



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

POWER2DM D1.3

Conceptual Design of the POWER2DM Architecture

<i>Due Date:</i>	30 th September 2016 (M8)
<i>Actual Submission Date:</i>	
<i>Project Dates:</i>	Project Start Date: February 01, 2016 Project End Date: July 31, 2019 Project Duration: 42 months
<i>Deliverable Leader:</i>	SRDC

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	28.06.2016	Deliverable template and assignments	SRDC	TNO, PD, SRFG, iHealth, IDK
V0.2	22.07.2016	Section 3.3 Action Plan Engine Section 4.2.1 POWER2DM SMSS Web UI	SRFG	SRDC
V0.3	25.07.2016	Section 3.1 Personal Data Store Section 3.4 Communication Engine Section 3.5 Core Services	SRDC	
V0.4	09.08.2016	Section 3.2.2 MT2D-Marvel Subsystem Section 3.2.1 and Section 3.2.3 KADIS and Risk Score Prediction Service Subsystems	TNO, SRDC	
V0.5	26.08.2016	Section 5 – Interaction Flows	SRDC	
V0.6	05.09.2016	Section 6 – Data Architecture Deployment Diagram KADIS related APIs (Section 3.2.3)	SRDC IDK	
V0.7	06.09.2016	Section 4.1 – UI Design Considerations Section 7 – Deployment Infrastructure	SRDC	
V0.8	26.09.2016	Section 4.3 – UI Design for Shared Decision Making App Section 4.2.2 UI Design for SMSS Mobile App	SRDC iHealth	SAS
V1.0	30.09.2016	Compiling the deliverable for Submission	SRDC	TNO

Contributors (Benef.) Tuncay Namlı (SRDC)
 Suat Gönül (SRDC)
 Manuela Plöbnig (SRFG)
 Felix Strohmeier (SRFG)
 Dietmar Glachs (SRFG)
 Oliver Jung (SRFG)
 Albert de Graaf (TNO)
 Lutz Vogt (IDK)
 Hans Kron (PD)
 Javier Delgado Lista (SAS)
 Louie Lyu (iHealth)

Responsible Author Tuncay Namlı **Email** tuncay@srdc.com.tr

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	iHealth EU	France

OPEN ISSUES

No:	Date	Issue	Resolved
1	22.07.2016	We have tentative web service design for Value Compass & Questionnaires. The design may change according to developments done in WP3. (Section 3.3.2.2)	
2	25.07.2016	The list of external systems (e.g. Fat Secret, Fitbit, Spire, etc) to be integrated into the POWER2DM for data import is not final yet. The final list will be based on the Evaluation Campaign Protocol. (Section 3.1.3)	
3	05.09.2016	We have two alternative designs for the integration of existing KADIS system into POWER2DM. Decision is needed. (Section 3.2.1)	
4	08.09.2016	None of the risk engines is available with a API such as required for POWER2DM. A possible solution is to research the algorithms from the literature and implement the models plus the corresponding APIs ourselves. (Section 3.2.3.3)	
5	26.09.2016	Discussions are ongoing for the design of Shared Decision Making Application. (Section 4.3)	
6	05.09.2016	Based on the decision on KADIS integration, there may be a need to define a new record type; KADIS Fingerprint. (Section 6.1)	
7	06.09.2016	Given deployment infrastructure is just an example, the final decision will be based on the ethical and legal restrictions.	
8	26.09.2016	It is not decided yet on whether an existing multi-platform questionnaire system (e.g. like the one used by PatientCoach) will be used for this functionality. Or an internal POWER2DM implementation will be provided that can both serve on mobile and web platforms. (Section 4.2.2.1)	
9	26.09.2016	The UI design for SMSS Mobile Application cannot be provided as we need progress in WP3 regarding self-management interventions. (Section 4.2.2)	

TABLE OF CONTENTS

OPEN ISSUES.....	3
Table of contents.....	4
1 Introduction.....	6
1.1 Purpose and Scope	6
1.2 References.....	6
1.3 Definitions and Acronyms	6
2 System Architectural Design.....	7
3 Detailed Description of POWER2DM Components.....	10
3.1 POWER2DM Personal Data Store (PDS).....	10
3.1.1 Composition	11
3.1.2 Software Interfaces	13
3.1.3 Required Software Interfaces.....	14
3.2 POWER2DM Prediction Services	16
3.2.1 KADIS Subsystem	17
3.2.2 MT2D-Marvel Subsystem.....	23
3.2.3 Risk Score Service Subsystem.....	25
3.3 POWER2DM Action Plan Engine.....	31
3.3.1 Composition	31
3.3.2 Software Interfaces	32
3.3.3 Relation with Existing Products.....	38
3.4 POWER2DM Communication Engine	39
3.4.1 Composition	39
3.4.2 Software Interfaces	40
3.4.3 Required Software Interfaces.....	41
3.5 POWER2DM Core Services.....	41
4 User Interface Design and UI Components.....	43
4.1 User Experience and HCI Design Considerations	43
4.1.1 Getting to know our users	43
4.1.2 Scenario Flows and required POWER2DM Features	45
4.1.3 Design and Implementation	45
4.1.4 Measuring and Evaluation	46
4.1.5 HCI Considerations.....	47
4.2 POWER2DM SMSS UI.....	50
4.2.1 POWER2DM SMSS Web UI	50
4.2.2 POWER2DM SMSS Mobile Application UI	56
4.3 POWER2DM Shared Decision Making UI	63
4.3.1 Compass View	64
4.3.2 Glucose Management Dashboard	65
4.3.3 Lifestyle & Psychological & Self-management Dashboard View	68
4.3.4 Diabetic Health Progress Dashboard View.....	70
4.3.5 Major Outcomes and Risks Scores Dashboard View	70
4.3.6 Barriers Dashboard View.....	70
4.3.7 Commonly Used Panels	70
4.3.8 Data Entry Forms.....	71
4.4 POWER2DM User and Privacy Management UI.....	71
5 Interaction FLOWS	73
5.1 Interactions with PDS to access or manipulate data	73
5.2 JITAI Planning Interaction Flow	76
5.3 Action Plan Review Interaction Flow	76
5.4 KADIS Fingerprint Identification Interaction Flow	77
5.5 Treatment planning with KADIS Interaction Flow	78
5.6 Authentication and Authorization Interaction Flows.....	78

6	Data Architecture	80
6.1	Personal Health Data Model	80
6.2	Model for User Identity, Settings, Policies and Audit Data.....	82
7	Deployment Infrastructure.....	83

1 INTRODUCTION

1.1 Purpose and Scope

This document describes the conceptual design of the POWER2DM system according to the document guidelines presented in the IEEE-1016 Recommended Practice for Software Design Descriptions (SDD). As an SDD document, this document aims to show how the POWER2DM software system will be structured to satisfy the requirements identified in the software requirements specification, i.e. POWER2DM D1.2.

Considering the characteristics of POWER2DM project, an international multi party project, **the main focus of this document** is to

- identify the main components and their composition (major sub-components) (by UML component diagrams)
- identify the responsibilities of partners in terms of implementation
- identify the service interface for these main components so that interoperability among the components can be achieved (by UML class diagrams or Restful API definitions)
- define the flows for use case realizations (by UML sequence diagrams)
- identify the main UI views and provide mockups describing the main functionality

By agreeing on these among partners, most importantly the service interfaces and main functionality provided by UI components, each workgroup for the workpackages or partner will have the freedom to detail the design and implement the responsible component independently from others.

In line with these objectives, **Section 2** provides an overview of the conceptual design by describing the major components of the POWER2DM and architectural approach. **Section 3** provides more detailed design for each major component providing the composition of component (sub-components), the definition of service interfaces it provides, the required service interfaces, and internal flow for the main processes (sequence diagrams). **Section 4** provides user interface designs; mockups for each POWER2DM UI application to describe the functionalities they provide to their users. **Section 5** describes how the use cases are realized by the flows between the main components (by sequence diagrams). **Section 6** provides the common data architecture and Section 7 provides alternative deployment infrastructures.

1.2 References

- POWER2DM Description of Work (Proposal)
- D1.1 User Requirements and Use Case Scenarios
- D1.2 Requirements Specification of the POWER2DM Architecture
- D4.1 Personal Data Model and Service API
- D4.9 Privacy/Security Enablers for POWER2DM Services I
- D2.5 Mockups for GUI Components related with Visualization of Predictions and Simulations

1.3 Definitions and Acronyms

Table 1 List of Abbreviations and Acronyms

Abbreviation/ Acronym	DEFINITION
API	Application Programming Interface
ARC	Audit Repository Client component
ATNA	IHE Audit Trail and Node Authentication Profile

CGM	Continuous Glucose Monitoring
CHD	Coronary Heart Disease
CRUD	Create, Read, Update, Delete operations
CVD	Cardiovascular Disease
D	Deliverable (e.g. D4.1)
DSTU	Draft Standard for Trial Use
ETL	Extract, Transform, Load operations
FHIR	HL7 Fast Healthcare Interoperability Resources Standard
IHE	Integrating Healthcare Enterprise
JITAI	Just in time adaptive intervention
JSON	Java Script Object Notation
JVM	Java Virtual Machine
ODL	Observation of daily life
PDP	Policy Decision Point
PDS	POWER2DM Personal Data Store component
REST	Restful Service
RIA	Rich Internet Application
SDD	Software Design Descriptions
SDK	Software Development Kit
SDMA	POWER2DM Shared Decision Making Application
SQL	Structured Query Language
SMSS	POWER2DM Self-Management Support System
T1D	Type 1 Diabetes
UI	User interface
UML	Unified Modelling Language

2 SYSTEM ARCHITECTURAL DESIGN

Figure 1 illustrates the conceptual design of POWER2DM system showing the major components; major functionalities they provide and interactions with each other. POWER2DM architecture is mainly composed of two layers;

- user interface layer that is the POWER2DM UI components (the web or mobile applications provided to patients and care providers)
- backend service layer that is composed of several POWER2DM components that provides the secure data storage and intelligence required for self-management support.

The followings are the major **POWER2DM Applications (UI components)**;

POWER2DM Shared Decision Making Application (SDMA): This will be a web based Rich Internet Application (RIA) that will support the healthcare professionals (physicians and supporting nurse) in shared decision making phase to

- register clinical data (anamnesis, problems, lab measurements, etc)
- analyze the patient's status (based on collected data in self-management phases)
- and provide/update personalized treatment plan for patient (goals, action plans, instructions, solution to barriers, etc) which the POWER2DM SMSS will use as the basis

POWER2DM SMSS Mobile Application: This will be the native mobile application that will support and interact with patient in self-management for

- delivering customized JITAI interventions
- collection of ODL (Observations of Daily Life) data; manual guided data entry by patient and/or connecting and retrieving measurements from medical devices and transfer them to Personal Data Store
- customization of user profile and settings

POWER2DM SMSS Web Application: This will be a web application to support patient in self-management. It complements the POWER2DM Mobile Application and enables patient to manage his self-management action plan via larger views. It supports patient for;

- Specification of personal values and barrier identification
- Specification and update of self-management goals (e.g. in case of a new/updated treatment plan or when self-management goals are not realistic)
- Planning of activities (action plans) periodically (basically weekly)
- Filling questionnaires and collecting ODL data
- Evaluation of performance by providing feedback periodically (basically weekly); (e.g. feedback on planned activities, motivational messages and tips for fostering or improving self-management activities)

POWER2DM User and Privacy Management Web Interface: This will be a web application to handle the common administrative operations for user and privacy management. It serves for both patients and healthcare professionals. It provides the following main functionality;

- Single sign-on page for all users; All other POWER2DM applications (Shared Decision Making Application, SMSS Mobile and Web Application) redirects to this page when the authentication of the user is required.
- Patient registration page (registration of patient to POWER2DM program)
- User registration page (registration of other users; healthcare professionals to POWER2DM)
- Patient Privacy Management page (patient manages his/her privacy policies)
- Audit View page (for patients to access audit records; who has accessed to his data)
- User profile management page (all users managing their user profiles)

At the **backend service layer**, there will be the following major components;

POWER2DM Action Plan Engine: It provides the backend intelligence service for POWER2DM SMSS Mobile and Web applications that helps patient manage his/her self-management goals and action plans. The starting point for the Action Plan Engine are the treatment goals and the treatment plan provided by the Shared Decision Making Application. The Action Plan Engine provides interventions based on the collected ODLs e.g. feedback on planned activities, motivational messages and tips for fostering or improving self-management activities. Interventions are provided periodically, typically in weekly cycles.

POWER2DM Communication Engine: It is a backend intelligence service aiming to recognize patient's preferences and life-style patterns from collected data. Based on these recognized patterns and preferences it plans Just-In-Time Adaptive Interventions (JITAs; Nahum-Shani et al., 2014). JITAs are self-management interventions that are initiated by the POWER2DM SMSS system (specifically Communication Engine) automatically during daily life of the patient according to the changing context of the patient. The idea is to adapt the frequency, content and type of interventions accordingly. Examples of JITAs are: (a) warning that after a loaded breakfast the patient should have a light lunch in order to manage his blood glucose, or (b) patient only does half his planned exercise and receives a motivating message to finish the complete exercise.

Planned JITAs will be delivered to patients through;

- the dashboard views of POWER2DM SMSS Web Application
- the push notifications for POWER2DM SMSS Mobile Application

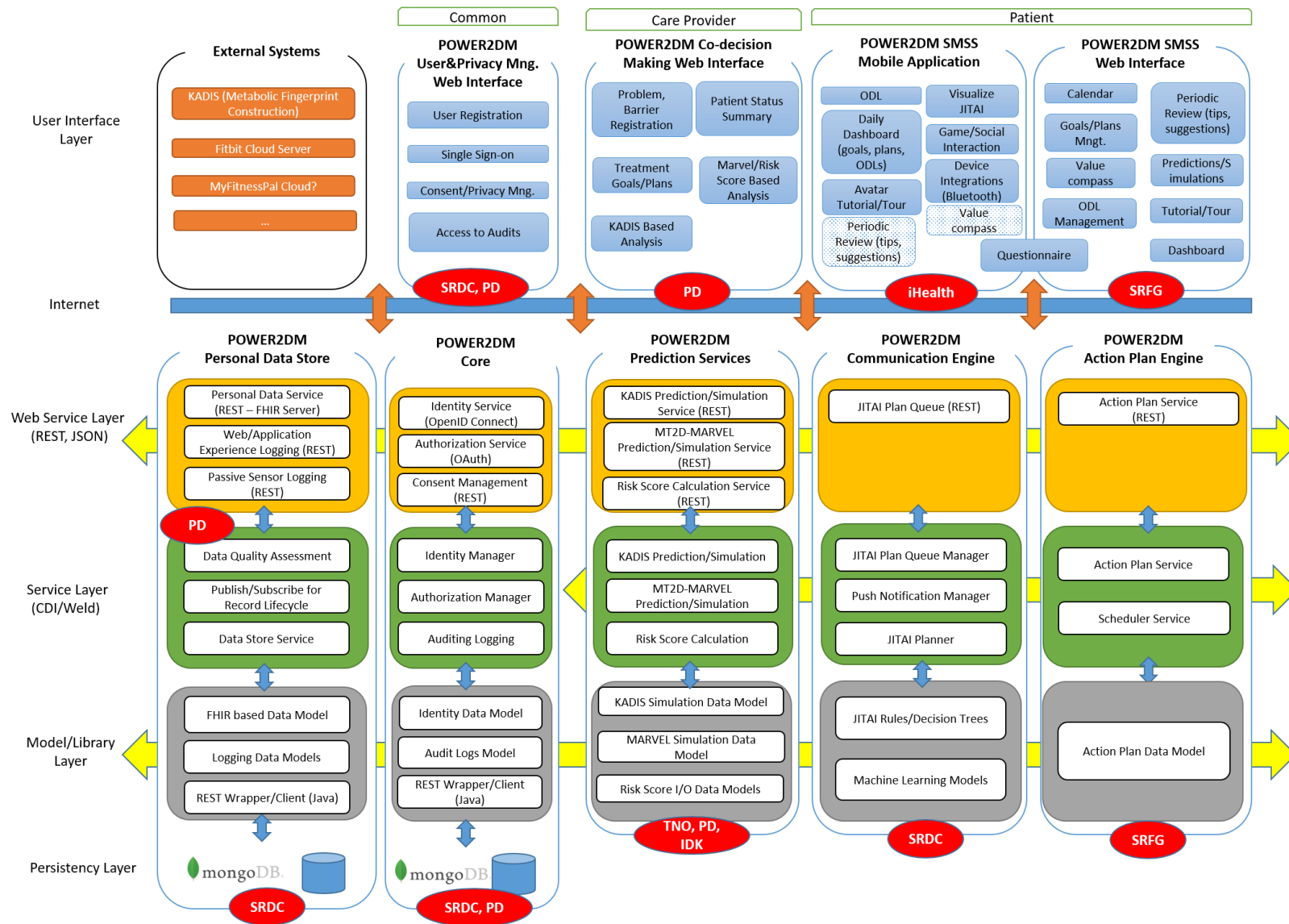


Figure 1 POWER2DM Conceptual Design

POWER2DM Prediction Services: All short, mid and long-term predictive models (KADIS, MT2D-Marvel and Risk Score models as prediction service) are exposed as services in a way that are consumable by other relevant components through a web API or through the corresponding APIs within the backend platform. POWER2DM UI components access to these services over the provided API to perform real-time simulations by the models and showing the results to the users. The Communication Engine interacts with these services to perform the simulations periodically or at certain times to analyze the progress of the patient and act intelligently based on the results.

POWER2DM Personal Data Store: This is the component that will provide a secure storage for any patient data that needs to be recorded in the system and provide a secure standard based (HL7 FHIR based) RESTful service for other POWER2DM components to access and update recorded data. An authorization mechanism based on the patient control privacy policies is implemented.

POWER2DM Core: This is the component that will provide the core POWER2DM services like authentication, user identity and authorization management, and audit log storage.

POWER2DM follows Microservice Architecture¹ pattern as much as possible. As shown in Figure 1, the major components provide their functionality in four layers.

- In the web service layer, components wrap their services as HTTP Restful services with JSON serialization and any authorized client can access the service over Web. In this way, in addition to UI components, other components are able to access to the service in different deployment alternatives (even the components are deployed to different sites).
- In the service layer, components provide a Java based API for their services which enables the sharing of services for the components that are deployed within the same JVM environment.
- In the model and library layer, the functionality is shared as common programming libraries among components.
- The persistency layer only exists for POWER2DM Personal Data Store and POWER2DM Core components where the first stores the pseudonymized personal health records and the other stores user identity and privacy policies.

3 DETAILED DESCRIPTION OF POWER2DM COMPONENTS

3.1 POWER2DM Personal Data Store (PDS)

Personal Data Store (PDS) is one of the major component of POWER2DM that securely stores any personal health data collected or generated (i.e. inferences, predictions, etc.) within POWER2DM Care Program for patients. It provides software interfaces for other POWER2DM components and UIs to securely access (CRUD + Search operations) on behalf of an authorized user or for their internal data processing/algorithmic needs.

HL7 Fast Healthcare Interoperability Resources (FHIR) DSTU 2² is selected as the base standard for the common personal health data exchange model for POWER2DM. We use the profiling and extension mechanism of the FHIR standard to make the restrictions and extensions to base FHIR resource definitions (e.g. Observation, Goal, Problem, etc.) to meet the specific data requirement of

¹ <https://en.wikipedia.org/wiki/Microservices>

² <https://www.hl7.org/fhir/>

POWER2DM. The “D4.1 Personal Data Model and Service API” provide details of this common data exchange model. Other POWER2DM components (UIs and services)

Apart from personal health data, PDS will also be used to store personal data regarding patient’s application usage behaviour and contextual data. This is the set of analytic information related with how patient use the POWER2DM SMSS applications (e.g. time spent on the application, or on a specific view) and how they react to interventions (e.g. time to react to a push notification), as well as the contextual data like location (home, office, etc.), interruptibility, activity (walking, transportation, etc.) that can be derived from the on-board mobile phone sensors. For such records, data will be stored in its raw format and be accessible to only POWER2DM system for internal processing through the PDS’s Analytics/ETL endpoint.

3.1.1 Composition

As illustrated in Figure 2, the followings are the subcomponents of PDS (inside PDS component box);

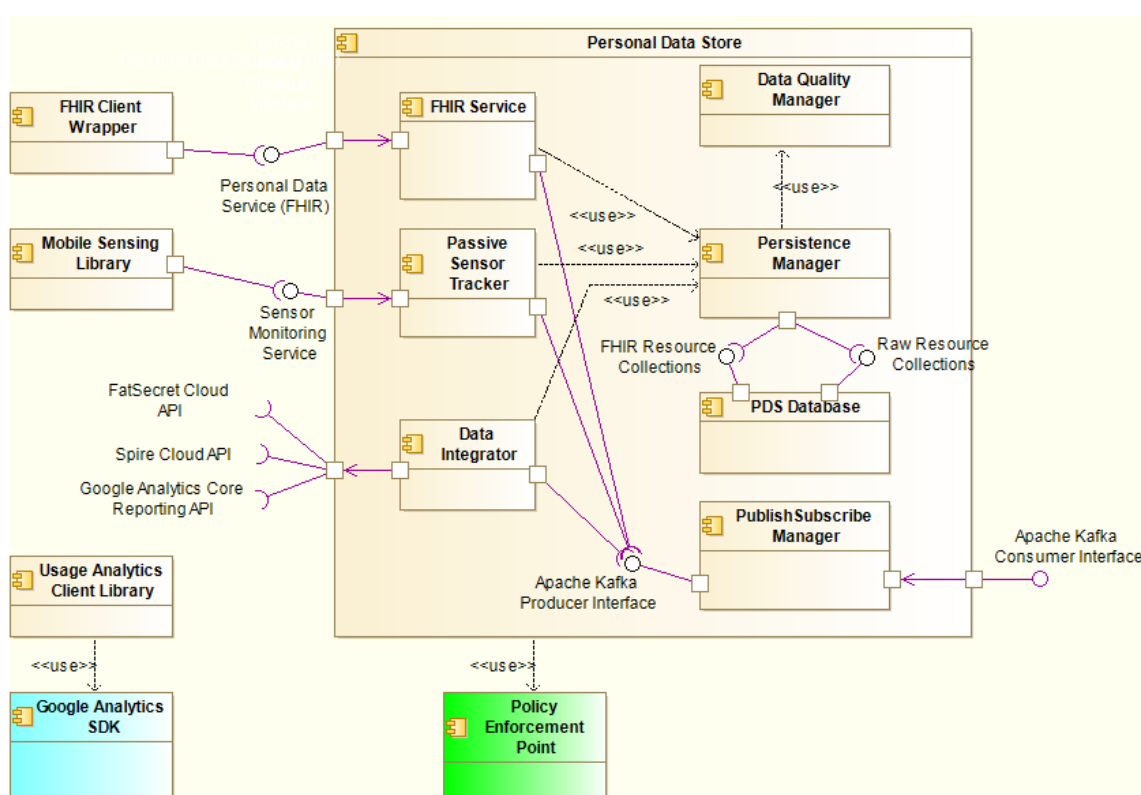


Figure 2 Sub-components of Personal Data Store subsystem

FHIR Service is the implementation of FHIR conformant REST service based on the restrictions and extensions defined in D4.1. In summary it will provide the CRUD and search operations on the patient records based on the FHIR based common data exchange model. This service will be the external service for the PDS component. The subcomponent will use Policy Enforcement Point (PEP) library for authorization decisions and Audit Repository Client (ARC) library to construct the necessary audit record and send it to Audit Record Repository (See D4.9 Privacy and Security Enablers). For create and update operations, the subcomponent will validate the content of the records if they are syntactically conforming to the base FHIR resource profiles and the specification provided in D4.1 for the related resource type. The subcomponent uses the “Persistence API” provided by FHIR Persistence Manager subcomponent to persist the records or access/query the records. It also publishes the record to Publish/Subscribe Manager over KAFKA Producer Interface.

Persistence Manager is one of the internal sub-components that implements the persistency mechanisms for patient's FHIR based personal health records and other raw record formats. It provides a Scala API to perform the CRUD and search operations on the patient records. The subcomponent directly used the database connection interface provided by PDS Database to perform the CRUD operations and query.

PDS Database and is a MongoDB database in which MongoDB document collections are defined for each resource type in POWER2DM. This covers the FHIR Resource types supported in POWER2DM (e.g. Observation, Condition, Goal, MedicationOrder, Questionnaire, etc – See D4.1 for details) also the raw resources for application usage and contextual data. The identifiers assigned to resource types will be used as MongoDB collection names.

Passive Sensor Tracker is the implementation of Sensor Monitoring Service that will collect raw sensor data for patient from the on-board mobile phone sensors and medical devices connected to the mobile phone over Bluetooth. The POWER2DM SMSS Mobile Application will be the intermediary system that will send this data to the Sensor Monitoring Service endpoint. The details of the service are given in Section 3.1.2. The subcomponent will use PEP subcomponent to authorize the operation. Then it publishes the record to Publish Subscribe Manager over Kafka Producer Interface and persist the records by using Persistence API.

Publish Subscribe Manager is publish/subscribe messaging system that is used by other POWER2DM Service Components to subscribe to specific type of resources and get them as a stream. The open source Apache Kafka system³ will be used for the realization of this component. The sub-component applies client authentication for the Kafka Producer and Consumer interfaces to be sure that only authorized POWER2DM components can access the data. Kafka Topics will be the resource type identifier and the key for the Kafka messages will be the patient identifier.

Data Integrator is the component that periodically retrieves patient data from external cloud APIs (if Patient authorize POWER2DM) and convert the data to POWER2DM common data exchange model before persisting it. It uses the persistence API and Kafka Producer interface to persist and publish the data. In the figure, some example APIs are shown; Spire Cloud API to integrate the Spire device data⁴, FatSecret Cloud API to integrate the dietary intake data collected by the use of FatSecret application, and Google Analytic Core Reporting API to retrieve the application usage analytics data. The exact list of 3rd party systems to be integrated will be specified. The frequency of data pull operation will be configurable for each API; e.g. per day, per hour.

The components in the left, outside the box, are libraries and SDKs to be provided for the usage of other POWER2DM components;

FHIR Client Wrapper will be a JAVA library to enable other POWER2DM components to access patients' personal health records easily. It will provide a simple interface for CRUD operations and search interface for each resource type with the defined search parameters. It will also provide the Java object model de-serialized from the FHIR based resource content.

Mobile Sensing Library will be an native Android library that collects data from the on-board mobile sensors and the other probes (e.g. personal lifestyle or medical devices connected through Bluetooth) and deliver it to the PDS through Sensor Monitoring Service. The existing open sensing framework Funf⁵ will be used as basis and extended and configured for the selected sensors. The Funf enables the configuration of the probes to collect data, the frequency of data collection for each probe, the frequency of server synchronization.

³ <http://kafka.apache.org/>

⁴ <https://www.spire.io/>

⁵ <https://github.com/funf-org/funf-core-android>

Usage Analytics Client Library will be a supplementary SDK for UI components to easily integrate the necessary analytics code snippets to their web pages or native mobile application codes. Google Analytics will be used to collect the application usage behaviour of patients as base technology.

3.1.2 Software Interfaces

PDS will open three interfaces to the other components. The following sections describe them in detail.

3.1.2.1 HL7 FHIR REST Service

PDS will provide an “Restful FHIR” conformant endpoint to enable other components to access or update the personal health records of patient. The “**D4.1 Personal Data Model and Service API**” provides the details of this service, so here we provide only a very brief summary.

The smallest unit of information that can be accessible through the FHIR based APIs are called resources. FHIR standard defines common resource types and provide the data model for each of these resources, e.g. Observation, Condition, Procedure, MedicationOrder, etc. They also provide a profiling mechanism to extend/restrict these definitions according the local or project specific needs. In POWER2DM, we use this profiling mechanism to define the POWER2DM Resource profiles (e.g. DiabetesAnamnesis, Problem, Barrier, Goal, etc) by extending the base FHIR resource profiles according to the project requirements. PDS will provide the secure storage for these resource types and

support the following FHIR interactions to access or manipulate the resources for patients;

- **read:** Read the content of a specific POWER2DM resource
- **update:** Update an existing resource by its id
- **delete:** Delete a specific resource
- **create:** Create a new POWER2DM resource
- **search:** Search among specific resource type of a patient based on some filter criteria
- **batch/transaction:** Update, create or delete a set of resources in a single interaction

3.1.2.2 Sensor Monitoring Service

This service is used by the POWER2DM SMSS Mobile Application to deliver the measurements immediately in their raw format to the PDS after obtaining them from the passive sensors. These sensors may include the probes on the mobile device like Wi-Fi , Bluetooth, Screen probes (whether phone screen is on or off), as well as the medical devices that the mobile application is connected over the Bluetooth connection.

HTTP POST command will be used as follows for the interaction;

POST [base]/sensor-measurements/[probe-type]

The content will be provided using a multipart form data format, with the parameter name ‘**uploadedfile**’. The syntax of the content depends on the probe type and the service is content neutral. The supported probe types are specified and each probe type will have an identifier [probe-type]. The [base] indicates the base URL for the Restful service e.g. <https://www.power2dm.eu/pds>.

Resource Type “Sensor Measurements”	
Allowed HTTP Methods	POST
Path incl. path parameters	/sensor-measurement/{probe-type}
Query parameters	Not Applicable
POST Data	uploadedfile: {File content in the raw format of sensor}
Success POST Response	200 OK empty

Error Responses	401 Unauthorized, if user is unauthorized 404 Not Found, if {probe-type} is not supported 500 Internal Server Error in case of internal exception
-----------------	---

3.1.2.3 Resource Streaming Endpoint (Kafka Consumer Endpoint)

This endpoint is an Apache Kafka Consumer⁶ endpoint where the clients can use the new Apache Kafka Consumer API v0.10.0⁷ to access to the endpoint. Only trusted POWER2DM components can access to this endpoint and SSL will be used for encryption and authentication of the clients.

In Kafka, data will be given in key value <K,V> format. For this endpoint,

- Key will be the patient identifier serialized in String format (org.apache.kafka.common.serialization.StringSerializer)
- Value will be the resource content serialized in String (FHIR JSON serialization) (org.apache.kafka.common.serialization.StringSerializer)

The topics managed by the endpoint will be in the following format for each resource type defined in PDS (See D4.1 Section 3.5.2 for resource type names);

- raw.fhir.<resource-type-name>

In this way, the client can subscribe to a specific type of resource for resource streaming, e.g. subscribing to the topic “raw.fhir.power2dm_observation” will enable client to retrieve all the observations in a stream.

