MarkLogic Data Workbench: Software Requirements Specification Version 1.0

Team Flash
Computer Science Department
California Polytechnic State University
San Luis Obispo, CA USA

October 9, 2018

Contents

Revision History **3**

Credits **3**

1 Introduction **5**
1.1 Purpose 5
1.2 Document Conventions 5

Conventions **5**
1.3 Intended Audience and Reading Suggestions 5
1.4 Project Scope 5
1.5 References 6

2 Overall Description **6**
2.1 Product Perspective 6
2.2 Product Features 6
2.3 User Personas 6
2.3.1 Data Analyst: Barry Allen 6
2.3.2 MarkLogic Sales Representative: Annette Caulfield 7
2.3.3 Health Care Business Analyst: Wally West 7
2.3.4 Business Analyst: Harrison Wells 7
2.3.5 Data Scientist Jojo Woleub 7
2.3.6 Business Analyst Jesse Quick 8
2.3.7 Climate Change Researcher Matt Sewall 8
2.4 User Classes and Characteristics 8
2.5 Operating Environment 9
2.6 Design and Implementation Constraints 9
2.7 User Documentation 9
2.8 Assumptions and Dependencies 9
2.9 Business Rules 9

3 Use Cases **9**
3.1 Use Case 1: Search Dataset 9
3.2 Use Case 2: Specify Custom Classifications 11
3.3 Use Case 3: View Data Lineage 13
3.4 Use Case 4: Modify Generated Categories 14
3.5 Use Case 5: Label Category as Sensitive 15
3.6 Use Case 6: Extract Data from CSV or JSON 16

4 System Features **17**
4.1 System Feature 1 17
4.1.1 Description and Priority 17
4.1.2 Stimulus/Response Sequences 17
4.1.3 Functional Requirements 17

5 External Interface Requirements **18**
5.1 User Interfaces 18
5.2 Software Interfaces 18
5.3 Communications Interfaces 18

6 Other Nonfunctional Requirements **19**
6.1 Performance Requirements 19
6.2 Security Requirements 19
6.3 Software Quality Attributes 19

7 Other Requirements **19**

A Glossary **19**

B Analysis Models **19**

C Issues List **19**

Credits

Name	Date	Role	Version
Bonita Golvan	October 9, 2018	Other Requirements	1.0
Diana Chiu	October 9, 2018	External Interface Requirements	1.0
Kent Tran	October 9, 2018	Overall Description	1.0
Tanner Villarete	October 9, 2018	User Persona, FR, NFR	1.0
Victoria Law	October 9, 2018	System Features	1.0
Joey Buelow	October 9, 2018	Introduction	1.0

Revision History

Name	Date	Reason for Changes	Version
Victoria Law	October 9, 2018	User Persona, Functional Requirement 1, Non-functional Requirement 1, Use Case 3.1	1.0
Joey Buelow	October 9, 2018	Introduction, User Persona, Use Case	1.0
Diana Chiu	October 9, 2018	External Interface Requirements, User Persona, Non-functional Requirement, Functional Requirement, Use Case	1.0
Kent Tran	October 9, 2018	Overall Description, User Persona, Functional Requirement, Non-functional Requirement, Use Case	1.0
Bonita Galvan	October 9, 2018	User Persona, Functional Requirement, Non-functional Requirement, Use Case, Basic Editing	1.0
Tanner Villarete	October 10, 2018	User Persona, Functional Requirement, Non-functional Requirement, Use Case	1.0

1 Introduction

1.1 Purpose

This document provides a detailed description of the requirements for the Data Classifier component of the Mark Logic Data Discovery Workbench. Contained in this document is the exhaustive outline for the development of the aforementioned system. This includes an overall description of the system, including definitions of the product features, users, operating environment, assumptions and business rules. Additionally this document includes requirements around external interfaces, nonfunctional requirements, and other general product information.

1.2 Document Conventions

Conventions

Term	Definition
User	Someone who interacts with the product
Stakeholder	Someone with a vested interest in the product
FE	Feature
FR	Functional Requirement
NFR	Non-Functional Requirement
AS	Assumption
DEP	Dependency
ML	MarkLogic
DC	Data Classifier
DDW	Data Discovery Workbench

1.3 Intended Audience and Reading Suggestions

This document was written for all Stakeholders of the Data Classifier component of the Mark Logic Data Discovery Workbench; primarily, developers, project managers, testers, and documentation specialists.

