H2020-PHC-2014-single-stage
PHC-26-2014: Self management of health and disease: citizen engagement and mHealth
Research and Innovation action

Mobile Therapeutic Attention for Patients with Treatment Resistant Schizophrenia

Deliverable 3.9

m-RESIST Prototype V1

Version	1.0.0
Status	Final
Date	14/11/2017
Filename	643552_m-RESIST_20171114_D3.9_m-Resist_Prototype V1 M35 V1.0.0.docx
Type	Demonstrator
Dissemination level	Public

Project reference No.	643552
Work package	WP3: Technical Design and Development

Authors

Date	Name	Partner
Document leader	Fotis Paraskevopoulos	ICCS
Participants		
	Jesús Berdún/Elisenda Reixach	TICSALUT
	Yoram Feldman	GERTNER
	Mathias Stevens	imec
	Xavier Constant	IBM
	Anna Triantafillou, Panagiotis Kokkinakis	ATC
	Valentina Simonetti	AB.ACUS
	Maria Bulgheroni	AB.ACUS
	Brunella Marzolini	AB.ACUS
	Iluminada Corripio	IR-HSCSP
	Eva Grasa	IR-HSCSP

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643552"

Document History

Version	Author	Change	Status
0.0.1	Fotis Paraskevopoulos	Outline and first proposal on procedures	Draft
0.0.2	Fotis Paraskevopoulos	Contributions	Draft
0.0.3	Valentina Simonetti	Contributions	Draft
0.0.4	Panagiotis Kokkinakis	Contributions	Draft
0.0.5	Yoram Feldman	Contributions	Draft
0.0.6	Xavier Constant	Contributions	Draft
0.0.7	Mathias Stevens	Contributions	Draft
0.0.8	Fotis Paraskevopoulos	Review	Draft
1.0.0	Jesús Berdún/Elisenda Reixach	Final review and harmonisation	Final

Table of Contents

Table of figures

1 Introduction

1.1 Purpose of the document

The following document presents deliverable D3.9 “m-RESIST Prototype V1”. m-RESIST is a mobile ICT solution addressed to empower patients with treatment-resistant schizophrenia (TRS), which encourages the patients and caregivers to actively participate in the therapeutic process and enables them to self-manage their condition. This deliverable is classified as a “demonstrator”, and presents the final version of the m-Resist system after piloting activities performed with TRS patients in WP5. The following document presents an overview of the system, describing the high level architecture and the different functionalities that it offers.

This deliverable does not provide an architectural description of each component, as this description has already been provided in previous WP3 deliverables. Rather, it presents the work done in terms of development, and the state of the final version of the system. Each component of the system is described briefly, and discussion is focused on issues and iteration which were done to the system during the piloting activities.

1.2 Relation with other deliverables

D3.9 is the “demonstrator” deliverable of the m-RESIST Prototype V1, the resulting system from the development carried out during the piloting of the system during WP5 activities. During this period and based on WP5 results, all components where improved in order to cater for edge cases, and scenarios presented which were not thought of initially. In this document we have included excerpts from previous technical deliverables (D3.1-8) providing a more detail overview of the architecture and technical components of the system.

2 m-RESIST Overview

2.1 m-RESIST tools

The m-RESIST solution is an innovative disease management system based in mobile technologies, addressed to empower patients with treatment-resistant schizophrenia, which will engage them, together with the professionals and caregivers, in active participation in therapeutic processes, enabling them to better self-manage their condition.

Figure 1: m-RESIST system tools

m-RESIST provides is made up of three main parts **a smartphone connected to a smartwatch** for patients and caregivers; a **web-based dashboard** for follow-up and monitoring for clinicians and a **back-end system for managing patient data**, interventions and communication between patients, caregivers and clinicians (m-RESIST back-end system).

Figure 2 provides a high level overview of the individual components that make up the entirety of m-Resist, and how the information flowing between them.

