are domain-dependant and can be activated or not depending on the characteristics of the system.

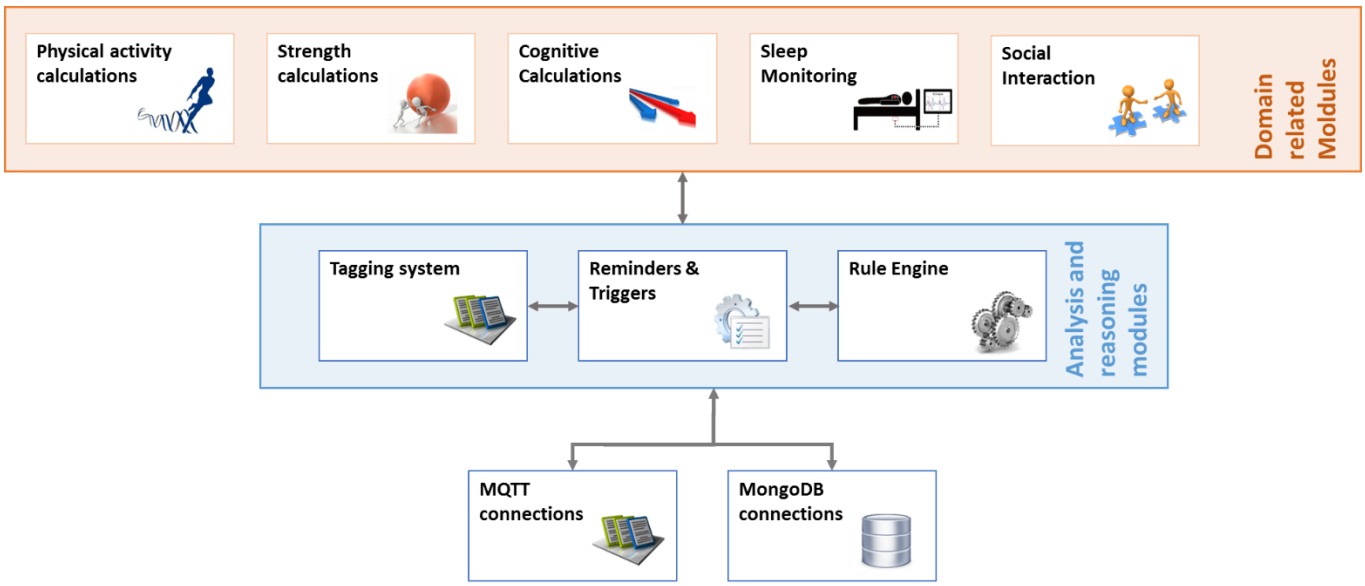

Figure 6. The DSS modules

The description of all the DSS modules shown in Figure 6 is explained extensively in the private version of this document.

Apart from the modules drawn in Figure 6, the DSS also contains the Emotive wellbeing affect API. It provides automated sensing and recognition of up to 24 emotion related affective states and moods from the chatbot/tangible coach (see WP5 deliverables). A number of longer-term emotion analytics on emotional engine detected affect (inter-user/intra-user based analytics) have been developed and are available via the wellbeing API endpoints /api/v1/wellbeing/affect_analytics/{function_name}. Given the inner nature of this dimension, it cannot be analysed on a daily basis to be aggregated over a longer period. The same goes for the sleep monitoring module, which is more a "control" dimension rather than a target domain.

In the private version of this document, there is an overview of available analytics POST methods in this API, followed with a table across subsequent pages detailing sample outputs and uses for each analytics.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643

5. Code repositories

The DSS is based on Python and Java. This decision comes from the fact that Python permits to explore the data, extract statistics and perform data analysis, while Java is more useful to perform complex calculations.

A Mongo Database was also set up to store partial calculations and DSS indicators.

The description of the code and the URLs of the different repositories is listed and described in the private version of this document.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769643