Small or medium-scale focused research project (STREP)

ICT SME-DCA Call 2013
FP7-ICT-2013-SME-DCA

Data Publishing through the Cloud: A Data- and Platform-as-a-Service Approach to Efficient Open Data Publication and Consumption

DaPaaS

Deliverable D5.2

Use case implementation, v1

Date:	31.10.2014
Author(s):	Ivan Berlocher (Saltlux), Seonho Kim (Saltlux), Tony Lee (Saltlux)
Dissemination level:	PU
WP:	WP5
Version:	1.0

Copyright © DaPaaS Consortium 2013-2015

Document metadata

Quality assurors and contributors

Quality assuror(s)	Amanda Smith (ODI) and Brian Elvesæter (SINTEF)
Contributor(s)	DaPaaS Consortium

Version history

Version	Date	Description
0.1	30.09.2014	Initial outline and Table of Contents (TOC)
0.2	27.10.2014	Ready for internal peer review
0.3	30.10.2014	Integrated version with suggested edits from the internal peer review
0.4	31.10.2014	Revision addressing reviewers' comments
1.0	31.10.2014	Final formatting and layout

Executive Summary

The main goal of the DaPaaS project is to provide an integrated Data-as-a-Service (DaaS) and Platform-as-a-Service (PaaS) environment, together with associated services, for open data, where 3rd parties can publish and host both datasets and data-driven applications that are accessed by end user data consumers in a cross-platform manner.

This report represents Deliverable D5.2 "Use case implementation, v1" of the DaPaaS project. Based on the outcome of D1.2, D2.2 and D4.1 this deliverable provides the details of the implementation of the DaPaaS use case.

As outlined in the previous version of this document - Deliverable D5.1 "Use case definition and requirements analysis", the use case implements PLUQI (Personalized and Localized Urban Quality Index). PLUQI is a customizable index model that can be used to represent and visualize the level of well-being and sustainability for given cities based on individual preferences.

The PLUQI prototype has been implemented as a web application using the datasets and APIs from the DaPaaS platform. This implementation illustrates the semantic integration of heterogeneous open data from the Korean Statistical Information Service (KOSIS)[1] and data from social sensors linked through the CITI-SENSE[2] platform.

[1] http://kosis.kr/

[2] http://www.citi-sense.eu/

Table of Contents

List of Acronyms

API	Application Programming Interface
CSV	Comma Separated Values (format)
DaaS	Data-as-a-Service
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation (format)
PaaS	Platform-as-a-Service
PLUQI	Personalized and Localized Urban Quality Index
REST	Representational state transfer
RDF	Resource Description Framework
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SPARQL	SPARQL Protocol and RDF Query Language
SSH	Secure Shell
UML	Unified Modeling Language
XML	eXtensible Markup Language

List of Figures

Copyright © DaPaaS Consortium 2013-2015

List of Tables

1 Introduction

This report represents supporting documentation for the prototype developed for Deliverable D5.2 “Use case implementation, v1”. The goal of this deliverable is to provide:

- An initial implementation of the use case which will make use of the DaPaaS platform released at the end of the first year.

The rest of this report is structured as follows:

- Section 2 describes the development of the PLUQI use case prototype.
- Section 3 outlines future work for the second version of the PLUQI use case.
- Appendix A summarises how the main PLUQI functional requirements are addressed.

2 PLUQI Use Case Prototype

2.1 Datasets

As stated in Deliverable D5.1, PLUQI (Personalized Local Urban Quality Index) is defined as a customizable index model that shows the level of well-being and sustainability for cities, based on individual preferences.

Sustainability is a notion that is too macro-level and multi-faceted to be measured by any single metric. Several domains (or dimensions) and measurable and available metrics have to be defined and then aggregated into a single summary indicator.

The well-being index model has been benchmarked from several models (for details, see Deliverable D5.1):

- OECD Better Life Index
- Gallup-Healthways Well-Being Index (WBI)
- The Urban Sustainability Index
- UK Sustainable Cities Index
- London Ward Well-Being Scores
- UK experimental subjective well-being estimates

For this use case, we define the model in the following way:

1. Environmental needs and efficiency: green space, water and electricity consumption, air quality.
2. Level of opportunity: job, education, re-education, economic dynamic.
3. Cultural satisfaction: number of theatres, museums, art centres, etc.
4. Daily life satisfaction: weather, transportation, community, living density, shopping space, entertainment venues.
5. Safety and security: number of police station, crime per capita, fire station.
6. Healthcare level: numbers of doctors, hospitals, sport clubs.

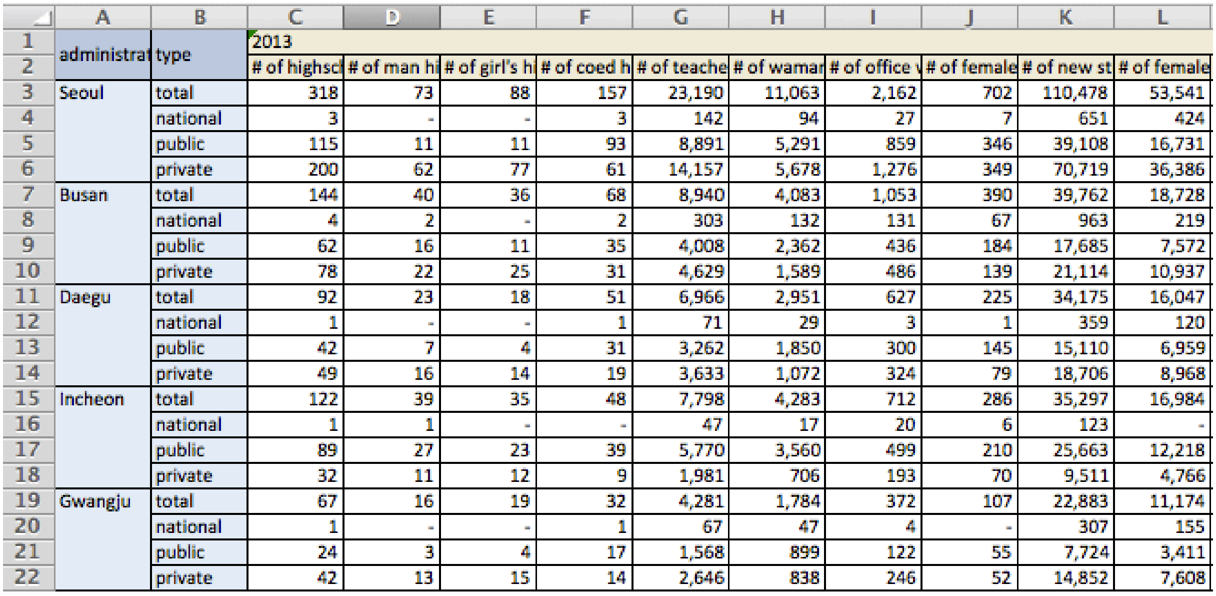
For the PLUQI use case implementation, a survey of available datasets has been undertaken for South Korea.

