POWER2DM

"Predictive model-based decision support for diabetes patient empowerment"

Research and Innovation Project
PHC 28 – 2015: Self-management of health and disease and decision support systems based on predictive computer modelling used by the patient him or herself

Deliverable 4.7

D4.4.2 Basic Data Processing Framework

Workpackage:	WP4
Task:	Task 4.4
Due Date:	30 April 2018 (M28)
Actual Submission Date:	31 January 2019 (M36)
Last Amendment:	31 January 2019 (M36)
Project Dates:	Project Start Date: February 01, 2016 Project End Date: July 31, 2019 Project Duration: 54 months
Deliverable Leader:	PD (Prime Data)

Project co-funded by the European Commission within H2020 Programme (20015-2016)		
Dissemination Level		
PU	Public	X
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)	

Document History:

Version	Date	Changes	From	Review
V0.1	4.1.2019	Initial version	PD	
V1.0	10.1.2019	Compilation of deliverable for submission	PD	TNO

Contributors (Benef.) Nielsen Vermij (PD)
Michel Sluis (PD)

Responsible Author Nielsen Vermij **Email** n.vermij@primevision.com

EXECUTIVE SUMMARY

Processed data is validated in PDS as main database for POWER2DM but also on the client side, as this makes the whole configuration more responsive. When a user enters a wrong value for a specific entry in one of the POWER2DM applications, the applications immediately notify the user without preparing the whole record, sending it to PDS and getting an error saying that value is not in valid range.

This deliverable describes for each POWER2DM component how data validation and processing are implemented in such a way that data validity and quality are guaranteed without losing sight of user behaviour.

POWER2DM Consortium Partners

Abbv	Participant Organization Name	Country
TNO	Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek	Netherlands
IDK	Institute of Diabetes “Gerhardt Katsch” Karlsburg	Germany
SRDC	SRDC Yazilim Arastirma ve Gelistirme ve Danismanlik Ticaret Limited Sirketi	Turkey
LUMC	Leiden University Medical Center	Netherlands
SAS	SAS Servicio Andaluz de Salud	Spain
SRFG	Salzburg Research Forschungs Gesellschaft	Austria
PD	PrimeData	Netherlands
iHealth	iHealthLabs Europe	France

OPEN ISSUES

No:	Date	Issue	Resolved
		no known issues	

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Purpose and scope

Data measured by devices or gathered through clinical questionnaires or entered by patient and clinicians needs to be validated for correctness, accuracy and precision to ensure correct action plans for the patient as well as reliable modelling results.

Several data quality checks and detections are being implemented in the POWER2DM configuration, such as anomaly detection, outlier detection and min-max analysis, in order to overcome first order statistical issues within the datasets.

Moreover, basic data processing and data validation will take place in order to guarantee data quality of the available data.

The purpose of this document is to describe the implementation choices which have been made to guarantee quality of the measured data.

2 REFERENCE DOCUMENTS

The following documents were used or referenced in the development of this document:

- D4.5 Data Quality Analysis Framework I

2.1 Definitions and Acronyms

Table 1 List of Abbreviations and Acronyms

Abbreviation/ Acronym	DEFINITION
API	Application Programming Interface
ARC	Audit Repository Client component
FHIR	HL7 Fast Healthcare Interoperability Exchange
PDS	POWER2DM Personal Data Store
SDMA	POWER2DM Shared Decision Making Application

3 VALIDATION AND PROCESSING

3.1 Overview

In the project proposal time, the idea of having basic data processing component was about the quality of measurements coming from medical devices and patients. After new insights the consortium decided to store measurements as FHIR Observation records in PDS as central database. Implementation is based on a secure FHIR repository product Onfhir.io which opens the FHIR defined APIs (FHIR CRUD operations) as restful services for all resource types (observations, medications, etc) as a generic mechanism. Onfhir.io is using the *validation* library provided by HL7 FHIR for DSTU2 version (which was the one only available during the project start) which validates the records according to the schema specified for different resource types.

Nevertheless, validation is also implemented on the client side, as this makes the whole configuration more responsive. When a user enters a wrong value for a specific measurement, applications can immediately notify the user without preparing the whole record, sending it to PDS and getting an error saying that value is not in valid range. Thus, current POWER2DM applications SDMA, SMSS Mobile and SMSS Web do not handle PDS errors in such a way. By default the applications use client side data validation and if there is an error while submitting the data to PDS, either it is a bug of client application or a transient error (PDS is down temporarily).

4 DATA VALIDATION IN PDS

4.1 Validation mechanisms

In PDS as central data component for POWER2DM the followings are the validation mechanisms used to validate the input from POWER2DM applications to the FHIR CRUD and search services.