1.4 Project Scope

The software specified within this document is intended to be used as a piece in the greater MarkLogic Data Discovery Workbench Project, namely: the Data Classifier. The responsibility of the DC is to fit input data sets to a pre-defined ontological set of classes, as well as accept new, user defined categories. The DC also makes special note of datum representing sensitive information, with the definition of sensitive provided in the ontology. Lastly, the DC publishes a machine readable description of the data sets to the next piece in the ML DDW: the Data Catalog.

1.5 References

TBD

2 Overall Description

2.1 Product Perspective

Data has become a crucial part of virtually every industry; so much so that the integrity of data has now become the subject of scrutiny. As MarkLogic put in their project proposal, "Transparency, reproducibility, and verifiability are key requirements to establishing trust [in data]." To achieve this goal, MarkLogic hopes to use our system to provide a better and more secure service to manage their customers' data. Our product is a data classifier that will use machine learning techniques to identify classes of information from inputted data sets. The classifier will also identify fields that may represent sensitive information such as social security numbers to further product customer data.

2.2 Product Features

The major feature of the system is the ability to feed data sets into the classifier which will then use machine learning techniques to categorize the data into classes. Predefinied element categories will be already embedded in the system, but it should have the capacity to learn more. The user will then be able to modify these categories through a web-based graphical user interface before a machine-readable description of the data set is published. As an additional safety feature, the classifier will also identify sensitive information.

2.3 User Personas

2.3.1 Data Analyst: Barry Allen

(Victoria Law) **Barry Allen** is 52 year old male ***Data Analyst***. He has a Masters of Information and Data Science from UC Berkeley. He has worked in the data analytic industry for over 25 years and has experience coding and scripting. His job involves using multiple database software and languages including, SQL, SAS, and BigQuery. He is familiar with statistical modeling, machine learning, and data mining. His goals are to be able to apply advanced statistics and mathematical techniques to huge data sets. he finds frustration in the time it takes to normalize various data sources and formats into one data catalog to work with. He would like our program to be able to take large data volumes with new and evolving information and categorize the data into one cohesive catalog for him to use to uncover new insights.

2.3.2 MarkLogic Sales Representative: Annette Caulfield

(Joey Buelow) **Annette Caulfield** is a 36 year old female ***Mark Logic Sales Representative***. She has an undergraduate degree from New York University. She has worked in sales for technology companies for fourteen years, and is familiar with important aspects of dealing with big data, but is not very technical. Her job involves demonstrating to a customer the features of MarkLogic products, and must be able to highlight impressive and technologically important features. She dislikes having to spend extensive amounts of time learning new technical concepts just to be able to show off the product. She would like the Data Classifier to be easy to use for a fairly non- technical person. She also wants an easy sell, so she would like to have a good looking user interface.

2.3.3 Health Care Business Analyst: Wally West

(Bonita Galvan) **Wally West** is a 30 year old male ***Health Care Business Analyst*** with a Masters in Health Administration from Cornell University. He has no coding experience. He has always used the same software program to visualize his data, but it can only take in one data set at a time. His job includes sifting through information to determine what opportunities to pursue that would relate to the needs of his clients and his company. He needs to be able to combine the data from multiple clinical trials, each with their own format, into one to determine which approaches are viable for the company. His frustrations include using a new program and trusting that it will combine information correctly. He will need the program to be easy to understand, create categories and determine which ones should be marked as sensitive, and hide those sensitive categories from people who receive his reports and should not have access to them.

2.3.4 Business Analyst: Harrison Wells

(Diana Chiu) **Harrison Wells** is a 45 year old ***Business Analyst***. He does not have a university degree, but he has been working as a business analyst for over 25 years. Harrison typically uses spreadsheets to run his calculations, but has very little experience in other kinds of programs or programming. He spends a lot of extra time trying to discover and digest datasets from various sources so that he can run his assessments of business practices and would like a solution that tags datasets with relevant information so that he can circumvent this pain point.

2.3.5 Data Scientist Jojo Woleub

(Kent Tran) **Jojo Woleub** is a 60 year old father who works as a ***Data Scientist***. He graduated from a local community college with a culinary degree, but has now been working as a data scientist for the past 40 years. Jojo has worked as the sole data scientist for the same local software architecture company for his entire career, and therefore only has experience using out-dated software. He would like the new system to be very user-

friendly but also to be easily trained to operate in a similar way the old software worked since Jojo does not adapt to change well.