The Korean Statistical Information Service (KOSIS)[3] is providing numerous datasets in Excel format. For the implementation of the prototype, each dataset used in PLUQI has been chosen based on the following rules:

- Data should have the same granularity of all locations (Korean Cities) and
- Data should be as recent as possible

The datasets (in row format) are transformed into RDF so that the data can be accessible in a uniform manner - through SPARQL queries. For PLUQI, a simple ontology has been designed for representing Quality Index from various sources (e.g., open data, open API and social data) and various formats (e.g., CSV, Excel, JSON and RDF).

[3] http://kosis.kr



	A	B	C	D	E	F	G	H	I	J	K	L
1	administrat	type	2013									
2			# of highsc	# of man hi	# of girl's hi	# of coed h	# of teache	# of wamar	# of office v	# of female	# of new st	# of female
3	Seoul	total	318	73	88	157	23,190	11,063	2,162	702	110,478	53,541
4		national	3	-	-	3	142	94	27	7	651	424
5		public	115	11	11	93	8,891	5,291	859	346	39,108	16,731
6		private	200	62	77	61	14,157	5,678	1,276	349	70,719	36,386
7	Busan	total	144	40	36	68	8,940	4,083	1,053	390	39,762	18,728
8		national	4	2	-	2	303	132	131	67	963	219
9		public	62	16	11	35	4,008	2,362	436	184	17,685	7,572
10		private	78	22	25	31	4,629	1,589	486	139	21,114	10,937
11	Daegu	total	92	23	18	51	6,966	2,951	627	225	34,175	16,047
12		national	1	-	-	1	71	29	3	1	359	120
13		public	42	7	4	31	3,262	1,850	300	145	15,110	6,959
14		private	49	16	14	19	3,633	1,072	324	79	18,706	8,968
15	Incheon	total	122	39	35	48	7,798	4,283	712	286	35,297	16,984
16		national	1	1	-	-	47	17	20	6	123	-
17		public	89	27	23	39	5,770	3,560	499	210	25,663	12,218
18		private	32	11	12	9	1,981	706	193	70	9,511	4,766
19	Gwangju	total	67	16	19	32	4,281	1,784	372	107	22,883	11,174
20		national	1	-	-	1	67	47	4	-	307	155
21		public	24	3	4	17	1,568	899	122	55	7,724	3,411
22		private	42	13	15	14	2,646	838	246	52	14,852	7,608

Figure 1: An example of open data used in PLUQI – High schools in Korea

Other sources of data come from social sensors through the CITI-SENSE[4] platform. A Social Sensor is a software agent that provides observations about an environment via communicating to other agents. The role is to understand citizen's experience about their environment. By analyzing a stream of messages from a citizen, an agent can detect citizen's intention and expectation. Then, by using ontologies, data can be linked semantically to other open data, geo-spatial information and other physical or social sensors data. A social sensor provides good integrated indicators about various aspects of the environment.

Social sensors are defined for getting data from 7 cities and 9 regions: Seoul, Busan, Daegu, Incheon, Gwangju, Daejeon, Ulsan, Gyeonggi-do, Gangwon-do, Chungcheongbuk-do, Chungcheongnam-do, Jeollabuk-do, Jeollanam-do, Gyeongsangbuk-do, Gyeongsangnam-do, Jeju-do.

2.2 Ontology Model

The PLUQI ontology schema is depicted in Figure 2. It can represent sub-indices and its values with location and date-time information. It reuses other standard ontology models such as *WGS84* and *Korean Administrative Divisions* ontology[5].