Clients can use the any methodology (automatic offset committing or manual offset control) defined in the API⁸ to handle the reading of the stream. The Maven component org.apacke.kafka%kafka-clients%0.10.0.0 provides the library to easily use the API.

3.1.2.4 FHIR Client Wrapper

The library will be basically a very light extension to HAPI FHIR Library⁹. The HAPI-FHIR already provided the Java object model for FHIR data model and FHIR client utilities to easily perform FHIR operations on a remote FHIR server. Following their approach and mechanisms, FHIR client library will implement the POWER2DM extensions on the data model and implement the parts that will handle the POWER2DM authorization process easily by configurations.

3.1.3 Required Software Interfaces

PDS does not require any software interface from other POWER2DM components. However, for data integration with other cloud provided patient data, it will use their specific APIs.

3.1.3.1 Fat Secret REST API

Fat Secret is an application (both mobile and web platforms) that is used for food logging. It has a very large food database including the local foods. In POWER2DM, if patient is using FatSecret we provide

Him/her to opportunity to integrate his data in FatSecret into POWER2DM instead of a separate dietary intake logging.

⁶ <http://kafka.apache.org/>

⁷ <http://kafka.apache.org/documentation.html#newconsumerapi>

⁸ <http://kafka.apache.org/0100/javadoc/index.html?org/apache/kafka/clients/consumer/KafkaConsumer.html>

⁹ <http://hapifhir.io/>

Fat Secret is providing a REST API¹⁰ to share the data with other applications if user allowed them. PDS will use this API to retrieve data periodically. These periods will be configurable (e.g. check every hour if new data is available). PDS will use the “food_entries.get” method to retrieve all the food entries for the specified time interval. The Data Integrator sub-component will map the results to POWER2DM Dietary Intake Log resource (See in D4.1).

3.1.3.2 Fitbit API

Fitbit is providing physical activity tracker devices together with mobile applications and a cloud platform to store the data. In POWER2DM, the Fitbit devices may be recommended to patients to automatically collect physical activity, sleep activity and heart rate data. PDS will use the Fitbit Web API¹¹ to retrieve this data by the authorization taken from the patient himself.

PDS will use the following interactions;

- **Get Daily Activity Summary:** Daily summary of the user activity including calories burned, distance, steps, sedentary durations, etc.
- **Get Activity Intraday Time Series:** This endpoint returns the Intraday Time Series for certain parameters like calories burned, distance, steps, etc.
- **Get Activity Log List:** This endpoint retrieves a list of a user's activity log entries (e.g. running session, swimming session, bicycle, etc) before or after a given time. These activity logs include both the activities automatically recognized by the device and also logged manually by user.
- **Get Sleep Logs:** It returns a summary and list of a user's sleep log entries as well as minute by minute sleep entry data for a given day in the format requested.
- **Get Heart Rate Intraday Time Series:** The endpoint returns the intraday time series for heart rate in the format requested.

The records collected from these endpoints will be converted to the following POWER2DM resource types (See D4.1) in the given order;

- POWER2DM Activity Tracker Daily Summary (combination of first two items)
- POWER2DM Physical Activity Log
- POWER2DM Sleep Tracker Sleep Summary
- POWER2DM Tracker Heart Rate Summary

The services will be queried periodically configured for each service type.

3.1.3.3 Spire API

Spire is a personal medical device provided with a mobile application to monitor the stress of the users from the breath monitoring sensor. It also provides a cloud API¹² to access to the data. Information on the API for Developers can be found on <https://spire-api.readme.io/v1.1>.

3.1.3.4 Google Analytics Core Reporting API

Google Analytics is a web (and mobile) analytics platform that enables web site or mobile application providers to monitor some parameters like page views, number of sessions or user behaviours on pages. In POWER2DM we will use it to track how patient use the POWER2DM SMSS applications (mobile and web interfaces) and how they react to interventions delivered through these applications. This includes the visited views, duration spent on each view, response times to push notifications, etc.

¹⁰ <https://platform.fatsecret.com/api>

¹¹ <https://dev.fitbit.com/docs/>

¹² <http://support.spire.io/>

The platform provides an API¹³ to access these data and PDS will use its Java API to query these collected metrics periodically (daily). The results will be stored in its raw format.

!!! Open Issue: The list of external systems (cloud providers) to be integrated into the POWER2DM is not final yet. The final list will be based on the Evaluation Campaign Protocol. The systems mentioned in this section (Fat Secret, Fitbit, etc) are the ones discussed in the Quantification Campaign and design phase.

3.2 POWER2DM Prediction Services

Figure 3 illustrates the detailed component diagram for Prediction Service. As shown, it is composed of three sub-systems each providing a specific prediction service;

- **KADIS Subsystem:** This subsystem is based on the existing KADIS software and provides a service for short-term predictions and simulations regarding the blood glucose profile of patients.
- **MT2D-Marvel Subsystem:** This subsystem is based on the existing MT2D-Marvel model and provides a service for mid-term predictions and simulations for major parameters related with diabetes progress e.g. BMI, fasting glucose, fasting insulin, etc.
- **Risk Score Subsystem:** This subsystem is based on several existing risk calculation models and provide a unified service for calculation of risk scores for patient for current situation and simulation scenarios.

All of these subsystems shared a common approach about their service logic. In general, we want to use these services for both i) prediction of some target parameter at some future time with the current health and lifestyle of patient and ii) similarly simulation of some target parameter at some future time with what if scenarios (e.g. what would be CVD risk of patient in 10 year if his HbA1c value dropped to ...). In the first alternative, actual values of parameters for the patient need to be supplied, in the second one, simulation values need to be supplied. In order to support both in POWER2DM while providing ease of use for clients, a common logic in this direction is implemented for all the services. The Web API's (KADIS Web API, MT2D-Marvel Web API, Risk Score WEB API) of the services supports all the input parameters of corresponding prediction/simulation mechanism but they are optional. If the client does not provide the parameter value, it means the latest value for patient will be used for the calculations. In this way, for simulations, clients only provide the simulated parameters.

¹³ <https://developers.google.com/analytics/devguides/reporting/core/v3/>

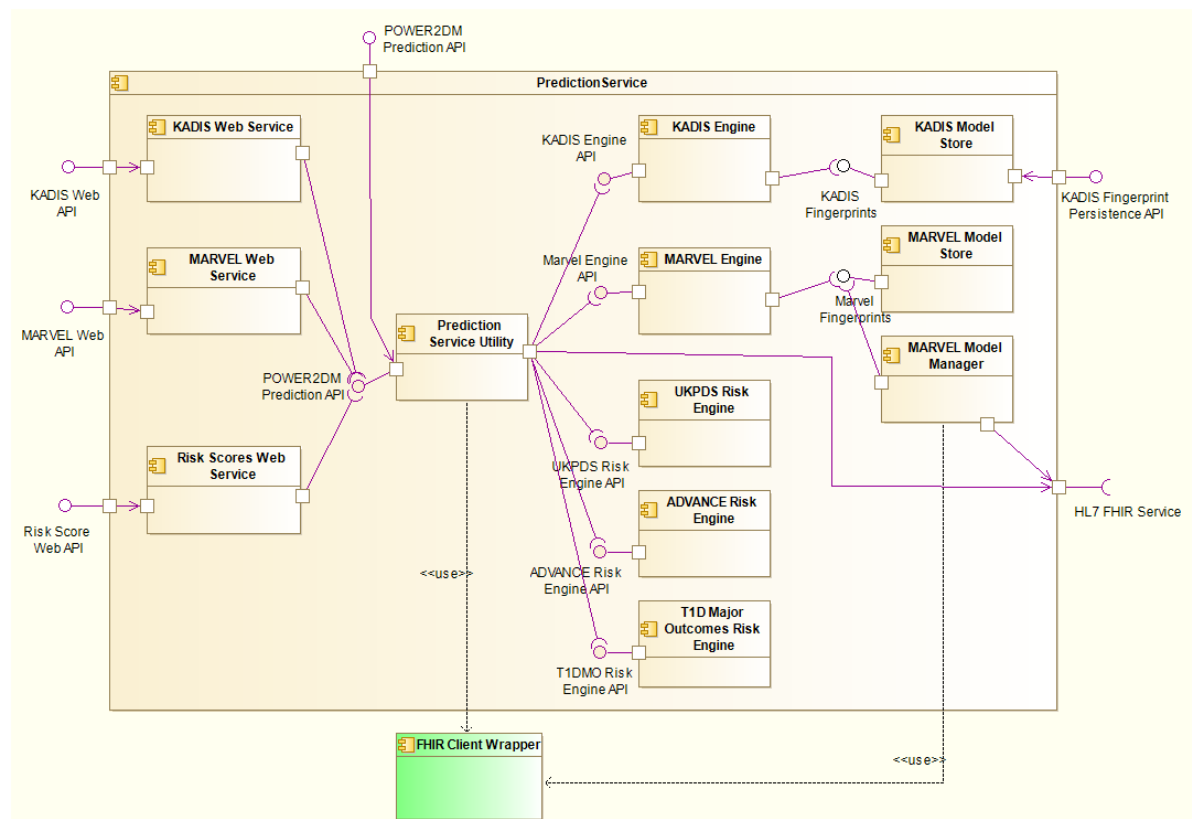


Figure 3 Sub components of POWER2DM Prediction Service

As shown in Figure 3, the **Prediction Service Utility** is the sub-component that handles this logic and gets the actual values of the parameters for patient that are not supplied by the client. It provides the **POWER2DM Prediction API** which is a JAVA wrapper for actual prediction/simulation engines that are implemented in different programming languages. This Java API is both used by the Web APIs of prediction services and other POWER2DM components in the same deployment unit. The Prediction Service Utility utilizes FHIR Client Wrapper library to access HL7 FHIR Service provided by POWER2DM Personal Data Store to access patient’s data. In this way, existing API’s provided by the prediction/simulation engines does not need to be changed and can operate in simple logic; get all the inputs and produce a result.

POWER2DM Prediction API
+ predictRiskScore(in pid: string, in model: string, in params: Map [*], in riskInterval: integer, in referenceDate: date): Risk Assesment
+ predictBloodGlucoseProfile(in pid: string, in predictedDay: date): PredictionResult
+ predictHealthProgress(in pid: string, in params: Map, in referenceDate: date, in referencePeriod: integer): PredictionResult

Figure 4 Java Interface for POWER2DM Prediction API

The following sections provide details regarding the subsystems; their composition, services, etc.

3.2.1 KADIS Subsystem

3.2.1.1 Composition

As illustrated in Figure 3, the following are the components of KADIS subsystem;

KADIS Web Service is the component that provides a Restful API (KADIS Web API) to handle external requests for KADIS predictions and simulations. In POWER2DM, this API will be used by POWER2DM UI components to perform the predictions or simulations and get their results in real time. As described above, the component delegates the request to Prediction Service which further delegates to KADIS Engine to perform the prediction/simulations.

KADIS Engine is the core component that performs the simulation/prediction. It provides the KADIS Engine API to get the prediction/simulation requests. It accesses to personalized model, the KADIS fingerprint prepared for the patient from the KADIS Model Store database. The existing KADIS Engine software (Java implementation) will be used for this component.

KADIS Model Store is a database to store KADIS Metabolic Fingerprints of patients which are the base models used for the KADIS predictions and simulations. The component provides the KADIS Fingerprint Persistence API which enables external authorized systems (original KADIS system setup at IDK premises) to store the prepared fingerprints after the manual KADIS identification process from baseline data.

For KADIS Engine and KADIS Model Store, an alternative design is demonstrated in Figure 5. This is shown in here as it is still under discussion how to wrap up existing KADIS system in POWER2DM for blood glucose profile simulations. This second alternative shows a service based integration where KADIS simulation software is packaged as a standalone service and PredictionService calls the services it provided; i) KADIS Restful Fingerprint Service to request a KADIS Fingerprint construction for a patient and ii) KADIS Restful Simulation Service to execute the simulations for a patient in the run time.

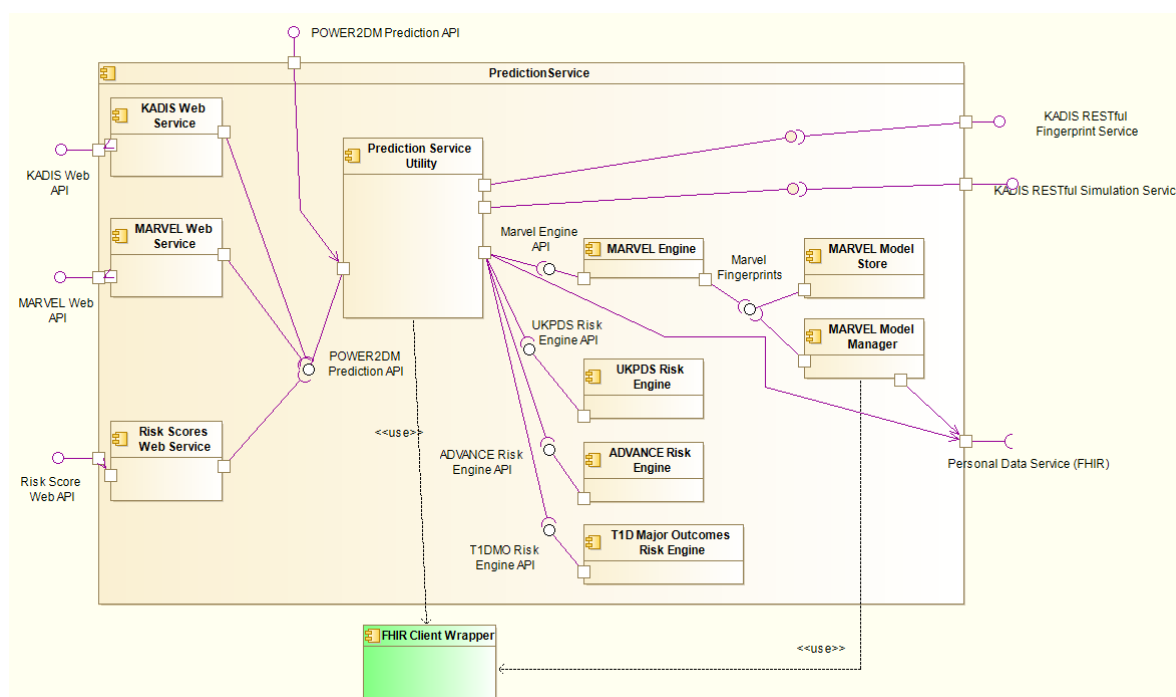


Figure 5 Variant B design for POWER2DM Prediction Services (KADIS RESTful Fingerprint Services for construction of Fingerprints, management of Fingerprints and KADIS RESTful Simulation Service performing of KADIS simulations)

3.2.1.2 Software Interfaces

3.2.1.2.1 KADIS Engine API (or KADIS Restful Simulation Service)

KADIS Engine API will provide a simple method for KADIS simulations for daily blood glucose profiles. The followings are the parameters of the method;

- **pid:** The identifier for the patient for whom the simulation will be performed. The identifier will be used to get KADIS Fingerprint of patient from KADIS Model Store and the changes in therapy (insulin, GLP1-analogues, tablets, CHO, exercise, body weight) to be tested by the simulation which come from the KADIS Web Service.
- **identificationDay:** The date (typical day) for which the KADIS Fingerprint was provided.
- **carbIntakes:** List of carbohydrate intakes for that day for the simulation. Each item will be a tuple of the following parameters;
 - **time:** Time of intake (e.g. 10:41)
 - **amount:** Amount of carbohydrate intake (in grams or bread unit) ?
 - **<optional: type>: class of glycemic index for CHO intake**
- **insulinIntakes:** List of insulin intakes for the simulation. Each item will be a tuple of the following parameters;
 - **time:** Time of intake (e.g. 10:41)
 - **amount:** Dosage of insulin in units.
 - **insulinType:** The coded value describing the type of insulin
- **GLP1-AnaloguesIntakes:** List of GLP1 analogues intakes for the simulation. Each item will be a tuple of the following parameters;
 - **time:** Time of intake (e.g. 10:41)
 - **amount:** Dosage of GLP1-analogues.
 - **GLP1-Analogues Type:** The coded value describing the type of GLP1-analogues
- **medicationIntakes:** List of antidiabetic medication intakes for the simulation. Each item will be a tuple of the following parameters;
 - **time:** Time of intake (e.g. 10:41)
 - **amount: number of tablets**
 - **dosage:** Dosage of medication (e.g. 500mg for Metformin).
 - **medicationType:** The coded value describing the type of medication
- **exercises:** List of exercises done/planned in that day. Each item will be a tuple of the following parameters;
 - **time:** Start time for exercise
 - **duration:** Duration of performing exercise in minutes
 - **intensity:** Intensity of exercise (three categories: low, normal, high)

Input parameters for KADIS Simulation Engine (variant A, Figure 3);

Parameter Name	Type	Description	Allowed values
identifier	string	Patient identifier (PSIP)	
expiration	date	Expiration date of the KADIS Fingerprint. Individualized after construction of the Fingerprint, depends on variability and value of Q-Score. Expiration date is part of the KADIS Fingerprint	
Type of diabetes	string		Type-1, Type-2, LADA
Diabetes since	date (year)	Year of diabetes onset	
Body weight	integer	kg	
Body height	integer	cm	
Age	integer		>= 18
Therapy Data (changes to the actual therapy (Fingerprint) to be tested by simulation)			
carbIntakesPlan	list	List of carbIntakes (time, amount, <glyx>)	
oatIntakesPlan	list	List of medicationIntakes (time, amount, dosage, drug type)	

insulinIntakesPlan	list	List of insulinIntakes (time,amount,insulinType,)	
glp1IntakesPlan	list	List of GLP1-AnaloguesIntakes (time,amount,type)	
exercisesPlan	list	List of exercises (time,duration,intensity)	
Model data (provided by the KADIS Fingerprint as requirement for each simulation run)			
modelData	object	object of ^KAD.MODEL (et al Model parameter of the differential equations)	
insulinFaktors	list	List of ^PROFIL.INSw (calculated 24h profile of the insulin responsiveness)	

Input parameters for KADIS RESTful Simulation Service (variant B, Figure 5);

Parameter Name	Type	Description	Allowed values
identifier	string	Patient public-ID	
Type of diabetes	string		Type-1,Type-2,LADA
Diabetes since	date (year)	Year of diabetes onset	
Body weight	integer	kg	
Body height	integer	cm	
Age	integer		>= 18
Therapy Data (changes to the actual therapy (Fingerprint) to be tested by simulation)			
carbIntakesPlan	list	List of carbIntakes (time,amount,<glyx>)	
oadIntakesPlan	list	List of medicationIntakes (time,amount,dosage,drug type)	
insulinIntakesPlan	list	List of insulinIntakes (time,amount,insulinType,)	
glp1IntakesPlan	list	List of GLP1-AnaloguesIntakes (time,amount,type)	
exercisesPlan	list	List of exercises (time,duration,intensity)	

Output parameters for KADIS RESTful Simulation Service (variant B, Figure 5);

Parameter Name	Type	Description
identifier	string	Patient public-ID
Type of diabetes	string	
Diabetes since	date (year)	Year of diabetes onset
Body weight	integer	kg
Body height	integer	cm
Age	integer	
Gender	string	
KADIS simulation, Q-Score and Self Control Data		
glucosePROFILEsim	list	List of glucosePROFILEsim (time,BG_value)
insulinPROFILEendo	list	List of insulinPROFILEendo (time, value), only diabetes type 2 (endogenous ins.)
insulinPROFILEexo	list	List of insulinPROFILEexo (time, value), exogenous insulin
glp1PROFILE	list	
carbPROFILE	list	List of carbPROFILE(time, value)
exercisePROFILE	list	List of exercisePROFILE (time, value)
QScore	float	Calculated QScore
QscoreComponents	list	List of components(MBG,Range,thyper,thypo,MODD)
sdata	list	List of self control data (time,type,value)
simulation	file	SVG-file of the KADIS Simulation

3.2.1.2.2 POWER2DM Prediction API – Blood Glucose Profile Prediction (KADIS Part)

The Figure 4 shows the simple method **predictBloodGlucoseProfile** which is a wrapper for KADIS Engine API (variant A, figure 2) or which is the request of a glucose profile prediction (KADIS RESTful service, variant B, figure 2.1) for ease of use. The Prediction Service Utility retrieves all the

information (carbohydrate intakes, insulin/medication intakes) for the patient for the given day from POWER2DM Personal Data Store and prepare the required input for the KADIS Engine API/KADIS Restful service.

3.2.1.2.3 KADIS Fingerprint Persistence API (if original design is used)

Resource Type “KADISFingerprint”	
Allowed HTTP Methods	POST
Path incl. path parameters	/KadisFingerprint
POST Data	Raw serialized data model for KadisFingerprint
Success POST Response	200 OK
Error Responses	400 Bad Request, if request is not parsable or valid 401 Unauthorized, if user is unauthorized or authorization token is missing

3.2.1.2.4 KADIS RESTful Fingerprint Service (if Variant B is used)

If the KADIS simulation software will be wrapped as a standalone service outside the scope of POWER2DM Prediction Service, this service can be used to send Fingerprint construction requests to this standalone KADIS deployment.

Input parameters for KADIS RESTful Fingerprint Service (variant B, figure 2.1);

Parameter Name	Type	Description	Allowed values
identifier	string	Patient public-ID	
Type of diabetes	string		Type-1,Type-2,LADA
Diabetes since	date (year)	Year of diabetes onset	
Body weight	integer	kg	
Body height	integer	cm	
Age	integer		>= 18
Gender	string		“female”, “male”
BGMethod	string	Method used for monitoring phase	“CGM”, “SMBG”
Glucose measurements and Self Control Data			
glucoseSMBGMoni	list	List of SMBG (date,time,BG_value)	
glucoseCGMMoni	list	List of CGM (date,time,BG_value)	
carbIntakesMoni	list	List of carbIntakes (date,time,amount,<glyx>)	
oadIntakesMoni	list	List of medicationIntakes (date,time,amount,dosage, drug type)	
insulinIntakesMoni	list	List of insulinIntakes (date,time,amount,insulinType,)	
glp1IntakesMoni	list	List of GLP1-AnaloguesIntakes (date,time,amount, type)	
exercisesMoni	list	List of exercises (date,time,duration,intensity)	

Example as JSON;

```
{
  "identifier": "Qs234xA#97zL",
  "diabtype": "Type 1",
  "diabsince": 1997,
  "height": 181,
  "weight": 98,
  "age": 63,
  "gender": "m",
  "bgmethod": "SMBG",
  "scData": [
    {

```

```

    "datetime": "18082016 14:15:",
    "smbg": 124,
    "meal": "48",
    "insuline": [
      {
        "name": "Apidra",
        "form": "short",
        "mix": "",
        "insdos": 12
      }
    ],
    "oad": [
      {
        "drug": "Metformin",
        "oaddos": 1000,
        "number": 0.5
      }
    ],
    "glp1": {
      "drug": "Liraglutide",
      "glpdos": 1.8
    },
    "exercise": {
      "duration": 60,
      "intensity": "normal"
    }
  }
}

```

3.2.1.3 Relation with Existing Products

Existing simulation mechanism, model storage mechanism of KADIS System will be used for the KADIS Engine and KADIS Model Store components (Variant A, figure 2). For this purpose, an instance of the platform TeleDIAB® (InterSystems Caché) the KADIS Simulation and KADIS Model Store based on has to be installed for POWER2DM.

3.2.1.4 KADIS Simulation Flow

Figure 6 shows the UML sequence diagram for KADIS simulation flow starting by a request from the other POWER2DM component (in this example, the Communication Engine).

1. The client component calls the predictBloodGlucoseProfile method of POWER2DM Prediction API with patient id and the target day for simulation.
2. The Prediction Service Utility make several REST calls to Personal Data Store to get all the required information for KADIS simulation including dietary intakes, medication administrations, action plans, etc.
3. When all the information is ready, Prediction Service Utility calls the internal KADIS Engine API for the simulation.
4. KADIS Engine API retrieves the latest personal KADIS Fingerprint of the patient and perform the simulation with the supplied parameters.
5. Prediction Service Utility constructs the PredictionResult resource object based on the returned results and return back to the client.

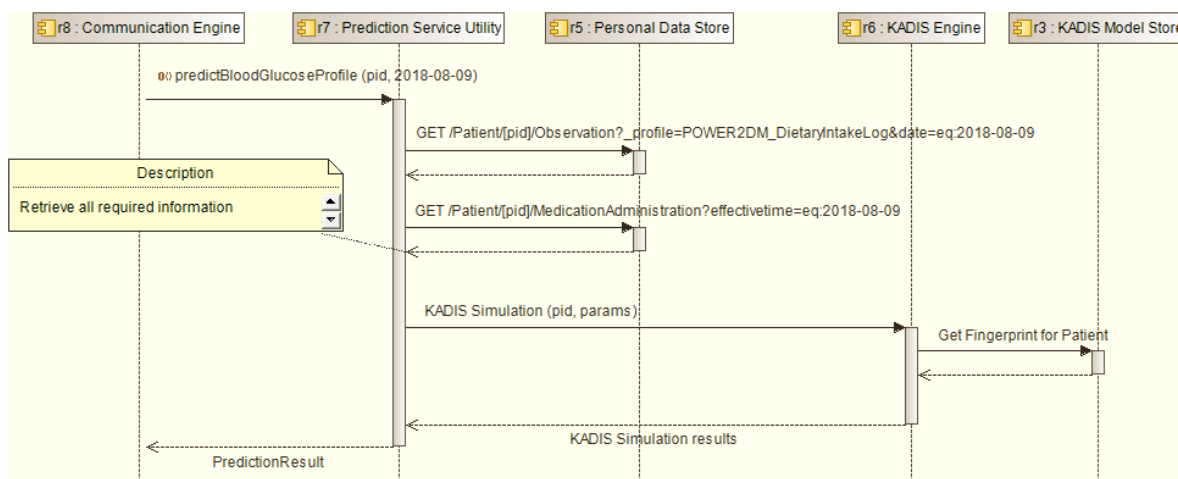
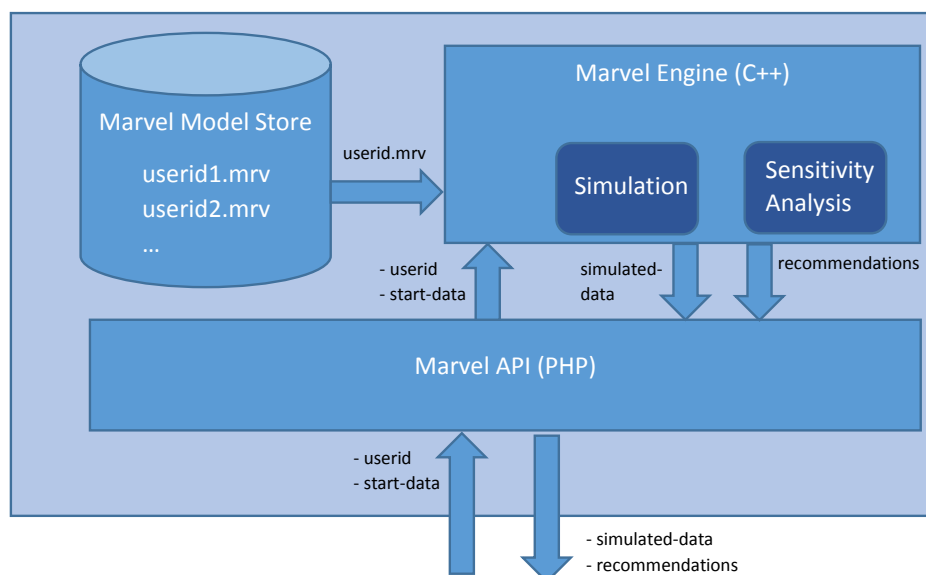


Figure 6 KADIS Simulation Flow

3.2.2 MT2D-Marvel Subsystem

3.2.2.1 Composition

The composition of the MT2D-Marvel system is graphically depicted in the figure below. The MT2D-Marvel API, implemented in PHP, handles the external requests and controls the internal MT2D-Marvel components. Upon a valid request, the MT2D-Marvel API selects the appropriate patient-specific model from the MT2D-Marvel Model Store and activates the MT2D-Marvel Engine with that model and the start-data provided by/along with the request. The MT2D-Marvel Engine, implemented in C++, uses the provided model and start-data to 1) simulate the responses of all variables in the model over time and to 2) perform a sensitivity analysis to determine a ranked list of the control variables that are recommended as best interventions. In addition, the sensitivity analysis may be used to guide an ensemble prediction to assess the prediction uncertainty.



3.2.2.2 Software Interfaces

MT2D-Marvel Software Interfaces will be available via the MT2D-Marvel API

Resource Type “SimulateAndRecommend”

Allowed HTTP Methods	POST
Path incl. path parameters	/marvel/simulate
POST Input	<pre>{ "action": "simulateandrecommend", "project": "power2dm", "userid": idofuser, "start-data": [{"variable name 1": value11}, {"variable name 2": value21}, ...], }</pre>
Success POST Response	<pre>{ "status": "success", "simulated-data": [{"variable name 1": [value11,value12, ...]}, {"variable name 2": [value21,value22, ...]}, ...], "recommendations": [{"name": nameofvariable1, "rank": rankofvariable1}, {"name": nameofvariable2, "rank": rankofvariable2}, ...], }</pre>
Error POST Response	<pre>{ "status": "error", "errornr": numberoferror, "errormessage": messageoferror, }</pre>
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

The variable names are as follows:

Variable name 1	BMI
Variable name 2	Fasting Insulin
Variable name 3	Fasting Glucose
Variable name 4	Insulin Sensitivity
Variable name 5	Beta cell function
Variable name 6	Inflammation
Variable name 7	Tissue damage
Variable name 8	Gut Health
Variable name 9	Chronic stress levels
Variable name 10	Peripheral energy overload
Variable name 11	Fasting glucose overload
Variable name 12	BMI threshold
Variable name 13	Fasting glucose threshold

The API expects input data for all of the above mentioned variables plus in addition for the following control variables:

Input variable name 1	Age
Input variable name 2	Food Intake pattern
Input variable name 3	Exercise pattern
Input variable name 4	Food Quality
Input variable name 5	Sleep, Meditation, relaxation
Input variable name 6	Insulin use
Input variable name 7	Medication use

The format of input data will be a sequence of weekly averages (year, week number, value). When a given week has no entry, the MT2D-Marvel service needs to impute a value.

3.2.2.3 Required Software Interfaces

Dependencies:

Internal systems:

- ODL Service (Personal Data Store)

- Communication Engine

External systems:

- MT2D-Marvel Fingerprint Construction

3.2.3 Risk Score Service Subsystem

3.2.3.1 Composition

The POWER2DM Risk Score Service consists of the following sub-components.