4.2 Validation in FHIR Create and Update services

Data content given in these services in related FHIR resource format is validated using the FHIR Validator (https://www.hl7.org/fhir/DSTU2/downloads.html) Java library. The library validates the content according to the special schema definition which is called StructureDefinition in FHIR. PDS configures itself using the StructureDefinition[1] files for each resource type and these are provided to FHIR Validator so it can validate against POWER2DM data model restrictions. A part of such definition which provides the restrictions for an element is given in Table 2. The library for the DSTU2 version (which is used in POWER2DM) is not supporting validations for defined 'Slicing' components[2] which we are used in some POWER2DM resource models. Therefore, library is extended to cover these also. The module performs the following validations;

- Validity of given JSON content
- Validity of the cardinalities for each element
- Validity of types for each element value (compliance to FHIR type restrictions)
- Validity of codes used as value of elements (for FHIR code type elements)
- Validity of slicing restrictions for component elements

In addition to this, PDS performs the following validations on service API level;

- Check if FHIR interaction (update or create) is defined for the given resource type
- Check if resource type given in the content matches the resource type indicated in service URL
- For FHIR Update, check if resource id given in content matches the id indicated in service URL

Table 2 A part of StructureDefinition of Observation resource type – Definition of an element Observation.categoty


```
{
    "path": "Observation.category",
    "short": "Classification of  type of observation",
    "definition": "A code that classifies the general type of observation being made.  This is used  for searching, sorting and display purposes.",
    "comments": "The level of granularity is defined by the category concepts in the value set.   More fine-grained filtering can be performed using the metadata and/or terminology hierarchy in Observation.code.",
    "min": 0,
    "max": "1",
    "type": [
      {
        "code": "CodeableConcept"
      }
    ],
    "binding": {
      "strength": "example",
      "description": "Codes for high level observation categories .",
```


[1] https://www.hl7.org/fhir/DSTU2/structuredefinition.html

[2] https://www.hl7.org/fhir/DSTU2/profiling.html

```
        "valueSetReference": {
          "reference": "http://hl7.org/fhir/ValueSet/observation-category"
        }
      },
      "mapping": [
        {
          "identity": "rim",
          "map": ".outboundRelationship[typeCode=\"COMP].target[classCode=\"LIST\", moodCode=\"EVN\"].code"
        },
        {
          "identity": "w5",
          "map": "class"
        }
      ]
    }
```


4.3 Validation in FHIR Search service

For FHIR search service, given search parameters are validated as follows;

- Check whether the given parameter is supported for the given resource type as indicated in the FHIR Conformance[3] file which PDS configures itself with. A part of the file is given in Table 3, which shows the search parameters supported for Condition.
- Check whether the parameter value, prefix and suffixes are given according to restrictions based on parameter type which are described in SearchParameter[4] definitions. Table 4 provides an example definition. PDS configures itself with the base SearchParameter definitions supplied by FHIR and also further parameters defined specifically for POWER2DM.

Table 3 Part of POWER2DM Conformance statement – Defining restrictions on API for Condition resource type


```
{
        "type": "Condition",
        "interaction": [
          { "code": "read" },
          { "code": "create"},
          {"code": "update"},
          {"code": "delete"},
          {"code": "search-type"},
          {"code": "history-instance"}
        ],
        "searchInclude": ["Condition.asserter", "Condition.related"],
        "searchParam": [
          {
            "name": "patient",
            "type": "reference",
            "target": [
              "Patient"
            ]
          },
          {
```


[3] https://www.hl7.org/fhir/DSTU2/conformance.html

[4] https://www.hl7.org/fhir/DSTU2/searchparameter.html

```
          "name": "code",
          "type": "token"
        },
        {
          "name": "category",
          "type": "token"
        },
        {
          "name": "clinicalstatus",
          "type": "token"
        },
        {
          "name": "related",
          "definition": "http://www.power2dm.eu/pds/SearchParameter/Barrier-related",
          "type": "reference",
          "target": [
            "Condition"
          ]
        },
        {
          "name": "asserter",
          "type": "reference",
          "target": [
            "Patient",
            "Practitioner"
          ]
        }
      ]
    }
```

Table 4 POWER2DM Search Parameter Definition for Goal.addresses

```
{
  "resourceType": "SearchParameter",
  "id" : "Goal-addresses",
  "url": "http://www.power2dm.eu/pds/SearchParameter/Goal-addresses",
  "name": "addresses",
  "status": "active",
  "publisher": "SRDC Corp.",
  "code": "addresses",
  "base": "Goal",
  "type": "reference",
  "description": "The Personal Value or Problem record that this Goal is addresing",
  "xpath": "f:Goal/f:addresses",
  "xpathUsage": "normal",
  "target": [
    "Condition"
  ]
}
```

4.4 Validation in FHIR Operations

For all FHIR Operations implemented specific to POWER2DM, a OperationDefinition[5] file is prepared and provided as configuration to PDS. Table 5 provides an example definition for POWER2DM Upload FSL Service. This file provides the definition for the operation; defines the input and output parameters,

[5] https://www.hl7.org/fhir/DSTU2/operationdefinition.html

their FHIR types, cardinalities, and possible value set. PDS validates the given input for each operation based on the corresponding operation definition;

- Check if operation is defined on the resource type and/or instance
- Validate if all required parameters are given and validate the cardinality of each parameter
- Validate if given parameter values are matching the type restrictions
- If parameter is coded, validate if its value is within the defined value set
- If operation is on a resource instance (e.g. on a specific Patient), check if that instance exists in PDS