[4] http://www.citi-sense.eu/

[5] http://lod.seul.go.kr

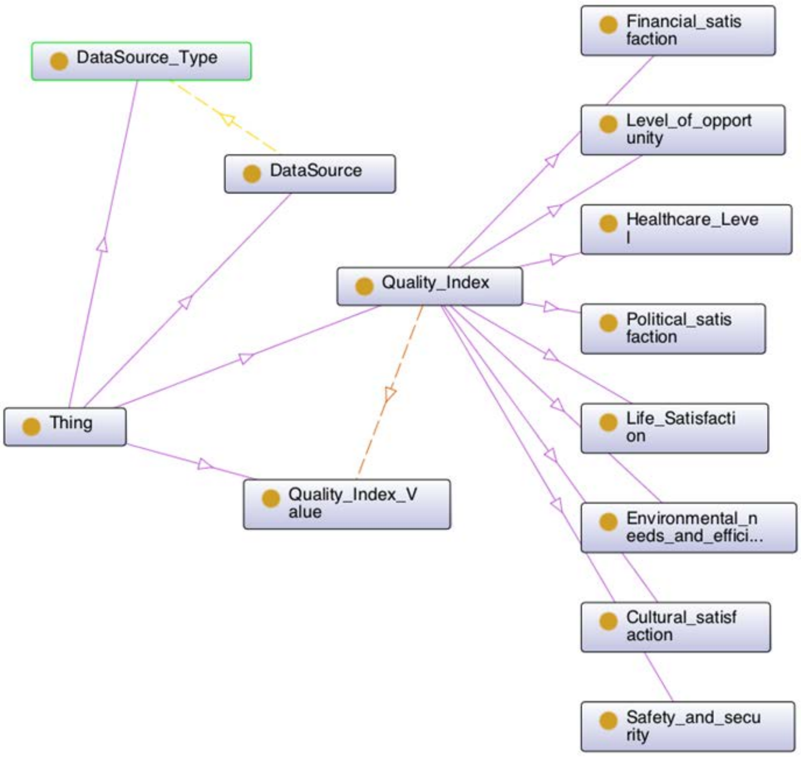


Figure 2: Class model from PLUQI use case

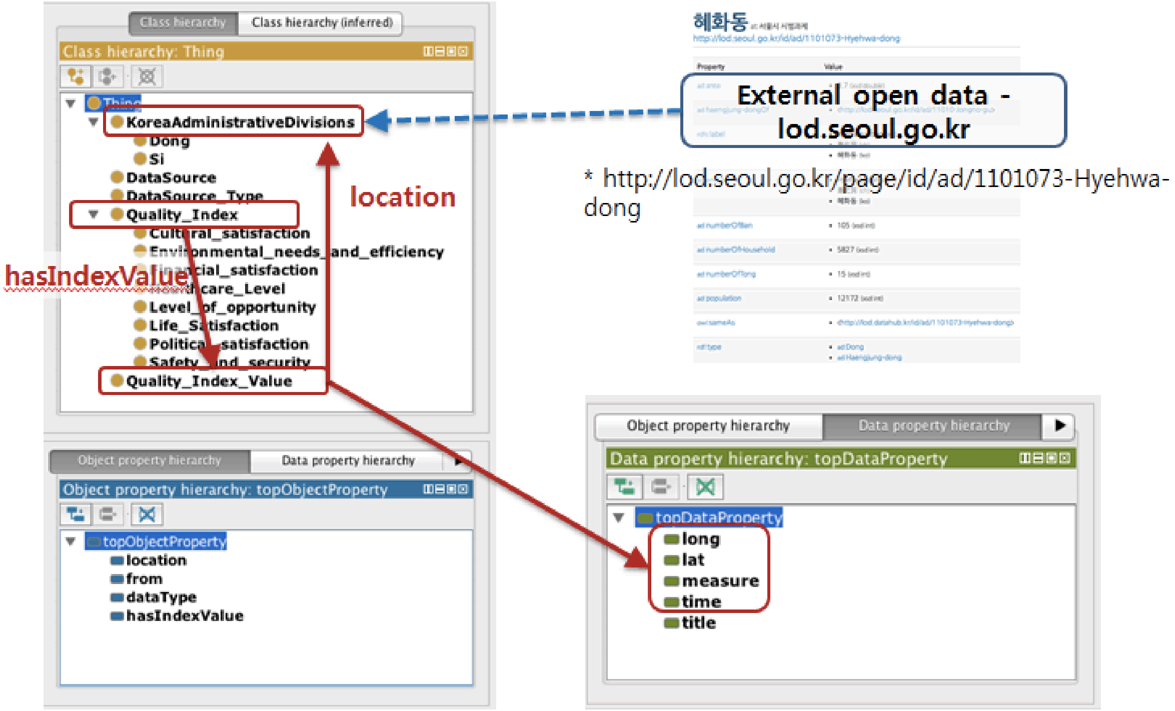
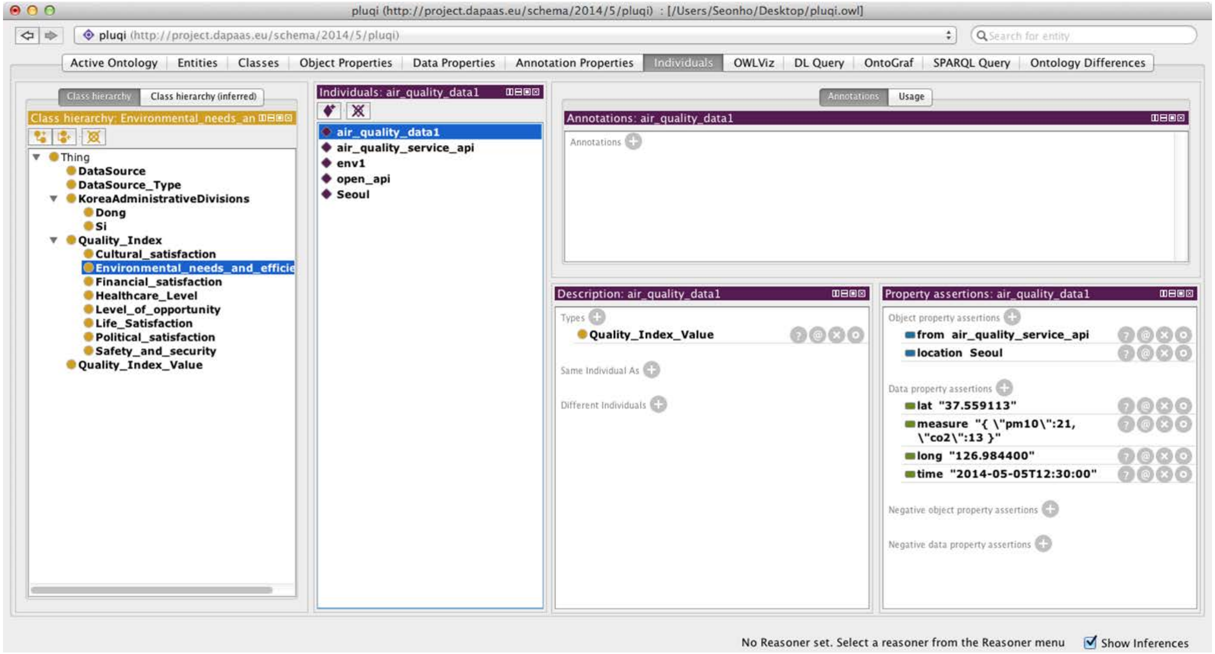


Figure 3: Properties of classes from PLUQI use case



pluqi (http://project.dapaas.eu/schema/2014/5/pluqi) : [/Users/Seonho/Desktop/pluqi.owl]

pluqi (http://project.dapaas.eu/schema/2014/5/pluqi)

