|  |  |
| --- | --- |
| Module | rDQ Analyser |
| Sub-Module | Explorer |
| Author | D Twaddell |
| Date | 19th Aug 17 |
| Reviewer |  |
| Review Date |  |
| Review Status |  |
| Review Comments |  |

# Design Summary

## Design Log

Contents

[Design Summary 1](#_Toc490993596)

[Design Log 1](#_Toc490993597)

[Sub-Module Description 1](#_Toc490993598)

[External Packages Used 1](#_Toc490993599)

[Data Model 1](#_Toc490993600)

[Design Details 1](#_Toc490993601)

[Function – List Datasets 1](#_Toc490993602)

[Input parameters: 1](#_Toc490993603)

[Logic 1](#_Toc490993604)

[Outputs 2](#_Toc490993605)

[Function – New Table Profile 2](#_Toc490993606)

[Input parameters: 2](#_Toc490993607)

[Logic 2](#_Toc490993608)

[Outputs (see data model for details): 2](#_Toc490993609)

[Function – New Attribute Profile 2](#_Toc490993610)

[Input parameters: 2](#_Toc490993611)

[Logic 2](#_Toc490993612)

[Outputs (see data model for details): 2](#_Toc490993613)

[Function –Attribute Report 3](#_Toc490993614)

[Input parameters: 3](#_Toc490993615)

[Logic 3](#_Toc490993616)

[Outputs (see data model for details): 3](#_Toc490993617)

[Appendix – Module List 3](#_Toc490993618)

## Sub-Module Description

1. Allow user to connect with a data source, list available datasets, and then analyse selected datasets.
2. Store the analysis output, the table ‘profile’, for later reference. For example, the profile will be used by the Rules module to suggest appropriate data quality rules.

## External Packages Used

Look into using the DataExplorer package for this module

## Data Model

List of database structures used (refer to rDQ Data Model design document):

* **sysParameters** – stores information about parameters used in the system
* **rdqModule** -reference data, list of rDQ modules and sub-modules
* **Table:** **antTableProfile** – stores the results of a table profile
* **Table: anaAttributeProfile** – stores the results of a column profile

# Design Details

## Function – List Datasets

### Input parameters:

* conId – Identifies a data connection

### Logic

* Attempt connection, if error return error gracefully
* If connection successful, return outputs

### Outputs

* a list/vector containing the name, date, filetype of each dataset at the connection. Also contains the Date Last Profiled, the number of rows, and the number of columns from the most recent profile: **antTableProfile** table (select highest antDate).

## Function – New Table Profile

### Input parameters:

* conId – identifies the connection to be used
* antDatasetName – identies the dataset to be profiled

### Logic

* Attempt connection, if error return error gracefully
* If connection successful, analyse and return outputs

### Outputs (see data model for details):

* antId
* antDate
* antDatasetName
* antDatasetDescription
* antRows
* antCols

The outputs are automatically stored in the antTableProfile

## Function – New Attribute Profile

### Input parameters:

* conId – identifies the connection to be used
* antId – identies the dataset to be profiled

### Logic

* Attempt connection, if error return error gracefully
* If connection successful, establish and return outputs
* Create new anaAttribute record

### Outputs (see data model for details):

* anaId
* anaDate
* anaAttributeName
* anaDescription
* anaDataType
* anaDetailReport

The outputs are automatically stored in the antTableProfile

## Function –Attribute Report

### Input parameters:

* anaId – identifies the attribute profile to be used

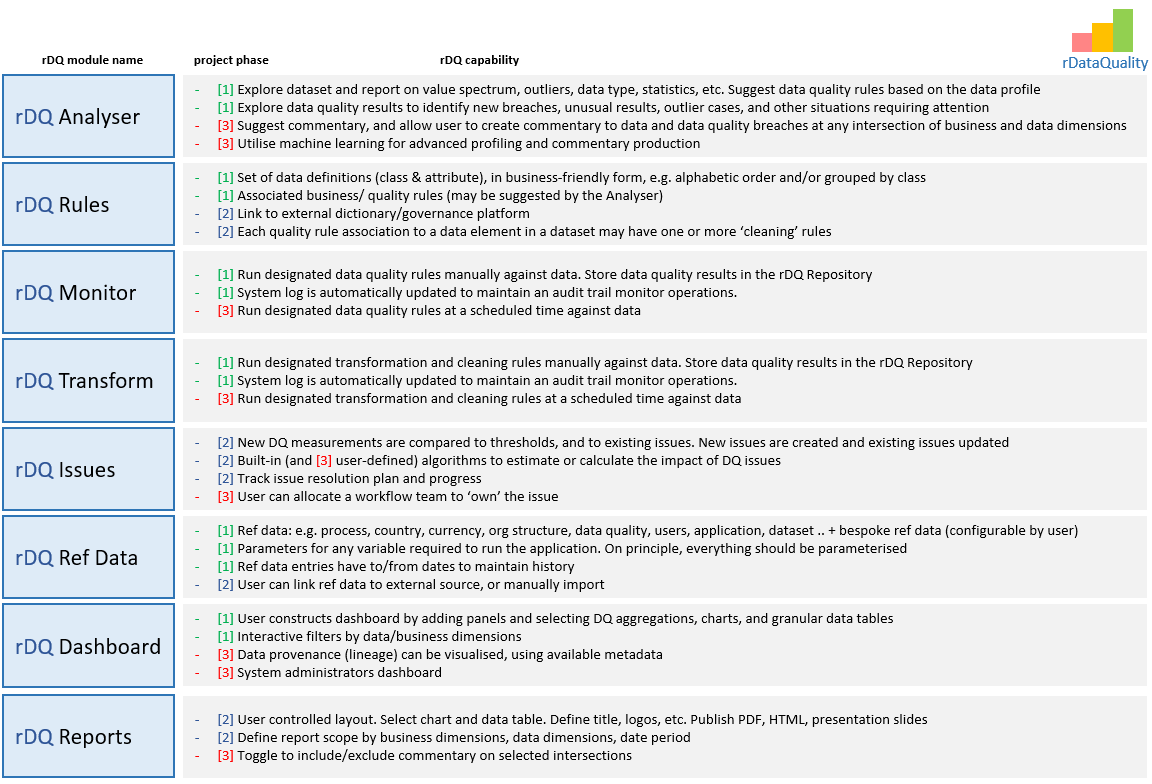
### Logic

* Pass the anaDetailedReport string as parameters to the report generator (DataExplorer?)
* If report generation successful, return the report. Otherwise gracefully return an error

### Outputs (see data model for details):

* Detailed report for the selected table attribute

# Appendix – Module List



In addition to these ‘public’ modules we may need some private modules to help organise work, e.g.:

* **rDQDataManagement** – common functions relating to data management, dealing with files and database operations. Define functions that are used in other modules. Define global variables
* **rDQUI** – implements the user interface, using Shiny