Table 5 OperationDefinition for POWER2DM Upload FSL Results Service


```
{
  "resourceType": "OperationDefinition",
  "id": "upload-fsl-results",
  "url": "http://www.power2dm.eu/pds/OperationDefinition/upload-fsl-results",
  "version": "1.0",
  "name": "srdc.p2dm.pds.operations.FSLResultsUploadHandler",
  "status": "final",
  "kind": "operation",
  "publisher": "SRDC Corp.",
  "contact": [
    {
      "name": "System Administrator",
      "telecom": [
        {
          "system": "email",
          "value": "tuncay@srdc.com.tr"
        }
      ]
    }
  ],
  "date": "2017-10-17",
  "description": "POWER2DM Operation for uploading FreeStyle Libre's FSL results by mapping it to POWER2DM FHIR based model",
  "code": "upload-fsl-results",
  "system": false,
  "type": [
    "Patient"
  ],
  "instance": true,
  "parameter": [
    {
      "name": "resource",
      "use": "in",
      "min": 1,
      "max": "1",
      "documentation": "Excel file encoded in FHIR Binary resource",
      "type": "Binary"
    },
    {
      "name": "return",
      "use": "out",
      "min": 1,
      "max": "1",
      "documentation": "The created observations in Bundle",
      "type": "Bundle"
    }
  ]
}
```

As defined in FHIR, in case of an error, PDS return the validation results in a OperationOutcome resource indicating the element and description of error as shown in Table 6.

Table 6 Example OperationOutcome with validation result


```
{
  "resourceType": "OperationOutcome",
  "id": "validationfail",
  "issue": [
    {
      "severity": "error",
      "code": "structure",
      "details": {
        "text": "Error parsing resource JSON (Unknown Content \"label\""
      },
      "location": [
        "/f:Patient/f:identifier"
      ]
    }
  ]
}
```

5 DATA VALIDATION IN DCC KADIS® SERVICES

5.1 Introduction

The DCC KADIS® Services are embedded into the TeleDIAB® eHealth system with general access to the data model, program logic moduls and the KADIS® engine, which is a highspeed calculator for simulation as well as model identification (KADIS® setup) procedures. Based on this complexity and a performant data base, the DCC KADIS® Services will be provided as RESTful Web-Service. For the POWER2DM Software system (Prediction Service, PDS) the various DCC KADIS® Services could be easy to use by requests using the HTTPS protocol.

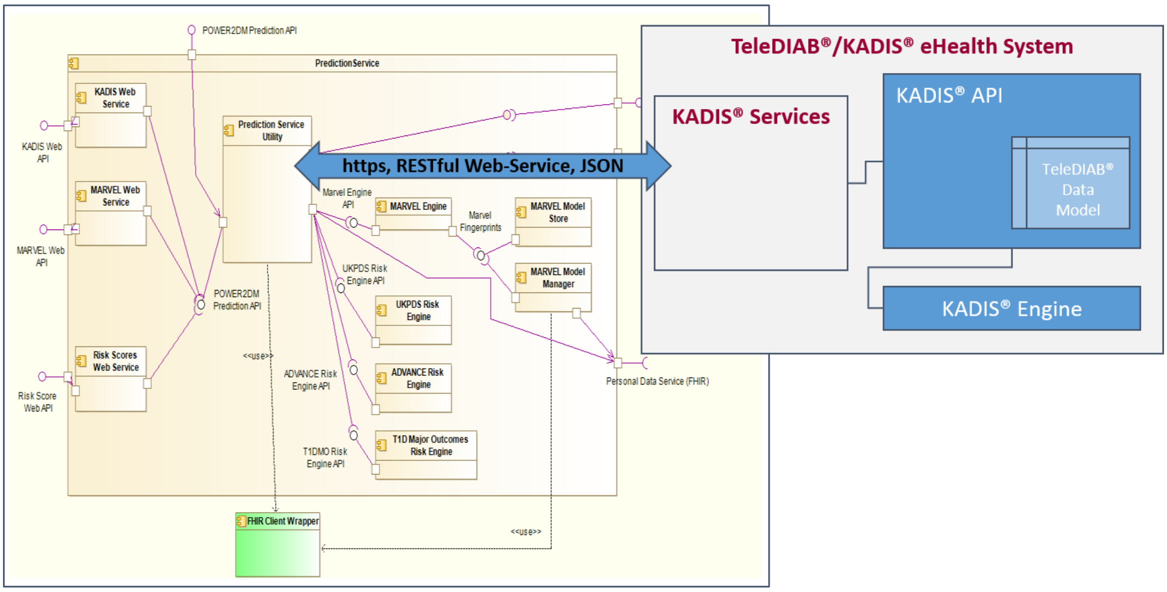


Figure 1 – The DCC KADIS® Services are embedded to the TeleDIAB® eHealth System with access to the KADIS® Engine and are mapped to the TeleDIAB® data model

5.2 Data validation

Data validation procedures are built into all DCC KADIS services. Due to the respective specifics (verification of input data, verification of model identification, verification of simulation results, etc.), methods for data validation are encapsulated with the respective REST services and not implemented as a central API. The two most important procedures for data validation in DCC KADIS Services are described below.

