

Data Manager Web Application

Business Requirements Document

Rebecca Ramnauth

2019-03-14

Contents

1	Executive Summary	2
2	Project Description	2
3	Project Scope	2
3.1	Technical Constraints	2
3.2	Assumptions	3
4	Requirements	3
5	Views	4
5.1	Content Views	4
5.1.1	Job Tracking	4
5.1.2	File Tracking	4
5.1.3	Regulation Tracking	6
5.2	File Views	7
5.2.1	Sub-project Source Code	7
5.2.2	Sub-project Configuration	7
5.2.3	Sub-project Log Files	7
5.3	Dashboards	8
6	Use Cases & Data Flow	9

List of Figures

1	Example of a Job View	5
2	Examples of Job Summaries	5
3	Example of a Files View	6
4	Examples of Files Summaries	6
5	Example of a Regulations View	7
6	File View of LexisNexis XML Parser Source Code	8
7	File View of LexisNexis XML Parser Source Content	8
8	File View of LexisNexis XML Parser Logs	9

9	Example of a Dashboard View	9
10	Roles, Permissions, and Use Cases	10

1 Executive Summary

The purpose of this Business Requirements Document (BRD) is to express the requirements of the Compliance Management team and stakeholders to be served by the deliverables of the Data Manager project. The primary audience for the document is the project team as well as the functional groups that must determine how to implement the application. This BRD contains both functional and non-functional requirements, an overview of the current process, as well as the proposed process once the solution is implemented.

2 Project Description

Data Manager is a web application that is responsible for the end-to-end management and administration of compliance regulation data processes. The management model involves four general components:

1. the configuration of feed subscriptions for regulatory content,
2. a comprehensive inventory of regulations for Compliance Tracking System (CTS)-Archer data feed,
3. file publication and high-level version control of sub-projects (e.g., XML parsers, batch loads, and web forms) on the resources server, and
4. status tracking of all related jobs.

In all, the objective of the Data Manager project is to create an integrated web site that projects a consolidated and consistent regulatory inventory for the Compliance Management team before CTS.

3 Project Scope

3.1 Technical Constraints

The following are relevant technical constraints that are involved with some of Data Manager's features:

1. For features that allow exporting to Microsoft Office 2016 (.xlsx),
 - a *maximum* of 1,048,576 rows allowed on a given worksheet
 - a *maximum* of 32,767 characters allowed in a given cell
2. For features that allow `null DateTime` values for ASP.NET calendars and calls a stored procedure with a `DateTime` parameter,
 - the *minimum* of a `SqlDateTime` object is January 1, 1753 12:00:00 AM whereas the *minimum* of a `C# DateTime` object is January 1, 0001 12:00:00 AM

- the *maximum* of a `SqlDateTime` object is December 31, 9999 11:59:59 PM whereas the *maximum* of a `C# DateTime` object is December 31, 9999 11:59:59 PM

3.2 Assumptions

The following are assumptions regarding Data Manager features:

1. Data Manager is responsive and should successfully display with full functionality in the following browsers. Display details may vary:
 - Internet Explorer 8.0⁺(Windows)
 - Firefox (Windows, Mac)
 - Chrome (Windows, Mac)
 - Safari (Mac)
 - Mobile Browsers (e.g., iOS Safari and Android browsers) on current devices

4 Requirements

	Category	Value	Description
R1	Accounts	critical	Users are authenticated using windows integrated security and Con Edison network account (via the Active Directory).
R2	Accounts	critical	Permissions are assigned using distribution lists for each user role (i.e., Administrator, Compliance Management (CM) Team, and Guest).
R3	Content Views	high	The application will permit authorized users to view, insert new, and update existing data rows to specific data tables of a secure connected database.
R4	Content Views	low	The application will allow users to extract a view of a data table as a Excel spreadsheet.
R5	Content Views	medium	Users can filter certain columns of a view of a data table through the application.
R6	File Views	medium	The application will allow administrators to view certain files on a specified server-located directory ¹
R7	File Views	high	The application must allow administrators to upload and replace files on a specified server-located directory ¹ .
R8	Code	high	Administrators can manage sub-sites, modules, and extensions of the Data Manager application through the website using a File View as previously described.

5 Views

As previously discussed, there are two primary views, 'content' and 'file' views. A *content* view allows users to view, insert, and update (based on their specific user permissions/role) rows in an existing data table with an existing database connection. On the other hand, a *file* view allows users to view, insert, and update files of a specified server-side directory. Access to a file view, however, is more exclusive to the more knowledgeable users (i.e., administrators).