- Sensor module: collects sensor data from the smartphone and the wearable (smartwatch), processes it and stores it in the m-RESIST Information Repository, providing the needed information to trigger therapeutic interventions and to visualize and monitor the patient status through the dashboard.
- m-RESIST Information Repository: acts as the central data repository for the entire system, allowing other modules to store and retrieve any kind of information in a flexible format.
- Clinical Decision Support System (CDSS): rule engine that uses decision algorithms under given clinical conditions collected either automatically (by sensors) or manually (user interaction). Patient data is interpreted according to pre-determined rules, triggering series of actions in three therapeutic interventions, namely, symptom management, treatment adherence and healthy lifestyle.

- Recommender: using information stored in the system, it provides a set of recommendations addressed to the patient, caregiver or professional, according to therapeutics interventions. Recommendations also consider previous user feedback and outcomes from the predictive module.
- Predictive module: provides predictions for different measures of a patient's status based on the information currently available in the system about the patient. A set of models have been developed for the predicting variable "Patient Evolution".
- Integration layer: it conforms the internal structure of m-RESIST back-end that results in a system flexible enough to support the coupling and effective communication between various components.

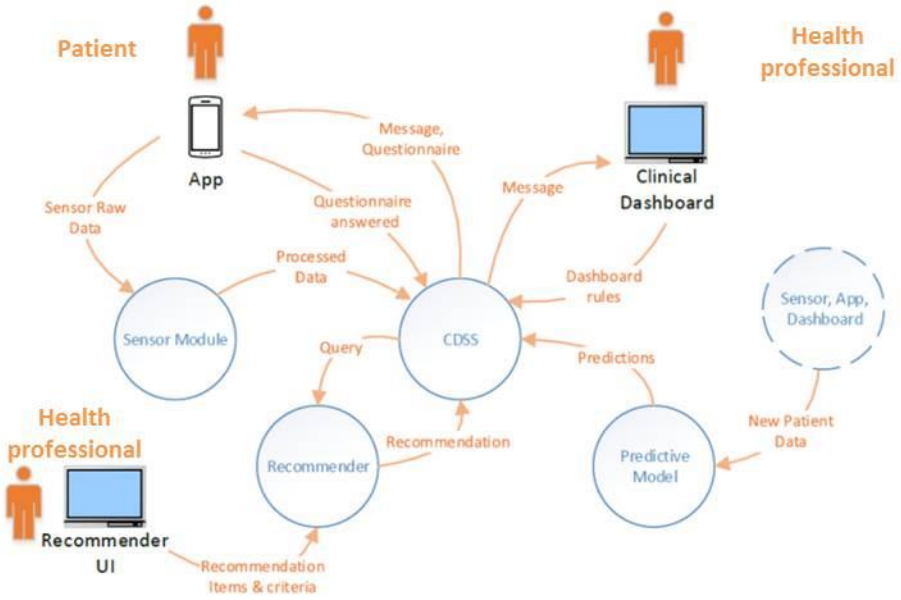

Figure 2: m-RESIST high-level functional architecture

The flow of information between modules is set up so that the system can cater to specific mHealth intervention modules which are specially designed for patients with treatment-resistant schizophrenia. Interventions are split into two categories ***Basal modules*** (Treatment Adherence, Healthy Lifestyle, and Symptom Management-CBTp) that are oriented to develop abilities in the patient; and ***Risk modules*** (Symptom Management-Risk).

2.2 Pilot Setup & Review Process

Pilot participants were selected from the University of Semmelweis (Semmelweis), the Sheba Medical Center – Psychiatric Division (Tel Aviv), and the outpatient service of Department of Psychiatry, Hospital de la Santa Creu i Sant Pau (Barcelona). All participants were fully informed about the nature of the

study (aims, methodology) and the system tested (data protection, expected risks/benefits, possible adverse events) and signed the informed consent for their participation in the study.

Recruitment activities started for each clinical partner in March 2017 and finalized on June 30th. In May 2017, m-Resist tools (Smartphone and Smartwatch) were handled to patients recruited to start collecting data for the baseline and to allow users to familiarize with the system. Some caregivers also installed the m-Resist app and used it during the pilots. Once patients and caregivers started using the system under real life conditions, clinicians of each site followed up data collected from the patients and interacted with them through the m-RESIST dashboard. M-RESIST partners also tested the system in order to detect bugs of the system.