Class hierarchy: Environmental_needs_an

- Thing
  - DataSource
  - DataSource_Type
  - KoreaAdministrativeDivisions
    - Dong
    - Si
  - Quality_Index
    - Cultural_satisfaction
    - Environmental_needs_and_efficie
    - Financial_satisfaction
    - Healthcare_Level
    - Level_of_opportunity
    - Life_Satisfaction
    - Political_satisfaction
    - Safety_and_security
  - Quality_Index_Value

Individuals: air_quality_data1

- air_quality_data1
- air_quality_service_api
- env1
- open_api
- Seoul

Annotations: air_quality_data1

Annotations

Description: air_quality_data1

Types
Quality_Index_Value

Same Individual As

Different Individuals

Property assertions: air_quality_data1

Object property assertions
from air_quality_service_api
location Seoul

Data property assertions
lat "37.559113"
measure "{ \"pm10\":21, \"co2\":13 }"
long "126.984400"
time "2014-05-05T12:30:00"

Negative object property assertions

Negative data property assertions

No Reasoner set. Select a reasoner from the Reasoner menu ☑ Show Inferences

Figure 4: Instance example of PLUQI use case

Following the schema, we can describe the information of high schools in Seoul as an instance of Level of opportunity, in Turtle format as following:

```
:v1 rdf:type :Value ,
    owl:NamedIndividual ;
    :measure 151 ;
    :time "2011-00-00T00:00:00Z"^^xsd:dateTime ;
    rdfs:comment "type:total"^^xsd:string ;
    :hasValue <http://project.dapaas.eu/schema/2014/5/pluqi/2011-2013_highschools> ;
    :location :Seoul .

:number_of_coed_high_schools rdf:type :Level_of_opportunity ,
    owl:NamedIndividual ;
    :hasValue :v1 .
```

Figure 5: Instance example serialized in Turtle format

To assist and simplify the RDFization process, the Grafter[6] tool provided by the Work Package 4 has been used. Future versions of this tool enhanced by new functionalities and a Graphical User Interface (GUI) will further facilitate this process.

2.3 Indices

In order to calculate PLUQI indices, several data representing different variables need to be gathered. After this data is collected, normalization, aggregation and weighting of the underlying variables are needed. Normalization is usually applied to single variables in order to make them comparable, i.e., transforming the various scales of variables into one unique scale. The normalized indicators are then

[6] http://grafter.org

aggregated using specific formulas (e.g. arithmetic mean). If one indicator is more “important” than another, the former is assigned a stronger weight than the latter within the aggregation procedure.

Scientifically sound methods for normalization (to make data ‘comparable’), weighting (to specify the ‘correct’ interrelationships), and aggregation (to get the ‘right’ functional relationship) are pre-requisites for the construction of meaningful indices.

In order to implement the PLUQI application we referred to the paper “An approach to measure sustainability: Comparative Evaluation of States of India”[7]. As the authors stated in their survey, datasets were often incomplete and at different scales. Sustainability indicators are often criticized because they aim to ‘measure the immeasurable’. Anyways, in this survey authors did provide a clear methodology to deal with datasets and we apply their proposed methodology in our implementation of PLUQI.

2.3.1 Z-Score

To make various parameters comparable a methodology has been adopted in the current implementation. Various indicator values computed for different cities in South Korea have been normalized using Z-score.

Z-score is a dimensionless quantity derived by subtracting the population mean from an individual raw score and then dividing the difference by the population standard deviation. This conversion process is called standardizing or normalizing. The standard score indicates how many standard deviations an observation is above or below the mean.

The standard score is: $Z = \frac{X - \mu}{\sigma}$

where:

- X is a raw score to be standardized
- σ is the standard deviation of the population
- μ is the mean of the population.

Z-score values computed for each of the parameters have been averaged for different themes (assuming equal weightage). Finally, a sustainability index has been formed using average of the averaged Z-score values of different themes.

Sustainability Index = Average (ZEnv, ZOpportunity, ZCultural, …)

Sustainability index values have been analysed and interpreted for various states of Korea to adjudge their performance.

Table 1: Themes, parameters and indicators selected for the current study

Domain	Theme	Indicators
Environmental needs and efficiency	green space	count
		area
	Air quality	Yellow Sand & PM10 (social sensor)
	Temperature	Social Sensor
	Humidity	Social Sensor
Level of opportunity	Education	Number of schools
		Number of professors
		Number of students
Cultural satisfaction	Cultural facilities	Number of theatres
		Number of museums
		Number of Art galleries
		Number of science museums
Daily life satisfaction	Traffic equipment	number of railroad stations

[7] http://umconference.um.edu.my/upload/163-1/Paper%2048.doc

		number of railroad line
		Length of road
		Area of road
		number of railroad stations
	Happiness	Social Sensor
Safety and security	Criminality	violent crimes
		theft
		narcotic crimes
		military related crimes
		traffic related crimes
	Anxiety	Social Sensor