The following sections outline examples of views that are essential to the Data Manager web application.

5.1 Content Views

These views concern regulatory content as extracted by parsers such as the LexisNexis XML Parser and necessary for CTS.

5.1.1 Job Tracking

The **job view** allows authorized users to browse current or past 'jobs', the execution of related projects. Figure 1 demonstrates requirements R4 and R5. This content view is of the data table `dbo.[Reg_Change_Job_Logs]` of the `dbo.[Lexis Export]` database.

Filtering **Job Type** (by dropdown menu), **job Start Time** (by AJAX Calendar Extender), or **job End Time** (by AJAX Calendar Extender) causes the ASP.NET gridview to refresh using the stored procedure `GetJobsByFilters`.

Requirement R1 is demonstrated using Windows Authentication. The username is located at the top right of every page in Data Manager.

Additionally, summary cards are provided on the same page below the view. As shown in figure 2, a summary of the job distribution, a bar chart of the number of each job attempted, and a list of incomplete jobs can offer a quick overview of the data in the view.

5.1.2 File Tracking

The **files view** allows authorized users to browse current or previously parsed 'files', usually `.txt` documents containing XML or feed data from an external source and downloaded into a server-side repository using the LexisNexis Bulk Download or similar processes. Figure 3 demonstrates requirements R4 and R5. This content view is of the data table `dbo.[Reg_Change_Extract_Tracking]` of the `dbo.[Lexis Export]` database.

Filtering **Subscription** (by dropdown menu), **file Start Time** (by AJAX Calendar Extender), or **file End Time** (by AJAX Calendar Extender) causes the ASP.NET gridview to refresh using the stored procedure `GetFilesByFilters`.

Similar to that of the Job View, summary cards are provided on the same page below the Files View. As shown in figure 4, a summary of the subscription distribution, a bar chart of

¹Specifically for related projects of CTS, such as the LexisNexis XML Parser and LexisNexis Bulk Download. This allows for the easy detection and resolution by administrators of exceptions or errors as recorded in log files on the server