All technical issues encountered during the pilots were collected in a document shared between clinical and technical partners. The issues were reported indicating: issue number, username (who experienced the issue), date, m-Resist module (dashboard, app, messaging system, CDSS, recommender, sensors, Smartwatch or appointment) and the problem description (steps to reproduce, expected behavior, actual behavior, app version) and a section for comments between clinical and technical partners and a section for links to pictures/videos. The list of the issues was revised weekly during WP5 tele-conference between clinical and technical partners to update/clarify their status. A total of 120 issues were reported during pilot study and were solved. On the other side, during 6th GA meeting in October 2017, a list of improvements was also agreed between technical and clinical partners to enhance m-Resist system before finalizing the pilots.

3 m-RESIST Components

The purpose of this chapter is to describe the technical aspects of changes and improvements made to the components to cater to issues, or edge cases discovered during the pilot activities in WP5.

3.1 Dashboard

The Dashboard is a set of UI components that display patient statistics, profiles, predictions, recommendations, appointments and messages. It allows the clinicians to see an overview or details of their assigned patients, activate interventions, edit patients' profiles, send messages, receive notifications, manage appointments, share educational content, configure the recommendation engine and view help pages.

During the pilot activities, the following improvements have taken place:

- **New roles and permissions**

  Added new field to patient profile, which allows case managers to assign alternative clinicians to their patients. Alternative clinicians can view and check the status of their patients, but they have reduced permissions, since they aren't allowed to update their profiles or activate interventions.

- **Multilingualism**

  Besides English, the dashboard is translated in Hungarian, Spanish, Catalan and Hebrew. In case of Hebrew, the RTL (Right to Left) direction is supported.

- **New graphs**

  Embedded new graphs that visualize more statistics and allow clinicians have a more detailed image of their patients' status. These graphs are:
  - HR (Heart Rate) graph, which shows (in line chart) the patient's heart rate during the last six hours
  - Activity graph, that visualizes (in both line and pie charts) if the patient is outside or at home
  - Risk Level graph, which added in the symptoms management intervention and shows if patient is at risk or not.

- **New functionalities and components**

  Displayed predictive statistics in the overview window, like improvement, worsening, stability, resistance and low-insight. Also, added the list of caregivers inside the users-window and allowed clinicians to edit their patients' profiles. Finally, displayed baseline assessment and added date-time ranges on graphs.

- **Educational Content**

  Clinicians are allowed to create or edit educational content and share it with their colleagues. The content is available in English, Hungarian, Spanish, Catalan or Hebrew language.

- **Recommender**

The Recommender is integrated in the dashboard, which allows clinicians to configure the recommendation engine.

- **Help Pages**

Implemented help pages, that allow clinicians to view a full list of functionalities supported by the dashboard, and details on how to use them.

3.2 Mobile Application

The mobile application (app) serves as the primary interface to the m-RESIST system for patients and their caregivers. The app allows them to communicate with the clinicians involved with their treatment, respond to requests (short, on-screen questionnaires) sent out by the system, receive recommendations generated by the system, view upcoming appointments and consult educational resources. In the case of patients the app also acts as a relay which forwards sensor data collected by the patient's smartwatch to the m-RESIST back-end (specifically the Sensor data module and the mIR).

Based on feedback received during the pilot activities several small and larger improvements were made to the app. The most noteworthy ones are:

- Support for always available ("self-assigned") questionnaires;
- Improved support for caregivers;
- Hiding of duplicate unanswered questionnaires;
- Added notifications (with sound) for incoming questionnaires;
- Various cosmetic improvements to the UI;
- Various bug fixes, including mitigation of some common crashes;
- New feedback screen with charts that visualize the patient's physical activity (step counter), time spent at home vs. elsewhere, and engagement with the system

3.3 m-RESIST Information Repository (mIR)

The m-RESIST Information Repository (mIR) acts as the central data repository for the entire system. mIR allows external modules to store and retrieve any kind of information and does not impose a particular schema. mIR exposes an HTTP Restful interface, which accepts input and provides responses in the JSON[1] format. HTTP verbs such as PUT / GET / POST / DELETE are mapped to the corresponding data storage actions Create / Retrieve / Update / Delete (CRUD).