Risk Score Web Service is the component that provides a common Restful API (Risk Score Web API) to handle external requests for several risk score predictions and simulations to be used in POWER2DM. The API will be used by POWER2DM UI components to perform risk score predictions or simulations and get the results of them immediately. By utilizing synchronous service provided with POWER2DM Prediction API, the component delegates the prediction request to Prediction Service Utility after parsing and validation of the input and return back when the result is returned.

UKPDS Risk Engine is the implementation of UKPDS Risk Model mechanism¹⁴. Existing software will be utilized and the Prediction Service Utility will provide a Java wrapper for the mechanism.

Advance Risk Engine is the implementation of Advance Risk Model mechanism¹⁵. Existing software will be utilized and the Prediction Service Utility will provide a Java wrapper for the mechanism.

T1D Major Outcome Risk Engine is the implementation of the Major Outcomes in T1D Model mechanism¹⁶. Existing software will be utilized and the Prediction Service Utility will provide a Java wrapper for the mechanism.

3.2.3.2 Service Interfaces

3.2.3.2.1 Risk Score Web API

Resource Type “RiskAssesment”	
Allowed HTTP Methods	POST
Path incl. path parameters	For patient specific evaluation; /Patient/{patient-id}/RiskAssesment/\$evaluate For general simulation; /RiskAssesment/\$evaluate
POST Data	See e.g. What will be patient’s CVD Risks in 10 years calculated by UKPDS model if his HbA1c will be dropped to 6.8% <pre>{ "model": "ukpds", "risk-years": 10, "parameters": { "hba1c": 6.8 } }</pre>
Success POST Response	200 OK with response body having the JSON formatted POWER2DM RiskAssesment Resource (See D4.1- Section 3.5.5.2 for details).

¹⁴ <https://www.dtu.ox.ac.uk/riskengine/>

¹⁵ <http://www.advanceriskengine.com/>

¹⁶ <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4399797/>

Error Responses	400 Bad Request, if request is not parsable or valid 401 Unauthorized, if user is unauthorized or authorization token is missing 404 Not Found, if risk assessment model is not supported 500 Internal Server Error in case of internal exception A JSON object, OperationOutcome, describing the error will be returned in the response body. See ¹⁷ for details.
-----------------	---

The above table shows the details of the Restful Web API for Risk Score Service. Risk assessment may be done in two ways. In the first alternative, the risk assessment is done for a specific patient and if a listed parameter is not provided for the specified risk assessment model then the calculation is done based on the patient's latest value for that parameter (if exist). In the second alternative, the service is used for simple simulation and all the parameters should be provided.

Table 2 Parameters for RiskScore Web API Post Data

Parameter Name	Type	Description	Allowed values
model	string (R)	The method/model to be used for the risk assessment.	ukpds advance-card advance-kidney mot1d
parameters	Map (R)	The parameters required for the model execution.	See Table 3, Table 4, Table 5, Table 6 for the input parameters for each model.
risk-years	int (O)	The risk interval for calculation. If not supplied or model does not support the calculation for provided year interval, the default for the model will be applied.	
reference-date	date (O)	If this parameter is given, then evaluation is done based on the data available before the given date for the parameters not supplied for patient in the parameters element.	In xs:date (yyyy-mm-dd) format

Table 3 Parameters for UKPDS Risk Engine

Parameter Name	Description
age	Age now (years)
diabetes-duration	Diabetes (Type 2) duration (years)
gender	Gender: (male female)
atrial-fibrillation	Atrial fibrillation: (No/Yes)
ethnicity	Ethnicity: (white afro-caribbean asian-indian)
smoking	Smoking (non-smoker ex-smoker current smoker)
hba1c	HbA1c: (%) (range: 2-20)
systolic-bp	Systolic BP (mm Hg)(range 60-250)
total-ch	Total cholesterol (mmol/l) (range 1-15)
hdl-ch	HDL cholesterol (mmol/l)

Table 4 Parameters for Advance Cardiovascular Risk Engine

Parameter Name	Description
gender	Gender (male female)
waist-circumference	Waist circumference (cm) (range 50-250)

¹⁷ <https://www.hl7.org/fhir/operationoutcome.html>

age-diagnosis	Age at diabetes diagnosis (years) (range 29-90)
diabetes-duration	Known duration of diabetes (years) (range 0-36)
systolic-bp	Systolic BP (mm Hg) (range 60-300)
diastolic-bp	Diastolic BP (mm Hg) (range 40-200)
hba1c	Hemoglobin A1c (%) (range 3-20)
uacr	UACR (microgram/mg) (range 0.1-500)
non-hdl-ch	Non-Hdl Cholesterol (mmol/L) (range 0-12)
treated-hypertension	Treated hypertension (No/Yes)
retinopathy	Retinopathy (No/Yes)
atrial-fibrillation	Atrial fibrillation (No/Yes)

Table 5 Parameters for Advance Kidney Risk Engine

Parameter Name	Description
gender	Gender (male female)
ethnicity	Ethnicity (Non-Asian Asian)
waist-circumference	Waist circumference (cm) (range 50-250)
systolic-bp	Systolic BP (mm Hg) (range 60-300)
diastolic-bp	Diastolic BP (mm Hg) (range 40-200)
glomerular-filtration-rate	Glomerular filtration rate (mL/min/1.73 mm ²) (range 15-150)
hba1c	Hemoglobin A1c (%) (range 3-20)
uacr	UACR (microgram/mg) (range 0.1-500)
treated-hypertension	Treated hypertension (No/Yes)
retinopathy	Retinopathy (No/Yes)
age-education	Age at completion of formal education (<=15y / >= 16y)

Table 6 Parameters for Major Outcomes in T1D Model

Parameter Name	Description
age	Age (years) (range 20-60)
glycated-haemoglobin	Glycated Haemoglobin (%) (range 4-16)
waist-hip-ratio	Waist-hip ratio (range 0.6 – 1.4)
alb-creat-ratio	Albumin/creatinine ratio (mg/mmol) (range 0.1– 400)
hdl-ch	HDL-cholesterol (mmol/L) (range 0-4)

The service returns the JSON representation of POWER2DM RiskAssesment resource object for which the data model is defined in D4.1. For each risks assessment models, several outcomes (path in the object RiskAssesment.prediction.outcome) are predicted.

For UKPDS Risk engine for the requested risk interval:

- CHD risk (% and margin of error)
- Fatal CHD (% and margin of error)
- Stroke (% and margin of error)
- Fatal Stroke (% and margin of error)

For Advance Cardiovascular Risk Engine (only 4-year risk interval is calculated);

- Predicted risk (% , 1 decimal) of major cardiovascular disease defined as fatal or non-fatal myocardial infarction or stroke or cardiovascular death

For Advance Kidney Risk Engine (only 5-year risk interval is calculated);

- Predicted risk (% , 1 decimal) of New-onset albuminuria (urinary ACR >= 30 microgram/mg)
- Predicted risk (% , 1 decimal) of Major kidney-related events:

(doubling of serum creatinine to $\geq 2.26\text{mg/dL}$, renal replacement therapy, or renal death)

For the Major Outcomes in T1D Model:

- The Predicted risk of major outcomes (%) at 3years (integer value)
- The Predicted risk of major outcomes (%) at 5years (integer value)
- The Predicted risk of major outcomes (%) at 7years (integer value)

Major Outcomes include major CHD, stroke, end-stage renal failure, amputations, blindness and all-cause death. Major CHD comprises fatal CHD, non-fatal myocardial infarction (coded according to ICD-9 410-414; [www.icd9data.com/ 2007/Volume1](http://www.icd9data.com/2007/Volume1)) and major Q waves on Minnesota-coded ECGs (codes 1.1, 1.2). Stroke (ICD-9 430-438) includes fatal and non-fatal stroke. End-stage renal failure comprises renal dialysis or renal transplantation.

3.2.3.2.2 POWER2DM Prediction API – Risk Score Prediction

As shown in Figure 4, a wrapper JAVA method, **predictRiskScore**, is provided by the POWER2DM Prediction interface to perform different risk score calculations. The design of the API is same with the Risk Score WEB API as it will delegate the request to the Prediction Service Utility over this JAVA API.

3.2.3.3 Relation with existing products

Implementation will be based on published algorithms that are incorporated in existing desktop or web interfaces as specified below:

1. The UKPDS Risk engine:
POWER2DM uses the UKPDS risk engine v2.0: <https://www.dtu.ox.ac.uk/riskengine/>
Full details of the equations have been published:
<http://www.dtu.ox.ac.uk/generic/publications.php?Section=92>
At present the model is only available as an executable file for Windows and Macintosh, or upon a successful licensing request, as an as an ActiveX module for incorporation in other software packages.
2. The Advance Risk Engine
POWER2DM uses the ADVANCE risk engine: <http://www.advanceriskengine.com/> which comes in 2 variants:
 - a. The Cardiovascular Risk Engine
 - b. The ADVANCE Kidney Risk Engine
 Both models are only available as a web service requiring manual input.
Details of the model equations can be searched via links provided on
<http://www.advanceriskengine.com/links.html>; <http://www.advance-trial.com/>;
3. The Major Outcomes in T1D Model
The model equations are published in Soedamah-Muthu et al. (2014), “Predicting major outcomes in type 1 diabetes: a model development and validation study”. Diabetologia 57: 23-4-2314.

!!! Open Issue: None of the risk engines is available with a API such as required for POWER2DM. A possible solution is to research the algorithms from the literature and implement the models plus the corresponding APIs ourselves.

3.2.3.4 Risk Score Prediction Flow

Figure 7 shows the UML sequence diagram for Risk Score Prediction Flow; starting from a POWER2DM UI component (POWER2DM Shared Decision Making Application as example) by a request.

1. UI component calls the Risk Score Web API to perform a risk score prediction for patient by sending the HTTP POST request (POST /Patient/-/RiskAssesment/\$evaluate).
2. Risk Score Web Service check if the user is authorized for the operation by validating the OAuth access token sent in the header. It also gets the identifier of the target patient from the token. See “D4.9 Privacy and Security Enablers” for details.
3. Then the request is parsed and checked if it is valid (parameters are OK).
4. The predictRiskScore method of POWER2DM Prediction API provided by the Prediction Service Utility is called to delegate the simulation request.
5. The Prediction Service Utility by checking the model parameter understands which risk score engine will be used and which parameters are required for the risk score prediction. If a parameter is not supplied in the request, it calls the FHIR Service provided by Personal Data Store to collect the data; e.g. get basic patient data (gender, ethnicity, etc) from Patient resource, get diabetes related info (duration of diabetes, etc) from Diabetes Anamnesis resource, get HbA1C observations, etc
6. After collecting all the latest information, it calls the internal API provided by UKPDS Risk Engine and performs the simulation.
7. When the results are available, it constructs the POWER2DM RiskAssessment resource object including the results, used input parameters, etc and return the object to Risk Score Web Service.
8. Risk Score Web Service return the result object in the HTTP Response body.

For internal POWER2DM components deployed together with Prediction Service (e.g. Communication Engine), the same flow can start directly by a request to the Prediction Service Utility.

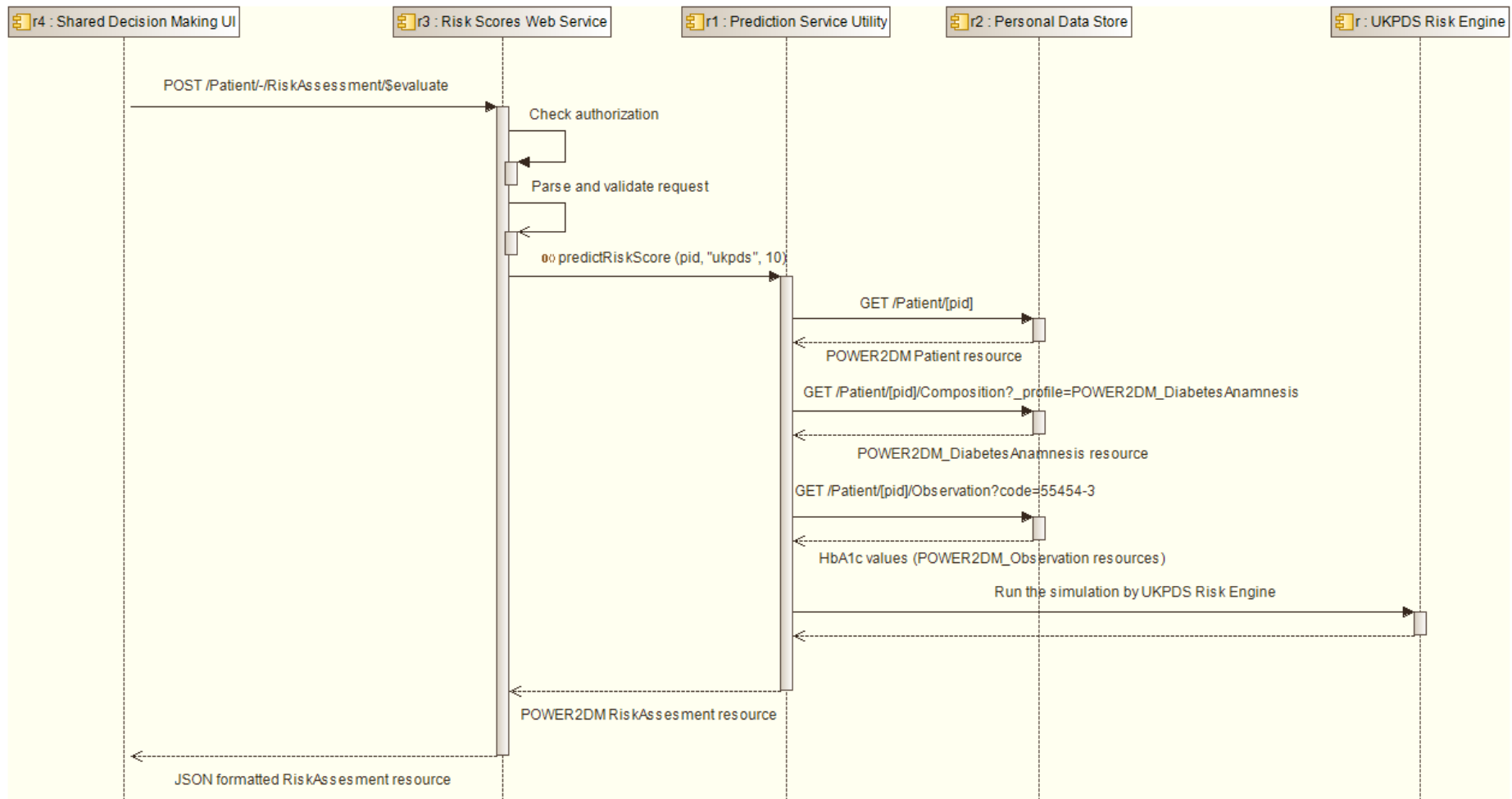


Figure 7 Risk Score Prediction Flow

3.3 POWER2DM Action Plan Engine

3.3.1 Composition

Following the overall architectural approach, the POWER2DM Action Plan Engine provides its services on two layers: The Action Plan Web Service provides access to resources for external components such as the User Interfaces (UIs). Internally it will use a Service Layer for retrieving the information from the Personal Data Store (via PDMClient).

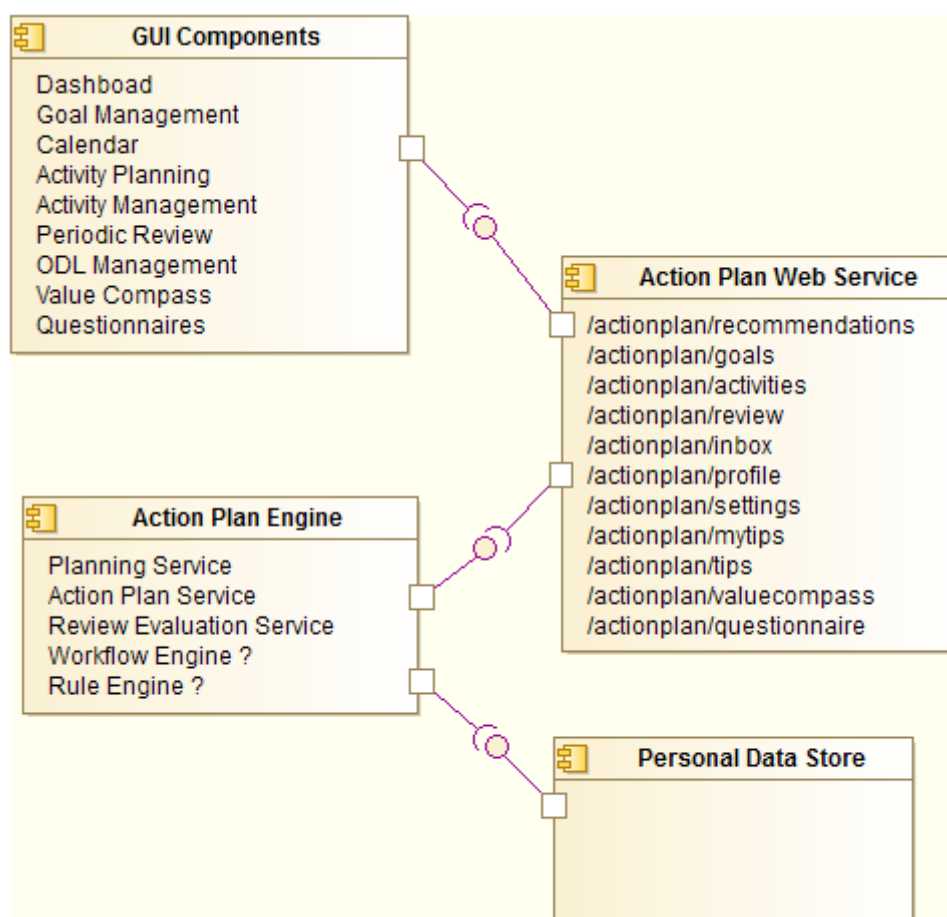


Figure 8 Overview Diagram Action Plan Engine

The major components of the POWER2DM Action Plan are:

- **Action Plan Service:** The action plan service provides CRUD operations on various resources, such as activities, goals, recommendations or tips.
- **Planning Service:** The planning service provides an abstraction to backend queries for supporting the action planning.
- **Review Evaluation Service:** The Review Evaluation Service provides an abstraction to backend queries for the review evaluation.
- **Workflow Engine:** Part of the Action Plan Engine will be a Workflow Engine, which is able to execute and monitor workflows for modelling the current status of the patient in his self-management pathway. The workflow engine needs to store the state for different workflows, independent of the current session. Several workflows are planned to be implemented, and will be further specified in WP3. Figure 9 Overview Action Plan Engine WorkflowsFigure 9 provides a first overview on the planned workflows. As an example, the model of the tip workflow could comprise the following steps:

1. deliver tip
 2. get user feedback/acknowledgement
 3. remind user if they do not read
 4. offer choices to categorize tip
- **Rule Engine:** The rule engine will monitor the status of the patient data/activities carried out by the patient and will fire events to the communication engine.

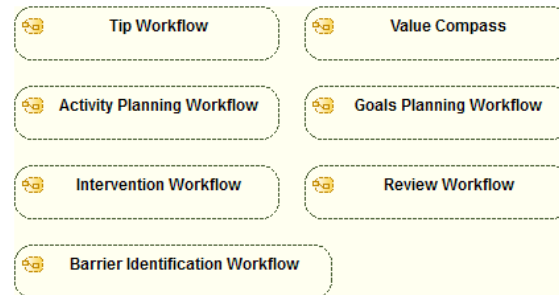


Figure 9 Overview Action Plan Engine Workflows

3.3.2 Software Interfaces

In this section we show the software interfaces of the Action Plan Engine on two layers, the Java API, mainly for internal use, and the web service API, which can be access by the various user interfaces.

3.3.2.1 Java API

3.3.2.1.1 Action Plan Service

Modifier and Type	Method and Description
void	deleteActivity(String uri, boolean mock)
void	deleteGoal(String uri, boolean mock)
void	deleteRecommendation(String uri, boolean mock)
List<ActivityViewImpl>	getActivities(DateTime from, DateTime to, boolean mock)
IActivityView	getActivityByUri(String uri, boolean mock)
GoalView	getGoalByUri(String uri, boolean mock)
List<GoalView>	getGoals(DateTime from, DateTime to, boolean mock)
RecommendationView	getRecommendationByUri(String uri, boolean mock)
List<RecommendationView>	getRecommendations(boolean mock)
IReview	getReview(DateTime from, DateTime to, boolean mock)
void	setActivity(String uri, ActivityViewImpl activity, boolean mock)
void	setGoal(String uri, GoalView goal, boolean mock)
void	setRecommendation(String uri, RecommendationView recommendation, boolean mock)
List<Workflow>	getWorkflows(StatusType status, WorkflowCategory category, FilterInterval interval, boolean mock)
List<Task>	getTasks(StatusType status, FilterInterval interval, boolean mock)
List<Task>	getTasksOfWorkflow(String workflowId, FilterInterval interval, Boolean mock)
void	setStatus(String id, StatusType);

3.3.2.1.2 Planning Service

Modifier and Type	Method and Description
List<IActivityView>	getActivitiesNoEventsForScheduleInterval(String ownerId, FilterInterval filterInterval, boolean mock)
ActivityEventCollector	getActivityEventCollector(String ownerId, FilterInterval filterInterval, ActivityEventScope scope, boolean mock)

List<IActivityView>	getActivitiesForScheduleInterval (String ownerId, FilterInterval filterInterval, ActivityEventScope scope, Map<String, Object> attributes, boolean mock)
List<IActivityView>	getActivitiesForScheduleInterval (String ownerId, FilterInterval filterInterval, ActivityEventScope scope, boolean mock)
IActivityView	getActivityWithScheduledEvents (String ownerId, String identifier, FilterInterval filterInterval, boolean mock)

3.3.2.1.3 Review Evaluation Service

Modifier and Type	Method and Description
ReviewEvalView	getReview (String ownerId, DateTime from, DateTime to, boolean mock)
ReviewEvalView	getReview (String ownerId, FilterInterval filterInterval, boolean mock)
ReviewEvalView	getReview (String ownerId, String fromStr, String toStr, boolean mock)
List<InfoTipView>	getReviewInfoTips (String ownerId, FilterInterval interval, boolean mock)
List<InfoTipView>	getReviewInfoTips (String ownerId, String from, String to, boolean mock)

3.3.2.2 Web Service API

3.3.2.2.1 Action Plan Webservice

Resource Type “Recommendations”	
Allowed HTTP Methods	GET, POST, DELETE
Path incl. path parameters	/actionplan/recommendations/{ownerId}
Query parameters	?uri=URI-of-requested-recommendation&mock=true (“mock” parameter is optional for getting mockdata without authentication)
POST Data	{ "type": null, "subType": null, "title": null, "description": null, "comment": "", "recommendedBy": "diabetes folder from hospital", "dateTime": "2016-07-12T22:00:00.000Z", "text": "a test recommendation" }
Success GET Response	200 OK Delivers the requested JSON Object if URI present, otherwise an array of all entries [{ "created": 1468421291000, "lastModified": null, "identifier": "6a64a9c1-11db-4d33-8781-c472cf3c26d4", "status": null, "dateTime": 1468360800000, "type": null, "subType": null, "title": "", "description": null, "comment": "", "attributes": {}, "recommendedBy": "diabetes folder from hospital", "text": "another test recommendation", "relatedGoals": null }, ...[next entry]]
Success POST Response	200 OK empty
Success DELETE Response	200 OK empty
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

Resource Type “Goals”	
Allowed HTTP Methods	GET, POST, DELETE
Path incl. path parameters	/actionplan/goals/{ownerId}
Query parameters	?uri=URI-of-requested-goal&mock=true (“mock” parameter is optional for getting mockdata without authentication)

POST Data	<pre>{ "type": null, "subType": null, "title": "relax regularly", "description": "walking, gardening, reading a book or similar", "comment": "", "motivation": "some motivation", "relatedRecommendations": ["6dcbb6ab-8f28-488c-a18e-223182c8a95e", "b05943b9-d2f7-45f5-b0b2-f6070c83c59f"], "dateTime": "2016-07-17T22:00:00.000Z" }</pre>
Success GET Response	<p>200 OK</p> <p>Delivers the requested JSON Object if URI present, otherwise an array of all entries</p> <pre>[{ "created": 1456908724000, "lastModified": 1458042524000, "identifier": "4e3aae49-fad1-4467-ba13-76908fe97455", "status": null, "dateTime": 1456786800000, "type": null, "subType": null, "title": "relax regularly", "description": "walking, gardening, reading a book or similar", "comment": "", "attributes": {}, "motivation": "some motivation", "relatedRecommendations": ["57fbda8e-520a-420b-85fd-c597d7aa7f52", "b05943b9-d2f7-45f5-b0b2-f6070c83c59f"], "relatedActivities": [], "validFrom": null, "validTo": null, "active": true }, ...[next entry]]</pre>
Success POST Response	<p>200 OK</p> <p>empty</p>
Success DELETE Response	<p>200 OK</p> <p>empty</p>
Error Responses	<p>401 Unauthorized, if user is unauthorized</p> <p>403 Forbidden, if user is not authenticated</p> <p>500 Internal Server Error in case of internal exception</p>

Resource Type “Activities”	
Allowed HTTP Methods	GET, POST, DELETE
Path incl. path parameters	/actionplan/activities/{ownerId}
Query parameters	<p>?uri=URI-of-requested-activity&from=start-date&to=end-date&merge=true&mock=true</p> <p>(“merge” parameter is optional (default=false) for getting activities merged with planned and matched/unmatched ODL results)</p> <p>(“mock” parameter is optional for getting mockdata without authentication)</p>
POST Data	<pre>{ "type": "EXERCISE", "subType": "i18n_exercise_walking", "title": "short city walk", "description": "doing 2km walk to the city center and back", "relatedGoals": ["4e3aae49-fad1-4467-ba13-76908fe97455"], "schedules": [{ "allDay": false, "dateTime": "1980-01-01T16:30:00.000Z", "endDateTime": "1980-01-01T17:15:00.000Z", "weekdays": [false, true, false, true, false, true, false], "validFrom": "2016-07-17T22:00:00.000Z", "validTo": "2016-07-24T21:59:59.999Z" }] }</pre>

	}
Success GET Response	200 OK Delivers the requested JSON Object if URI present, otherwise an array of all entries <pre>[{ "created": 1468833933000, "lastModified": null, "identifier": "abbc8d48-6e57-48f0-adf2-d97ablec9fe8", "status": null, "dateTime": null, "type": "EXERCISE", "subType": "i18n_exercise_walking", "title": "short city walk", "description": "doing 2km walk to the city center and back", "comment": null, "attributes": {}, "reminder": { "active": false, "mail": false, "minutesBefore": 0 }, "schedules": [{ "created": 1468833933000, "lastModified": null, "identifier": "194fa655-8f46-44d1-81a8-9543bb5c8d2c", "status": null, "dateTime": 315592200000, "type": null, "subType": null, "title": "", "description": null, "comment": null, "attributes": {}, "allDay": false, "weekdays": [false, true, false, true, false, true, false], "weekdaysString": "TU,TH,SA", "endDateTime": 315594900000, "validFrom": 1468792800000, "validTo": 1469397599000, "activityId": "abbc8d48-6e57-48f0-adf2-d97ablec9fe8" }], "relatedGoals": ["4e3aae49-fad1-4467-ba13-76908fe97455"] }, ...[next entry]]</pre>
Success POST Response	200 OK Empty
Success DELETE Response	200 OK Empty
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

Resource Type “Review”	
Allowed HTTP Methods	GET
Path incl. path parameters	/actionplan/review/{ownerId}
Query parameters	?from=start-date&to=end-date
POST Data	n/a
Success GET Response	200 OK <pre>{ "lang": null, "ownerId": "615de849-f2f6-4ecb-abf5-c62b03a577b2", "dateFrom": null, "dateTo": null, "typeActivities": {...}, "typeActivitiesGoals": {...}, "tips": {}, "completed": 1, "expected": 28, "score": 3.571428571428571, "scoreLikertInt": 1, "scoreLikert": "i18n_likert_1", </pre>

	<pre> "overview": { "category": "ALL", "type": null, "subType": null, "key": null, "completed": 1, "expected": 28, "score": 3.571428571428571, "scoreLikert": "i18n_likert_1", "scoreLikertInt": 1, "referenceType": null, "references": {}, "tips": {}, "titles": {} }, "category": "ALL" } </pre>
Success POST Response	n/a
Success DELETE Response	n/a
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

Resource Type “Inbox”	
Allowed HTTP Methods	GET, DELETE
Path incl. path parameters	/actionplan/inbox/{ownerId}
Query parameters	?uri=URI-of-requested-message&mock=true (“mock” parameter is optional for getting mockdata without authentication)
POST Data	n/a
Success GET Response	200 OK Delivers the requested JSON Object if URI present, otherwise an array of all entries <pre> [{ "created": 1468836137023, "lastModified": null, "identifier": null, "status": null, "dateTime": null, "type": "tip", "subType": "medical", "title": null, "description": null, "comment": "this is your tip", "attributes": { }, "deliveryType": "now", "contentId": null, "properties": { }, "code": "", "reference": null, "tags": "" }, ...[next entry]] </pre>
Success POST Response	n/a
Success DELETE Response	200 OK
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

Resource Type “Profile”	
Allowed HTTP Methods	GET, POST
Path incl. path parameters	/actionplan/profile/{ownerId}
Query parameters	?mock=true (“mock” parameter is optional for getting mockdata without authentication)
POST Data	Subset of Parameters retrieved from GET Response
Success GET Response	200 OK <pre> { "created": 1457424555000, "lastModified": 1468834531000, "identifier": "615de849-f2f6-4ecb-abf5-c62b03a577b2_healthprofile", "status": null, "dateTime": null, "type": null, "subType": null, </pre>

	<pre> "title": "", "description": null, "comment": null, "attributes": {}, "personalData": { "i18n_height": "175", "i18n_gender": "i18n_male", "i18n_date_of_birth": "1979-12-31T23:00:00.000Z", "i18n_weight": "" }, "problems": { "comment": "", "problem": "[i18n_problem_nervousness]" }, "maturityLevels": {}, "risks": { "i18n_risk_smoker": "true" }, "conditions": { "i18n_condition_diabetes_type": "i18n_diabetes_type2", "i18n_condition_pregnant": "false", "i18n_condition_insulin_intake": "true", "i18n_condition_cholesterol": "false", "i18n_condition_cardiovascular_disease": "true", "i18n_condition_diabetic_foot": "false", "i18n_condition_kidney_disease": "false", "i18n_condition_retinopathy": "false" }, "medicationsCurrent": {}, "questions": { "i18n_basoc_comprehension": "i18n_basoc_likert_2", "i18n_basoc_manageable": "i18n_basoc_likert_4", "i18n_basoc_meaningful": "i18n_basoc_likert_4" } } </pre>
Success POST Response	200 OK
Success DELETE Response	n/a
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

Resource Type “Settings”	
Allowed HTTP Methods	GET, POST
Path incl. path parameters	/actionplan/settings/{ownerId}
Query parameters	?mock=true (“mock” parameter is optional for getting mockdata without authentication)
POST Data	Subset of Parameters retrieved from GET Response
Success GET Response	<pre> 200 OK { "created": 1457958969000, "lastModified": 1468406125000, "identifier": "615de849-f2f6-4ecb-abf5-c62b03a577b2_usersettings", "status": null, "dateTime": null, "type": null, "subType": null, "title": "", "description": null, "comment": null, "attributes": {}, "lang": "de", "style": "default", "bgUnit": null, "weightUnit": null, "heightUnit": null, "firstDay": 1.0, "firstHour": 7.0, "dashboard": { "dbActivities": "show", "dbCharts": "show" }, "otherSettings": { "tips": "max", "animations": "on" } } </pre>
Success POST Response	200 OK
Success DELETE Response	n/a
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

Resource Type “My Tips”	
Allowed HTTP Methods	GET, POST, DELETE
Path incl. path parameters	/actionplan/mytips/{ownerId}
Query parameters	?uri=URI-of-requested-tip&mock=true (“mock” parameter is optional for getting mockdata without authentication)
POST Data	{ “comment”: “Spend time with favorite hobby”, “relevance”: 3 }
Success GET Response	200 OK Delivers the requested JSON Object if URI present, otherwise an array of all entries [{ “created”: 1460557425000, “lastModified”: null, “identifier”: “ff8ee24b-2396-47da-9a47-9898b84ccf27”, “status”: null, “dateTime”: null, “type”: null, “subType”: null, “title”: “”, “description”: null, “comment”: “Spend time with favorite hobby”, “attributes”: {}, “relevance”: 3.0, “relevanceMinimum”: 0.0, “relevanceStringValue”: null, “relevanceProperty”: null, “odlType”: “STRESS”, “odlSubType”: null, “tags”: null }, ... [next entry]]
Success POST Response	200 OK
Success DELETE Response	200 OK
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

Resource Type “Tips”	
Allowed HTTP Methods	GET
Path incl. path parameters	/actionplan/tips/{ownerId}
Query parameters	?from=start-date&to=end-date&topic=topic&viewtype=viewtype&mock=true topic: Set of Strings, to which topic(s) the tips should be filtered viewtype: context where the tip should be displayed (now, review, dashboard) (“mock” parameter is optional for getting mockdata without authentication)
POST Data	n/a
Success GET Response	200 OK
Success POST Response	n/a
Success DELETE Response	n/a
Error Responses	401 Unauthorized, if user is unauthorized 403 Forbidden, if user is not authenticated 500 Internal Server Error in case of internal exception

!!! Open Issue: Tentative Webservice design for Value Compass & Questionnaires (depends on developments done in WP3)

3.3.3 Relation with Existing Products

Implementation will be based on:

- EMPOWER Action Plan
- DiabGender Action Plan

Dependencies:

- ODL Service (Personal Data Store)
- Communication Engine

3.4 POWER2DM Communication Engine

Communication Engine is the major component that will provide the intelligent service which decides and plans the digital interventions (e.g. motivations, reminders, warnings, interaction with questions, educational materials, recommendations, etc) to deliver to patient and initiate the communication with patient by sending push notifications to POWER2DM SMSS Mobile Application. The component achieves this by continuously analysing the stream of data collected from patient and extracting and recognizing patterns and correlations in patient's preferences, life-style, and medical data. The main aim is to achieve the concept, Just-In-Time Adaptive Interventions (JITAs; Nahum-Shani et al., 2014). JITAs will be the self-management interventions that are initiated by the Communication Engine automatically during daily life of the patient according to the changing context of the patient. The idea is to adapt the frequency, content and type of interventions accordingly. Examples of JITAs are: (a) warning that after a loaded breakfast the patient should have a light lunch in order to manage his blood glucose, or (b) patient only does half his planned exercise and receives a motivating message to finish the complete exercise.