2.3.6 Business Analyst Jesse Quick

() **Jesse Quick** is a female 25 year old ***Business Analyst*** who is fairly new to the industry. She has her Bachelors in Business Administration from Cal Poly. She is familiar with only a few business software programs and has not done much work with large amounts of data. While in college she took one class on data systems and has basic knowledge of SQL. Her job is to look for trends, interpret and evaluate data in order to gather critical information to hand over to various stakeholders and produce useful reports. She needs to find a clean and easy way to navigate large sets of data. She wants the ability to categorize data and flag categories important to her research.

2.3.7 Climate Change Researcher Matt Sewall

(Tanner Villarete) **Matt Sewall** is a 42 year old male ***Climate Change Researcher*** employed by the EPA to analyze weather patterns over the past decades. He has extensive experience using software programs to view climate anomalies. His job requires him to analyze heaps of NASA weather data to track the effects of human activity on the weather, but needs a better way of classifying information. His current source of data comes from the data.gov API and he has no good tool to extract key insights. He would like a program to that can accept a CSV file and find trends and determine which pieces of data should be marked as potentially important.

2.4 User Classes and Characteristics

Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy. You might use a table like this:

User Class	**Description**
Internal Employee	This is a MarkLogic employee who understands the system inside and out. They have worked with the engineers who developed the system, so they understand its capabilities.
Technical Customer	This is user of the system who is outside of MarkLogic and has a basic understanding of the technical topics and terminology.

Non-Technical Customer	This is a user of the system who is outside of MarkLogic but has little to no technical background knowldge.

2.5 Operating Environment

The operating environment of the software will be web-based as the graphical user interface of the system is web-based. It will be hosted on the MarkLogic servers, but any computations done by the categorization will be done client-side.

2.6 Design and Implementation Constraints

Due to the nature of machine learning, constraints will probably be caused by hardware that is able to run the training sets. In addition, AWS Services may be used in which case cost of the services will be a limiting factor.

2.7 User Documentation

The system will be developed open-sourced on GitHub under the Apache License.

2.8 Assumptions and Dependencies

Potential dependencies include AWS, Pandas Python library for data processing, Scikit Learn for machine learning techniques, and NLTK for natural language processing. More dependencies will come to light once development begins.

2.9 Business Rules

To be determined once meeting with MarkLogic on Thursday, October 11th.

3 Use Cases

3.1 Use Case 1: Search Dataset

Use Case ID:	1
Use Case Name:	Compare Results
Created By:	Victoria Law
Last Updated By:	Victoria Law
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018

Actors:	Data Analyst
Description:	A data analyst wants to search for a specific set of data based on a keyword to easily compare data results from different sources.
Preconditions:	1. Data analyst has preloaded data sets into the data classification model. 2. Data analyst knows a specific keyword they want to search up.
Postconditions:	1. The result will show data that matches the specific keyword that the user inputted.
Normal Flow:	1.0 Input a search string. 1. The data analyst wants to retrieve a set of data. 2. The system displays a search bar for the data analyst to enter in a search string. 3. The data analyst types in the keyword into the search string. 4. The data analyst indicates that the search string is complete. 5. The system feeds the search string into the data classification model to search for data that matches the search string. 6. The system displays the results that were found. 7. The data analyst finds the specific set of data that she needed.
Alternative Flows:	1.1 Data was not found.

	1. The data analyst wants to retrieve a set of data. 2. The system displays a search bar for the data analyst to enter in a search string. 3. The data analyst types in the keyword into the search string. 4. The data analyst indicates that the search string is complete. 5. The system feeds the search string into the data classification model to search for data that matches the search string. 6. The system does not find any data that matches the search string and returns "No Results Found. " 7. The data analyst sees that there was no results found.
Exceptions:	There is no data loaded into the system.
Includes:	None
Priority:	High
Frequency of Use:	High, continuous usage every day.
Business Rules:	TBD
Special Requirements:	1. Data must be inputted into the data classification model before searching for data.
Assumptions:	Assume that data has already been inputted into the data classification model before searching for data.
Notes and Issues:	NONE

3.2 Use Case 2: Specify Custom Classifications

Use Case ID:	2
Use Case Name:	Specify Custom Classifications
Created By:	Joey Buelow
Last Updated By:	Joey Buelow
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018
Actors:	Health care Data Analyst
Description:	A Health care Data Analyst wants to create their own data classifications that were not generated by the system.