Data of the 3-day test

The patented Karlsburg Diabetes Management System, KADIS® solved the dilemma of addressing metabolic behavior in individuals with diabetes, in a practical way, for the first time worldwide. To model individual metabolic behavior, the modeling system required data that could be collected readily under everyday conditions. These data included sequential blood glucose measurements, either measured periodically by the patient (self-monitored blood glucose [SMBG]) or continuously monitored with a continuous glucose monitoring (CGM) device; type of therapy (insulin injections and/or tablets and/or GLP1 analogs); food intake (grams of carbohydrate); and physical activity (sports). In addition, it required a few basic characteristics of the patient (eg, age, diabetes type, duration of disease, body weight, and height). These data were collected and documented by the patient during a 3-day monitoring period (3-day test), according to a structured protocol for measurements under everyday conditions. This test could optionally be extended up to 7 days (7-day test). All data from the 3-day test will be compiled in a JSON file by the POWER2DM PDS and will be transferred to the DCC Server by a https-request (Fig. 2).

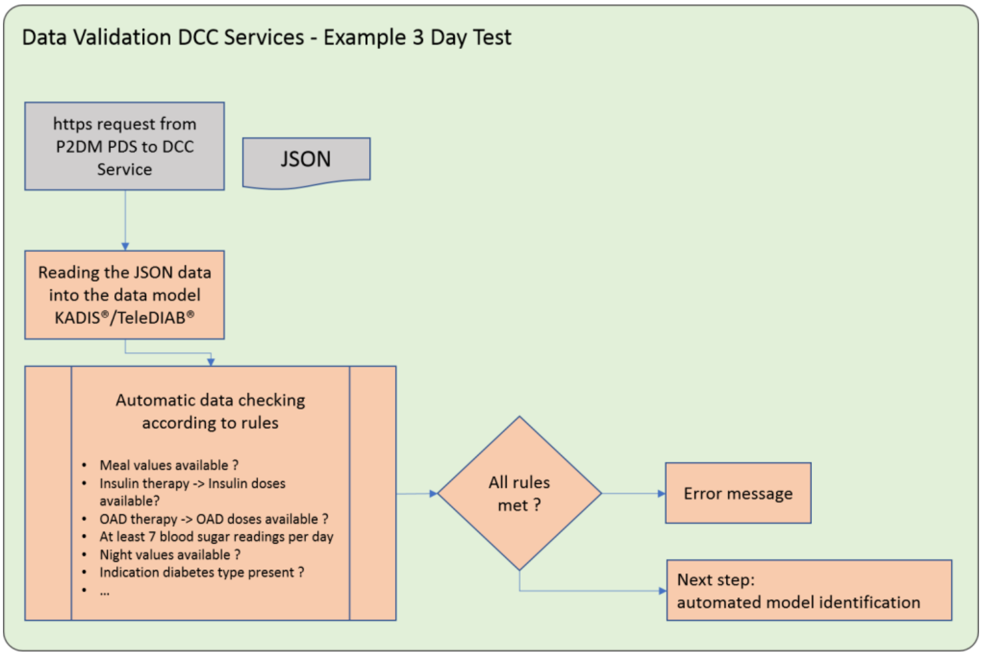


Figure 2 – Data validation of the 3-day test data compiled as JSON file by POWER2DM PDS

In order to prepare and secure a successful and optimal model identification, these input data must meet criteria and rules that are automatically checked by the DCC Service as shown in Fig. 2.

"Metabolic Fingerprint"

The result of a successful and optimal model identification is the "metabolic fingerprint".