mIR is schema less, which means that modules can store any information as long as they are in the JSON format. Each document stored in mIR is of a certain *type*, which is stored in a specific *index.* The *type* and *index* are the basis of the URLs which are used to perform actions in mIR. mIR's API is documented here **http://docs.mirmresist.apiary.io/**

During the piloting activities, mIR was deployed to a production environment which adhered to the following requirement:

- **Security –** We made sure mIR was only accessible through the ESB . Calls from the ESB towards mIR was done through the local network, using a JWT token to make sure that calls were being initiated from a trusted source
- **Redundancy –** We implemented a backup and restore system which maintains data backups for the last 7 days, the last 4 weeks and the last 12 months. This runs nightly and copies of the data where stored encrypted and password protected in a separate network location
- **Scalability –** We proxied mIR behind an NGINX[2] server, in order to allow us to scale the Application (Tomcat) layer of mIR

Furthermore we enhanced the query parameters of several endpoints to allow for sorting and date paginations, given that during the piloting activities, the UI needed a more granular control of the amount of data being fetched and presented to the user.

3.4 ESB / Message Bus

The integration layer of m-RESIST back-end is consisted of 2 core components: the WSO2 ESB and the WSO2 Message Bus. All the functionality that is exposed by the various back-end components, the mobile applications as well as the dashboard passes through the ESB promoting thus an event driven architecture. By adopting this kind of architecture design pattern we make sure that the resulting system is flexible enough to support the loose coupling of the various components.

3.4.1 ESB layer

The ESB layer is composed by a number of proxy services that encapsulate the actions that have to be performed transparently each time the m-RESIST service APIs are accessed. A proxy service is a virtual service that receives messages and applies various transformations and other functions before dispatching them to the target API endpoint. The building blocks of an ESB proxy service are called mediators. Custom ESB mediators have been developed and integrated to satisfy the needs of the back-end components.

During the pilot activities, the ESB has been switched to production mode and the embedded H2 database has been replaced by postgres, which can handle a huge number of requests required while on a production mode.

3.4.2 Mail API

The Mail API exposes a set a of web services that allow clinicians, patients and system to send messages to users. All API methods exposed perform a JWT token validation permitting access to only authenticated and authorized users.

[2] https://www.nginx.com/resources/wiki/

During the pilot activities, some changes made on API are:

- The API sends push messages (using the Notification API) to mail receivers in order to inform them that they have got got a new message to their inbox
- It supports charset UTF-8, in order to send correctly messages with Hungarian, Spanish, Catalan and Hebrew texts.

3.4.3 Calendar API

The Calendar API offers RESTful web services that allow clinicians to create, edit and delete appointments. In other words, it provides a set of CRUD operations on the appointment resource as well as a search API to effectively retrieve calendar appointments by date. All API methods exposed perform a JWT token validation permitting access to only authenticated and authorized users.

The API has been enhanced by implementing new methods based on the needs of the integrated system. Some of the functionality that was added includes:

- Attendance flag, which defines if patient attended the appointment that had with his/her clinician or not
- Statistics service, that provides the number of appointments missed by a patient
- The API sends message to the involved users when an appointment is updated by the clinician, and inform them that there was an update to the appointment

3.4.4 Notification API

The Notification API provides functionalities that allow both users and components to send instant push notifications to subscribed end users. Technically, it sends push notifications using the Firebase Cloud Messaging service. The end user has to subscribe to the system in order to receive a notification. The subscription is feasible from both web and android platforms, and the subscribed user could see a popup for every received notification.

During the pilot activities, a few bugs were fixed and the charset UTF-8 was supported, in order to send correctly notifications with Hungarian, Spanish, Catalan and Hebrew texts.

3.5 Sensor Module

The Sensor Module works as an autonomous Java application that has been deployed on the m-RESIST windows server given by ATC. It performs the digital processing over the raw data time series recorded by the App and computes the processed values needed by the CDSS and the dashboard (listed in D3.4) for visualization purposes. ESB messaging allows the synchronization between Sensor Module, App, and CDSS.