Figure 1: Example of a Job View

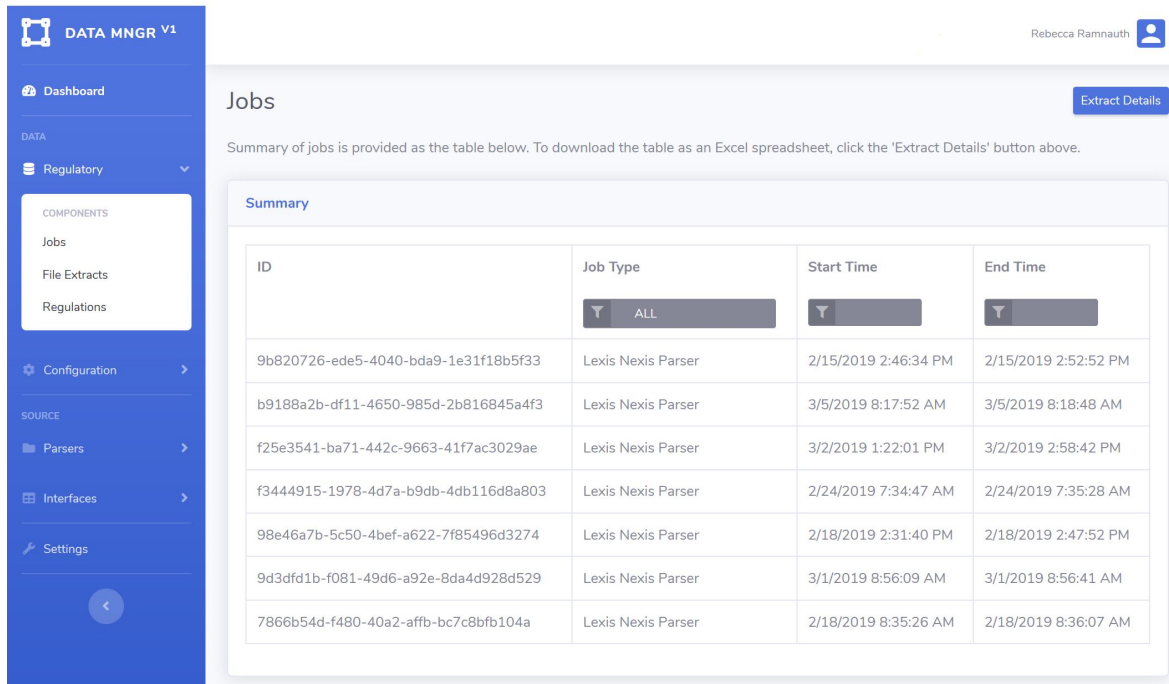
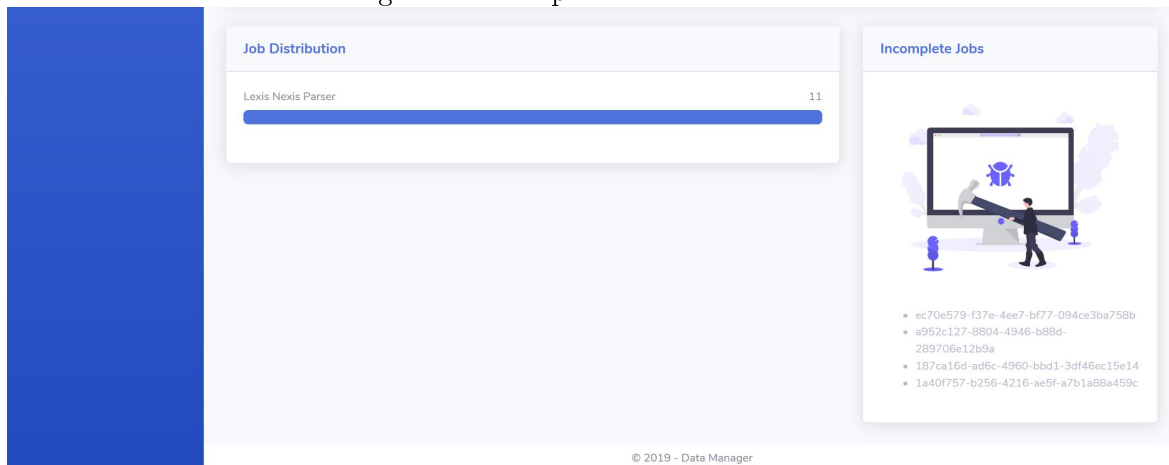


Figure 2: Examples of Job Summaries



the number of parsed files per subscription, and a list of files logged with exceptions can offer a quick overview of the data in the view as well as the status of that data on the server.