2.4 Web App

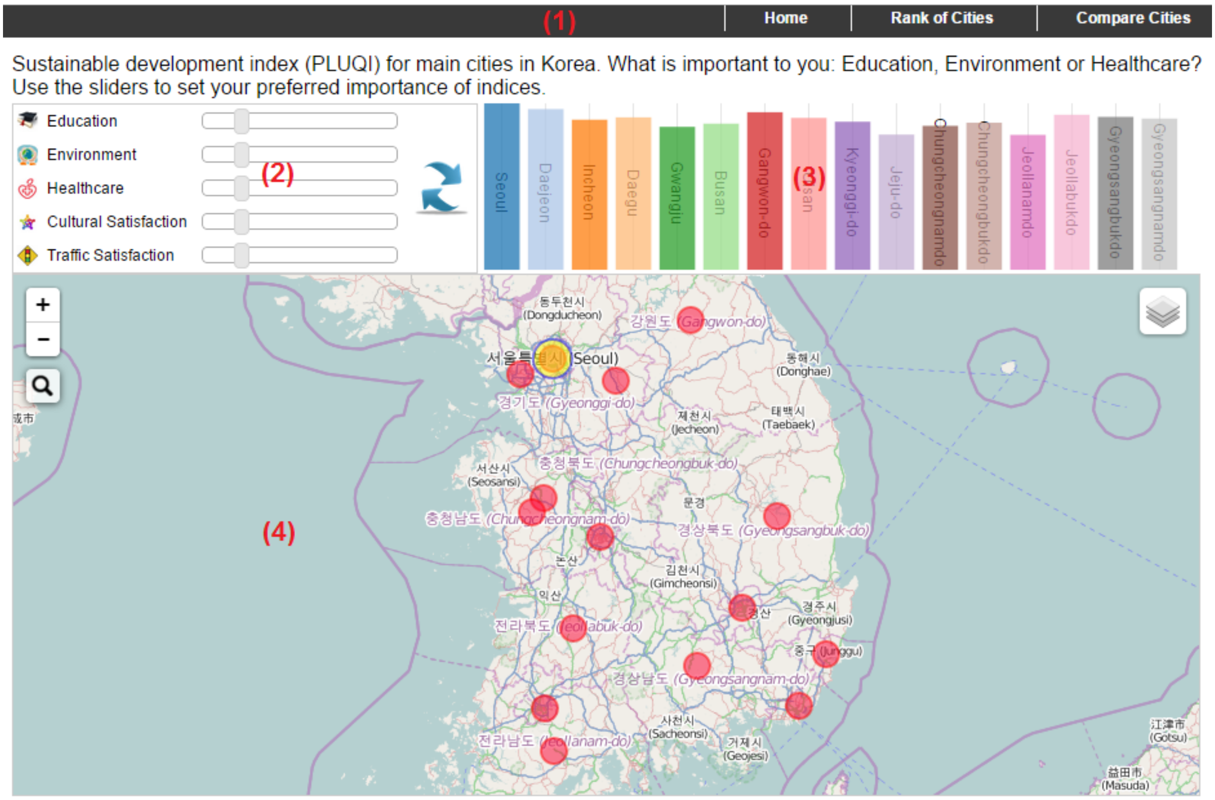
This section describes the web application prototype, which aims to demonstrate the strengths of DaPaaS in real applications. The web application visualizes data about PLUQI.

These indices are visualized in form of different graphs, e.g., map, bar chart and table, with intuitive user interactions. In this prototype, data of Korean cities has been used.

PLUQI indices are calculated from the indicators in the table above. Each index can have different weight/importance for different users. The web application consists of three pages (Home, Rank of cities and Compare cities) as described below.

2.4.1 Home Page

The main page is the starting point of the application from where users can see and interact with the PLUQI visualizations.



Sustainable development index (PLUQI) for main cities in Korea. What is important to you: Education, Environment or Healthcare? Use the sliders to set your preferred importance of indices.

Figure 6: PLUQI – Home page

The main page has following parts (marked as (1) to (4) in the above picture):

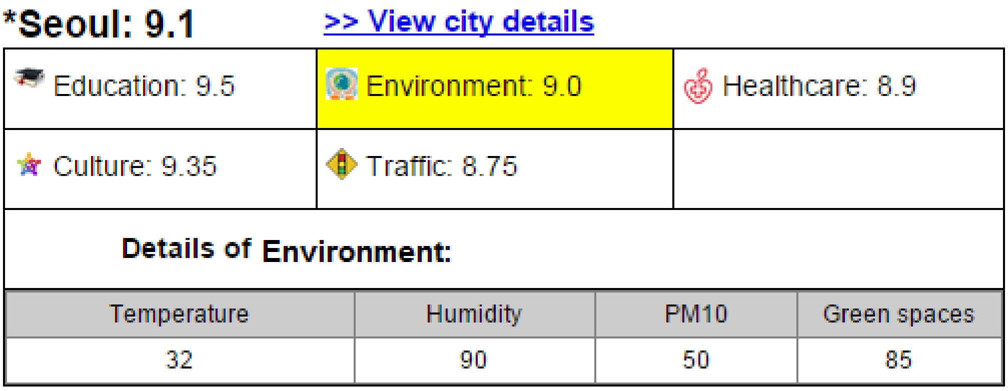
- Menu bar (1): on top of page, which allows user to navigate between pages.

- Index weight setting (2): For example, ‘environment’ may have higher importance to user A but less importance to user B. These sliders are used to set the individual’s preferred weight of each index. From here, PLUQI is calculated from the set of these weights and their corresponding index scores.
- PLUQI chart (3): visualizes the PLUQI of cities. A higher bar means there is a higher PLUQI.
- Map chart (4): visualizes cities’ PLUQI on map. The size of the circle represents PLUQI.

Page interactions:

- Changing sliders and clicking refresh button in (2) will redraw the PLUQI chart (3) with newly calculated values.
- By clicking a bar in PLUQI chart (3), the corresponding city will be highlighted on map (4).
- By clicking a city on map (4): the corresponding city in PLUQI chart (3) is highlighted and detail indices of the city will be shown below the map.

The city detail indices look as follows:



*Seoul: 9.1 >> View city details

Education: 9.5	Environment: 9.0	Healthcare: 8.9
Culture: 9.35	Traffic: 8.75	

Details of Environment:

Temperature	Humidity	PM10	Green spaces
32	90	50	85

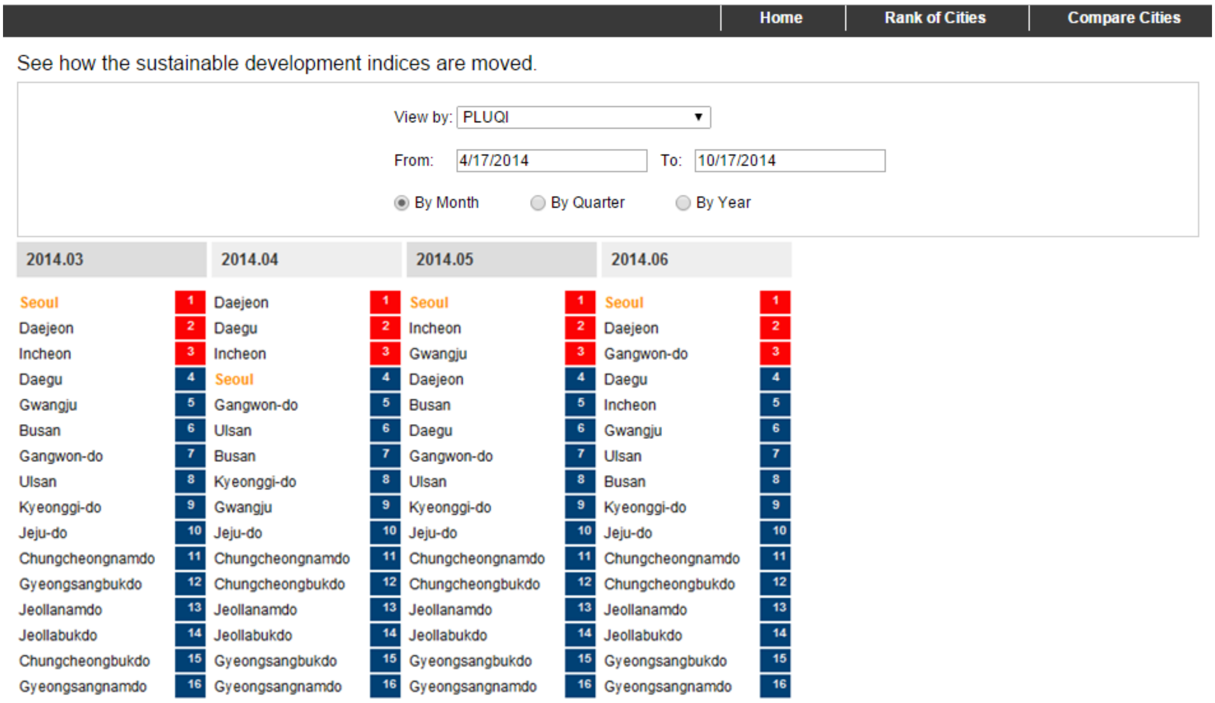
Figure 7: City detail indices for Seoul

The city detail view can be interacted as follows:

- Clicking an index will detail data of the index.
- Clicking ‘view city details’ will bring the city detail pop-up.

2.4.2 Rank of Cities Page

This page help user see how the sustainable development indices changed through time.



See how the sustainable development indices are moved.

View by: PLUQI ▾

From: 4/17/2014 To: 10/17/2014

● By Month ○ By Quarter ○ By Year

2014.03		2014.04		2014.05		2014.06	
Seoul	1	Daejeon	1	Seoul	1	Seoul	1
Daejeon	2	Daegu	2	Incheon	2	Daejeon	2
Incheon	3	Incheon	3	Gwangju	3	Gangwon-do	3
Daegu	4	Seoul	4	Daejeon	4	Daegu	4
Gwangju	5	Gangwon-do	5	Busan	5	Incheon	5
Busan	6	Ulsan	6	Daegu	6	Gwangju	6
Gangwon-do	7	Busan	7	Gangwon-do	7	Ulsan	7
Ulsan	8	Kyeonggi-do	8	Ulsan	8	Busan	8
Kyeonggi-do	9	Gwangju	9	Kyeonggi-do	9	Kyeonggi-do	9
Jeju-do	10	Jeju-do	10	Jeju-do	10	Jeju-do	10
Chungcheongnamdo	11	Chungcheongnamdo	11	Chungcheongnamdo	11	Chungcheongnamdo	11
Gyeongsangbukdo	12	Chungcheongbukdo	12	Chungcheongbukdo	12	Chungcheongbukdo	12
Jeollanamdo	13	Jeollanamdo	13	Jeollanamdo	13	Jeollanamdo	13
Jeollabukdo	14	Jeollabukdo	14	Jeollabukdo	14	Jeollabukdo	14
Chungcheongbukdo	15	Gyeongsangbukdo	15	Gyeongsangbukdo	15	Gyeongsangbukdo	15
Gyeongsangnamdo	16	Gyeongsangnamdo	16	Gyeongsangnamdo	16	Gyeongsangnamdo	16

Figure 8: PLUQI – Rank of cities page

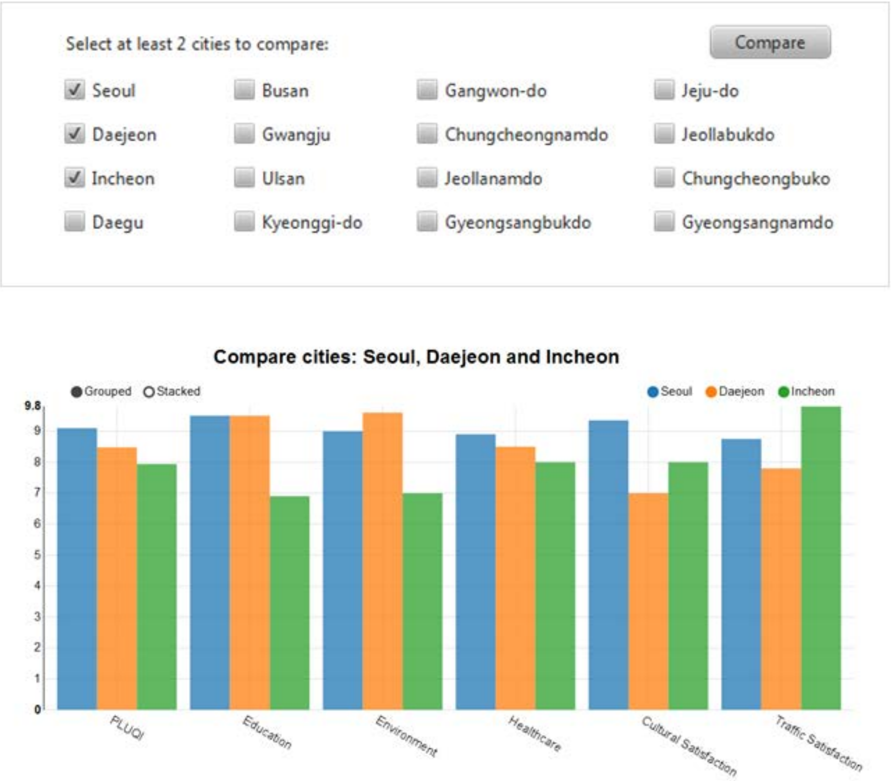
The user can select the preferred index and time period to how cities are ranked based on the selected conditions.