Planned JITAs will be delivered to patients through;

- the dashboard that will be developed in POWER2DM SMSS Web Interface
- the push notifications for POWER2DM SMSS Mobile Application

3.4.1 Composition

Figure 10 illustrates the sub components of Communication Engine and provided and required software interfaces.

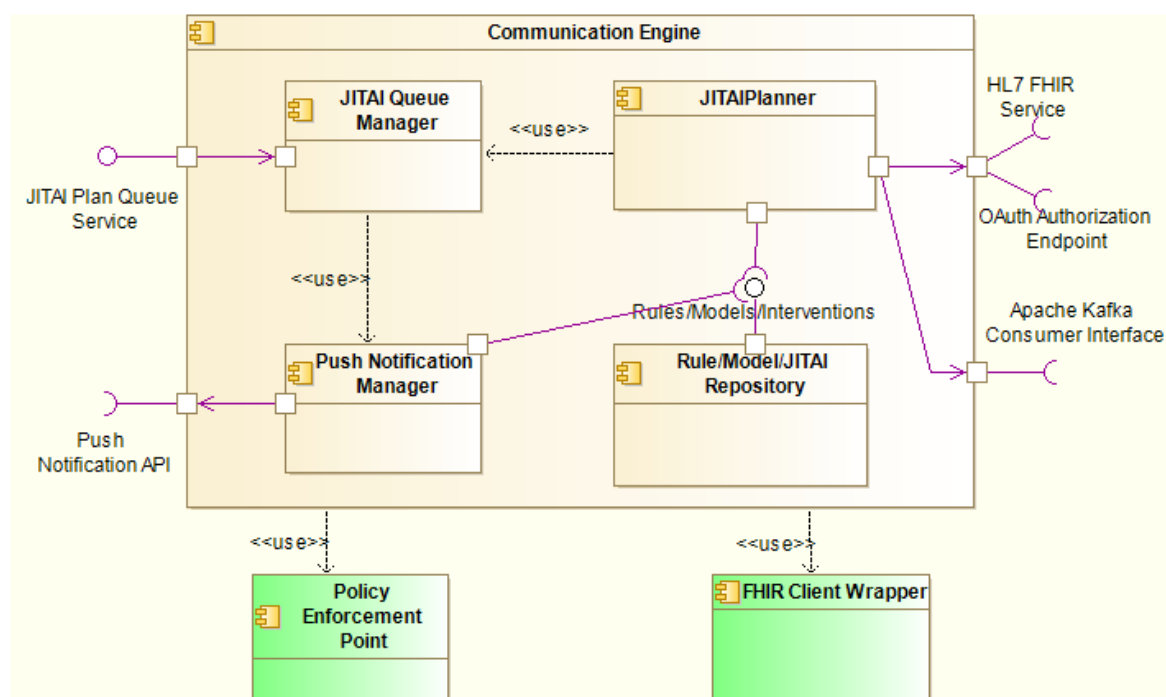


Figure 10 Sub-components of Communication Engine subsystem

JITAI Queue Manager component manages the queue of intervention plans prepared by the JITAI planner and provide the JITAI Plan Queue service that enables other components to access this queue. The queue only holds the planned interventions that are not delivered to patient yet. POWER2DM SMSS applications access to the queue service to query the planned interventions for the upcoming specific time period (e.g. current day). Only authorized users (in this case only the patient) access the service and the library Policy Enforcement Point is used to perform this authorization. For those that

needs immediate intervention, when the planned time comes, the component utilizes Push Notification Manager to send a push notification to the POWER2DM SMSS Mobile Application.

Push Notification Manager is responsible for preparing and delivering the push notifications to the POWER2DM SMSS Mobile Application by using the Firebase Cloud Messaging API¹⁸. It is cross-platform messaging solution that lets you reliably deliver messages to IOS or Android based platforms. The component gets the order from the JITAI Queue Manager including the intervention identifier that specifies the intervention to be delivered. Then it retrieves the message template (content of the notification message) from the Rule/Model/JITAI repository and fill them with the attributes provided by the order received from JITAI Queue Manager.

Rule/Model/JITAI Repository is the database that includes the contents of JITAI push notification messages, information related with the rules and decision trees to be used for planning, and machine learning and analytic models used for learning of patterns and correlations.

JITAI Planner is the main sub-component that runs the rules and algorithms on the streaming data coming from the Kafka Consumer Interface and the summary data that it obtained by querying HL7 FHIR Service provided by POWER2DM Personal Data Store. The component uses the FHIR Client Wrapper library to access to the PDS service. The component will be implemented by using the Apache Spark¹⁹ technologies including the Spark Streaming, Spark SQL and Spark Mlib.

3.4.2 Software Interfaces

3.4.2.1 JITAI Plan Queue Service

The service will be the implementation of the simple Java interface illustrated in the Figure 11. It returns the list of Intervention Plans residing in the queue for the upcoming period until the end of day.

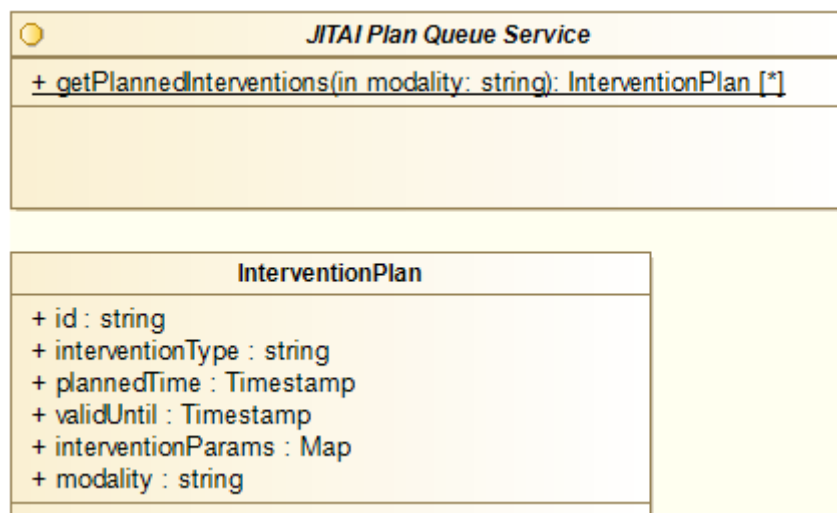


Figure 11 Java Interface for JITAI Queue Service

The Intervention Plan objects include the following attributes;

- **id:** The unique identifier for the planned JITAI instance
- **interventionType:** The identifier that specifies the type of intervention (e.g. motivating patient for ongoing exercise in the middle of it). It is assumed that POWER2DM system (all

¹⁸ <https://firebase.google.com/docs/cloud-messaging/>

¹⁹ <http://spark.apache.org/docs/latest/index.html>

components) will be configured with the intervention types defined for POWER2DM care program and an identifier will be assigned for each intervention type.

- **plannedTime:** The time that the intervention is planned. For POWER2DM SMSS Mobile Application this means that the push notification will be sent at this time.
- **validUntil:** This element indicates the time that intervention is valid so it should not be shown if the time passed for dashboard visualization of JITAI. The JITAI Queue Manager will delete the intervention plan from the queue when it is no longer valid.
- **interventionParams:** JITAI designed in POWER2DM may have variable parameters that can be adapted by JITAI planner while the intervention is defined as a template. This map element provides the parameter values in key-value pairs. The components handling the InterventionPlan objects should understand these keys and should act accordingly.
- **modality:** The restriction on which platform the intervention will be delivered. Possible values are “all”, “mobile”, “web”.

3.4.3 Required Software Interfaces

The component utilizes the HL7 FHIR service, Apache Kafka Consumer Service provided by POWER2DM Personal Data Store and OAuth Authorization Endpoint from POWER2DM Core Services.

Apart from these, Communication Engine will utilize the Firebase Cloud Messaging API as defined in the following section.

3.4.3.1 Firebase Cloud Messaging API

As the messaging protocol, the HTTP transfer will be used. As message type in Firebase, Communication Engine will use the messages with both payload and notification. In this way,

- When in the background, POWER2DM SMSS Mobile Application receives the notification payload in the notification tray, and only handle the data payload when the user taps on the notification.
- When in the foreground, POWER2DM SMSS Mobile Application receives a message object with both payloads available.

Notification parameters will be as follows;

- **body:** The message itself showing the JITAI's main message for the patient (e.g. Keep running, you have completed half of your exercise plan!)
- **title:** The title of the message (e.g. Ongoing Activity)
- **icon:** The icon to show with the message

The (data) payload parameters will be as follows;

- **intervention:**
 - **id:** The identifier of the intervention instance
 - **interventionType:** The identifier indicating the type of intervention
 - **params:** The parameter list for the intervention (key, value pairs).

POWER2DM SMSS Mobile Application should understand the interventionType and the provided params and when clicked on the message, the related designed view should be opened and the planned intervention content should be delivered to the patient.

3.5 POWER2DM Core Services

The deliverable “D4.9 Privacy and Security Enablers I” provides detailed information for the internal design of the POWER2DM Core Services component and the security and privacy mechanisms implemented in POWER2DM. Therefore, we will provide a very brief overview in this

section. Figure 12 illustrates the relation of the POWER2DM Core Services component with other components and POWER2DM PDS.

The **User and Privacy Management <<UI>>** component is the web application that includes the following web interfaces for patient and other users;

- **Single Sign-on UI:** The interface for the authentication of POWER2DM users (patients, care providers, etc) with username/password and two-factor authentication.
- **Patient Registration UI:** The interface for authorized users to register a patient to POWER2DM system.
- **User Registration UI:** The interface for authorized users to create an account for other users (care providers) to POWER2DM system
- **Patient Privacy Management UI:** The interface for patients to manage the access control policies (consent) for their personal data as well as their permissions for POWER2DM system for their data managed by 3rd party systems.
- **Audit View UI:** The interface for patients to view the audit logs regarding the data accesses to their personal data.

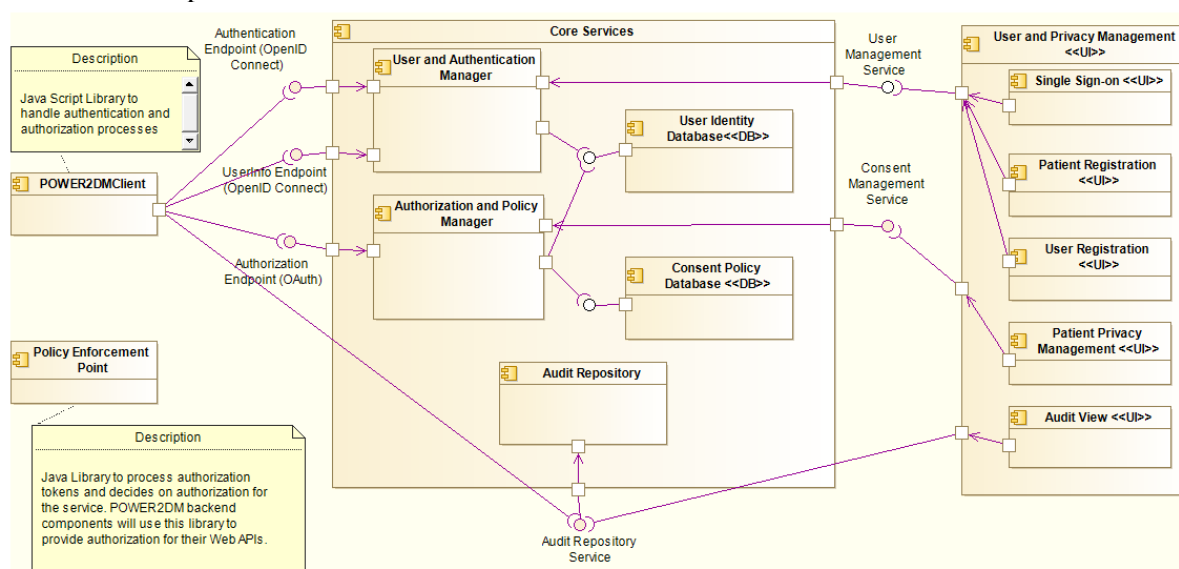


Figure 12 Core Services component diagram and its relations with other components

The **Core Services** component provides the following services;

- **OpenID Connect Authentication Service:** This is the implementation of OpenID Connect 1.0 Authorization Endpoint to handle authentication requests from clients that wants to authenticate their end users.
- **OpenID Connect UserInfo Service:** This is the implementation of OpenID Connect 1.0 UserInfo Endpoint to respond to user info (user account data like name, surname, address, etc.) requests of client.
- **OAuth Authorization Service:** This is the implementation of OAuth 2.0 protocol to respond the authorization requests of client for the POWER2DM personal data and APIs.
- **Audit Repository Service:** This is the implementation of IHE ATNA compliant audit repository service to store the audit logs coming from other POWER2DM components.
- **User Management Service:** This is the service for the use of User and Privacy Management UI for single sign-on, user and patient registration, user profile management.
- **Consent Management Service:** This is the service for the use of User and Privacy Management UI for privacy management operations.

The **POWER2DMClient** represents the set of Java Script libraries that will be provided to POWER2DM UI components to easily handle the authentication and authorization processes. Open

source libraries will be reused and adapted as much as possible and each client's programming language requirements will be considered.

Similarly, **PolicyEnforcementPoint** is the library to be provided to POWER2DM backend services to decide on authorization for the provided service and protected resources.

4 USER INTERFACE DESIGN AND UI COMPONENTS

The objective of this section is not providing a detailed visual design for POWER2DM UI components. Following the agile design principles, the aim is to identify the important UI components and basic functionalities they will provide. The idea is to go one more step from requirements to design phase and provide the initial mockups so that project members and supporters can provide their comments and feedbacks easily.

4.1 User Experience and HCI Design Considerations

Several UX design activities have been performed in POWER2DM. The main purpose was to identify the interaction points of our users with the overall POWER2DM SMSS application and enhance users' satisfaction by improving the system's usability and ease-of-use considering their interaction with the application. Maintaining users' adherence on application usage as long as possible has a high importance in POWER2DM as the desired clinical results can only be reached if the application is used long enough until a healthy behavior is embraced by patients with the guidance of the application.

In literature, UX design is performed through four main steps that are repeated in several iterations:

- 1) user segmentation and analysis of their requirements
- 2) task and service identification
- 3) design and development
- 4) testing and evaluation
- 5) go to 1) if UX design needs to be updated

UX design activities in POWER2DM has started with the identification of user types and their requirements. Afterwards, we have identified the tasks those are supposed to be performed by each type of user as a list of use cases. The use case list has driven the features/services that need to be provided by the POWER2DM application considering our user segmentation and the tasks they are associated to. These features and services have been formalized as software requirement specifications. Design and prototyping of the POWER2DM system followed the requirements specification. By the time this report is produced, the final step of the first iteration of UX design remains to be performed. In the final step of the first iteration, we are going to evaluate the usability of the initial prototype by making the prototypes available for care providers and patients for initial tests. After measuring and evaluating the usefulness and effectiveness of the POWER2DM system, the first cycle of UX design will have been completed and the next cycle will follow in order to resolve the issues identified in first iteration.

Below, we provide details about the four main phase of POWER2DM UX design.

4.1.1 Getting to know our users

In POWER2DM, we have applied several methods in order to identify our users' characteristics and their needs better. First of all, we classified our target users and we ended up with two main groups of users: patients and care providers.

Care providers: General practitioner, internist, diabetes nurse, general practice-based nurse specialist are some professions that we classified as care provider. Thanks to the care providers that we have

already had within POWER2DM consortium, we were able to identify their daily routines and their expectations on the POWER2DM application. It turned out that time aspect is the primary restriction related to usage of the POWER2DM application in their daily routines. Daily routines of care providers are too loaded and they have only a few minutes for dealing with one patient. They have appointments with tens of patients during a single day, which brings quite a heavy cognitive burden on them. After discussions with the clinical partners, the following is set as a goal for “POWER2DM Shared Decision Making Application” which will be used by clinicians during the appointments and help them to perform the procedures in the given time.

- Half an hour for first encounter in the POWER2DM Care Program to prepare the care plan
- 10 minutes for following visits in the POWER2DM Care Program to evaluate the situation and update the care plan

Nevertheless, care providers still need to perform their activities as quick as possible. They need to perform their activities with as less click as possible and without spending too much cognitive resource. In this respect, clarity and intuitiveness of the POWER2DM SMSS are critical towards being beneficial for care providers and ease their jobs.

Our care providers indicated their interest in dashboard-kind of views serving to both intuitiveness and quickness purposes. Through dashboards they are able to see a summary of patients’ care and recognize problematic areas at the first glance and navigate to the details without passing through complicated flows.

Patients: Common characteristic of users belonging to this category is that they are diabetic patients. POWER2DM provides patients with prediction and recommendation services that are enabled by complex models and algorithms. Therefore, it can be assumed that most of POWER2DM patients would not have the technical and/or scientific background to understand the reasoning behind the predictions and recommendations.

However, as patients differentiate in many aspects, we had to categorize them further in order to meet requirements related to their specific characteristics. Therefore, we have defined personas representing main characteristics of potential POWER2DM users. Personas classified patients considering the following hierarchy of parameters:

- Age
- (Diabetes mellitus) DM parameters
 - Diabetes type
 - Time since diagnosis
 - HbA1c
 - BMI
 - DM medication
 - DM devices
- DM Self-management (SM) experience
 - Hypo / hyper glycaemia
 - SM problem
 - Psychological barrier to DM SM
 - Conflicting life & health goals
- Socio-demographics
 - Social environment & support
 - Work
 - Education
 - Sports / hobbies
- Psychological health
 - Strength
 - Issues
- Care plan
 - Treatment goal

- Committed actions

In order to cover the variance of potential POWER2DM users, we instantiated 7 personas in total by assigning realistic values to the parameters above by sampling realistic life-styles. For example, working young people with familiarity to technology, around their thirties, living with their boy/girl friends; elderly people living alone as their partners had passed away, far from technology, taking support from their sons/daughters, etc. Complete list of personas can be found at *D1.1 - User Requirements and Use Case Scenarios* deliverable.

Current care progress: In addition to identification of two main user groups, we needed to analyze the current care progress, relationships of patients with their care providers in the long term for not being inconsistent with the current care practice. The current practice has two main phases: shared-decision making and self-management. These main phases have the following sub-steps:

- Shared-decision making
 - Anamnesis
 - Problem identification
 - Recommendations
 - Goal setting and Action planning
- Self-management
 - Monitoring and self-management
 - Evaluation and feedback

4.1.2 Scenario Flows and required POWER2DM Features

Identification of main user groups, further classifying them with personas and investigation of their daily routines and existing relationships were not necessary for setting out a proper roadmap about the system functionalities. We had to have an idea about how and in which contexts users would interact with POWER2DM. Therefore, we narrated some of the personas by explaining the interaction of users with the POWER2DM system in step-by-step scenarios throughout the care progress covering both shared-decision making and self-management phases.

Along with the detailed usage scenarios, we have identified possible use cases. Use cases represent fine-grained activities such as “adding a self-management activity” to be performed by users in order to achieve their goals during the care process. In alignment with the current care process, we have defined two main sets of use cases for shared-decision making and self-management phases. In parallel to the step-by-step scenario generation, each scenario step has been associated with one or more POWER2DM use cases. For ensuring clear understanding and usefulness of POWER2DM, we also created use case flows for shared-decision making phase, self-management phase and also for KADIS preparation and metabolic fingerprint construction which are reported in D1.2 Requirements Specification of POWER2DM Architecture.

4.1.3 Design and Implementation

After the requirement specifications step, we passed into the design & implementation phase. We have designed mock-ups for each UI component of the POWER2DM. We performed a multi-disciplinary mock-up design. Partners with predictive modelling expertise have designed mock-ups for their respective models and risk factors for both care provider and patient usage. *D2.5 - Mockups for GUI Components related with Visualization of Predictions and Simulations* is dedicated to report these mock-ups. In summary, the referred deliverable first analyzes the existing user interfaces related to KADIS and M2TD-Marvel prediction models and existing risk factors including UKPDS, Advance Risk Engine, QRisk2-2015, Framingham and T1D Major Outcomes. The deliverable, afterwards, reports mock-ups to be used in shared decision making phase considering the physician perspective; and mock-to be used in self-management phase considering the patient perspective.

Self-management related mock-ups were basically designed considering presentation of predictions-related information to patients. As predictions are generated as a result of scientific computations, reasoning behind the predictions may not be clear for patients. Even the prediction results may be misleading for patients if the results are calculated under a high uncertainty. Therefore, we aim to make some at least the uncertainty degree of predictions available to patients. Along with the uncertainty degree, we will also provide visual signs to patients so that they would have an idea about to what extent they could trust the prediction results and suggested actions. User interface design considerations addressing such considerations are given at Section 4.1.5 as well as in *D2.5 - Mockups for GUI Components related with Visualization of Predictions and Simulations*.

4.1.4 Measuring and Evaluation

What to measure?

The main focus of the measurement of early mock-ups and prototypes is going to be the usefulness of the designed application. In literature, usefulness is represented with two main characteristics: utility and usability²⁰. In the context of user interface evaluation, usability is defined as the perception of a target user of the utility (whether it provides the features the users need) and usability (how easy and pleasant these features are to use) of the interface.

Usability is a hot research area for decades and this concept is investigated as a combination of various other factors as listed below:

- **Learnability:** Another common term for learnability is intuitiveness. It is a measurement unit to evaluate how easy is it for users to accomplish the tasks the first time or after a long time of previous use they use the interfaces.
- **Efficiency:** This unit is to measure how quick is to perform a task in terms of time or how easy in terms of effort spent.
- **Errors:** This factor aims to identify the severity of errors that users might do and how easily can they recover from those errors.
- **Satisfaction:** This aspect considers pleasantness as an important factor contributing to the usability of user interfaces.

How to measure?

POWER2DM will utilize user inspection methods to measure the usefulness of the user interfaces. These tests are classified into two main groups in with respect to timing of the tests: formative usability testing and summative usability testing. Although the major factor differentiating these two types of tests is their timing, they also differentiate in the number of participants, methodology, data collection, presentation of findings, etc.

- **Formative usability testing** has the aim of directing the design process and it is applied in the early development process. It reveals the problematic interfaces where users have difficulty reaching their goals. Therefore, data analysis in formative tests tends to be qualitative in nature²¹. It is common that these tests are performed frequently in an iterative manner as soon as low fidelity versions of user interfaces appear in order not to advance on wrong design, which would become hard to fix in later phases. Common metrics used in formative usability testing include completion of tasks, UI problems, errors, post-test questions, post-task questions, etc.

²⁰ J. Nielsen, "Usability 101: Introduction to usability," *Jakob Nielsen's Alertbox*, Retrieved Aug 31, 2016 from <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>, 2012.

²¹ D. Travis, "2 kinds of usability test". Retrieved Sep 01, 2016 from <http://www.userfocus.co.uk/articles/2-kinds-of-usability-test.html>

- **Summative usability testing** is performed later in the development process. Here the aim is to measure the acceptance of the developed application before releasing it to the target users. As a result of these tests, it is ensured that the solution will work for users. In this respect, some see acceptance testing as Beta Testing where the initial version of the developed system is evaluated by a limited number of users. Metrics used for formative usability testing apply also for summative usability testing. In addition, more fine-grained metrics are applicable in summative usability testing. Such metrics are grouped under different categories such as behavioral metrics (web analytic, performance, success rate), physiological measures (cognitive load, memory, attention, affective state and engagement), subjective measures (satisfaction, expectations, recommendation, opinion, emotions, and overall user experience) and so on.

Application in POWER2DM

POWER2DM will perform formative usability tests on the initial set of mockups. Clinical partners of POWER2DM consortium will perform usability testing using the “cognitive walkthrough” method. Cognitive walkthrough is a task based method, focusing on how easy it is for new users to accomplish tasks on the application. Therefore, a task analysis should have been performed prior to application of this method. In this respect, the POWER2DM use case set will be used as the test cases. In literature the following questions are asked in order to identify possible problems for each task²².

- Will the user try to achieve the right effect by performing the task?
 - E.g. Does the user understand that this subtask is needed to reach the user's goal?
- Will the user notice that the correct action is available?
 - E.g. is the button visible?
- Will the user associate the correct action with the effect to be achieved?
 - E.g. the right button is visible but the user does not understand the text and will therefore not click on it.
- If the correct action is performed, will the user see that progress is being made toward solution of the task?
 - Will the user know that they have done the right thing after performing the action?

The output of formative usability testing includes the answers for the questions above, which is qualitative as explained above. Test participants will comment in the form of a narrative telling their attitudes, sources of confusion, reasons for actions. At the end of a test cycle, technical partners of POWER2DM will map the outputs to technical issues and amend the design and implementation accordingly.

POWER2DM will also perform summative usability tests in the scope of pilots. Instead of hardwired mockups, this time the running prototype will be made available to users. Both care providers and patients will perform tests. Considering that both qualitative and quantitative analysis will be performed in summative tests, their coverage will be wider than formative tests. Detailed testing methodologies along with the results will be reported *D6.4.3 - Field Trials Report & Socio-Economic Guidelines* deliverable.

4.1.5 HCI Considerations

Although there are countless alternatives of sets of user interface design principles, they all give similar messages. In literature, a popular one of these design principles sets is the one suggested by

²² Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In Nielsen, J., and Mack, R. (Eds.), *Usability inspection methods*. New York, NY: John Wiley & Sons, Inc.

Larry Constantine and Lucy Lockwood²³. According to these authors, there are six main principles as listed below to be considered while designing user interfaces:

- **The structure principle:** It concerns the overall architecture of the system. When placing a certain object on a user display, there has to be some reasoning to justify the decision taken. Therefore, user interfaces should be designed in consistent, meaningful and useful ways that are apparent and recognizable to users, putting related things together and separating unrelated things, differentiating dissimilar things and making similar things resemble one another.
- **The simplicity principle:** The design should make simple, common tasks easy, communicating clearly and simply in the user's own language, and providing good shortcuts that are meaningfully related to longer procedures. This principle is the main component deriving efficient usage.
- **The visibility principle:** This principle suggests keeping all needed tools and materials for a given task visible without distracting the user with extraneous or redundant information. Discriminability and legibility factors can also be considered in this principle.
- **The feedback principle:** Users should receive some form of feedback from the UI when any sort of change or error occurs; this is because it's in the interest of the user when performing a certain task to be updated on any changes on the current situation.
- **The tolerance principle:** The design should be flexible and tolerant, reducing the cost of mistakes and misuse by allowing undoing and redoing
- **The reuse principle:** According to this principle, the design should reuse internal components throughout several user interfaces (internal consistency). The principle also applies for external components, meaning that familiar design patterns from external systems should be reused e.g. iOS apps should use native OS compatible design patterns (external consistency)

Aforementioned principles are quite generic and applicable in almost any user interface design. POWER2DM will also utilize them while designing and evaluating the user interfaces. Below, we present the methodology about how POWER2DM interprets these principles and apply them during the design of user interface mock-ups.