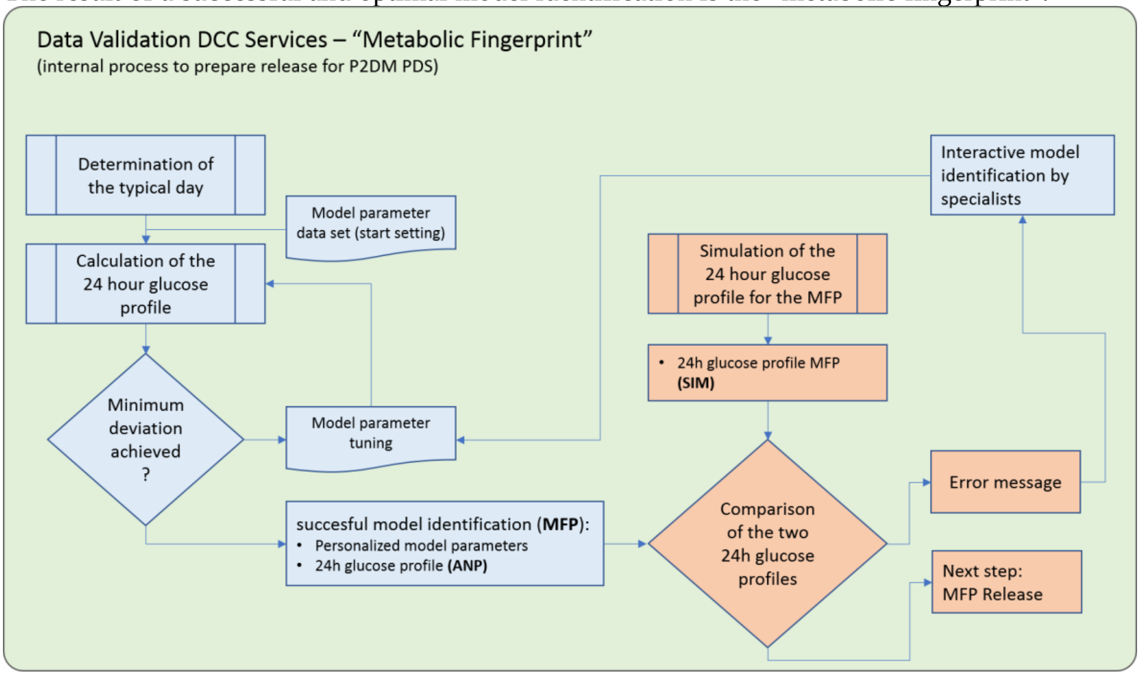
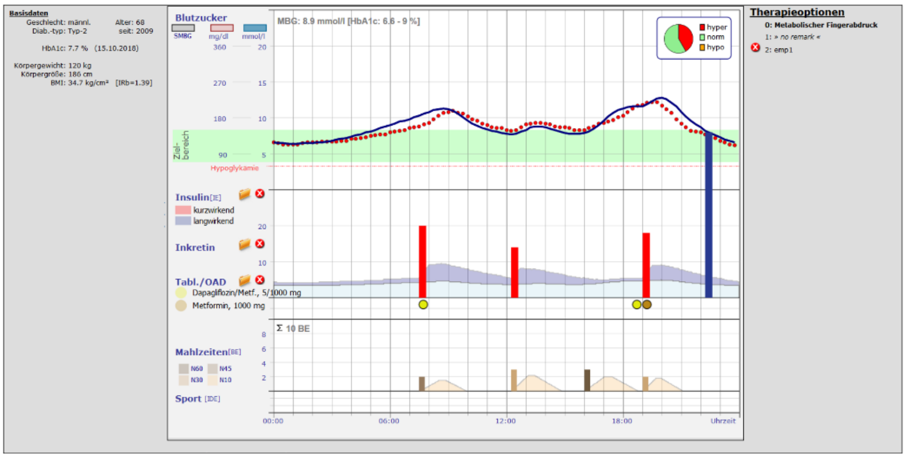
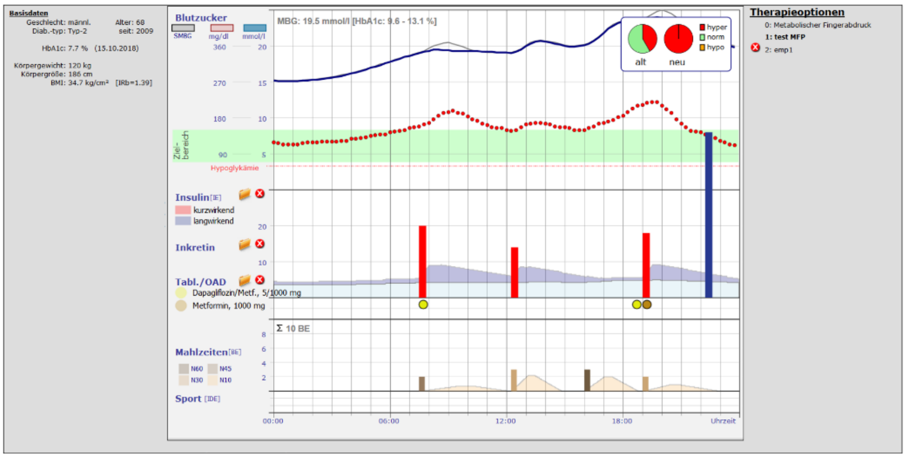


Figure 3 – Data validation of the MFP

The model parameters, which were determined after an automatic model identification run, are used for simulation tests, i.e. model-based calculation of the 24h glucose profile for the typical day (meals, therapy) as well as for simulative testing of therapy changes (e.g. increase or reduction of insulin dose, adjustment of food quantity (gCHO), etc.). Then the comparison is made with the 24h glucose profile of the MFP. If all tests are passed successfully, the MFP is released. In the other case, an interactive model eyelid identification can first be carried out by specialised personnel. In very few cases or if the input data has been falsified or manipulated, no model identification can take place. In this case, the 3-day test usually has to be repeated. To illustrate one part of the MFP data validation Fig. 4 shows a MFP (part A) and the test simulation for the MFP (part B) which in this case results in a rejection of the model identification for this MFP.



part A



part B

Figure 4 – Data validation of the MFP be test simulation (part B)

6 VALIDATION ON THE PREDICTION SERVICE

6.1 API input validation for Prediction Service

All of the models incorporated in the TNO Prediction Service APIs make use of the data stored in POWER2DM's personal data store (PDS). These data are retrieved from the PDS by the TNO Prediction Service itself. If the data retrieved are incomplete, this is reflected in the response given by the APIs together with the actual outcomes of the risk models, that may often be able to run even though some input data are lacking.
In addition, most of the APIs also provide intuitive error messages in case of malformed or otherwise unusable data. The KADIS simulation service, for instance, offers the possibility to simulate a patient's glucose profile on the basis of a patient's typical daily glucose curve and a set of simulation data. If the simulation data provided by the client is inconsistent or for another reason cannot be used for simulations, an intuitive error message is generated by the KADIS service and eventually passed back to the client addressing TNO's prediction service.