Preconditions:	1. The Health care Data Analyst has pre-loaded data sets into the data classification model. 2. The Health care Data Analyst has several new classifications he wants to add. 3. An ontology is already specified
Postconditions:	1. The data is now categorized by the new custom classifications. 2. The new classifications will be applied to any new data.
Normal Flow:	1.0 Specify a Custom Classification 1. The Analyst indicates that he wants to add a new classification. 2. The system displays field for the analyst to fill out in order to specify the form of data that fits in this classification and the name to display it as, as well as whether this field is sensitive. 3. When all required fields are complete, the system allows the Analyst to save his work. 4. If the Analyst is happy with what he has placed in the fields, he indicates that he would like to save his work. 5. The system displays a sample of the data that applies to the new classification and allows the Analyst to either confirm correctness, confirm correctness and make a new classification or fix mistakes in the definition. 6. The Analyst confirms correctness. 7. The system saves the new classification to be applied to new data. 8. The system applies the new category to the data sets. 9. The system displays the pre-loaded data in terms of the new classifications.
Alternative Flows:	1.1 Unhappy with Sample Data (Branch after step 5) 1. The Analyst indicates the sample data is not representative of his desired classification 2. The system returns to step 3 1.2 Specify Multiple new Classifications (branch at step 5)

	1. The Analyst indicates the sample is correct and would like to add another classification. 2. The system finishes steps 7 and 8 and returns to step 2. 1.3 Malformed Classification (branch after step 6) 1. If the new classification can not be translated into a well formed rule, the system displays an error message and returns to step 3.
Exceptions:	There is no data loaded into the system.
Includes:	None
Priority:	High
Frequency of Use:	Low, only applicable to new types of data sets, or new business requirements.
Business Rules:	TBD
Special Requirements:	1. The Analyst shall be able to cancel creating a new rule at any time and return to the home screen.
Assumptions:	The Analyst knows the "form" of their new classification
Notes and Issues:	TBD

3.3 Use Case 3: View Data Lineage

Use Case ID:	3
Use Case Name:	View Data Lineage
Created By:	Diana Chiu
Last Updated By:	Diana Chiu
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018
Actors:	Data Analyst
Description:	A data analyst wants to view the lineage of a particular dataset to better understand where the information came from.
Preconditions:	1. Data has been loaded and integrated into the workbench. 2. The data has some kind of metadata tag that indiciates it's origin.
Postconditions:	1. The user sees the origin of the data.

Normal Flow:	1.0 See Data Lineage 1. The Data Analyst selects a dataset to view the lineage. 2. The system fetches the data history. 3. The system displays a graph-like representation of the lineage.
Alternate Flow:	1.0 No Data Lineage 1. The Data Analyst selects a dataset to view the lineage. 2. The system fetches the data history and finds none. 3. The system indicates to the user that no history could be found.
Exceptions:	None
Includes:	None
Priority:	High
Frequency of Use:	Moderate, TBD
Business Rules:	TBD
Special Requirements:	None
Assumptions:	None.
Notes and Issues:	None

3.4 Use Case 4: Modify Generated Categories

Use Case ID:	4
Use Case Name:	Modify Generated Categories
Created By:	Kent Tran
Last Updated By:	Kent Tran
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018
Actors:	Data Analyst
Description:	A data analyst wants modify the categories generated by the data classifier.
Preconditions:	1. Data has been loaded and integrated into the workbench. 2. The data has successfully been classified into categories by the data classifier.
Postconditions:	1. The user is able to view the categories and the data under each category.

Normal Flow:	1.0 View Generated Categories 1. The Data Analyst selects a data set to categorize. 2. The system fetches the data history. 3. The system displays the categories generated and sample data for each category.
Alternate Flow:	1.0 No Categories Were Generated 1. The Data Analyst selects a data set to view the lineage. 2. The system fetches the data history is unable to categorize the data set. 3. The Data Analyst inputs categorizes, allowing the classifier to learn.
Exceptions:	None
Includes:	None
Priority:	High
Frequency of Use:	High, TBD
Business Rules:	TBD
Special Requirements:	None
Assumptions:	None.
Notes and Issues:	None

3.5 Use Case 5: Label Category as Sensitive

Use Case ID:	5
Use Case Name:	Label Category as Sensitive
Created By:	Bonita Galvan
Last Updated By:	Bonita Galvan
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018
Actors:	Health Care Business Analyst
Description:	A Health Care Business analyst wants label a category as sensitive to ensure patient privacy.
Preconditions:	1. Data has been loaded and integrated into the workbench. 2. The data has successfully been classified into categories by the data classifier.
Postconditions:	1. The user is able to view the categories and label them as sensitive.