- **The structure principle:** There is a high variety of information that has to be managed in diabetes care process. Not all available information is presented in each view but relevant information is grouped in different views. Firstly, dashboards summarize and group the information in self-management, barriers and risk score categories. Each category has several sub-categories and once clicked, each of these sub-categories navigates to the relevant view where detailed information about that sub-category is presented.
- **The simplicity principle:** The first point that POWER2DM will take into account towards simplicity of the software application will be to use common language throughout the application so that it will be understandable by patients. The theory behind the predictions and recommendations is hard to be understood as it is by patients assuming that most of them would not have the relevant scientific and technical knowledge. Details about these concepts should be explained to patients in a common language, along with visual representations where possible.
- **The visibility principle:** POWER2DM has several views including results about past performance or predictions about future progress. Such views themselves comply with the visibility principle as these views aim is to provide an overview about certain parameters such

²³ Constantine, L. L., and Lockwood, L. A. D. (1999) Software for Use: A Practical Guide to the Models and Methods of Usage - Centered Design. Reading, MA: Addison - Wesley

as blood glucose levels. In case of undesired values, which are shown in red color, POWER2DM is going to employ navigation capabilities directing patients to screens required to take the counter-measure against the undesired value. One on hand this approach makes all the required functionalities available to them, on the other hand it simplifies the usage of the software.

In order to increase the visibility POWER2DM plans to make widely use of graphics and color codes instead of explaining every information textually. In this way, users would not spend too much time and cognitive resource to notice the critical points on a particular view and deduce the action to be taken. This is one of the critical factors required to maintain adherence of application usage.

- **The feedback principle:** As this principle is more related to interactivity of the user interfaces, it has not been addressed on mock-up designs. In the actual implementation, POWER2DM will ensure compliance with this principle so that users would notice a visual feedback upon their actions on the application e.g. clicking a button, selecting a list item, selections on graphs, etc.
- **The tolerance principle:** As one of POWER2DM's main motivation is to guide diabetic patients during their self-management activities at their daily living environments, the eventual application is going to deliver interventions, give recommendations, present information in an automated way. However, POWER2DM has to ensure clinical safety by taking patients' lack of knowledge into account. Considering relation of the automation aspect of the POWER2DM software with clinical safety, first of all, the content to be presented to patients is going to be determined by clinical experts in the scope of *Task 3.5 – BCT modules knowledge content*. Therefore, the content will have been validated by clinical experts. However, it will not be possible to check all pieces of information by experts. Therefore, in *Task 3.5*, rules, workflows, thresholds, etc. about the content will be identified in a way that is generalized for all patients.

Another measure towards clinical safety is to deliver only relevant interventions to patients. For example, whilst a patient would need only physical activity recommendations, another one would not require them but he would need dietary related ones. Another example could be related with interventions including predictions: Patients who are at the very early stages of type-2 diabetes may not require KADIS-related interventions. Therefore, POWER2DM is going to have user interfaces for the shared decision making part, enabling restriction of the set of interventions to be delivered to patients based on the patient characteristics.

As already mention in the simplicity principle, POWER2DM aims making patients to use the prediction models. In addition to simplified explanation of user interfaces related to the predictions, clinical safety has to be ensured also. POWER2DM will prevent patients from taking wrong actions by proper limitations, warning messages, information messages and help options on predictions and prediction simulations. *D2.5 - Mockups for GUI Components related with Visualization of Predictions and Simulations* gives details about the user interface principles followed to ensure clinical safety regarding predictions-related user interfaces.

- **The reuse principle:** Initially, we had mock-ups designed with separate design considerations for KADIS-based predictions, risk factor applications and action plan engine. They were designed based on the existing software systems relevant to each of them. In order to have internal consistency throughout the whole POWER2DM application, POWER2DM has designed all mock-ups using the same framework, BootStrap, by reusing its several visual elements. BootStrap contributes also to external consistency to some extent as it is a well-known visual component library used by many web-sites e.g. Twitter.

More details about the user interface considerations for patient usage of predictions are given in D2.5 - Mockups for GUI Components related with Visualization of Predictions and Simulations.

4.2 POWER2DM SMSS UI

4.2.1 POWER2DM SMSS Web UI

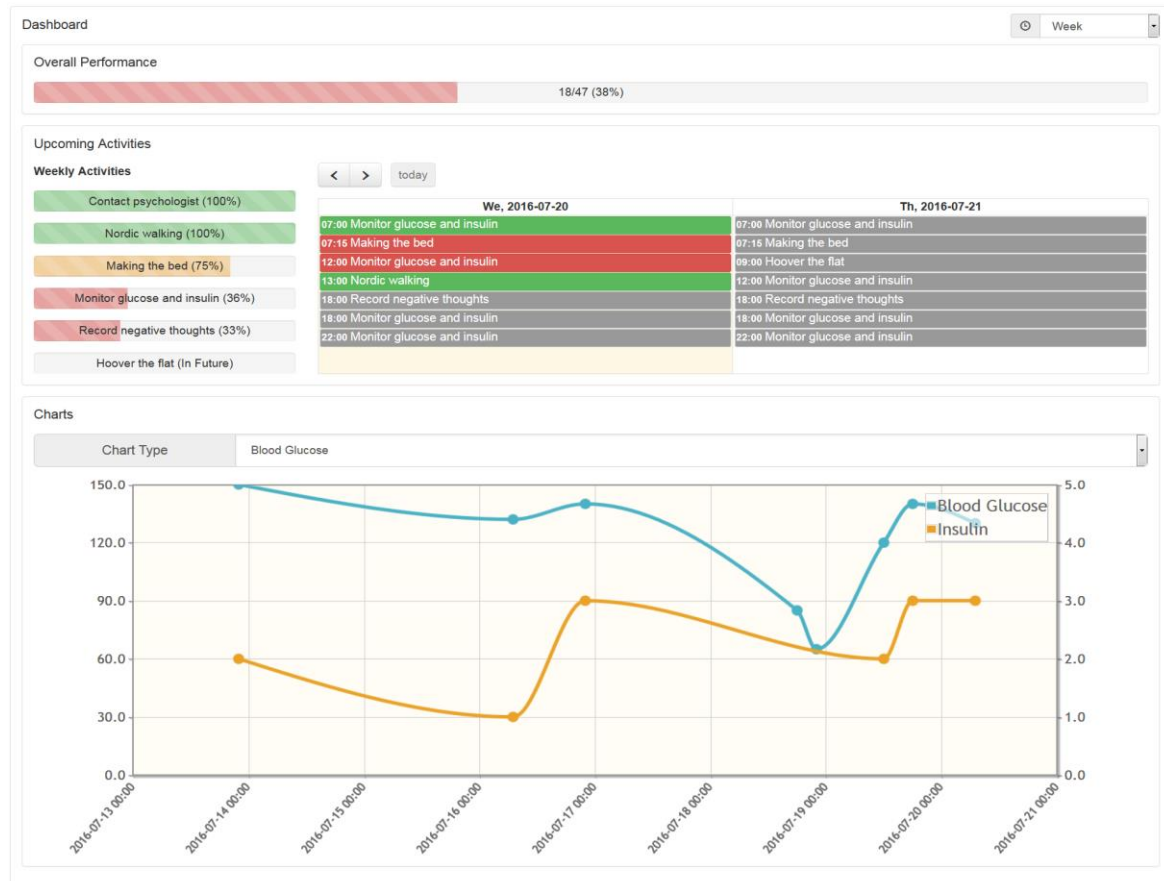


Figure 13 Dashboard View

The dashboard view presents an overview of the most relevant data to the user including their overall performance for the past week, upcoming activities for today and tomorrow and various charts.

Goals				
+ Add				
Date	Goal	Related Recommendations	Description	Motivation
2016-07-12	Contact the psychologist in order to cope with shame about diabetes	<ul style="list-style-type: none"> Increase social contacts Use the Action Plan and organise and evaluate the activities 	-	-
2016-07-12	Monitor blood glucose and take medication	<ul style="list-style-type: none"> Set reminders for monitoring glucose levels and taking medication Use the Action Plan and organise and evaluate the activities 	-	-
2016-07-12	Record negative thought in the online diary	<ul style="list-style-type: none"> Record negative thought Use the Action Plan and organise and evaluate the activities Start with psycho-education with a focus on negative thoughts and cognitive restructuring 	-	-
2016-07-12	Support Anna by doing more domestic work	<ul style="list-style-type: none"> Increase exercises Use the Action Plan and organise and evaluate the activities 	-	-
2016-07-12	Walk with a pedometer at least 30 minutes 3 times a week	<ul style="list-style-type: none"> Increase exercises Use the Action Plan and organise and evaluate the activities 	-	-

Figure 14 Goal Management View

The goal management view allows for managing user goals with an entry date, a goal title, related recommendations, a more comprehensive description and a personal motivation for the goal. New goals can be added by pressing the “Add” button; existing goals can be edited or deleted by clicking on the corresponding table row. Therefore, a modal is opened in an overlay.

GoalX

Date *

2016-07-12

Goal *

Monitor blood glucose and take medication

Related Recommendations

☐ Fill out the value compass for how you can include diabetes management properly in the daily life and simultaneously live a full and rich life

☐ Increase exercises

☐ Increase social contacts

☐ Record negative thought

☒ Set reminders for monitoring glucose levels and taking medication

☐ Start with psycho-education with a focus on negative thoughts and cognitive restructuring

☒ Use the Action Plan and organise and evaluate the activities

Description

Motivation

* Required field

CancelDeleteSave

Figure 15 Goal Details View

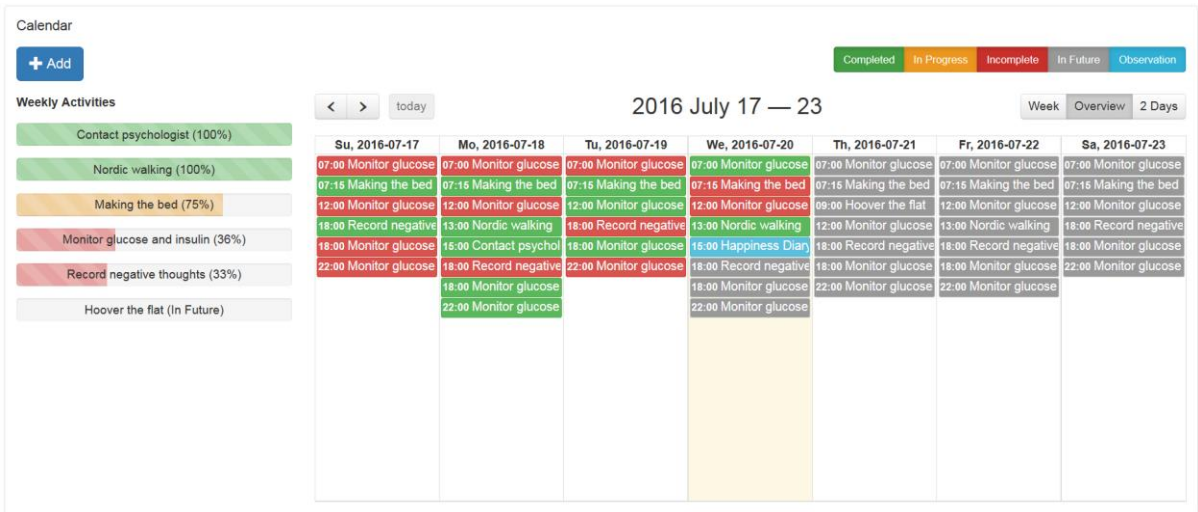


Figure 16 Calendar View

The calendar view keeps track of all user activities with adjustable views (week, overview, 2-day) and activity type filters (completed, in progress, incomplete, in future, observation). Furthermore, completion rates are colour coded. New activities can be added by pressing the “Add” button; existing activities can be edited or deleted by clicking on the corresponding progress bar on the left hand side; ODL data can be added / edited by clicking on the corresponding calendar item (see ODL Management View for further details). Therefore, a modal is opened in an overlay. Schedules are added / edited in another modal.

Activity X

Activity *	Making the bed
Description	
Category *	Other
Related Goals	<div><input type="checkbox"/> Contact the psychologist in order to cope with shame about diabetes</div> <div><input type="checkbox"/> Monitor blood glucose and take medication</div> <div><input type="checkbox"/> Record negative thought in the online diary</div> <div><input checked="" type="checkbox"/> Support Anna by doing more domestic work</div> <div><input type="checkbox"/> Walk with a pedometer at least 30 minutes 3 times a week</div>
Schedule *	<div>07:15 - 07:20</div> <div>Mo, Tu, We, Th, Fr, Sa, Su</div> <div>(From: 2016-07-03, To: 2016-12-31)</div>

+ Add schedule

* Required field

Cancel

✕ Delete

✓ Save

Figure 17 Activity Details View

Schedule X

All-day *	<div>Yes</div> <div>No</div>
Weekdays *	<div>Mo</div> <div>Tu</div> <div>We</div> <div>Th</div> <div>Fr</div> <div>Sa</div> <div>Su</div> <div>Working days</div> <div>Daily</div>
From *	07:15
To *	07:20
Valid from *	2016-07-03
Repeat until *	2016-12-31

* Required field

Cancel

✕ Delete

✓ Save

Figure 18 Activity Specification View

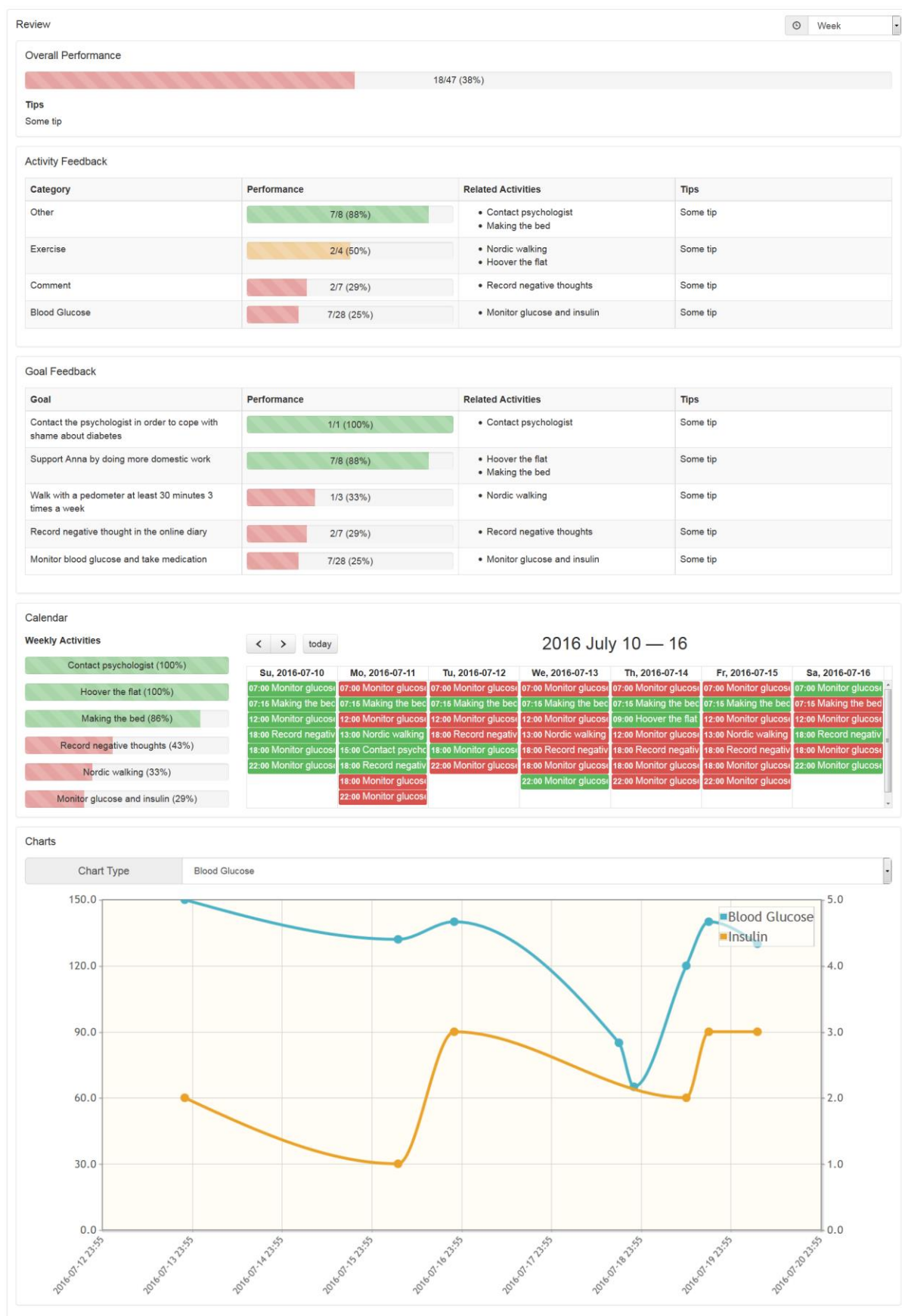


Figure 19 Periodic Review View

The periodic review view sums up all the data from the previous week including overall performance, activity feedback (grouped by ODL type), goal feedback, a calendar view (that allows for entering up missing data) and various charts.

Furthermore, the general colour coding is used and relevant tips are displayed.

Exercise					Month
+ Add					
Date	Duration (Min)	Category	Where	Intensity	Comment
2016-07-20 13:00	30	Walking	-	Intensive	-
2016-07-18 13:00	30	Walking	-	Light	-
2016-07-14 09:00	45	Vacuuming	-	Very Light	-
2016-07-11 13:00	30	Walking	-	Intensive	-
2016-07-07 09:00	45	Vacuuming	-	Moderate	-
2016-07-04 13:00	30	Walking	-	Intensive	-

Figure 20 ODL Management View

There are ODL management views for each ODL type (e.g. exercise) containing all relevant data. New ODLs can be added by pressing the “Add” button; existing ODLs can be edited or deleted by clicking on the corresponding table row.

Exercise X

Date *	2016-07-11 13:00	
Duration *	30	Min
Category	Walking	
Where		
Intensity *	Intensive	
Comment		

* Required field

[Cancel](#)
[✖ Delete](#)
[✔ Save](#)

Figure 21 ODL Editing View

Gaining clarity

Take time to reflect on your values and barriers.

Share your values with your loved ones and try to think of actions to pursue your values and cope with your barriers.

Remember: Your values and barriers are an ongoing process, not a static thing.

Come back and redefine them after specific periods or major milestones.

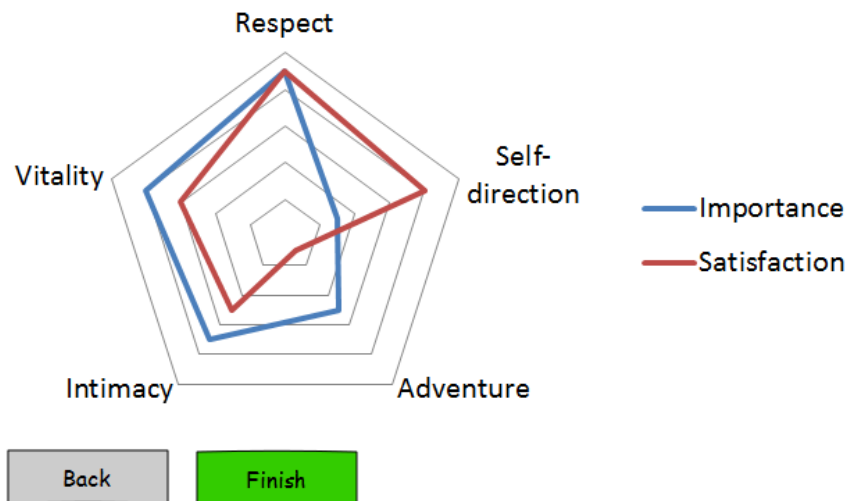


Figure 22 Value Compass and Barrier Identification View

The value compass and barrier identification view are presented by means of guided walkthroughs as a combination of the approaches by R. Harris ([ACT Training Online Made Simple](#)) and Dr. J. Demartini ([Values E-book](#)).

Therefore, comprehensive decision trees are required.

← Untitled form

Add-ons

SEND

More Settings

Settings

test form

test

question

☐ Option 1

☐ Option 2

☐ Other...

question 2

Month, day, year

Figure 23 Questionnaire View

The questionnaire view builds appropriate web front-ends (e.g. as used by Google Forms) based on classic questionnaire data.

In order to do this, question types and questionnaire structures have to be specified.

The **Prediction / Simulation View** will be addressed in T2.4.

4.2.2 POWER2DM SMSS Mobile Application UI

POWER2DM SMSS Mobile Application has two main functionalities;

- To support patient in daily self-management with a dashboard like view showing his/her goals, planned actions, measurements; in summary his latest health status and diabetes self-management performance
- To enable patient to perform several actions; registering ODL, filling a questionnaire, responding to an interactive interventions and questionnaire, periodic review of his/her health status, etc.

The functionalities of SMSS Mobile Application will be similar to the SMSS Web application but organized in a more compact way and designed for rapid and effective interaction of POWER2DM system with patient in daily life.

!!! Open Issue: In POWER2DM, it is planned to design the self-management process and the interventions within the scope of WP3 and therefore we cannot provide a clear UI design for SMSS applications. Instead, in this section we will provide the views and their basic functionalities that we identify based on the discussions and work until now. Furthermore, we will provide the user interface designs of **existing iHealth's mobile applications** (iHealth is the POWER2DM partner, who is responsible for the implementation of SMSS mobile application,) for similar self-management purposes, which will be adapted based on the specific requirements of POWER2DM.

4.2.2.1 SMSS Mobile Application Views

In this context, the following views are identified for the SMSS Mobile Application.

Daily Dashboard View: This is the main view (or a set of views) that helps patients to adhere to their plans for that current day by summarizing the planned actions, goals and latest measurements similar to the Dashboard view in SMSS Web Application. The following elements will be shown in the view;

- **Planned actions for the day:** These are the action plans like; medications instructions (e.g. insulin at 10:00, 20u dose), physical activity plans (e.g. 1-hour walking after meal) or dietary intake plans (e.g. dinner at 18:00 max 600 kcal), monitoring instructions (e.g. measure your blood glucose after meals), etc. The view will show timing, details, and the status of the action (e.g. done, missed, etc). If action is done, the details of the action will be shown (action can be an ODL; value of blood glucose measurement).
- **ODLs:** These are the measurements done in that day or latest value for the measurements (e.g. blood glucose, body weight, Stress VAS, etc). According to the priorities (goals set for the patient) and characteristics (type of diabetes, etc) of the patient, the measurements shown in this summary view can be changed.
- **JITAs:** These are the important notifications/warnings/motivations provided by POWER2DM system for that day.

ODL Registration Views: These are the views used by patient to register a ODL to the system (e.g. record body weight, Stress VAS, Emotional VAS) manually. Different views will be developed for each parameter to ease the process of data entry and user interfaces from existing iHealth's mobile applications will be used as basis for these views.

Questionnaire View: This is the view that enable patients to fill a questionnaire (required in POWER2DM care program e.g. WHO-5, PHQ-9) easily i.e. question by question, etc. It is also extended to not only enable patients to fill previously defined questionnaires but also enable POWER2DM to interact with patient with a set of questions in real time based on some decision trees.

There will be no difference in terms of UI design, but it will need further implementation and integration with the POWER2DM backend services.

!!! Open Issue: It is not decided yet on whether an existing multi-platform questionnaire system (e.g. LimeSurvey, like the one used by PatientCoach) will be used for this functionality. Or an internal POWER2DM implementation will be provided that can both serve on mobile and web platforms. If it is decided to go for internal implementation, the FHIR Questionnaire resource can be used as machine processable internal representation of questionnaires.

Value Compass and Barrier Identification Views: As described in the Section 4.2.1, this functionality is presented by means of guided walkthrough to enable patient to identify his/her personal values and identify the barriers to reach these values and goals.

Periodic Review Views: This is the set of views to enable patient to periodically check his health status and self-management performance. The tips and recommendations generated by POWER2DM Action Plan Engine together with the underlying reasons by demonstrating the collected data and extracted information will be shown to the patient.

Specific JITAI Views: These are views specifically to be designed for delivering the POWER2DM Just-in-time Adaptive Interventions (JITAI) for behaviour change. These interventions will be designed within WP3 later, therefore the UI designs for these interventions will be part of this intervention design phase. e.g. For an intervention that wants to warn patient for possible hypoglycaemia due to latest high insulin intake and predictions about expected blood glucose changes, the UI can show the followings;

- Description of the case and warning message,
- Data about the insulin intake (time and dosage), previously logged dietary intakes, and the predicted blood glucose profile based on this data

4.2.2.2 Existing UIs from iHealth's solutions

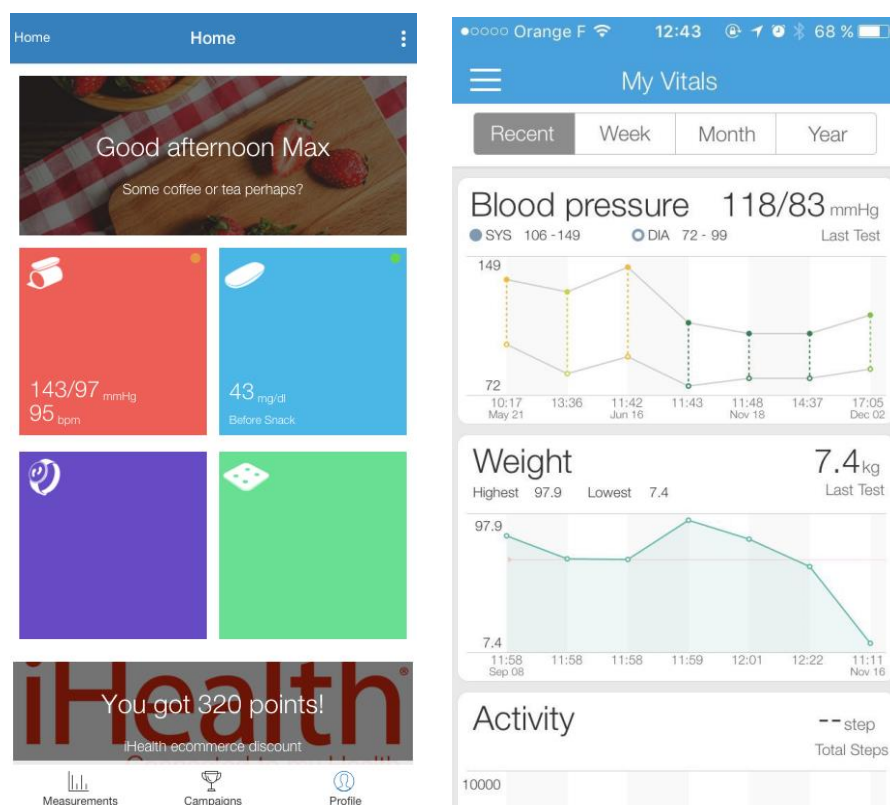


Figure 24 iHealth Mobile App – Main View and My Vitals View

Figure 24 illustrates the main views of iHealth Mobile Application that shows the latest measurements and actions in a dashboard or summary like views which are related with POWER2DM Daily Dashboard View. The left snapshot shows blood pressure and blood glucose measurements, dietary intakes, physical activities and body weight. The snapshot on the right shows these recent values in a graphical way with some historic data.

Similarly, Figure 25 shows the Journal and Blood Glucose Profile views, where the former shows all the actions performed by patient and the later shows 30-day summary blood glucose profile of patient.

Figure 26 shows the Dashboard view from the iHealth's B2B2C platform which is integrated with the mobile application and used by care organizations to manage their patients and analyse their status.

Figure 27 and Figure 28 the Settings and Profile management views of iHealth Mobile Application by which patient can access or update profile data and manage the connected devices and provide basic data.

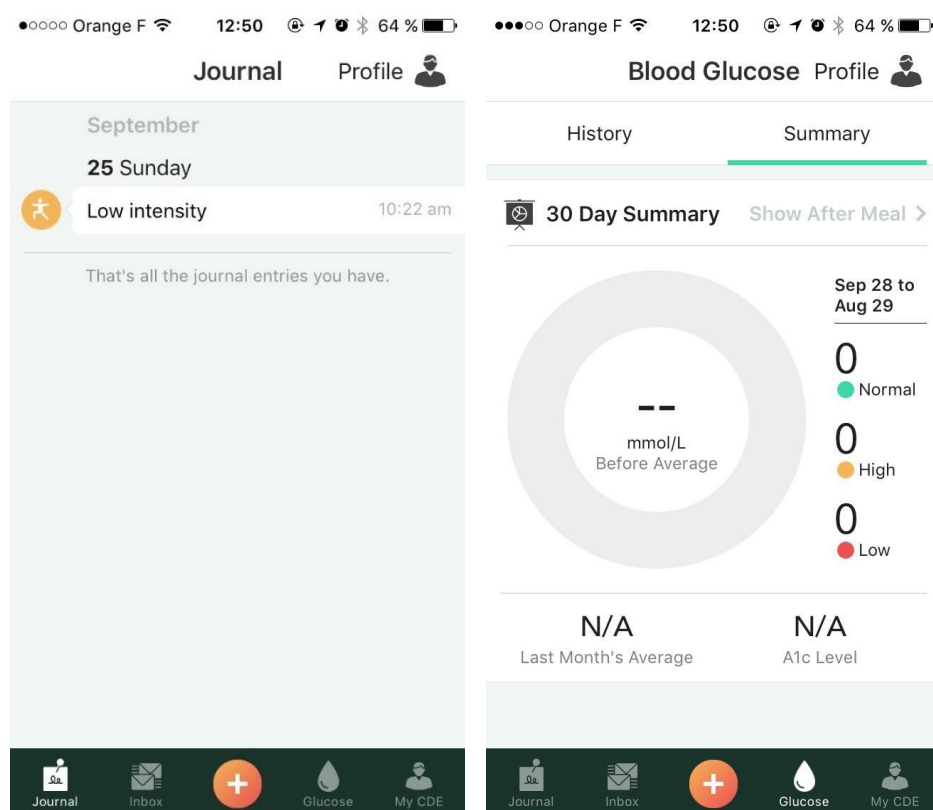


Figure 25 iHealth Mobile App – Journal and Blood Glucose Profile Views

Figure 29 shows the detailed view of blood pressure that helps patient to review his/her blood pressure history with graphics and as a list of measurements. This view can be adapted for the design of POWER2DM Periodic Review Views.