6.2 Internal Risk model validation

The risk models themselves (i.e. the core algorithms without the API) have all been validated by checking (for a list of scenarios) whether the exact intended outcomes are returned when given particularly defined inputs.

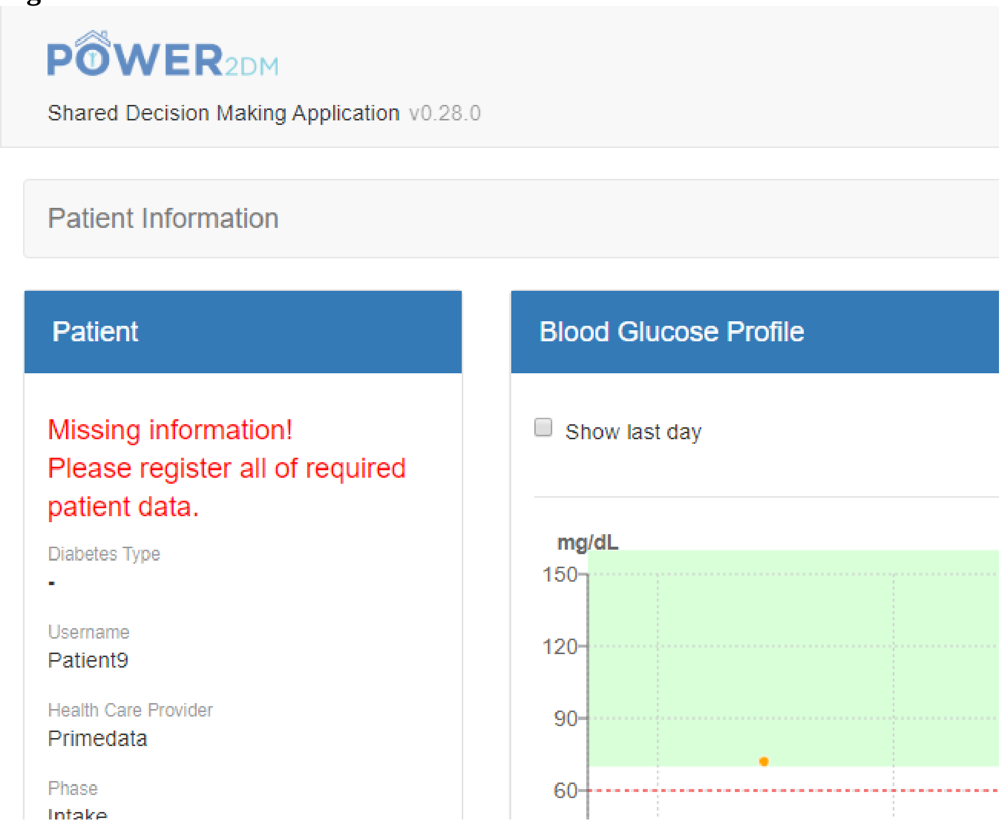
The POWER2DM risk scoring models were reprogrammed from the published equations.
The risk scoring services were subsequently validated by running predictions for a large set of randomly chosen values of the model input parameters, and checking the outputs versus the predictions done when using the available webservices.
The API's that form part of the Prediction Services check each input value for the specified valid range for the model and return an error message in case that the variable is out-of-range.