Figure 3: Example of a Files View

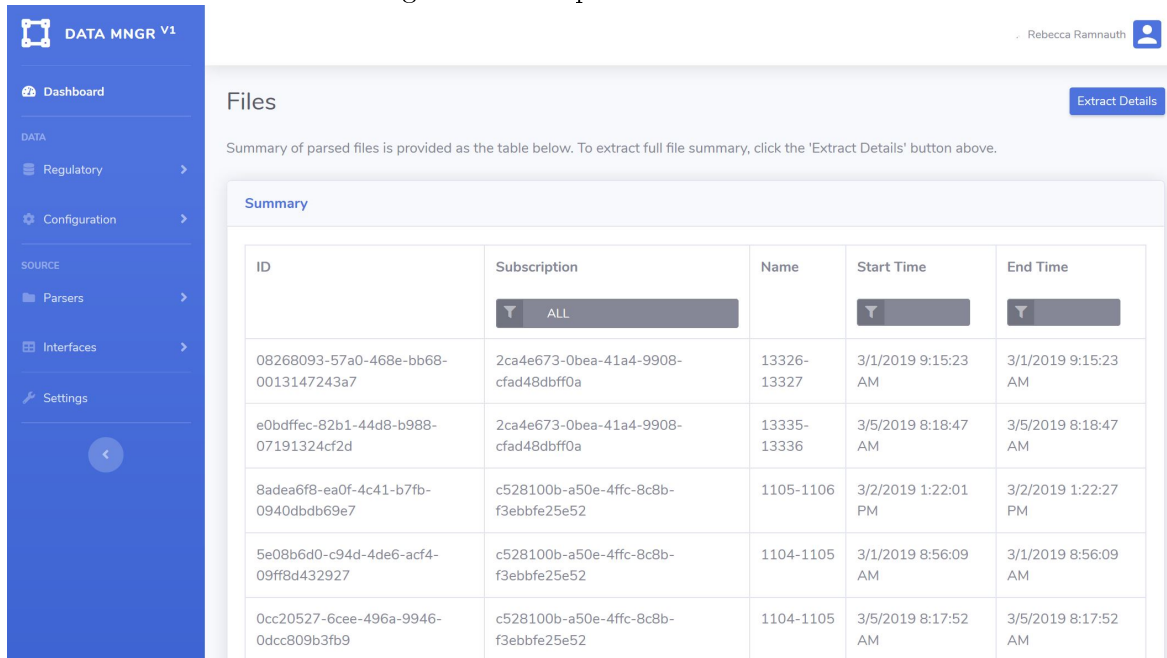
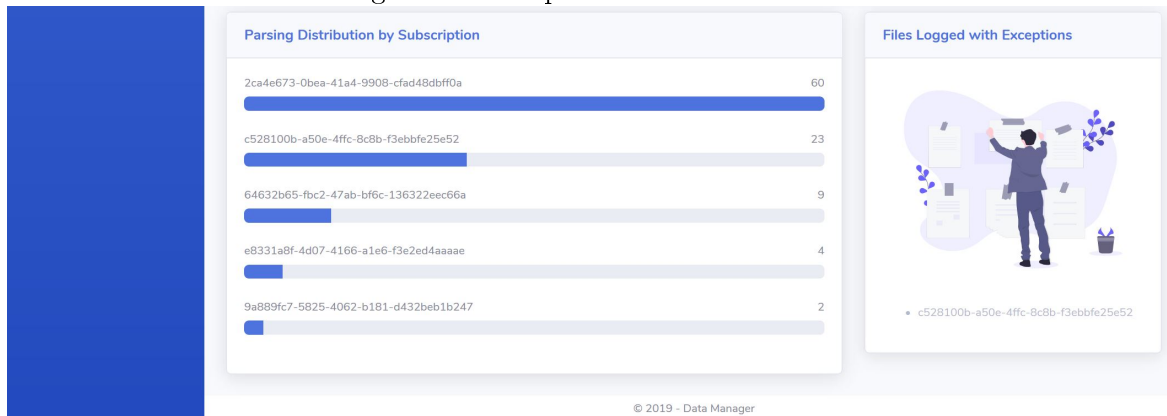


Figure 4: Examples of Files Summaries

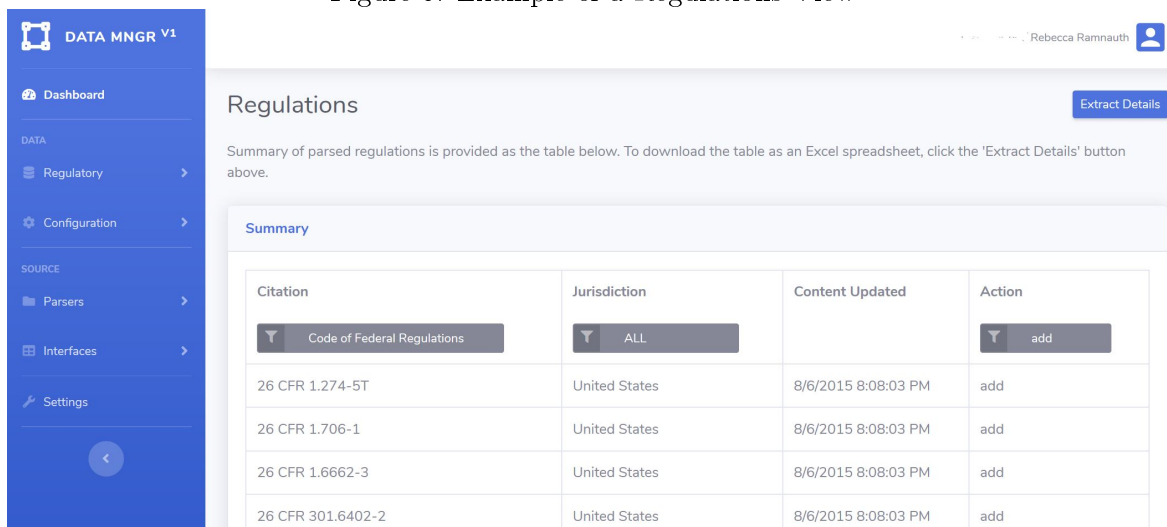


5.1.3 Regulation Tracking

The **regulations view** allows all users to browse current or previously parsed 'regulations', the result of a parser such as the LexisNexis XML Parser. Figure 5 demonstrates requirements R4 and R5. This content view is of the data table `dbo.[Reg_Change_Content]` of the `dbo.[Lexis Export]` database.