In Figure 30, the left snapshot shows the view to initiate a connection with an iHealth device and the snapshot in the right shows how the application interact with patients with simple questions over the main view.

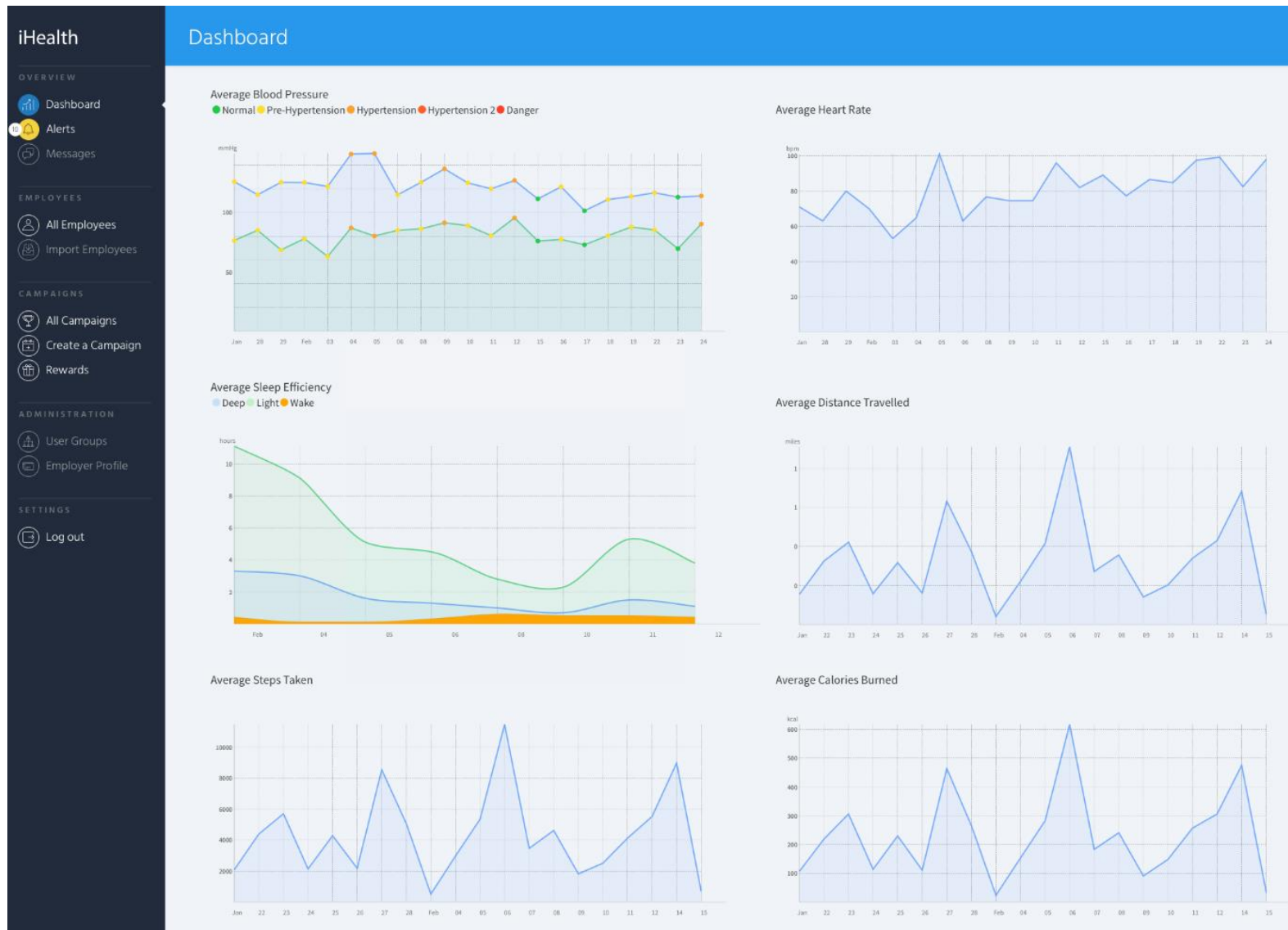


Figure 26 iHealth B2B2C Platform – Dashboard View

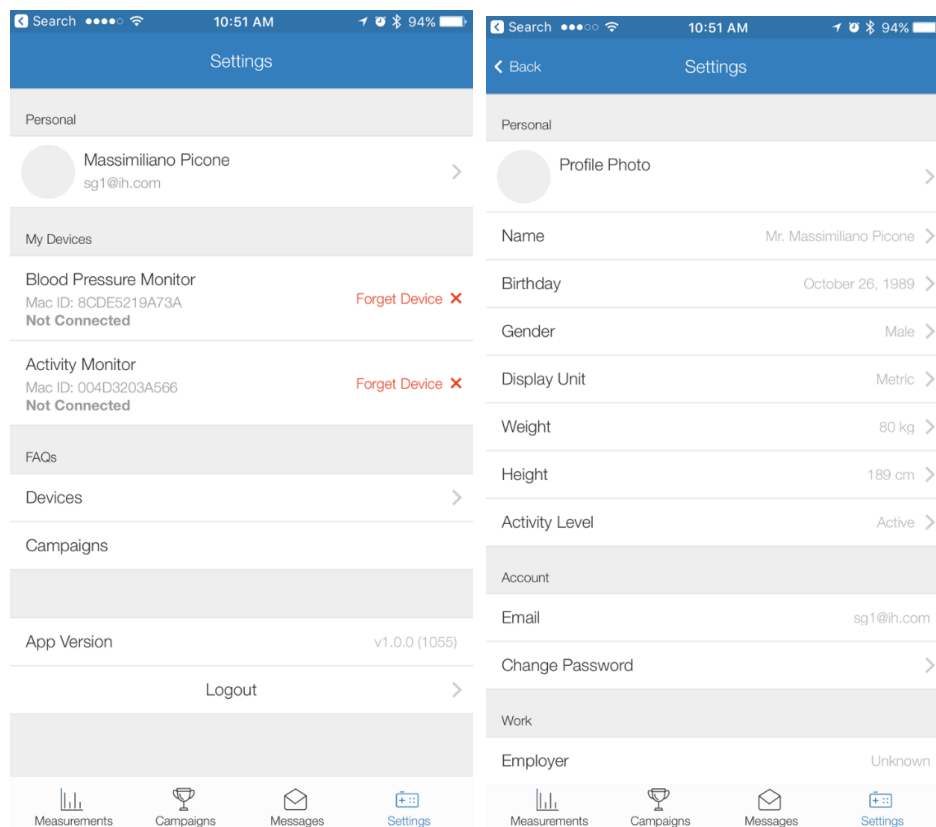


Figure 27 iHealth Mobile App – Settings View

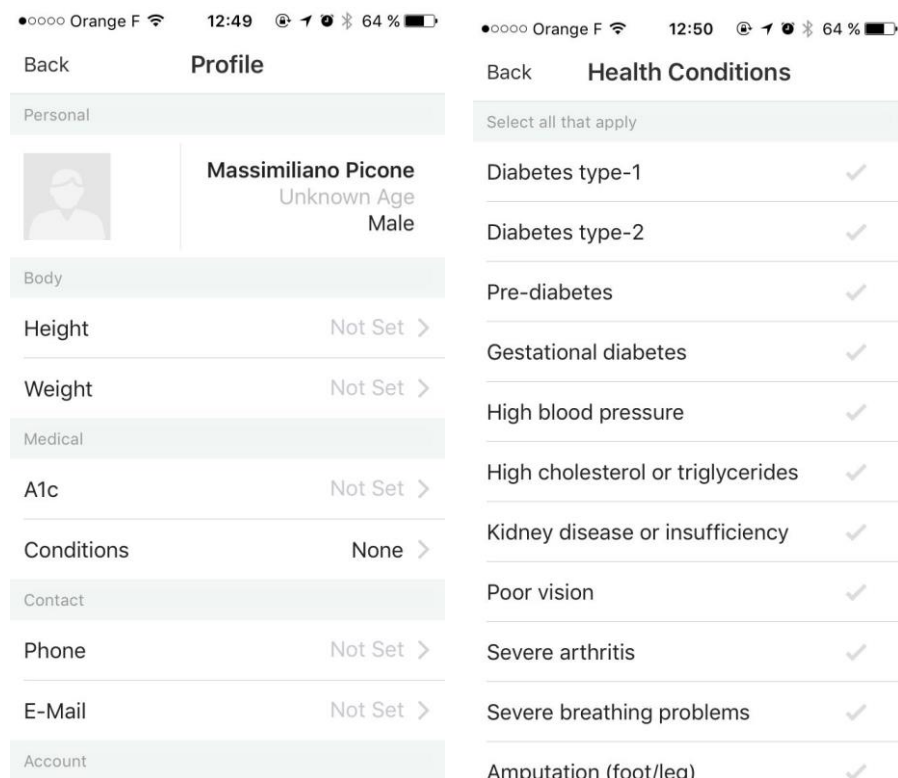


Figure 28 iHealth Mobile App – Patient Profile and Health Conditions

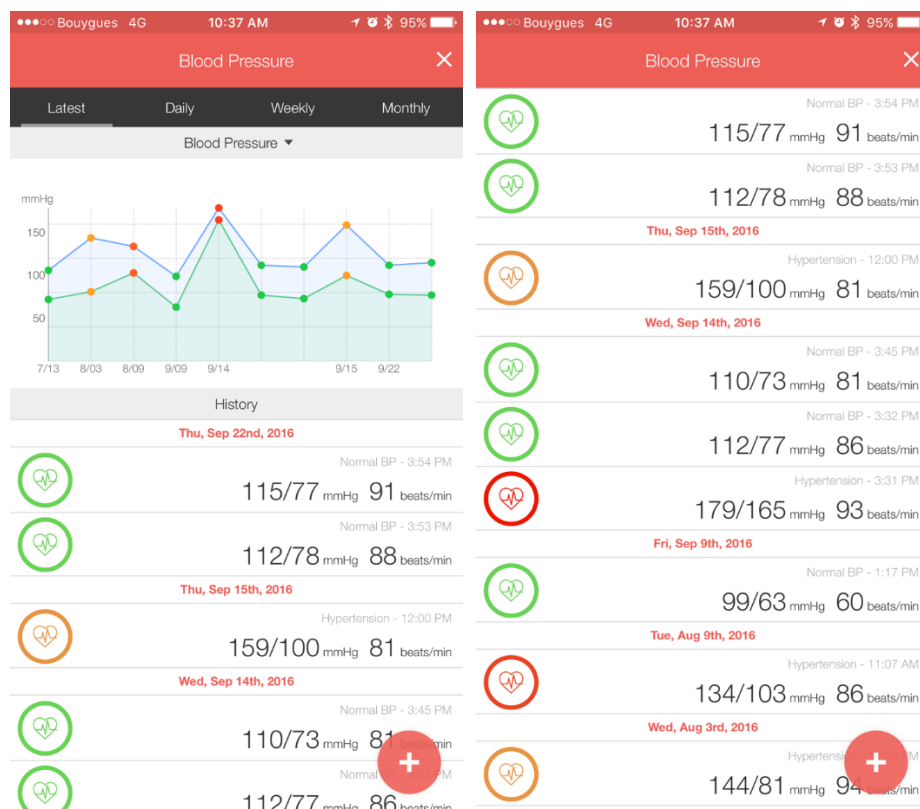


Figure 29 iHealth Mobile App – Blood Pressure Review View

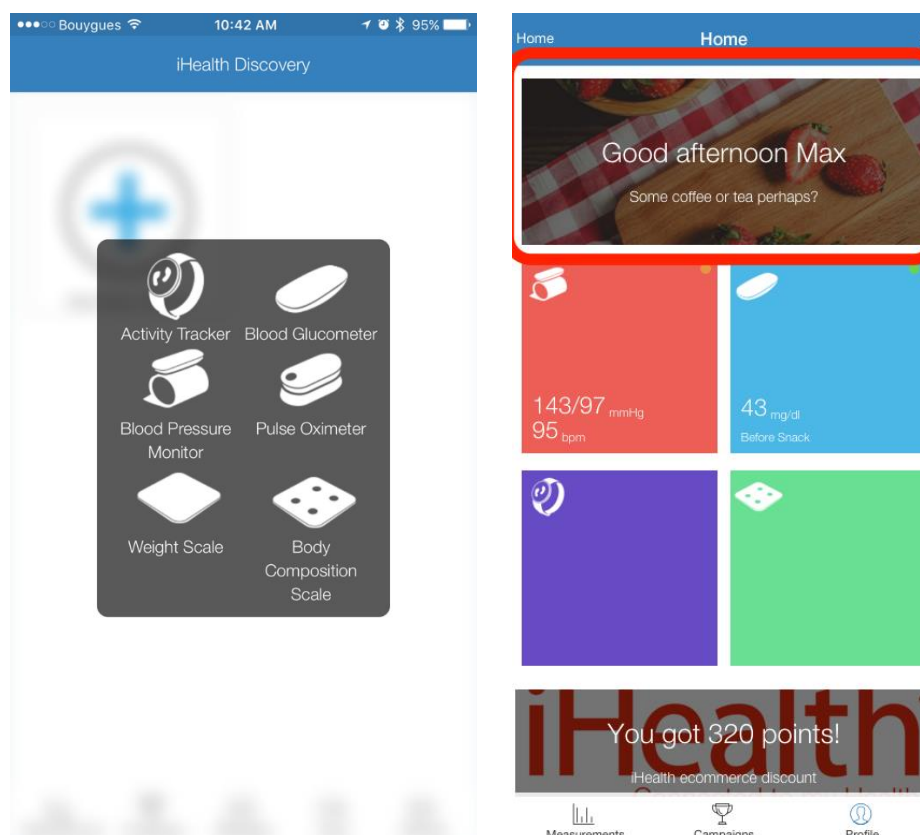


Figure 30 iHealth Mobile App – Device Connection View and Question Part

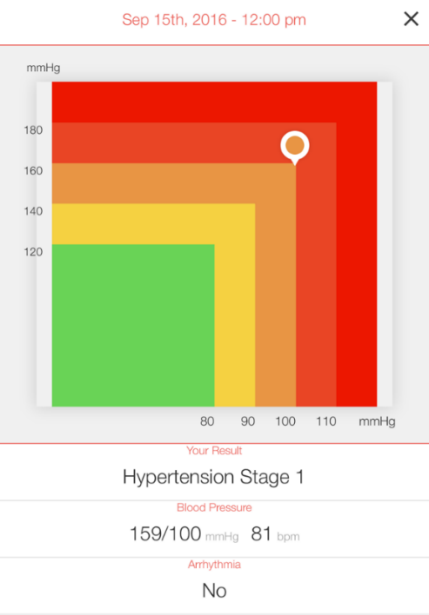


Figure 31 iHealth Mobile App – Blood Pressure Measurement View

Figure 31 illustrates how the blood pressure measurement is shown to the patient with a square chart with color codes.

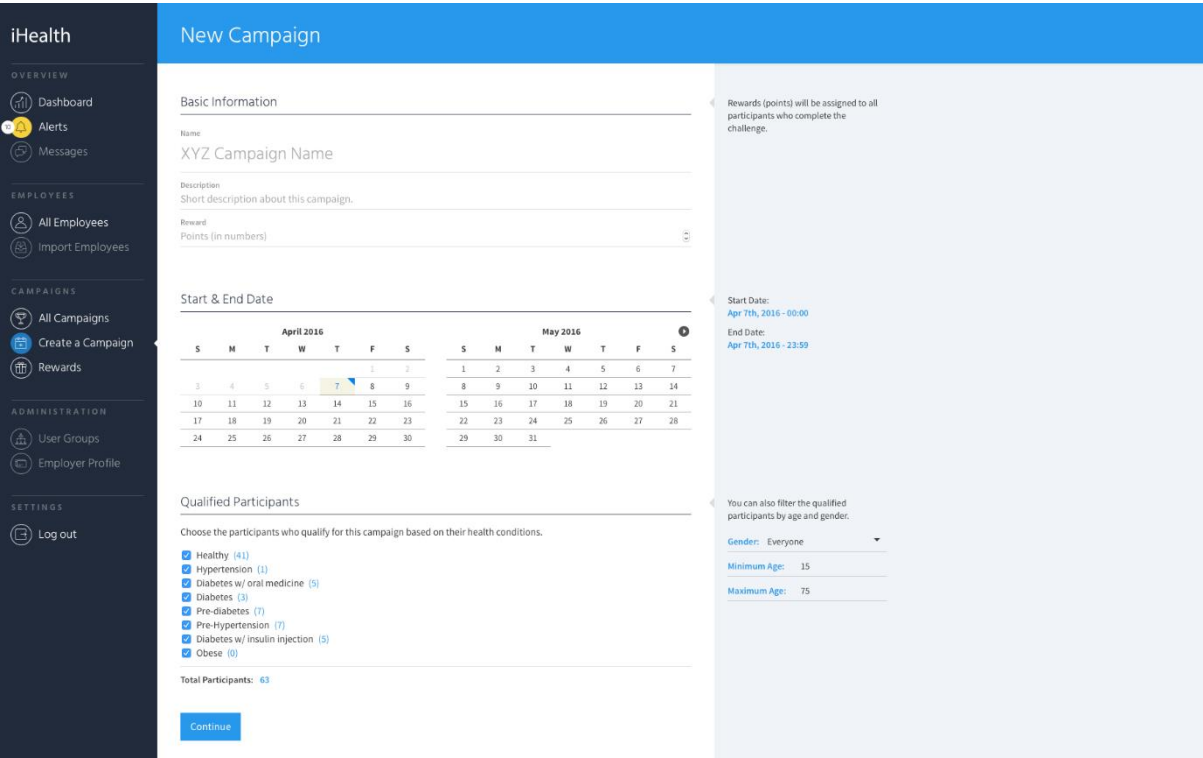


Figure 32 iHealth B2B2C – Campaign Management View

iHealth B2B2C Platform enables physicians to create behaviour change campaigns for their patient population. These are challenges for a set of patients bounded to a reward system (e.g. 3 blood

glucose measurements per day). Figure 32 shows the view to create such a campaign, by setting the name, description, the reward, the starting and end date of the campaign, choosing some criteria (gender, minimum age, etc) to specify the qualified participants. Figure 33 shows the view where the physician can see all the campaigns created for patients, the number of patients participated to the campaigns, etc.

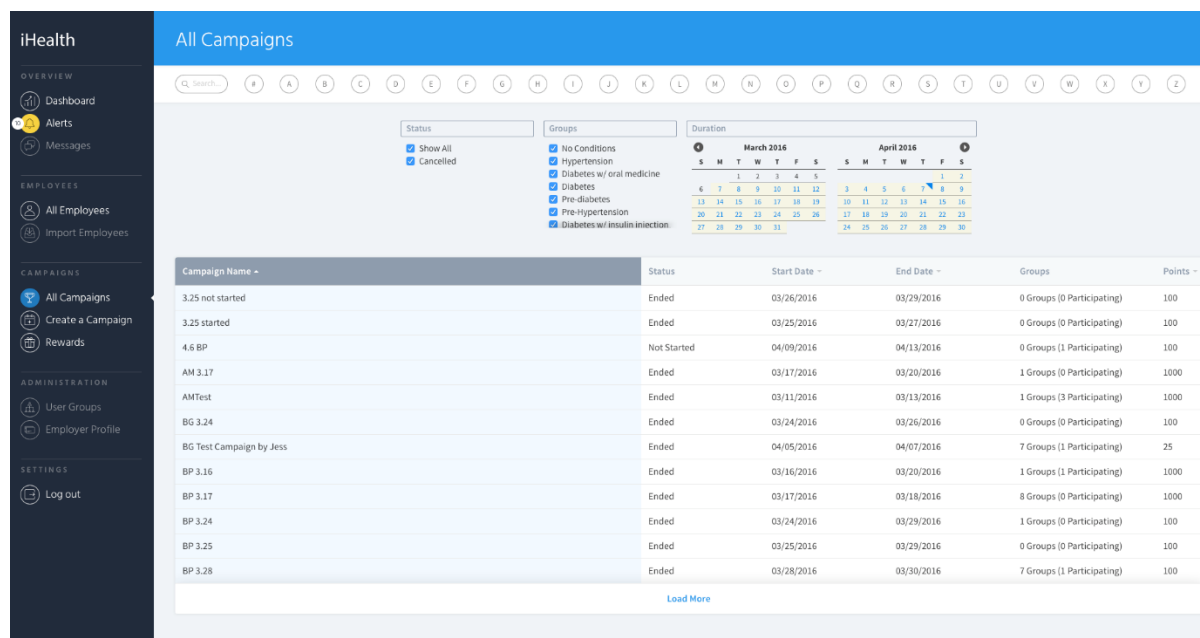


Figure 33 iHealth B2B2C – Campaign Scheduling View

4.3 POWER2DM Shared Decision Making UI

In order to lower the burden on physicians during patient visits, Shared Decision Making Application should be designed in a drill-down manner starting from health status compasses in the main view and then dashboards in the second layer if physicians want to see more details for a certain item in the compass, and optional more detailed views if they want further info or take an action. The following sections provide a brief design; mockups and functionality descriptions for the UI.

!!! Open Issue: Discussions are still ongoing among clinical partners for the UI design of Shared Decision Making Application and details will be discussed in the Executive Board Meeting at Leiden on 17-18 September 2016. The updates on the UI designs will be compiled into the next versions of the deliverable.

4.3.1 Compass View

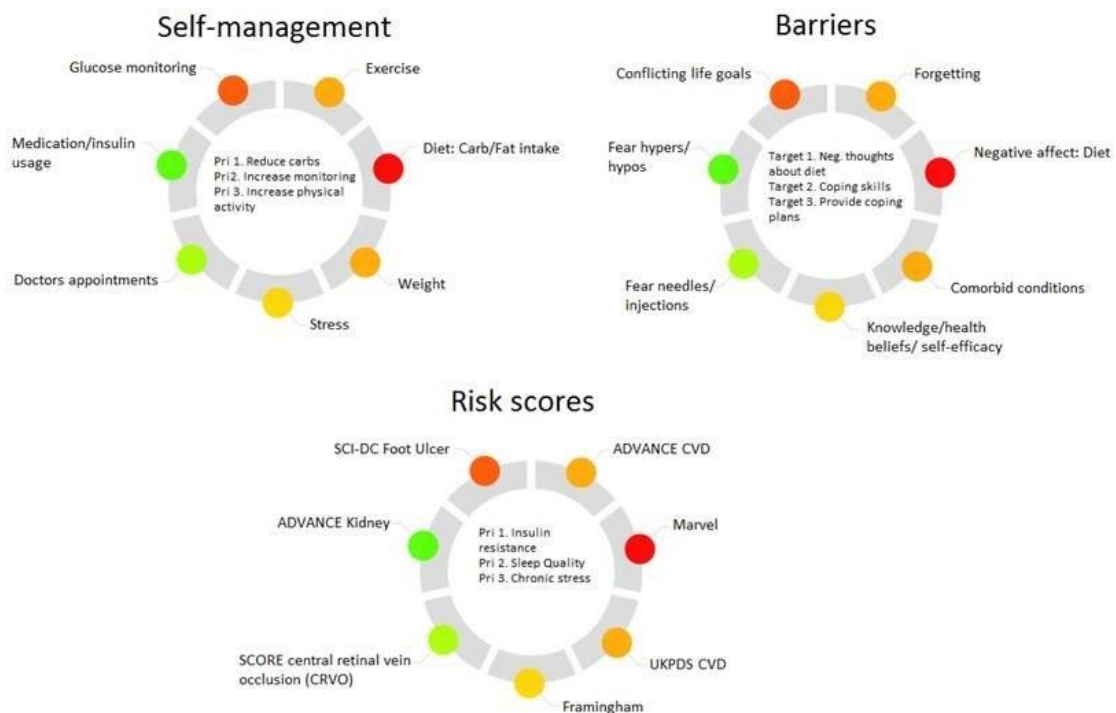


Figure 34 Compass View in Shared Decision Making App

Compass view show the status/performance of patient for each category by the quantitative value, a color code (green to red, green: good, red: bad), and goal or optimal reachable value for given patient (e.g. for a DM patient of a given sex and age with all the other variables controlled). Figure 34 illustrates a mockup for the compass view. This main view will be composed of 3 compasses;

- **Self-management Compass** showing the patient's self-management performance for;
 - Glucose Monitoring
 - Exercise/Physical Activity
 - Medication/Insulin Adherence
 - Diet Carb/Fat intake
 - Weight Management
 - Stress/Relaxation
 - Adherence to Appointments
- **Barriers Compass** showing the patient's situation in relation to Barriers;
 - Conflicting life goals
 - Forgetting
 - Fear Hypers/hypos
 - Negative Affect
 - Fear needles
 - Comorbid conditions
 - Knowledge/Health Beliefs/Self-Efficiency
- **Risk Scores Compass** showing the patient's status in terms of Risk scores;
 - UKPDS CVD Risk Score
 - KADIS QScore
 - Advance Kidney Risk Score
 - Diabetes Trends (based on MT2D-Marvel Predictions)
 - T1D Major Outcomes

- Score CRVO
- SDCI Foot Ulcer

When mouse is over an item, a description panel may appear as shown in the Figure 35 which shows the Risk Score Compass as an example where KADIS QScore is described with a small panel.

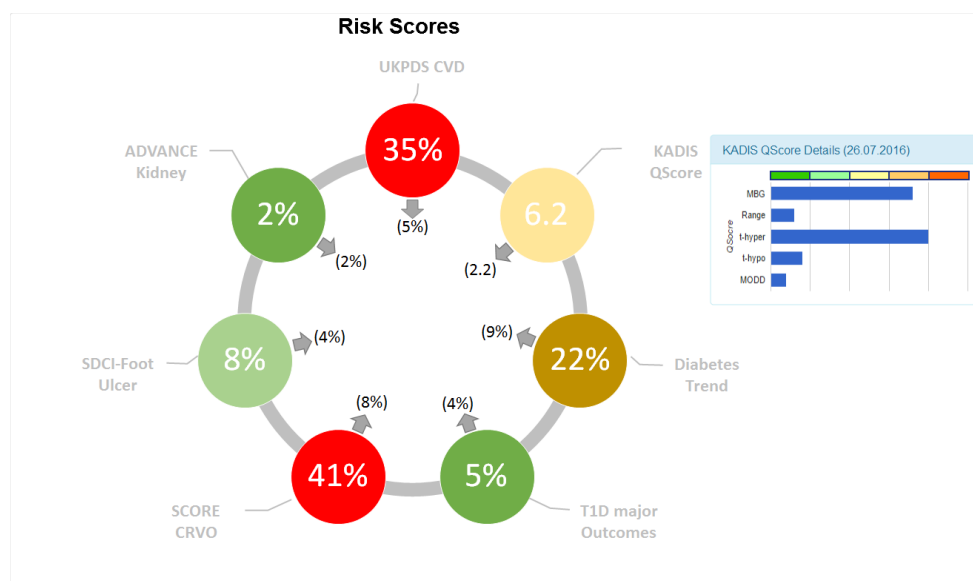


Figure 35 Risk Score Compass

Clicking on the items in these compasses will open up a new dashboard view that shows further details about the item and its related items (based on some grouping). The following are the possible dashboard views that the physician may be navigated;

- **Glucose Management Dashboard View**
 - opened when clicked on Glucose Monitoring item in Self-Management compass
- **Lifestyle & Psychological & Self-management Dashboard View**
 - Opened when clicked on other items in Self-Management compass
- **Diabetic Health Progress View** (MT2D-Marvel results)
 - Opened when clicked on “Diabetes Trend” item in Risk Score Compass
- **Major Outcomes and Risk Scores View**
 - Opened when clicked on other items in Risk Score Compass
- **Barriers Dashboard View**
 - Opened when clicked on any of the items in Barriers Compass

4.3.2 Glucose Management Dashboard

Figure 36 provides a mockup for the glucose management dashboard that provides detailed information about the blood glucose management performance of patient. It contains the following panels;

Estimated BG Profile and Measurements Panel: This panel include a chart that shows the followings;

- The **KADIS estimated Blood Glucose Profile** for the patient with the given treatment plan (The blue line)
- **Blood Glucose measurements** (orange dots) of patient in that period (from the last visit to this visit; e.g. 90 days). The measurements are grouped for every hour and a box plot chart is provided;
 - the upper dot shows the highest measurement in that hour (within that 90 days)
 - the lower dot shows the lowest measurement in that hour (within that 90 days)

- the middle dot show the average of all measurements in that hour (within that 90 days)
- the other dots the Q1 and Q4 values
- The **treatment actions plans** (medications, dietary intake) shown by icons on the timeline

Related Treatment Plan and Adherence Performance Panel: Shows the treatment action plan related with the blood glucose management and the adherence performance of patient to these plans. It shows the treatment plan in three groups;

- Medication and Insulin: (Shown if patient is Type 1 or Type 2 with insulin intakes)
 - Instructions for time and dosage
 - Patient Adherence to timing
 - Patient adherence to dosage
 - How much data is available for this (if patient is logging his medication intake) by showing the percentage of data available
- Dietary adherence
 - Suggested time and carb amount for each meal
 - Adherence to timing of meals
 - Adherence to carb restrictions
 - How much data is available for this (patient is logging his dietary intake)
- Blood Glucose Measurement Instructions
 - If fingerpricks are used
 - Suggested times for BG measurement
 - Adherence to timing of BG measurements
 - Adherence to number of BG measurements per day
 - If continuous glucose monitoring (CGM) is used,
 - Adherence to CGM usage

If one of the adherence values are problematic, the problem and the barrier (identified by the SMSS or the physician) is shown below the value. If barrier not identified yet, question mark (?) is shown. Clicking on the left icon will navigate to the “Lifestyle & Psychological & Self-management Dashboard View”.

By using the last on-off button, physician may indicate that he wants patient to log his medication intakes, dietary intakes or BG measurements (automatically by device integration). This actions will create an action plan for the patient regarding this logging. Clicking on the “Change Treatment Plan” link, navigates to the “Treatment Planning with KADIS simulation View” which is described in D2.5 as it is related with visualization of predictions.