Normal Flow:	1.0 Label Category as Sensitive 1. The health care analyst inputs a data set to categorize. 2. The system displays the categories generated and sample data for each category. 3. The health care analyst marks certain categories as sensitive.
Alternate Flow:	1.0 No Categories are Needed to be Labeled as Sensitive 1. The health care analyst inputs data sets into system 2. The system displays the categories generated and sample data for each category. 3. The health care analyst does not see any categories that need to marked as sensitive.
Exceptions:	There is no data loaded into the system.
Includes:	None
Priority:	High
Frequency of Use:	Dependent on User. High for Health Care Business Analyst
Business Rules:	TBD
Special Requirements:	None
Assumptions:	Data must be inputted into the data classification model before labeling categories.
Notes and Issues:	None

3.6 Use Case 6: Extract Data from CSV or JSON

Use Case ID:	6
Use Case Name:	Extract Data from CSV or JSON
Created By:	Tanner Villarete
Last Updated By:	Tanner Villarete
Date Created:	October 10, 2018
Date Last Updated:	October 10, 2018
Actors:	Climate Change Researcher
Description:	A Climate Change Researcher needs to view insights from a large CSV file.
Preconditions:	1. All information has already been collected. 2. The data is in either JSON or CSV format.

Postconditions:	1. The user is able to view weather insights using the data classifier.
Normal Flow:	1.0 View insights from CSV 1. The climate change researcher uploads a CSV to the system. 2. The system recognizes the CSV and extracts data to classify. 3. The climate change researcher views insights on the data classifier UI.
Alternate Flow:	1.0 No Categories are Needed to be Labeled as Sensitive 1. The climate change researcher uploads a JSON file to the system. 2. The system recognizes the JSON and extracts data to classify. 3. The climate change researcher views insights on the data classifier UI.
Exceptions:	The filetype is not a CSV or JSON file.
Includes:	None
Priority:	High
Frequency of Use:	Every time the system is used
Business Rules:	TBD
Special Requirements:	None
Assumptions:	Data must be correctly exported to CSV or JSON before importing.
Notes and Issues:	None

4 System Features

4.1 System Feature 1

4.1.1 Description and Priority

4.1.2 Stimulus/Response Sequences

4.1.3 Functional Requirements

1. REQ-1: (Victoria Law) The system shall identify classes of information contained in the data sets.

2. REQ-2: (Diana Chiu) The system shall put data into multiple categories.

3. REQ-3: (Bonita Galvan) The system shall allow users to label categories as sensitive.
4. REQ-4: (Kent Tran) The system shall allow the user to edit generated categories after being classified.
5. REQ-5: (Joey Buelow) The system shall allow the user to create their own categories.
6. REQ-6: (Tanner Villarete) The system shall allow users to import data using CSV or JSON filetypes.

5 External Interface Requirements

5.1 User Interfaces

The Data Workbench must be able to be used by technical users with varying levels of computer literacy. There must be a GUI interface so that a user without coding and scripting experience can still utilize all the features of the workbench: the data, the data lineage, the classifier, and the search. This GUI must work in Edge, Safari, and Chrome on at least Windows 7, Mac OSX Capitan, and Ubuntu 14. The UI should follow the colourscheme and feel of marklogic.com, as well as employ UI design standards such as Google's Material Design.

More details and screenshots will be added as the back-end features are better detailed.

5.2 Software Interfaces

Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

The Data Workbench will be built so it is integrated with an already existing database software. The exact software is to be determined. The database will store and provide data to the workbench. The Classifier component of the workbench is a machine learning

5.3 Communications Interfaces

The workbench GUI must work in a web browser. Sensitive fields should not be able to be accessed over the internet without HTTPS.

6 Other Nonfunctional Requirements

6.1 Performance Requirements

1. NFR-1: (Diana Chiu) The system should take no longer than 30 seconds to search for and present a tag to the user.

6.2 Security Requirements

1. NFR-1: (Joey Buelow) The system will require user authentication to access the system.

6.3 Software Quality Attributes

1. NFR-1: (Victoria Law) The system must be compatible with Linux and MacOS.
2. NFR-2: (Bonita Galvan) The system user interface should be viewable in a modern web browser (Chrome, Firefox, Edge)
3. NFR-3: (Kent Tran) The system should be scalable for all screen sizes including mobile, tablet, and desktop.
4. NFR-4: (Tanner Villarete) The system shall be modular to allow for further features to be added later on.

7 Other Requirements

TBD

A Glossary

B Analysis Models

C Issues List