2.4.3 Compare Cities Page

This page allows user to compare cities. Indices of the compared cities are visualized in a bar chart.

Each group of bars presents an index of cities side by side, and a higher bar means there is a higher score.

Copyright © DaPaaS Consortium 2013-2015



Select at least 2 cities to compare:

☑ Seoul □ Busan □ Gangwon-do □ Jeju-do
☑ Daejeon □ Gwangju □ Chungcheongnamdo □ Jeollabukdo
☑ Incheon □ Ulsan □ Jeollanamdo □ Chungcheongbuko
□ Daegu □ Kyeonggi-do □ Gyeongsangbukdo □ Gyeongsangnamdo

Figure 9: PLUQI – Compare cities page

2.5 Application API

In order to fetch data from the DaPaaS Platform and pass it to the visualization component, an intermediate API has been defined as follows:

2.5.1 Get City Data

HTTP method

GET CityData

Description

Get list of cities and their information.

Parameters

weights	*Set of 5 weights for 5 sub-indices (education, environment, healthcare, cultural_satisfaction, traffic_satisfaction).*

Response

Cities: An array contains cities data. If there is an empty slot, the slot will be set as empty object.

Sample Response:

```
{
  "response": [
    {
      "name": "Seoul",
      "PLUQI": 9,
      "Indices": {
            "education": 8,
            "environment": 9,
            "healthcare": 5,
            "cultural_satisfaction": 7,
            "traffic_satisfaction": 6
        }
    },
    {
        "name": "Busan",
        "PLUQI": 9,
        "Indices": {
            "education": 6,
            "environment": 7,
            "healthcare": 5,
            "cultural_satisfaction": 9,
            "traffic_satisfaction": 6
        }
    },
    … (more city objects are here)
  ]
}
```

2.5.2 Get Sub Index Details

HTTP method

GET SubIndexDetailData (city, [sub-indexname])

Description

Get the details about a city's sun-indexes, with evidences data.

Sample Response

(each sub index, will have different JSON object format)

For Education:

```
 {
   "response": {
          "name": "education ",
          "weight": 9,      "details": {
                  "number_of_schools": 538,
                  "number_of_graduates": 89500
     }
```

For Environment:

```
 {
   "response": {
          "name": "Environment",
          "weight": 9,
    "details": {
                 "temperature": 32,
                 "humidity": 90,
                 "PM10": 50,
                 "green_spaces": 85
           }
      }
}
```

Copyright © DaPaaS Consortium 2013-2015

2.5.3 Get City Rank

HTTP method

GET CityRank (time interval)

Description

Get city rank data.

Sample Response

```
{
    "response": [
       {
          "cities": [
             {
                "node": "Seoul"
             },
             {
                "node": "Daejeon"
             },
             {
                "node": "Incheon"

              }

       }

     ]

}
```

2.5.4 Compare Cities

HTTP method

GET Sub-Indices Rank Data

Description

Get cities information to compare them.

Sample Response

```
{
  "response": [
        {
           "city": "Seoul",
           "values": [
              {
                 "index": "PLUQI",
                 “value": 10
              },
              {
                 “index”: "Education",
                 “value”: 5
              },
              {
                 “index”: "Environment",
                 “value”: 8
              },
              {
                 “index”: "Healthcare",
                 “value”: 7
              },
              {
```

```
            “index”: "Cultural Satisfaction",
            “value”: 9
          },
          {
            “index”: "Traffic Satisfaction",
            “value”: 4
          }
        ]
      },
      {
        "city": "Daejeon",
        "values": [
           {
            “index”: "PLUQI",
            “value”: 9
          },
          {
            “index”: "Education",
            “value”: 6
          },
          {
            “index”: "Environment",
            “value”: 9
          },
          {
            “index”: "Healthcare",
            “value”: 7
          },
          {
            “index”: "Cultural Satisfaction",
            “value”: 9
          },
          {
            “index”: "Traffic Satisfaction",
            “value”: 4
          }
        ]
      },
      {
        "city": "Daegu",
        "values": [
          {
            “index”: "PLUQI",
            “value”: 8
          },
          {
            “index”: "Education",
            “value”: 6
          },
          {
            “index”: "Environment",
            “value”: 5
          },
          {
            “index”: "Healthcare",
            “value”: 7
          },
          {
            “index”: "Cultural Satisfaction",
            “value”: 9
          },
          {
            “index”: "Traffic Satisfaction",
            “value”: 4
          }
        ]
```

Copyright © DaPaaS Consortium 2013-2015

```
        },
        … (more cities are here)
      ]

}
```

2.6 Open Source Libraries Used

The PLUQI application uses the following 3rd party libraries:

- **Leaflet**[8], which is a modern open-source JavaScript library for mobile-friendly interactive maps. This library is used to visualize data on the map in Main page.
- **NVD3**[9], which is an attempt to build re-usable charts and chart components for d3.js without taking away the power that d3.js. This Javascript library is used to draw charts.

2.7 Application Deployment

The application intended to run on Tomcat so PLUQI has been developed in JSP / Java and packaged in a war file. DaPaaS provides a Graphical User Interface (GUI) for uploading a war file packaging the application. Through this GUI, metadata are provided for managing and retrieving the application.

[8] http://leafletjs.com

[9] http://nvd3.org

3 Future Work

This first prototype illustrated the usage of functionalities provided by DaPaaS platform. The PLUQI use case did use services (APIs) for importing datasets, transforming and publishing them in RDF format, then querying them uniformly for creating an API for feeding the visualization part. Then the application was deployed on the DaPaaS platform. The application can be retrieved by search based on metadata provided during registration.

During data importing, it is clear that an easy interface for importing, transforming data is important for minimizing the cost of development, data quality checking and maximizing the reuse of Open Data. Datasets are often missing values, containing lot of spelling errors, incomplete etc. In order for the PLUQI use case to be more attractive, more data is planned to be integrated and additional features for social support should be taken in account.

As Open Data is currently at the beginning stage in South Korea, we could not find as many datasets as expected but we plan to apply the same process for others cities, especially for cities in Europe. Also, because DaPaaS is still on-going work, we could not use some of datasets expressed in JSON but with next versions of DaPaaS as more data formats will be supported we will integrate them in the next version. One future work planned is to provide also a mobile application.