Sensor module functionalities did not change from the ones listed in D3.8 and are:

- Receive ESB messages from the topic ***mresist.sensor.raw.all*** that are generated every time raw data acquired in a certain time interval are stored in mIR.

- Retrieve the corresponding **patientId** and **endTime** (unix milliseconds of the last record) from the "_link" in the message received
- Retrieve from mIR home coordinates (latitude and longitude) input in the dashboard in patient registration phase.
- Check whether it is the first time for that specific **patientId** and, if that is the case, generate a record for it. If the **patientId** is within the patientIds already processed the software updates the cumulative values for that patientId for **lowActivityTime, remTime, nonRemTime, timeSpentOutside** and updtes the **startTime** and other training parameters.
- Perform a date range query by patientId in order to retrieve the raw sensor values.
- Parse the returned JSON for each sensor and store the values in specific data structures (see D3.5). Due to some repeating values in raw sensor data from the App, the Sensor Module performs a control over the timestamp of each parsed time series, in order to reduce computational time and occupied memory.
- Process the raw sensor values using the algorithms reported in D3.5 and compute: **timeSpentOutside** (seconds spent outside home for that time range), **timeNonRem** (seconds of non rem sleep for that time range), **timeRem** (seconds of rem sleep for that time range), **timeLowActivity** (seconds spent in low activity for that time range), **location** (array with 0: outside, 1:at home), **distance** (array of distances in meters), **activityAmplitude** (array of accelerometer amplitude values), **sleep** (array with 0:non rem sleep, 1: rem sleep, 2: awake).

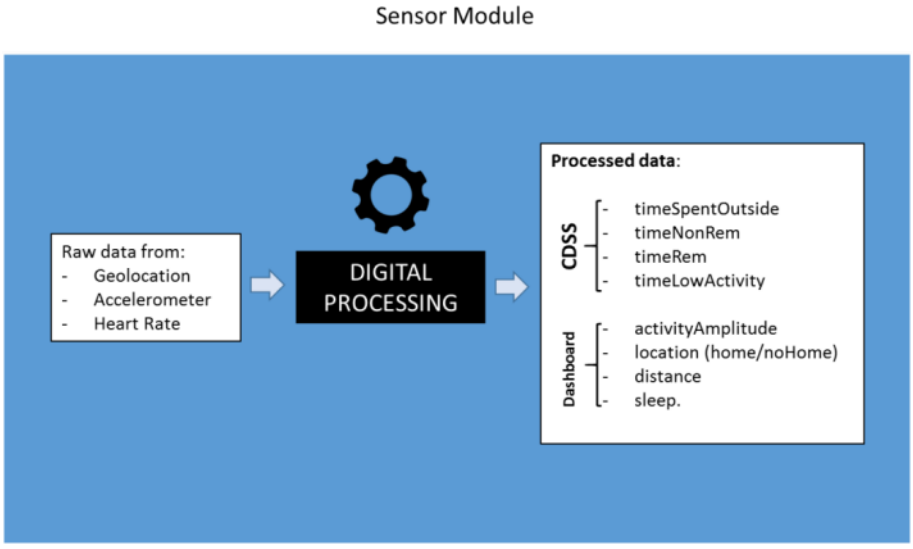

Figure 3 Sensor Module

- Update the cumulative values for **lowActivityTime, remTime, nonRemTime, timeSpentOutside.**

- Put the cumulative value for **lowActivityTime, remTime, nonRemTime, timeSpentOutside,** publishing to the topic ***mresist.sensor.processed.***
- Put in mIR the time series for **location, distance, activityAmplitude, sleep**

Robustness of the processing algorithms has been tested and monitored during the first months of trials and the evaluation of real data also helped in making slight changes and add some controls, especially for the sleep evaluation, that, for some users, showed spurious values for the classification.

The amount of data collected allowed the beginning of the computation of daily statistical indexes for each user over the trial period. This process will continue until the end of the trials and will allow the evaluation of data integrity, along with possible correlation between sensors parameters and clinical outcomes. In depth study of sensor parameters will evaluate the use of these parameters as inputs of the predictive model.