7 DATA VALIDATION IN SDMA

7.1 Validation For Data Registration

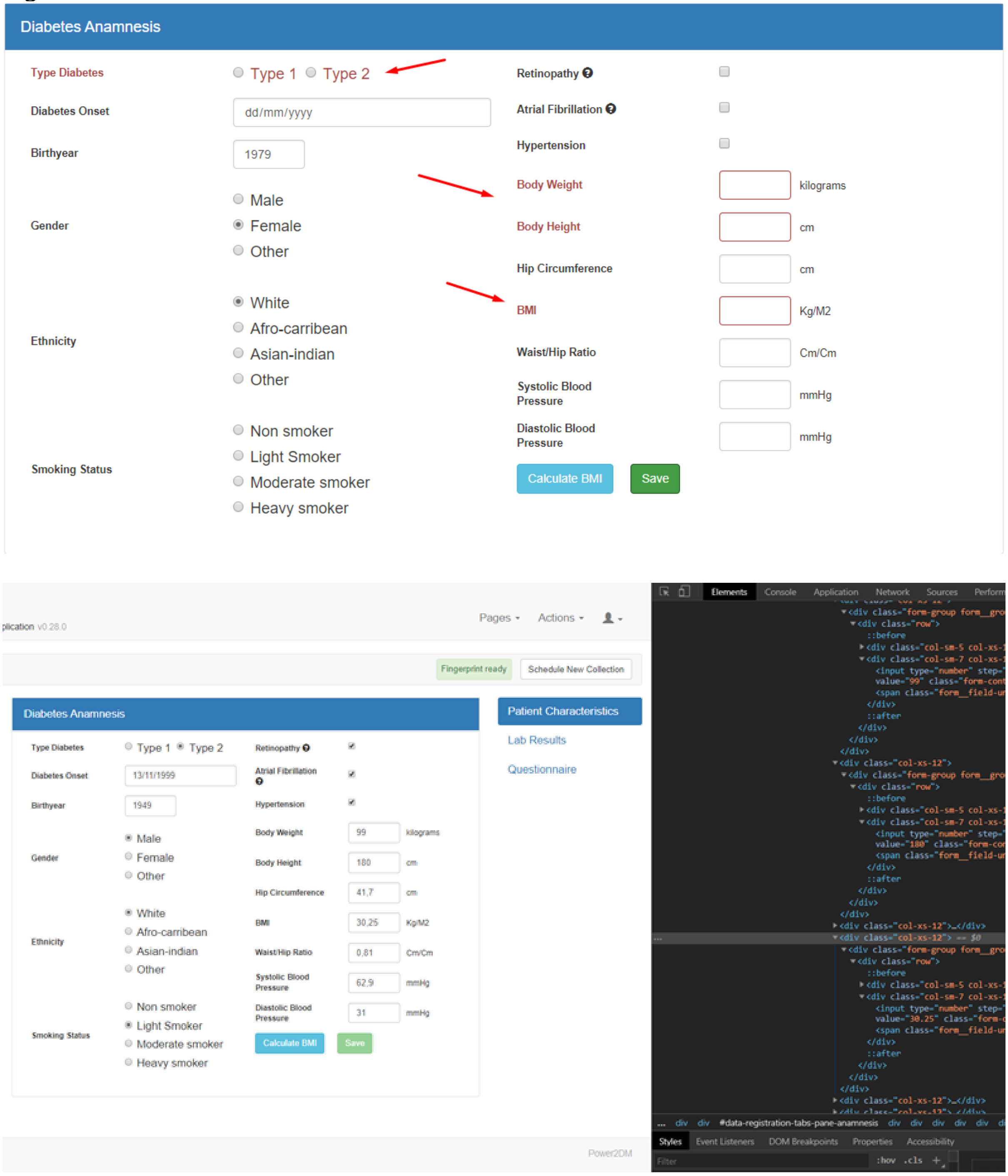
In SDMA we are validating data registration filed with all maximum and minimum value ranges, that was part of SDMA for long time. Lately we have added another validation step(required) for specific fields. The problem was that BMI is necessary for the KADIS calculation and to calculate it, we needed to have Body Weight and Body Height. So in order to prevent this problem, we have added warning in Patient Information (Figure 1).

Figure 1



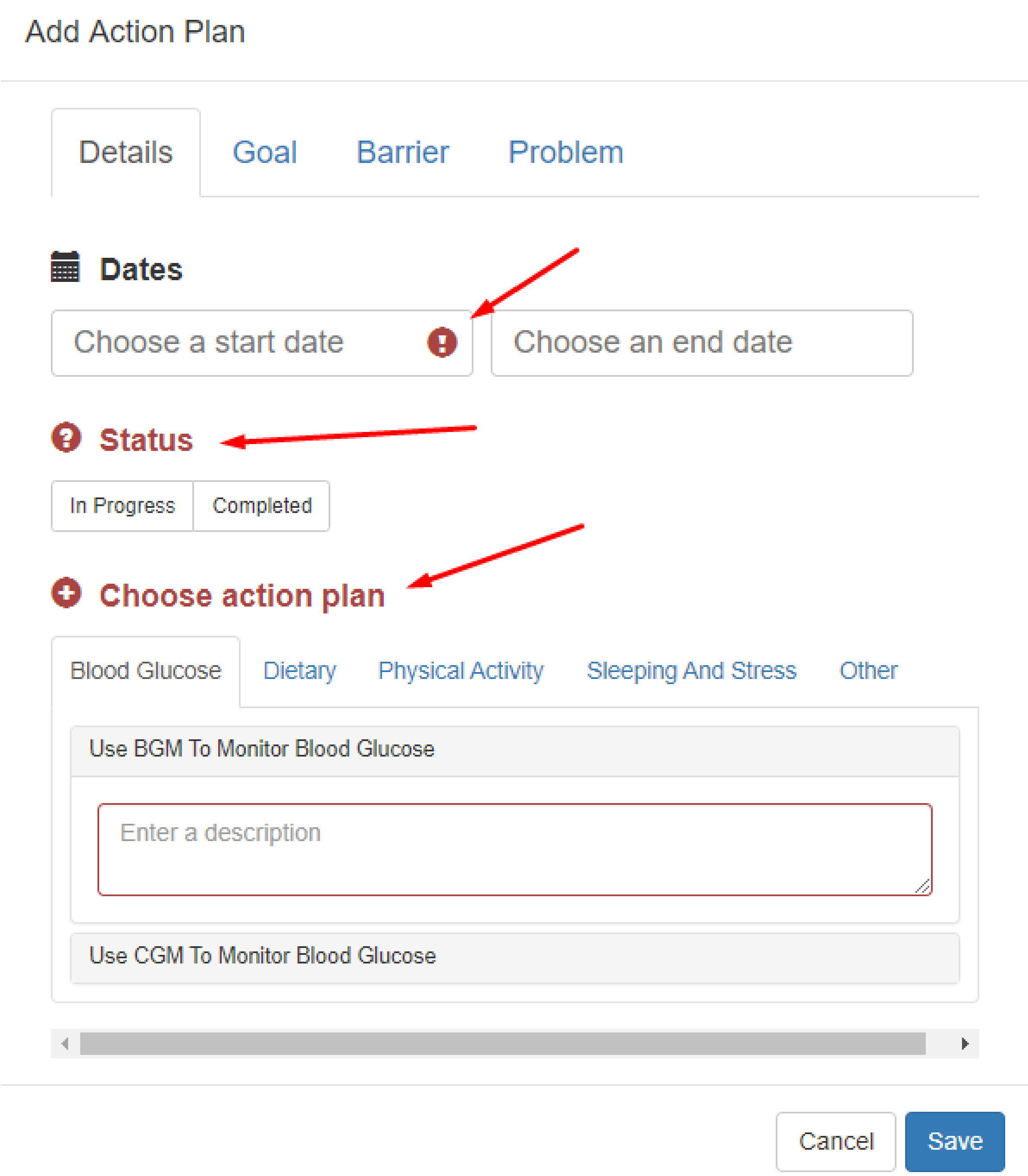
We have also made that specific fields which are important for the KADIS calculation required, so data registration is not possible to accomplish without filling them. See figure 2