4 Appendix A: Requirements Addressed by PLUQI

Deliverable D5.1[10] outlined a set of requirements for the use case. The tables below summarize how the main PLUQI (Personalized Local Urban Quality Index) functional requirements are addressed by the current version of the use case implementation.

Table 2: PLUQI functional requirements addressed

ID	Functionality	Description	Priority	Addressed
FU-01	User Management	System shall provide functionalities for managing user information (CRUD operations on users).	Core	Covered by the DaPaaS Platform requirements (IO-02): Addressed in Deliverable D1.2.
FU-02	User Management	System shall support authentication from other Services like Facebook, Google, Twitter, LinkedIn etc.	Optional	Covered by the DaPaaS Platform requirements (DP-10, AD-01, EU-01): Addressed in Deliverable D2.2.
FU-03	Import Dataset	System shall support various connectors/adapters for importing datasets of various types (textual, tabular or RDF data) and formats (e.g. txt, html, rdf, csv, xls).	Core	Covered by the DaPaaS Platform requirements (DP-01): Addressed in Deliverable D1.2 and D2.2.
FU-04	Import Dataset	System shall support import for single or multiple files.	Optional	Import of multiple files is to be addressed in the next version of the prototype.
FU-05	Import Dataset	System shall support import for a directory of files.	Optional	To be addressed in the next version of the prototype.
FU-06	Import Dataset	System should provide support to bulk upload in parallel.	Optional	To be addressed in the next version of the prototype.
FU-07	Import Dataset	System should provide support to resume upload in case of interruption (temporary network or system failure).	Optional	To be addressed in the next version of the prototype.
FU-08	Import Dataset	System shall support various connections to Open API Services like Twitter Streaming/Search API, RSS etc.	Core	To be addressed in the next version of the prototype.
FU-09	Import Dataset	System shall support various connections to Legacy Systems like RDMS/EDMS/CRM/Sear	Optional	Partially covered by the DaPaaS Platform requirements (DP-01): Addressed in Deliverable D1.2.

		ch Engine.		
FU-10	Import Dataset	System shall support connections to other already published data from other Open data platforms (C-KAN, D-KAN, Socrata).	Core	To be addressed in the next version of the prototype.
FU-11	Refresh Dataset	System shall be able to refresh/update data for sources of real-time datasets (like Twitter, Sensors data) or frequently changing datasets (RSS).	Core	Related to Citi-Sense Platform integration.
FU-12a	Dataset Aggregation	System shall provide functionalities for aggregating several datasets.	Core	To be addressed in the next version of the prototype.
FU-12b	Dataset Mashup	System shall provide functionalities to link to other datasets.	Core	Covered by the DaPaaS Platform requirements (DP-04): Addressed in Deliverable D1.2.
FU-13	Data Manipulation	System shall provide functionalities to clean/refine datasets.	Core	Covered by the DaPaaS Platform requirements (DP-05): Addressed in Deliverable D1.2.
FU-14	Data Storage	System shall support various storage repositories depending on properties of datasets (Triple Store, RDBMS, NoSQL, Search Engine).	Core	Covered by the DaPaaS Platform requirements (DP-02): Addressed in Deliverable D1.2.
FU-15	Dataset Management	System shall offer functionalities for managing metadata of datasets.	Core	Covered by the DaPaaS Platform requirements (DP-07): Addressed in Deliverable D1.2.
FU-16	Dataset Management	System shall offer functionalities for managing access (read/write permissions) depending on user permissions (individual, group, universal).	Optional	Covered by the DaPaaS Platform requirements (DP-07): Addressed in Deliverable D1.2.
FU-17	Export Dataset	System shall export to various formats (RSS/Atom, text, CSV, RDF, etc) and provide a URL for remote access.	Core	Covered by the DaPaaS Platform requirements (EU-05): Addressed in Deliverable D1.2.
FU-18	Sociality	System should provide functionalities for creating and managing communities of users.	Optional	To be addressed in the next version of the prototype.
FU-19	Sociality	System should allow user to follow and/or vote for datasets.	Optional	To be addressed in the next version of the prototype.

FU-20	Sociality	System should provide connectivity to other social services (Google+, Twitter, Facebook, E-mail) for sharing datasets links.	Optional	To be addressed in the next version of the prototype.
FU-21	Management	System should provide notifications (e.g. change in datasets).	Core	Covered by the DaPaaS Platform requirements (EU-03): Addressed in Deliverable D1.2 and D2.2
FU-22	Notification	System should provide notification in case of change in datasets.	Core	Covered by the DaPaaS Platform requirements (EU-03): Addressed in Deliverable D1.2 and D2.2
FU-23	Curation & Management	System should provide functionalities for adding/modifying data in datasets when allowed.	Core	Covered by the DaPaaS Platform requirements (DP-02): Addressed in Deliverable D1.2
FU-24	Management	System should provide statistics of usage/access of published datasets.	Core	Covered by the DaPaaS Platform requirements (DP-07): Addressed in Deliverable D1.2
FU-25a	Visualization	System should provide functionalities for viewing full datasets or previewing parts of datasets with adapted visualizations.	Core	Covered by the DaPaaS Platform requirements (EU-04): Addressed in forthcoming Deliverable D3.3.
FU-25b	Visualization	Visualization should provide support for tabular form, charts (line, plot, histograms etc.) for displaying 2D data, time series, plotting data on a map for geo-spatial data.	Core	Covered by the DaPaaS Platform requirements (EU-04): Addressed in forthcoming Deliverable D3.3.
FU-26	Search	System should provide federated search to repositories by providing adapted repository querying (Search Engine, RDF-Store, SQL etc.).	Core	Covered by the DaPaaS Platform requirements (DP-03, EU-02): Addressed in Deliverable D1.2.
FU-27	Discovering	System should provide filter results by tags/criteria (faceted search).	Optional	Covered by the DaPaaS Platform requirements (DP-03, EU-02): Addressed in Deliverable D1.2.
FU-28	API	System should provide RESTful Open API for all functionalities.	Core	Covered by the DaPaaS Platform requirements (DP-02, AD-01): Addressed in Deliverable D1.2 and D2.2.