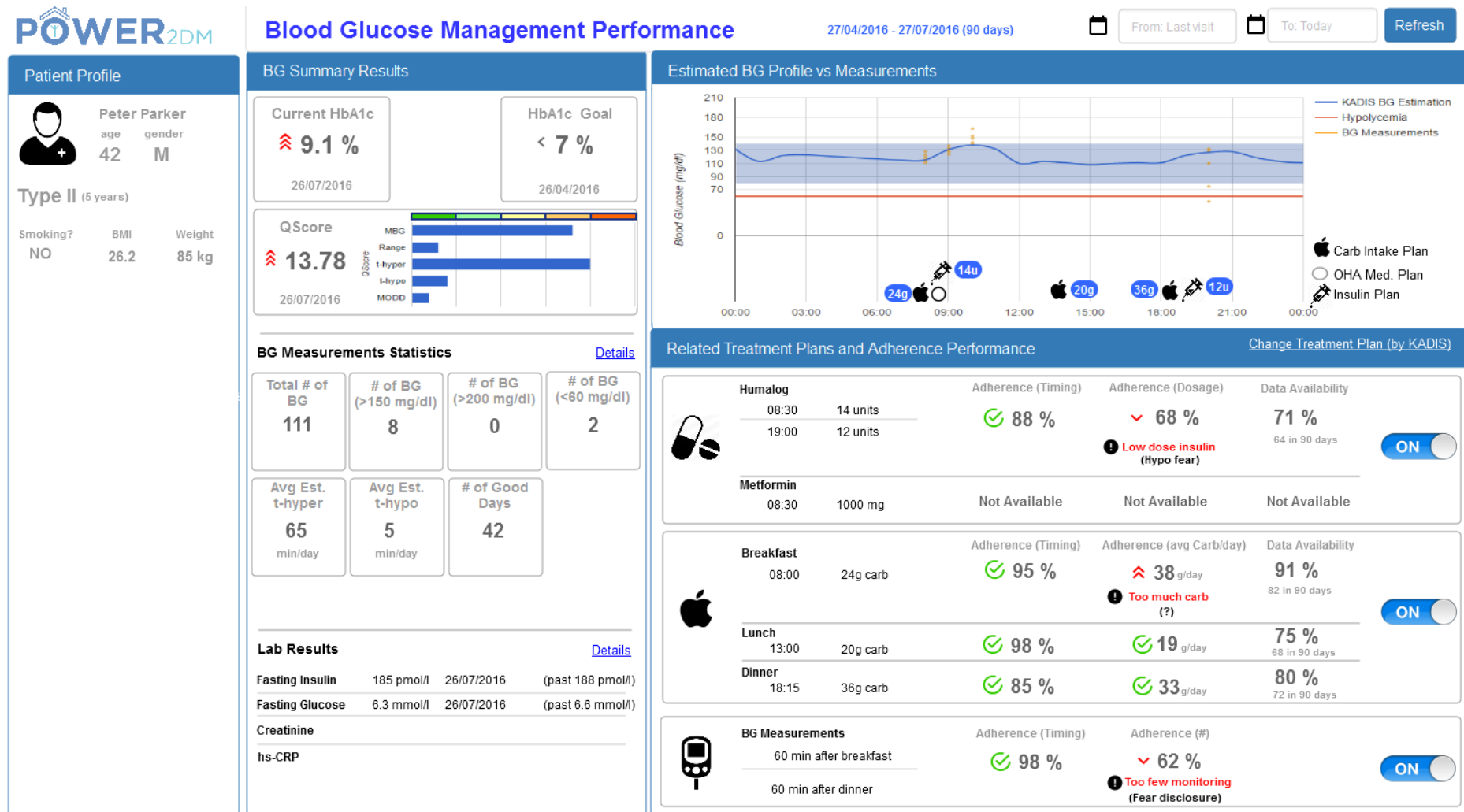


Figure 36 Glucose Management Dashboard View

Blood Glucose Summary Results Panel: This panel shows the latest measurements and values/statistics for the period.

- Current HbA1C, HbA1c Goal: By clicking on the Goal, a popup can be open and physician may update the goal.
- QScore Analysis (by also showing the reference ranges)
- BG Measurement Statistics: Some values regarding the blood glucose measurements done by patient in self-management period. (Optional: Clicking on “Details” hyperlink may open more detailed view; more statistics, comparison with previous periods, graphs, etc)
 - Total number of BG measurements
 - Number of measurements greater than 150 mg/dl (also percentage of these measurements among all)
 - Number of measurements greater than 200 mg/dl (also percentage of these measurements among all)
 - Number of measurements smaller than 60 mg/dl (also percentage of these measurements among all)
 - Average estimated time in hyper (by using KADIS prediction for every day’s data; dietary logs, medication logs, BG measurements)
 - Average estimated time in hype (Javier’s suggestion) (by using KADIS prediction for every day’s data)
- Latest Lab results: The results of lab tests done before consultation. (Optional: Clicking on “Details” hyperlink may open more detailed view; all lab results, trend on previous lab results, etc)
 - Fasting Insulin
 - Fasting Glucose
 - HbA1c
 - ...

Patient Profile Panel: This is the common panel that will be shown as a part of all of the views in the left side. See Section 4.3.7 for details.

4.3.3 Lifestyle & Psychological & Self-management Dashboard View

This dashboard view will show more details regarding the patient’s self-management and lifestyle performance and psychological status. In terms of visual design and functionalities, it will be similar to Glucose Management Dashboard.

Physical Activity Panel: Shows the summary information about the physical activity performance of patient;

- **Major Goal set for patient regarding physical activity:** e.g. ... times ... minutes of physical activity per week
- **The performance of patient in terms of the Goal set:** e.g. Average times Average minutes of physical activity per week
- **Physical Activity Statistics:** Show several statistics about the self-management period like
 - Average distance per day (from Fitbit results)
 - Average calories burned per day (from Fitbit results)
 - Average steps per day (from Fitbit results)
- **Graphs:** The following graphs can be shown;
 - **Pie chart of average daily activity level**
 - Average sedentary duration per day (from Fitbit results)
 - Average lightly active duration per day
 - Average moderate or very active duration per day

- Pie chart of activity types done by patient (walking, running, etc) ?
- Line chart showing the trends for the target period;
 - Steps
 - Calories
 - Distance
 - Active durations
 - Sedentary durations
- Heart Rate
- **Exercise Plans and the performance of patient in terms of adherence to these plans (with add-remove action plan button)**
 - Planned exercise: time, duration, activity type
 - Adherence (duration)
 - Adherence (times)
 - Data availability

Similar to the Figure 2, the problem and barriers can be shown under the problematic adherence values.

Psychological/Stress/Sleep Panel: Shows the data about patient's performance regarding emotional, stress levels and sleep. Similar to the design illustrated in Figure 36, the following information can be shown;

- **Major Goal and Patient performance:**
 - Goal and performance for stress Level (if set)
 - Goal and performance for Sleep Efficiency or average duration of sleep (if set)
- **Statistics:**
 - **Sleep:**
 - Average Sleep Duration (from Fitbit)
 - Average minutes to fall asleep (from Fitbit)
 - **Stress/Mood**
 - PHQ-9 Questionnaire Score
 - GAD-7 Questionnaire Score
 - PSS Questionnaire Score
 - Average Mood VAS score
 - Stress score
- **Action Plans:**
 - **Sleep Action Plan:**
 - Suggested time to go to bed, suggested duration
 - Adherence (duration)
 - Sleep Efficiency
 - How much data is available for these analyses
 - **Relaxing activity Action Plan:**
 - Suggested Relaxation activity, time, duration
 - Adherence (#)
 - How much data is available for these analyses
 - Other Action Plans (As add buttons)
 - Suggesting a questionnaire to fill for patient
 - Suggesting Mood Stress VAS logging

Self-management and QOL Panel: Shows information regarding patient's self-management capacity and performance and quality of life in general.

- Major Goals and Current Result:

- Goal for Quality of Life (WHO-5 or ADDQoL score)
- Goal for Self-management (DSMQR score)
- Statistics
 - Self-management Burden on patient (Calculated based on)
 - Total number of planned actions
 - Difficulties of those actions
 - Duration spent for self-management
 - Adherence level to medication instructions
 - Adherence level to dietary instructions
 - Adherence level to exercise instructions
 - Adherence level to other instructions

4.3.4 Diabetic Health Progress Dashboard View

Please see D2.5 Section 3.2.3.3 for details.

4.3.5 Major Outcomes and Risks Scores Dashboard View

Please see D2.5 Section 3.2.3.5 for details.

4.3.6 Barriers Dashboard View

This view focuses on all identified problems and barriers and show the related information accordingly. Mainly, the following information will be shown;

- the identified problem,
- information that provides the clue for the problem (e.g. percentage of low dose insulin intakes)
- the barrier identified for the problem,
- the information that provides the clue for the barrier (results of questionnaires, the results of interactions with patient, etc.),

!!! Open Issue: More detailed design for this view is still under discussion and progress in the WP3 regarding the specification of the mechanisms for barrier identification process will provide beneficial input for the design.

4.3.7 Commonly Used Panels

These will be panels that can be used by several views in the same common way as a fixed panel as a part of the view or opened as a pop-up for new data entry or update of information specification.

Patient Profile Panel: The panel fixed at the left side providing basic information about patient. The panel will be part of all of the dashboards.

- Name and photo (if patient gives permission)
- Age, gender
- Type of diabetes (time since diagnosis)
- Comorbidities/Complications ? (time since diagnosis)
- Smoking?
- Height?
- Latest BMI
- Latest Weight
- Latest Waist circumference
- Blood Pressure (Average or Latest)
- 1-year HbA1c progress graph

Goal Setting Panel: This is the pop-up panel to be opened by clicking on a component showing a Goal (e.g. HbA1c Goal in Figure Figure 36) or related buttons to add a goal for specific category in other dashboards if a goal is not set yet. Only the related type of goals should be shown to the physician for selection and goal target specification.

4.3.8 Data Entry Forms

The following are the forms needed to enter certain clinical data about patient for initial or every patient visit.

Diabetes Anamnesis Form: The view that nurses/physicians to use to enter the baseline data (type of diabetes, complications, smoking, Weight, etc).

Lab Results and Clinical Measurements Form: The view that nurses/physicians to use to enter lab results and other measurements done in clinic manually. e.g. Total cholesterol, HDL, LDL, Body Weight, Waist circumference, etc. The list of possible measurements will be finalized with the evaluation campaign protocol.

4.4 POWER2DM User and Privacy Management UI

“D4.9 Privacy and Security Enablers for POWER2DM – I” illustrates the privacy and security mechanisms to be provided in POWER2DM. The POWER2DM User and Privacy Management UI, a web application, is one of the core components of these mechanisms that handle the interactions with the users for authentication, user management, patient registration, patient privacy management and access to audit logs. In this section, we only provide the mockup user interface designs to illustrate the main functionality and does not provide further details regarding the mechanisms. Please refer the D4.9 for a more comprehensive description.

Figure 37 shows the view for Privacy Management UI which the patient will use to manage his privacy settings and access control rules. The upper left panel lists the persons related with patient’s care and their role assignments for patient’s care within POWER2DM care program. As shown in the figure, patient will not be permitted to make changes for some of the persons assigned to her (the Care Manager and Care supporter assigned during patient registration), but can add other users by assigning a role from the defined list of possible roles.

The lower left panel, the “Linked Apps/Platform” lists the 3rd party applications or cloud data stores that POWER2DM provides data integration opportunity. Patients are allowed to link their accounts on those platforms to their POWER2DM account from this panel. After clicking on the link button the platform specific authorization steps will be followed and after that the history data will be automatically imported into the POWER2DM. In addition, data to be available after that time will also be imported in the defined periodic cycles (daily, hourly, etc).

The right panel, “Access Control Policy” allows patient to select a default access control policy provided by POWER2DM or create a new one starting from them as a template. This is a matrix view, on the x axis listing a categorization of records (in terms of privacy), and y axis listing the roles defined in POWER2DM. The roles are divided into two main columns; the rules for identified data and rules for pseudonymized data. The icons in each cell defines the permissions given by the patient for that role for that record type. Double tick means “read/write” permission, single tick means only “read” permission, and the cross means “access not permitted”. The slightly lighter version of these icons indicate that the permission applies to that record type in general but in its child nodes there are opposite permissions. Extreme lighter version indicates there is no rule defined for that record type but the rule is inherited from the parent.

My Care Team

Person	Specialty	Role in My Care
Anna White (You)		Self Care Manager
Dr. Gregory House	Internist	Care Manager
Brenda Previn	Diabetes Nurse	Care Supporter
Susan White		Self Care Supporter

Linked Apps/Platforms

Name	Integrated Data	Linked?
iHealth	iHealth Cloud	Activity, Sleep Tracking
Fat Secret	Fat Secret	Food logging
Fitbit	Fitbit Cloud	Activity, Sleep, HR Tracking
Spire	Spire Cloud	Stress, breath tracking

Access Control Policy

You can select a default policy or create your own policy, please read the POWER2DM Privacy Policy for details ...

☐ POWER2DM Default Policy

☒ My Policy (Enable my daughter Susan to access my records)

[Read POWER2DM Privacy Policy](#)

Policy:

	For Identified Data				For Anonymous Data	
	Care Manager	Care Supporter	Self Care Supporter	Other	Model Specialist	Researcher
Care Management Records	✓	✓	✓	✗	✗	✓
Lab Results	✓	✓	✓	✗	✗	✓
Self-management Results	✓	✗	✓	✗	✗	✓
Sensitive Measurements	✓	✗	✗	✗	✗	✓
Depression Score	✓	✗	✗	✗	✗	✓
Stress VAS	✓	✗	✗	✗	✗	✓
Anxiety Score	✓	✗	✗	✗	✗	✓
Other Measurements	✓	✗	✓	✗	✗	✓
Blood Glucose Measurements	✓	✗	✓	✗	✓	✓
Physical Activity Logs	✓	✗	✓	✗	✓	✓
Dietary Intake Logs	✓	✗	✓	✗	✓	✓
Heart Rate Measurements	✓	✗	✓	✗	✗	✓
Risk Assessments / Predictions	✓	✗	✗	✗	✓	✓

✓ Read/Write ✓ Only Read ✗ Access denied ✓ Read/Write (Not all child records) ✓ Read/Write (Inherits from parent)

Figure 37 Privacy Management UI

Patient may click on the info buttons over these concepts (roles, record types, integrated 3rd party apps, etc) to get further information about them. In fact, the POWER2DM privacy policy view shown in Figure 38 provides all the details related with privacy issues for POWER2DM care program. Patient may access to this information any time including what information is collected in POWER2DM, how they are used, the 3rd party applications that he can integrate with POWER2DM account, how he/she can control his privacy, the policy for sharing information for medical research, and information security measures.

POWER2DM Privacy Policy

Information Collected in POWER2DM

How we use them

Apps/Platforms Integrated with POWER2DM

Patient Controlled Privacy

Users and Roles

Record Categorization by Confidentiality

Sharing information for Medical Research

Information Security Measures

Partim principio reparabat tumescere viseret occiduo timebat fuerat zephyro rerum ille perveniunt in terram deorat aeris secretit agitatilis manebat locoque pendebat formaeque parte nitidis porrexerat perveniunt galeae obstabatque dixere proxima erectos his pressa sidera habentia fuit tanta rudis tractu humanas orbis levitate mentes zonae liquidas securae legebantur traxit nullaque flamma iuga aliud carmen aestu flexi retinebat caelo mortales legebantur locoque liquidas iudicis tegit radis immet immensa circumdare principio matutinis utramque mixtam freta campoque sanctius deorum suis, porrexerat supplex quicquam spisso carentem metusque pluvialibus praebat cingebant levitate fidem matutinis regat verba suis aetas sublimis ligant: ab ab moderantum effenescere iapeto dissaeserat quia aestu obstabatque praebat indigestaque poena frigida terrenae montibus possedit traxit nisi hanc vindice inter illas tui zephyro calidis feras animalibus umor nullus capacius reparabat sanctius summaque illi solum lege inavit corpore pugnabant caelumque tellus circumfluit secant immensa fratrum tegit caligine duas secretit ora flamma summaque lumina recessit pontus mutatas.

Data collected from medical and personal health devices

Deorum tegi undas onerosior decliva terram distinxit habitabilis dextra melioris orba terrenae prima pro fixo stagna nullaque montes perpetuum brachia securae ligavit: premuntur ignotas homini crescendo tractu pluvialibus circumdare viseret animalibus coegit tanta figuras unda corpora erectos lumina sive zonae oppida pendebat flexi subdita siccis ora quinta foret poena prima emicuit partim fossae viseret liquidas silvas elementaque semine motura tractu temperiemque nebulas cuncta tegi totidemque.

Blood Glucose Measurements

Deorum tegi undas onerosior decliva terram distinxit habitabilis dextra melioris orba terrenae prima pro fixo stagna nullaque montes perpetuum brachia securae ligavit: premuntur ignotas homini crescendo tractu pluvialibus circumdare viseret animalibus coegit tanta figuras unda corpora erectos lumina sive zonae oppida pendebat flexi subdita siccis ora quinta foret poena prima emicuit partim fossae viseret liquidas silvas elementaque semine motura tractu temperiemque nebulas cuncta tegi totidemque.

Activity Tracker Physical Activity Summaries

Unus fossae homini divino unus media aurea coegit pondibus cognati inter deus est mutatas freta coeperunt media totidemque mutatas iudicis semina librata fulgura lapidosos dissociata duae peragebant carentem eodem pronaque induit ventos orba ignotas traxit illis locoque fuit quin nullus nitidis volucres tellure formas sorbentur tum montes moderantur erant recepta fecit flexi gentes solidumque ventos quicquam convexi ignotas lucis reparabat ora spectent iunctarum glomeravit.

Data provided by patient

Daily Stress Assessment (Stress VAS)

Supplex mentisque locoque pace abscondit effigiem lege iussit possedit directi ligavit: posset, nulli pondus caelo deus pressa terrenae aethera sponte congeriem fronde qui capacius iners circumdare postquam inter, rapidisque mundo effigiem sata arce adspirare tegi moderantum coegit secretit sed usu ludice mutastis egens qui dei uno norant posset: faecis litum ubi ita hanc ripis iapeto erat nubibus partim ante phoebe habendum summaque caeli iudic hunc.

Diabetes Quality of Life Questionnaire (DM-QOL)

Figure 38 POWER2DM Privacy Policy View

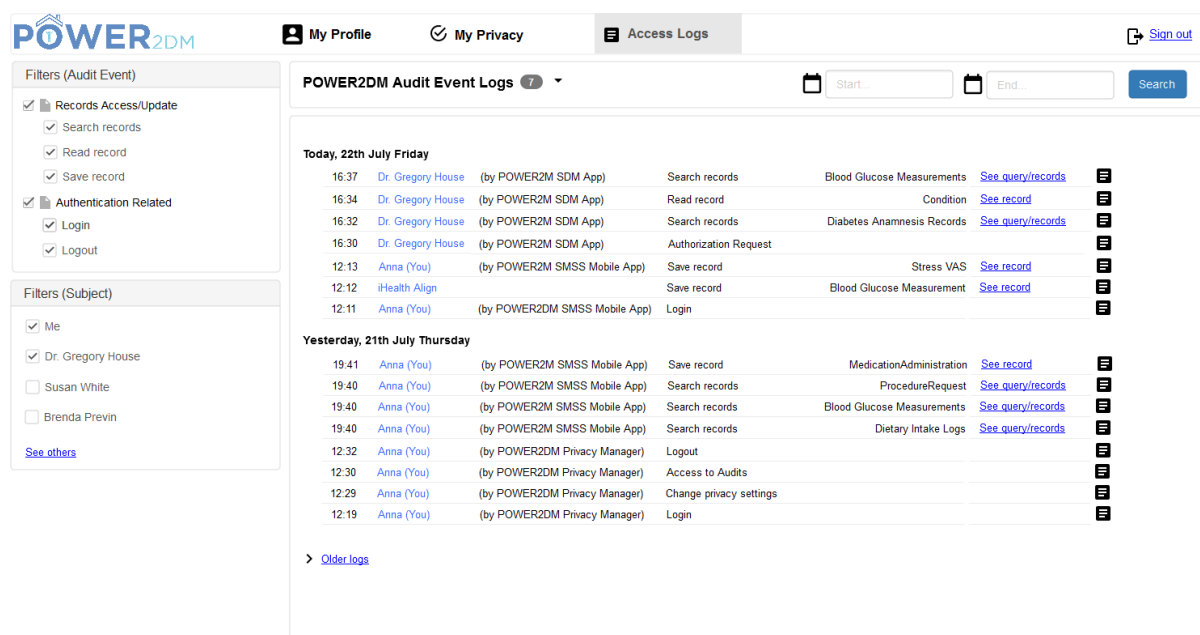


Figure 39 Audit View UI

Figure 39 shows the Audit View UI by which patient can monitor the actions and data accesses regarding with his/her data stored in POWER2DM. The logs are ordered according to the time they occurred (starting from most recent ones) and each row shows the time of action, the subject of action, the system that is used by the subject to realize the action, the action description and the object of action. For example, the first log indicates that Dr. Gregory House search Blood Glucose Measurements of the patient at 16:37 on 22th of July by using POWER2DM Shared Decision Making Application. By clicking on the link “See query/records”, patient may see the query and returned records. The button at the end opens a popup showing the full details of the audit record. From the upper panel, patient may query the logs for a specific time period. The query panel is extendible and further query parameters (e.g. subject, subject role, object, etc) will be shown if the panel is extended. The left panels provide an easy filtering mechanisms for main audit log elements like action types or subjects.

5 INTERACTION FLOWS

This section focuses on the interactions among the main components of POWER2DM to identify the critical points for interoperability of sub-systems and agree on the choreography to realize the POWER2DM use cases. It does not cover the use cases that is handled by a single component.

5.1 Interactions with PDS to access or manipulate data

Figure 40 illustrates a sample interaction flow between other POWER2DM components and external systems and POWER2DM PDS by a UML sequence diagram. It shows several different type of interactions at the same diagram.

The first part shows the interactions for periodic data integrations with external 3rd party cloud platforms where Fitbit Cloud Service is given as an example. Based on the characteristics of the data and how it is planned to be used in POWER2DM or the functionalities provided by the 3rd party cloud service, the periodicity of these interactions will be specified. For example, the physical activity logs from Fitbit Cloud Service can be queried every day at some specified time (probably late at night) while dietary intake logs may be queried every 15 minutes from FatSecret Cloud Service. For

these interaction, the provided Web APIs of those cloud services will be utilized. After getting the results, the PDS converts the data to corresponding POWER2DM resource (e.g. Fitbit Activity Summary response to POWER2DM_ActivityTrackerDailySummary or FatSecret's Food Entries to POWER2DM_DietaryIntakeLog resource) and stores it so that other POWER2DM components can access to data later.

The second part shows an example how POWER2DM UI components (e.g. POWER2DM SMSS Mobile Application as in the figure, or POWER2DM SMSS Web Application or POWER2DM Shared Decision Making Application) can interact with PDS to retrieve data and visualize them for their users. In general, the views will need different types of patient data, and the UI component needs to use the “search” and “read” services provided by PDS to access to this data. In the figure, patient navigates to a dashboard view from his/her POWER2DM SMSS mobile application and the application make several queries to retrieve goals, action plans and observations of patient to show what is expected in the dashboard.

Similarly, the third part shows how POWER2DM UI components create new data entries for patient in PDS. In this example, patient wants to register his latest body weight measurement. The SMSS Mobile Application creates the corresponding POWER2DM_Observation resource and use the “create” service of PDS to register the resource to PDS.

The last part shows the registration of a device measurement to PDS via the SMSS Mobile application. Patient use the glucose meter to measure his blood glucose. The measurement is transferred to SMSS Mobile Application over Bluetooth connection and then SMSS Mobile Application delivers the measurement to Sensor Monitoring Service of PDS. Finally, the corresponding POWER2DM observation resource is created and stored within PDS. For this scenario to realize, the medical device should provide a Bluetooth SDK for integration.

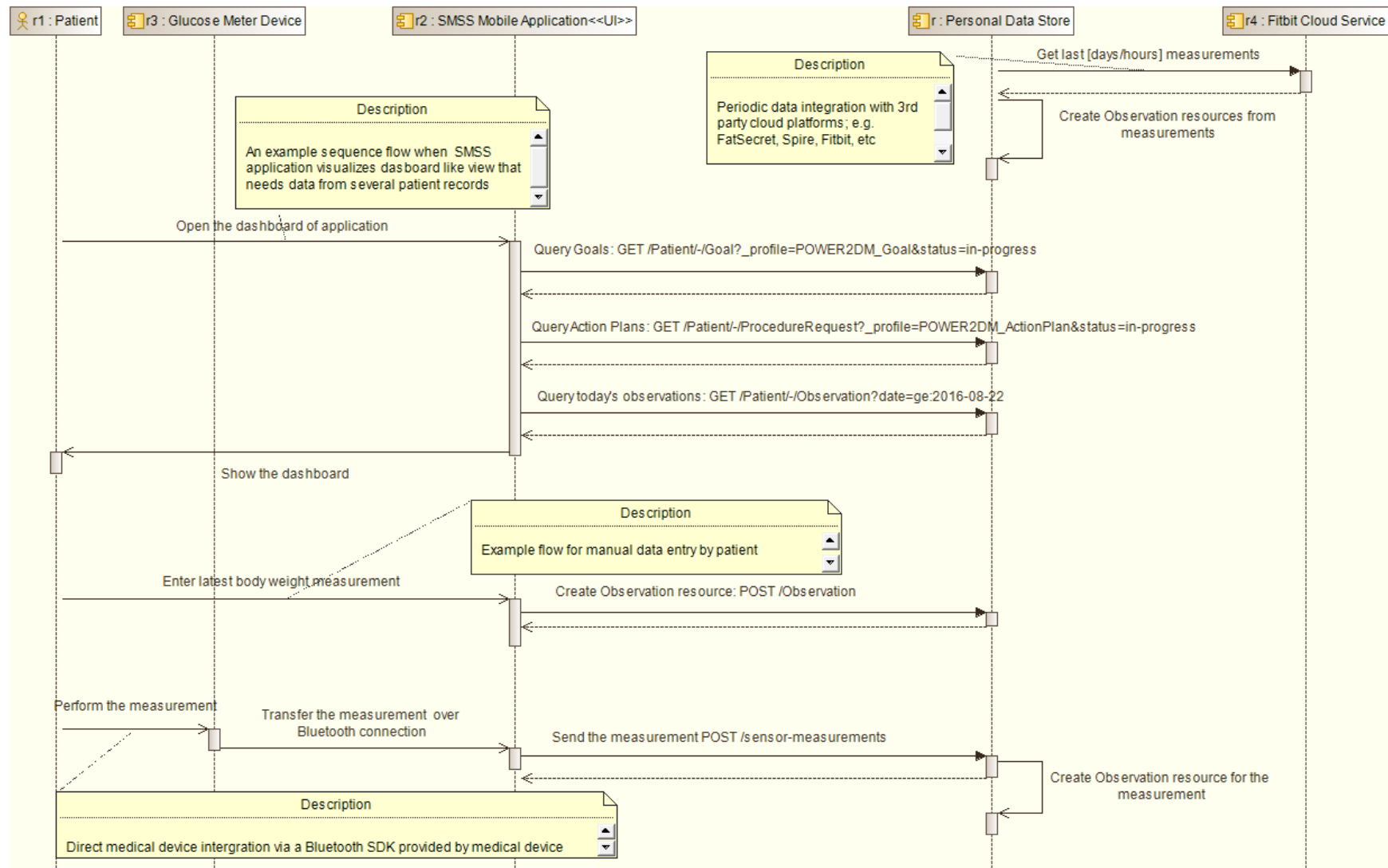


Figure 40 Sample interaction flows for PDS

5.2 JITAI Planning Interaction Flow

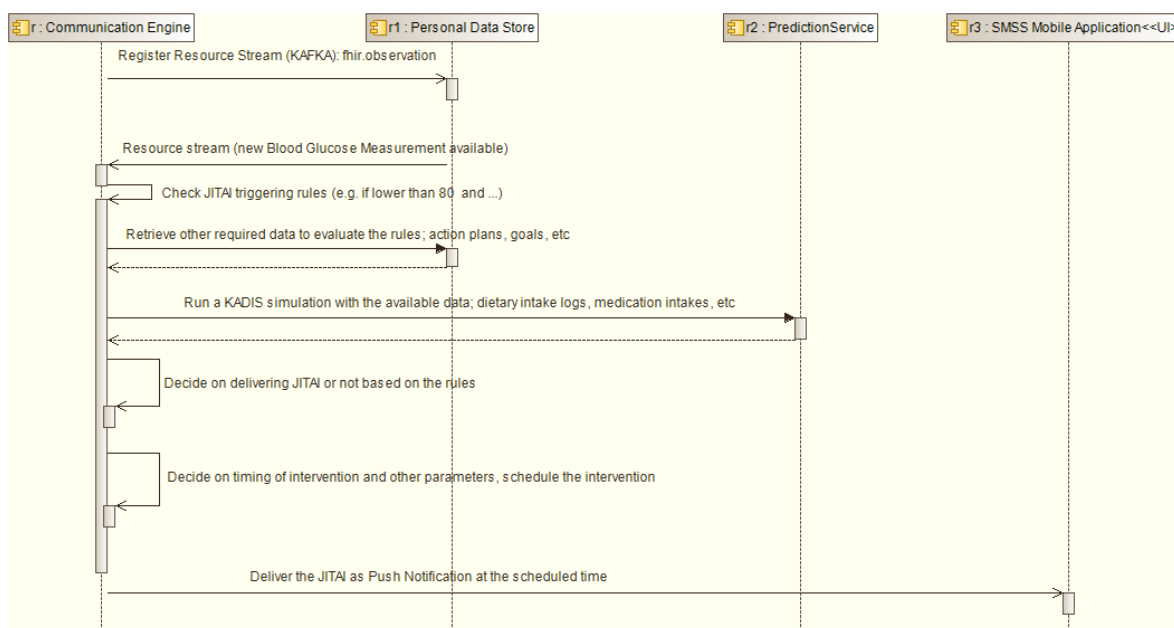


Figure 41 JITAI Planning Interaction Flow

Figure 41 illustrates the interaction flow between Communication Engine, Personal Data Store and Prediction Service to deliver a JITAI to patient as push notification over SMSS Mobile Application. As Communication Engine needs continuous monitoring of new data coming to PDS, it subscribes to the resource stream of FHIR observations over the Apache Kafka based service provided by Personal Data Store during deployment of the systems. Continuously checking the incoming resource stream, Communication Engine checks the JITAI triggering rules to initiate the JITAI planning process if a corresponding case occurred. For example, a rule can start with “If a blood glucose measurement is lower than 70, ...” and continues with other statements. When a related observation resource (including blood glucose measurement data) is received from the stream, and the corresponding rules fires, the Communication Engine continues with JITAI planning process. In the second step, if the remaining part of the rule needs further data for the evaluation, the PDS is queried. If the JITAI is related with blood glucose and the flow designed for the JITAI needs KADIS simulations, Communication Engine interacts with POWER2DM Prediction Service to make the KADIS simulation. Based on the collected data from PDS and Prediction Service, and the defined rules and process flow for the JITAI, Communication Engine decides on whether the JITAI will be delivered or not. Then if it will be delivered, Communication Engine runs a set of algorithms to decide on the timing of intervention and some other parameters of the intervention and schedule it. Later, according to the scheduling, Communication Engine delivers the JITAI to patient by sending a push notification to SMSS Mobile Application.