Figure 2



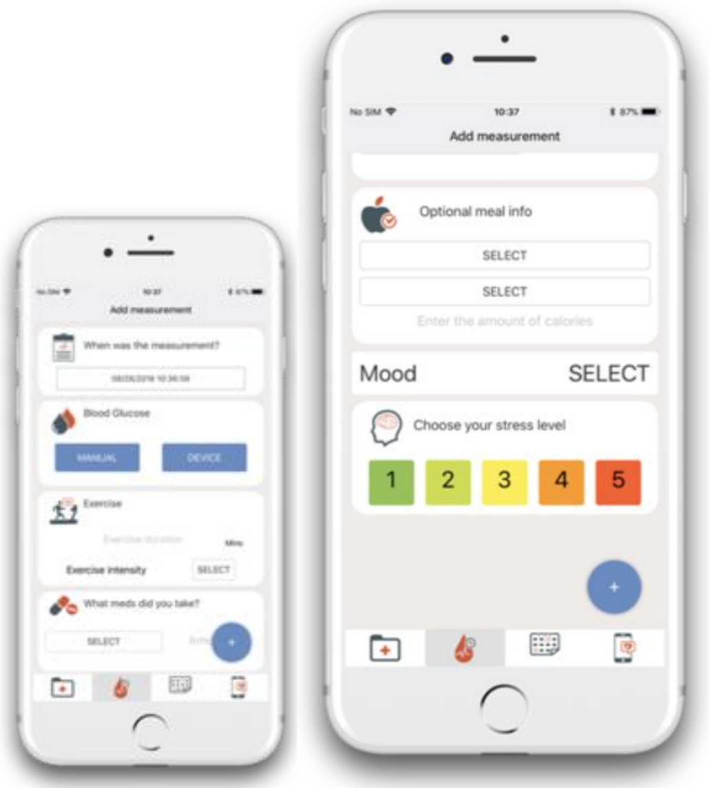
In SDMA we also validate every required field, and prevent to save/create something without required information. See figure 3

Figure 3



8 DATA VALIDATION IN MOBILE APP

In the mobile App, data validation is done when users enter their data in the mobile App. Most of the entries use a user interface that prevent from data entry errors.



Date is entered through a calendar.

For blood glucose, data can be entered either manually or automatically from the connected iHealth Gluco (BG5) glucometer. In case of automatic entry, no data validation is required since there is no risk of error. For manual entry, users enter their data through a numerical keyboard to limit data entry errors. For exercise, exercise duration is entered through a numerical keyboard and intensity is a a list (low, med, high).

For medication, the name of medication is presented as a list which includes only selected medications for a given user and the amount is entered through a numerical keyboard.
For meal information, type of meal can only be selected from a list (breakfast, lunch, dinner, snack), grams quantity is also a list (Very high - 73 grams or more ; High - 49-72 grams ; Medium - 25-48 grams ; Low - 0-24 grams) and the amount of calories.
Mood is selected using pictures representing the possible options (tense, excited, cheerful, irritated, neutral, relaxed, sad, bored or calm).
Stress is selected using color coded buttons (from 1 in green to 5 in red).

9 CONCLUSION

During the project the technical consortium partners decided, this in contrast with the original design which agreed on a central generic data validation component, that every application within POWER2DM does its own validation and processing. By default the applications use client side data validation and if there is an error while submitting the data to PDS, either it is a bug of client application or a transient error (PDS is down temporarily). This makes the whole configuration more responsive.