3.6 Predictive Modeling

The predictive module provides predictions for different measures of a patient's status based on the information currently available from the patient. Before starting the pilot the module contained a set of predictive models that had been pre-trained with patient data provided by the clinical partners (a total of 450 patients).

The variables available as input for the models were the following:

- Gender	- Level of education	- Marital status
- Living environment	- Employment status	- Number of relapses
- PANSS (P, N, G)		

The models received these variables as input for each patient, and output probabilities for each of the different possible outcomes of a list of target variables defined by the clinicians. A set of models were developed and deployed for the following target variables:

Target variables	Possible values	Description
Evolution	Improvement / Stable / Worsening	Describes the change in PANSST scale of a patient with respect to basal.
Low insight	Yes / No	Describes if the patient presents low insight.

Resistance	Yes / No	Describes if the patient is resistant.

Based on the available input variables, the models predicted the patient's probability of improvement, worsening, remaining stable, presenting resistance and presenting low insight in the following days. These values were made available to the rest of the modules by storing them in mIR. When any of the input variables of a patient changed, these probabilities were computed again and posted to mIR through the ESB.

These are some statistics and outcomes about predictions made during the pilot:

- A total of 32 patients had all the input data and received predictions.
- Most patients had only one prediction (demographic data and PANSS did not change during the trials).
- Predicted probabilities for improvement and worsening were low.
- Predicted resistance and low insight probabilities take a wide range of values.

After the pilot, we are improving existing predictive models to include data from sensors and responses from questionnaires, so the prediction of evolution (worsening, improvement or remaining stable) will be recalculated on a daily basis based on sensor inputs.

3.7 Clinical Decision Support System (CDSS)

The functionality of the CDSS is based on the workflows developed by the clinical team, reflecting the process of interaction between the system and its users, in order to establish new healthcare pathways. The CDSS activation is triggered by an event (i.e., arrival of sensor data), that is interpreted in a context of additional information that exists regarding a specific patient, and a series of pre-defined conditions and actions.

The development process throughout the piloting activities period can be summarized as following:

1. Improvement of Baseline Assessment calculation

The calculation of the baseline assessment was changed according to the following definitions:

Baseline period is the amount of seconds the CDSS waits from first receive of the sensor data (steps, time outside, low activity time, REM time, non-REM time) of a given user until start measuring changes from baseline data. The timing of calculating the baseline value is when the first received measurement after the **baseline period** elapsed. Then the CDSS calculates:

Units (current) – Units (previous)

Elapsed time (current) – Elapsed time (previous)

The above gives the average **units per day** within the measured period. The CDSS calculates each unit average for the **baseline period** and then for each **observation period**.

When an **observation period** average goes 30% beyond or below the **baseline period** average, the CDSS starts an intervention, based on the deviated unit and the magnitude of deviation.

The baseline period default duration was set as 15 days and observation period was set as 3 days. These periods can be edited manually by clinicians through the dashboard, in order to allow personalization of assessment period per patient.

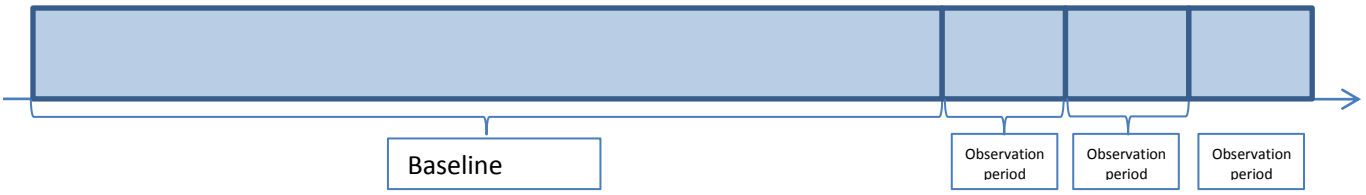

Figure 4 CDSS Process

2. Continuation of implementation of questionnaires & notifications.

The CDSS includes 2 types of interactions with the user: (1) Questionnaires – a series of question requiring user's response, which results are calculated, interpreted and communicated to other system modules, such as the Recommender. (2) Notifications – messages that are sent to the user according to interpretation of the system input (i.e., sensor data) and the user's response to the questionnaire (i. e., reply or expiry). Each questionnaire and notification has been translated to Spanish, Catalan, Hungarian and Hebrew and implemented in the system according to workflows defined by the clinical team. The translations were checked by the representatives of the clinical team and corrected in case of mistakes.