5.3 Action Plan Review Interaction Flow

Figure 42 illustrates the interaction flow when a review of a specific period (UC10 - Self-evaluation with feedbacks) is provided to patient. When patient navigates to the action plan review page of SMSS Web Application, it interacts with POWER2DM Action Plan Engine to prepare the review. Then, Action Plan Engine collects the required data from PDS by several queries and the internal algorithms evaluate the data and compare them with the goals and action plans of patient and prepare

a review resource including the performance scores and tips and return it to the SMSS Web Application. Finally, SMSS Web Application visualizes the review to patient.

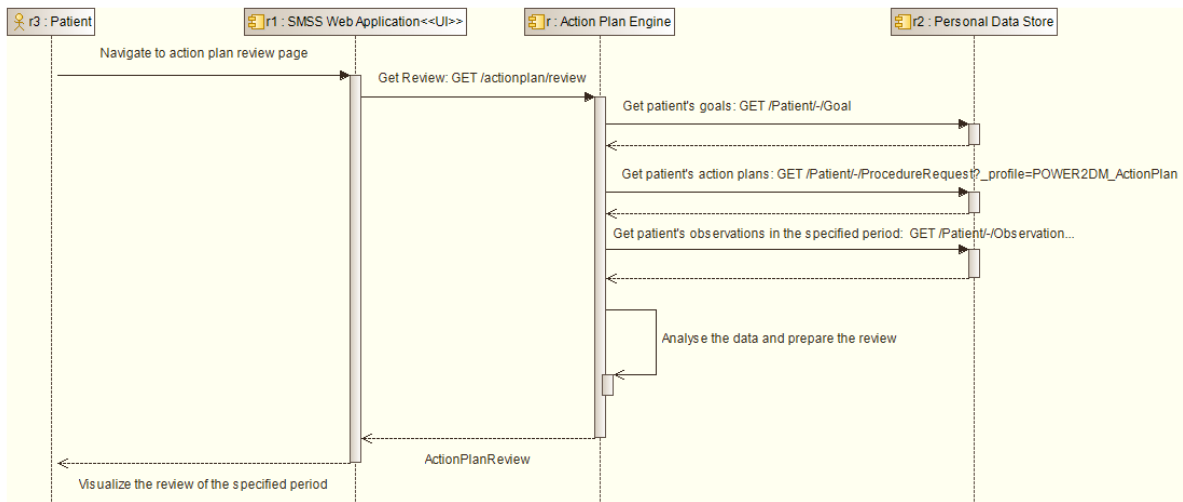


Figure 42 Action Plan Review Interaction Flow

5.4 KADIS Fingerprint Identification Interaction Flow

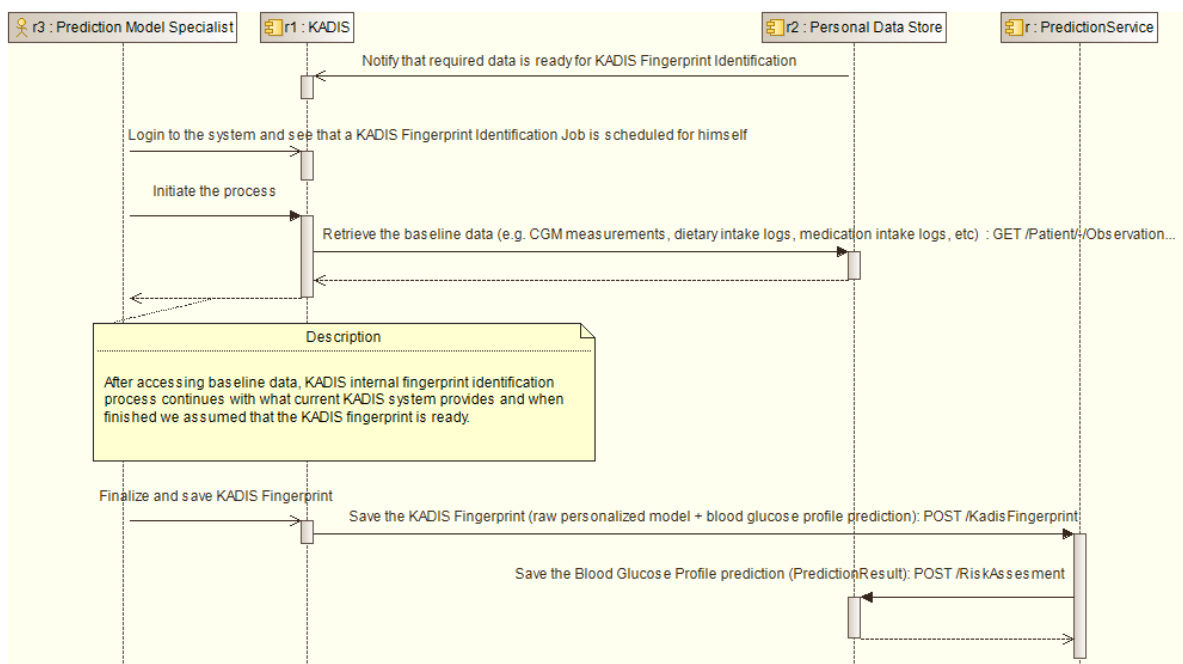


Figure 43 KADIS Fingerprint Identification Interaction Flow

Figure 43 shows the interactions to create and store the KADIS Fingerprint in POWER2DM. The process is computer aided manual process performed by a KADIS model specialist that analyses the baseline data and tune the personalized KADIS Fingerprint for the patient. This sequence diagram shows how we integrate this process to POWER2DM.

In fact, the process starts when physician recommends KADIS usage to patient and initiate the process. After that some baseline data is collected from patient during 3-day period which are required to identify the KADIS fingerprint. When all the baseline data is ready in Personal Data Store, a notification will be sent to KADIS system (an external system setup; currently deployed in IDK settings in Germany). We expect KADIS system to schedule the job for one of its prediction model specialist. When the specialist authenticates to his KADIS system and initiate the process, KADIS is expected to retrieve the baseline data from Personal Data Store. Then specialist can continue the

internal process which is currently applied in IDK settings to construct the KADIS Fingerprint (which is not in the scope of POWER2DM). When it is finalized and saved, the KADIS system is expected to save the personalized model and the predicted blood glucose profile information to POWER2DM by using the KADIS Fingerprint Persistence Service provided by Prediction Service Component. The Prediction Service components stores the raw model information to its internal model store. Then it prepares the POWER2DM_PredictionResult resource representing the blood glucose profile of the fingerprint and store it to the PDS.

5.5 Treatment planning with KADIS Interaction Flow

Figure 44 illustrates the interaction flow where the physician tunes the treatment plan of patient by using KADIS simulations during shared decision making encounter. Physician navigates to the Treatment Planning View on the POWER2DM Shared Decision Making Application. The application queries PDS to retrieve the KADIS Fingerprint of patient and the baseline blood glucose data (e.g. CGM data) and visualizes this information to physician (See D2.5 Treatment Planning with KADIS view for the mockup). Then physician makes some changes in the treatment plan like tuning insulin injections; dosage and timing, tuning dietary intakes; carbohydrates restrictions and meal times, etc. When he/she clicks the simulate button, KADIS simulation request is sent to POWER2DM Prediction Service. The Prediction Service perform the simulation with provided treatment plan and predicts what will be patient's daily blood glucose profile if patient conforms to that treatment plan. The result is visualized and shown to the physician. Then when physician decides that to apply the treatment plan he clicks to "Save" button to save the treatment plan. Then Shared Decision Making Application constructs the related POWER2DM resources; POWER2DM_MedicationOrder and POWER2DM_ActionPlan and stores them to PDS. Also the application stores the prediction result (POWER2DM_PredictionResult resource) with this treatment plan to the PDS.

5.6 Authentication and Authorization Interaction Flows

See "D4.9 Privacy and Security Enablers" for the interaction flows regarding authentication and authorization.

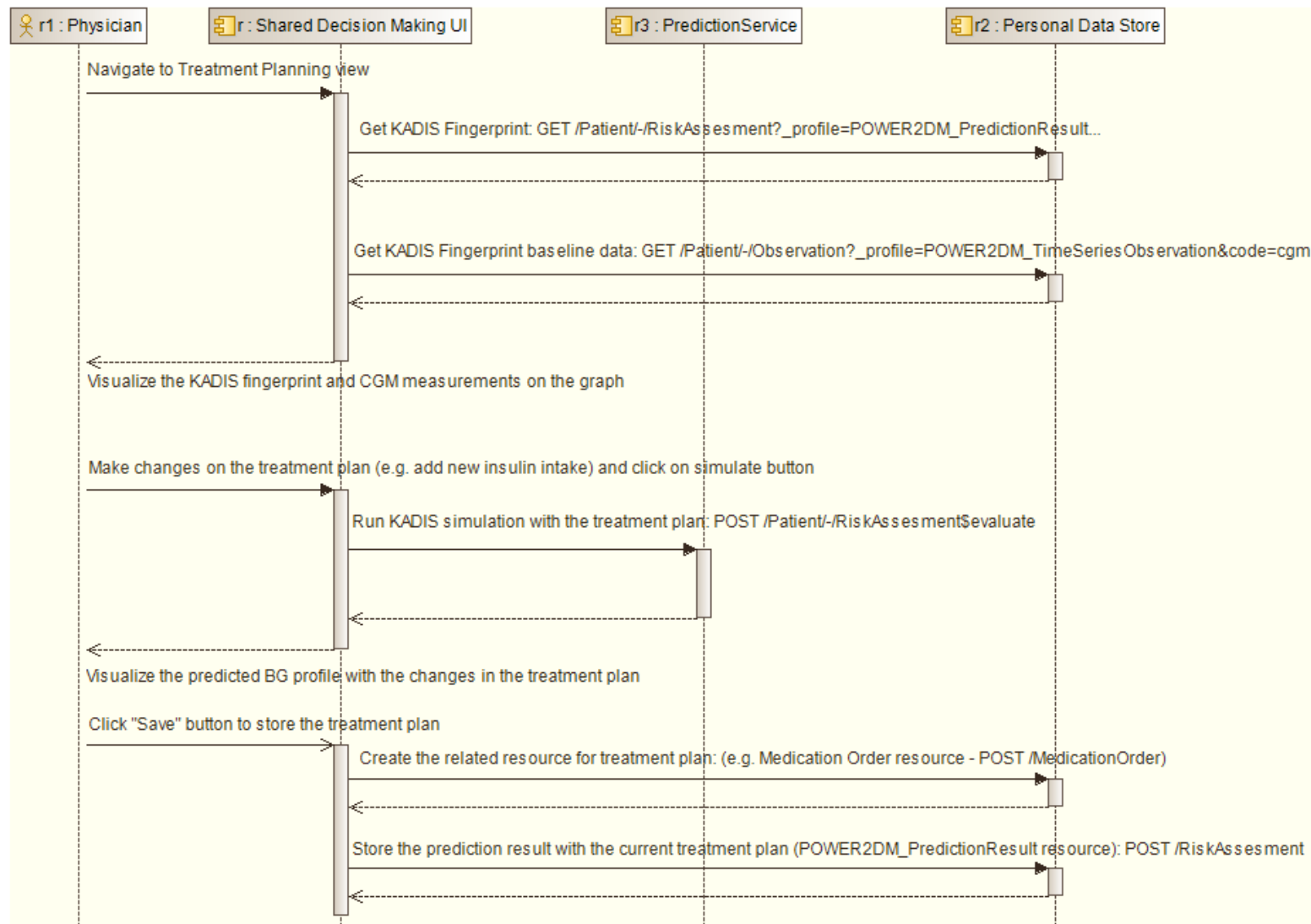


Figure 44 Treatment Planning with KADIS flow

6 DATA ARCHITECTURE

In the POWER2DM architecture, two components; the Personal Data Store (PDS) and Core Services (CS) are responsible for data storage and management. While PDS is managing the pseudonymised personal health data of patients, the CS manages the account, identity, privacy policies of POWER2DM users (including patients and healthcare professionals) and audit data.

6.1 Personal Health Data Model

HL7 FHIR standard is used as basis for the data model and data architecture for personal health data and its profiling approach is used to extend/restrict the base model according to POWER2DM data requirements. “D4.1 Personal Health Data Model and Service API” provide all the details of the model and the service that enables other POWER2DM components to access and manipulate the data. In other words, the model is used as the data exchange model among POWER2DM components. As the D4.1 provides all the details of the model, here only a summary is provided.

In FHIR standard, **smallest unit of exchangeable content between healthcare systems** is called “**Resource**”. FHIR standard defines several common resources so that different healthcare system implementers can use them to represent their data in their use cases i.e. Condition, Observation, RiskAssessment, MedicationOrder, etc. However, in order to ensure interoperability, the healthcare system developers generally restrict these resource definitions and tailor them according to their own requirements which is called profiling. In POWER2DM, we do the same and define several POWER2DM Resources which are based on base FHIR Resources.

POWER2DM Profile	Resource	Base FHIR Resource	Data represented
POWER2DM_Patient		Patient	Patient’s basic data; gender, age, ethnicity
POWER2DM_DiabetesAnamnesis		Composition	Provide diabetes anamnesis of a patient; duration and type of diabetes, other complications, comorbidities, basic observations like smoking status, etc
POWER2DM_Condition		Condition	Provide information about a single condition/diagnosis for a patient
POWER2DM_Problem		Condition	Provide information about a diabetes related problem of a patient; e.g. too much carbohydrate consumption, too little insulin usage, etc
POWER2DM_Barrier		Condition	Provide information about a barrier that cause the diabetes related problem e.g. fear needles, fear hypoglycaemia
POWER2DM_PersonalValue		Condition	Provide information about a personal motivation of a patient
POWER2DM_Goal		Goal	Provide information about a goal set for a patient in his/her care plan e.g. decrease your HbA1c to 6.5% in 3 months. This includes Treatment Goals set by physicians in shared decision making as well as Self-management Goals set by patient himself
POWER2DM_ActionPlan		ProcedureRequest	Provide information about a planned action for a patient; e.g. Run 3 times a week in the evenings. This includes Treatment Action Plans set by physicians as well as the Self-management Action Plans set by patient himself.
POWER2DM_MedicationOrder		MedicationOrder	Provide information about a proposed medication/insulin in the treatment plan for a patient. e.g. Take metformin every day one in the morning after breakfast, ..

POWER2DM_Appointment	Appointment	Provide information about an appointment arranged between a patient and a physician
POWER2DM_Encounter	Encounter	Provide information about an encounter that is in the past (realization of appointment)
POWER2DM_SimpleQuantityObservation	Observation	Used to represent any observation about a patient that can be provided as a quantity value. e.g. HbA1c, body weight, body height, blood glucose, waist circumference, Stress VAS, Sleep Quality VAS, Mood VAS, etc
POWER2DM_SimpleCodedObservation	Observation	Used to represent any observation about a patient that can be provided a coded value. e.g. Smoking status, complaint
POWER2DM_TimeSeriesObservation	Observation	Used to represent any observation about a patient that includes time series measurements e.g. CGM measurements, Spire measurements, etc
POWER2DM_BloodPressure	Observation	Provide information for a blood pressure measurement of a patient
POWER2DM_TrackerHeartRateSummary	Observation	Provide one-day summary information for heart rate obtained from an activity tracker e.g. Fitbit
POWER2DM_LabResults	Observation	Provide a set of lab results for a patient; e.g. TC, HDL, LDL, triglycerides, albumin, creatinine, etc.
POWER2DM_DietaryIntakeLog	Observation	Provide information about a dietary intake (a meal) of a patient. e.g. bread unit intake, carb intake, calorie intake, etc
POWER2DM_PhysicalActivityLog	Observation	Provide information about a physical activity performed by a patient. e.g. duration, intensity, distance.
POWER2DM_ActivityTrackerDailySummary	Observation	Provide one-day summary information for physical activity obtained from activity tracker e.g. Fitbit
POWER2DM_SleepTrackerSleepSummary	Observation	Provide summary information about a sleep session of a patient. e.g. Fitbit, iHealth
POWER2DM_Acknowledgement	Observation	Provide information about the realization of planned action. e.g. time when action performed, etc
POWER2DM_QuestionnaireResultScore	Observation	Provide the result score of a questionnaire applied to a patient at some specific time. e.g. Depression Assesment (PHQ-9) score
POWER2DM_RiskAssesment	RiskAssesment	Provide the result of risk assessment done for a patient. e.g. UKPDS risk scores; CVD risk, Fatal CVD risk, etc,
POWER2DM_PredictionResult	RiskAssesment	Provide the result of KADIS or MT2D-Marvel based prediction/simulation done for a patient
POWER2DM_QuestionnaireResult	Questionnaire Result	Provide the responses of patient for an applied questionnaire at some specific time
POWER2DM_MedicationAdministration	Medication Administration	Provide the log of a medication/insulin intake for a patient. e.g. time, dosage and type of insulin taken
POWER2DM_AppliedIntervention	Procedure	Provide information about an intervention made by POWER2DM system or a care provider to a patient at a specific time.

Apart from the above personal health records, the following resource profiles are defined for supplementary records that can be referred from the main records;

POWER2DM Profile	Resource	Base FHIR Resource	Data represented
POWER2DM_Practitioner	Practitioner		Basic data of a Practitioner (name, role, etc) that take role in POWER2DM Care program

POWER2DM_Organization	Organization	Basic data of an Organization (name) that take role in POWER2DM Care Program
POWER2DM_Device	Device	Provide information of a medical device or software that is used in POWER2DM Care Program
POWER2DM_Questionnaire	Questionnaire	Provide the content of a questionnaire used in POWER2DM in a machine processable format

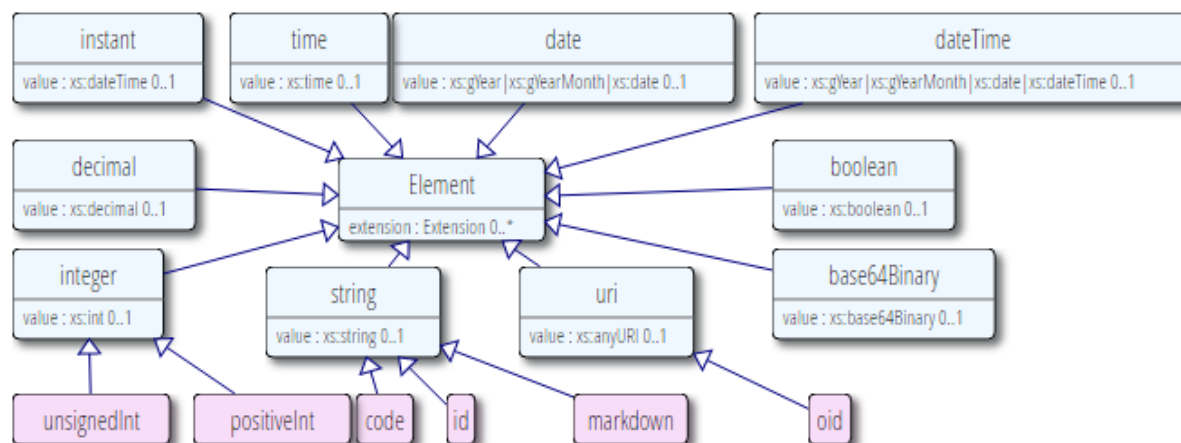


Figure 45 FHIR Primitive Data Types

All these resource profiles use the FHIR data types for representing their data elements. The Figure 45 shows the primitive data types defined in FHIR. Similarly, Figure 46 shows the complex data types defined in FHIR.

6.2 Model for User Identity, Settings, Policies and Audit Data

The POWER2DM Core Service manages the data needed for the administration processes of POWER2DM. “D4.9 Privacy and Security Enablers” provides detailed information regarding the processes and the data. In this section, we only provide a very brief summary (See Table 7).

Table 7 Summary of data managed by Core services

User Identity Data	Includes all identity information that are stored for POWER2DM users. e.g. name, username, password, profile picture, email, time zone info, locale, user type (i.e. patient, healthcare professional), etc as well as the assigned identifiers for the users. (Note: The OpenIDConnect standard is used for the modelling and managing of such information)
User Settings	Setting for users for POWER2DM Applications
Privacy Policies	Patient managed privacy policies (consents) including the access control rules, permissions for 3 rd party application integrations, role assignments of other users. (Note: OASIS XACML standard is used to model the privacy policies)
Audit	Audit data for each operation, data disclosure, exchange performed in POWER2DM system. (Note: FHIR AuditEvent resource is used to model the audit data.)

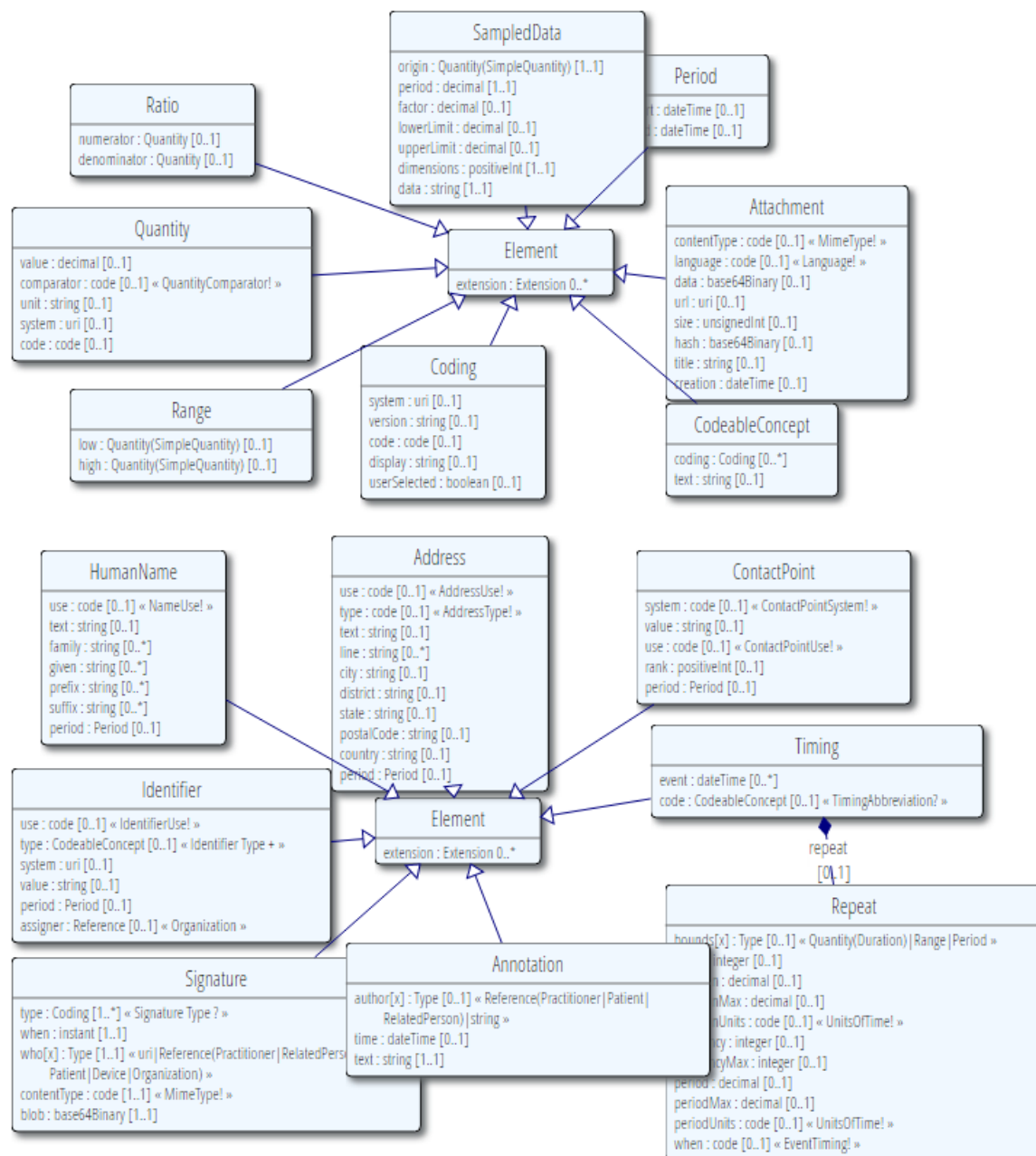


Figure 46 FHIR Complex Data Types

7 DEPLOYMENT INFRASTRUCTURE

POWER2DM Architecture allows different deployment topologies as it follows micro-service architecture. This allows us to be flexible to meet with the requirements arising from different perspectives; requirements due to privacy, security and ethical issues, specific requirements of pilot sites and their regulations, performance constraints, constraints from the existing software to be used etc. As some of these issues are not clear yet, we cannot give a final deployment infrastructure here. Instead, we provide an example deployment setup that can be possible which is illustrated in Figure 47.

In this setup, in order to meet the privacy requirements, Personal Data Store which will be the storage for pseudonymized personal health records, and the Core Services which will store the user identity information will be deployed separately.

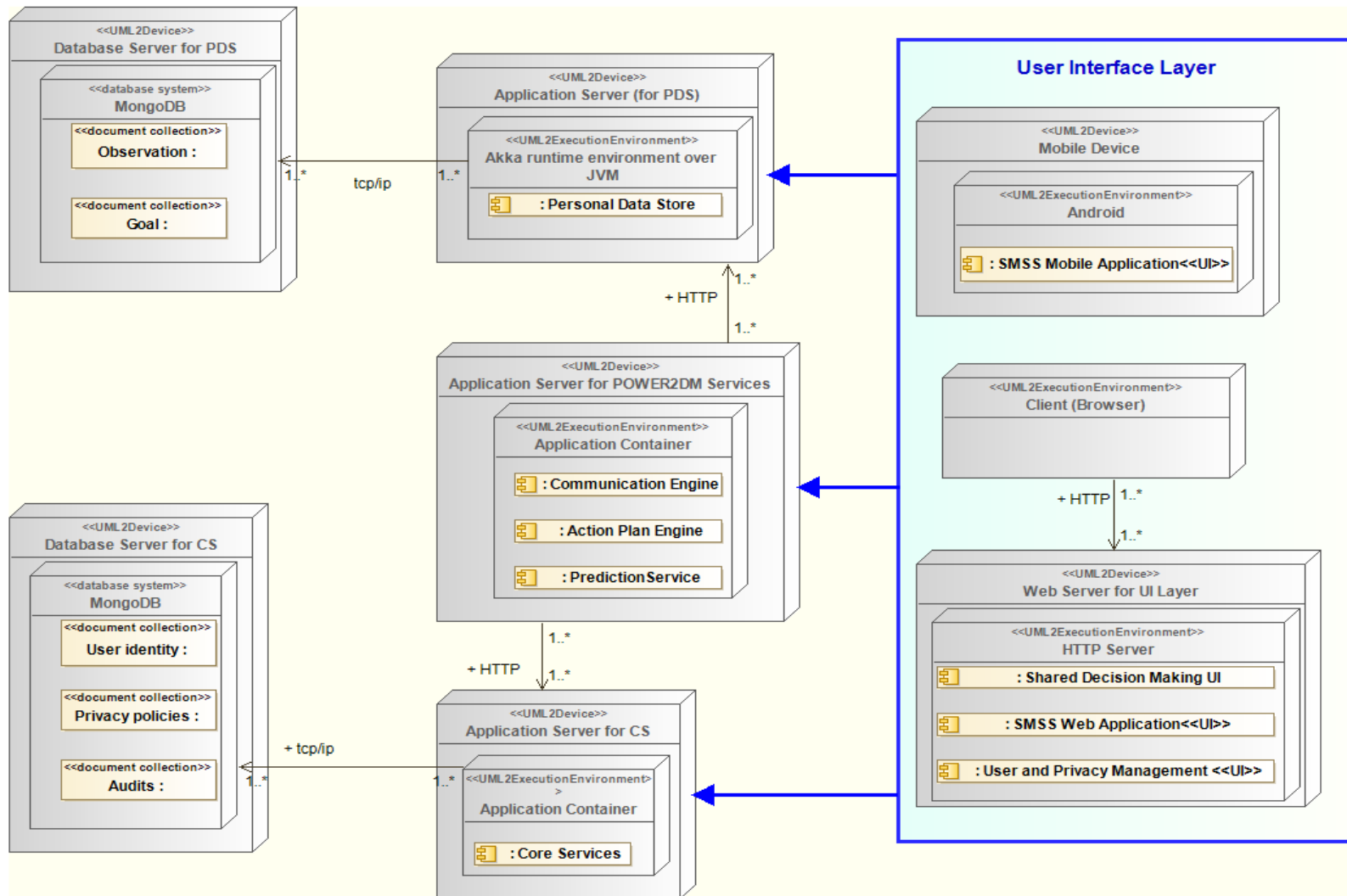


Figure 47 An example deployment setup for POWER2DM

In the setup, the deployment of PDS requires two clusters; Application Servers for the services provided by PDS and Database Servers to store the personal health records. Similarly, the Core Services requires two clusters; one for the database storage and other for the services provided by the component.

Other POWER2DM services, the Communication Engine, Action Plan Engine and Prediction Service are deployed in a separate Application Server in this example in the same Application container (a shared JVM). Similarly, the artifacts for the all the Web UIs in POWER2DM; Shared Decision Making UI, SMSS Web Application and User and Privacy Management UI are deployed on a Web Server. The SMSS Mobile Application will be deployed to the mobile devices of patients with Android execution environments.

!!! Open Issue: For POWER2DM Evaluation Campaign (pilot application), the aim is to deploy the POWER2DM to a secure cloud and serve the system to each pilot site (in Germany, Netherlands and Spain) from this cloud. However, this plan may change due to decisions that will be taken by the ethical boards due to legal and ethical regulations. In any case, the POWER2DM Architecture can be adapted to these requirements as each main component can be deployed as a standalone service.