3. Testing activities

Following the implementation of the CDSS questionnaires and notifications, testing activities were performed throughout the pilot period. Test profiles for patient, caregiver and clinician were created, and each CDSS flow related to a certain starting point (questionnaire with specific answers, events such as two consecutive missed appointments, clinically significant deviation from baseline and observation averages) was checked according to the design specifications. Furthermore, the three clinical sited systematically reported issues and bugs that emerged from the users' interaction with the system, and these issues have been checked and fixed within the CDSS component or through collaboration with other system modules, such as the Recommender, Dashboard and App. Throughout the pilot, several improvements of the system were proposed by the clinicians, these proposals were prioritized according to potential for usability improvement, and implemented.

3.8 Health Recommendation Module

In order to provide recommendations the system follows this sequence

a) How to provide recommendations
b) When to provide recommendations
c) **What to provide as recommendations**

The Health Recommendation Module (recommender) is responsible for **what** to provide.

Due to privacy restrictions concerning historical data we could not create recommendations implicitly, therefore our implementation provides:

- An interface which allows clinical experts to populate the system with recommendations items.
- The recommendation engine which contains the necessary algorithms to provide recommendations
- A feedback mechanism so that recommendations can be personalized based on the patient recommendation selection.

The recommender provides a user interface which allows the clinical experts provide and classify recommendation through a dynamic set of classification criteria.

Recommendation UI

The recommendation UI provides the following functionality:

- Manage the classification criteria which will be used to classify recommendations, and consequently used by the CDSS to perform recommendation queries. There are two types provided, range criteria, and boolean criteria.
- Manage recommendation categories which are used only throughout the interface and allows the clinical experts to group and organize recommendations.
- Manage recommendation item groups

  Clinical experts are able to add, remove and update recommendation items. They use classification criteria to classify recommendations, by setting specific values to the range and boolean criteria
- Provide language specific recommendations.

During the pilot activities the following improvements were performed:

- We provided a tighter integration with the m-Resist Dashboard as a result of the production deployment. To achieve this, we needed to re-architecture the UI in order to provide the entire functionality through a single HTTP context **/ui**
- Even though the recommendation engine provides language specific recommendations, these do not act as a one to one direct translation between the languages. The assumption was that each language environment may use their own lingo, or classify recommendations in their own way. However it turned out that a correlation between some (or all) recommendation is necessary between languages. As such we introduced a recommendation “code” in order to allow the CDSS to correlate several languages into a single recommendation
- One aspect of the recommendation which was not implemented is the type of recipient of the recommendations. In prototype V0, the recommendation did not consider whether the recipient was

a patient, a care giver or a clinician. The recommendation engine did provide the available functionality to achieve this through Boolean classification criteria, however due to the weighting aspect of the algorithm, it did not act as expected. As such we introduced three explicit Boolean criteria which pre-filters recommendations before the recommendation for a patient a provided. This way the CDSS is able to explicitly determine the type of recipient it is querying recommendations for. This option can be omitted, which will just revert to the original algorithm implementation.

4 Conclusions

This document provides a general description of the “m-RESIST Prototype V1”, the improved and final version of the m-RESIST system, and as a result of pilot testing activities (in WP5) under real life conditions with TRS patients, their caregivers and the associated clinicians. This deliverable complements D3.8: m-RESIST Prototype V0. The refining of m-RESIST system has been based in the identification of technical malfunctions of the system experienced over the pilot period by patients, caregivers, clinicians and testing partners and taking into account unforeseen needs of clinical partners during the testing of the system. All technical issues detected were solved by technical partners leading to new versions of the App and dashboard.

m-RESIST Prototype V1 is a stable refined prototype which could be deployed and consumed by institutions outside the consortium.