Filtering regulation **Citation** (by dropdown menu on attribute `[Reg_Body]`), regulation **Jurisdiction** (by dropdown menu), or regulation **Action** (by dropdown menu) causes the ASP.NET gridview to refresh using the stored procedure `GetRegulationsByFilters`.

Figure 5: Example of a Regulations View



5.2 File Views

These views allow authorized users to view, insert, and update files in a server-side directory that is related to a CTS sub-project (e.g., LexisNexis XML Parser source code and resulting logs).

5.2.1 Sub-project Source Code

A file view of a directory containing the source code of a related CTS project allows authorized users to browse the files in that directory and make changes to source code within reason. All changes to the view are recorded. Figure 6 implements requirements R6 and R7 through a file view of the LexisNexis XML Parser source code. For each file in the directory, there are the options to view, download, or delete the given file.

An authorized user can view a file on the same page below the ASP.NET gridview such as shown in figure 7.

5.2.2 Sub-project Configuration

A file view of a directory containing the configuration files or all files of a specified directory matching a certain pattern (e.g., *.config) allows authorized users to view and, if/when needed, change the configuration of a CTS-related project. All changes to the view are recorded. This configuration file view follows the same design as a source code view, such as figure 6.

5.2.3 Sub-project Log Files

A file view of a directory containing the logs of a related CTS project allows authorized users to browse or delete the logs stored in that directory. All changes to the view are recorded. Figure 8 implements requirements R6 and R7 through a file view of the LexisNexis XML

Figure 6: File View of LexisNexis XML Parser Source Code

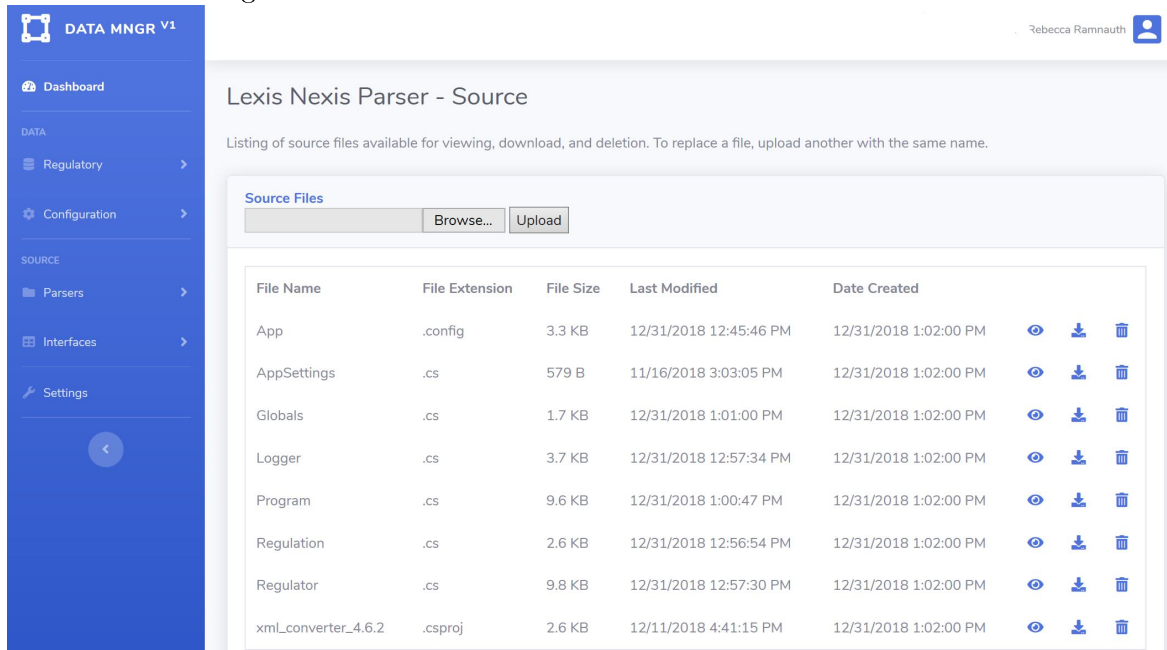
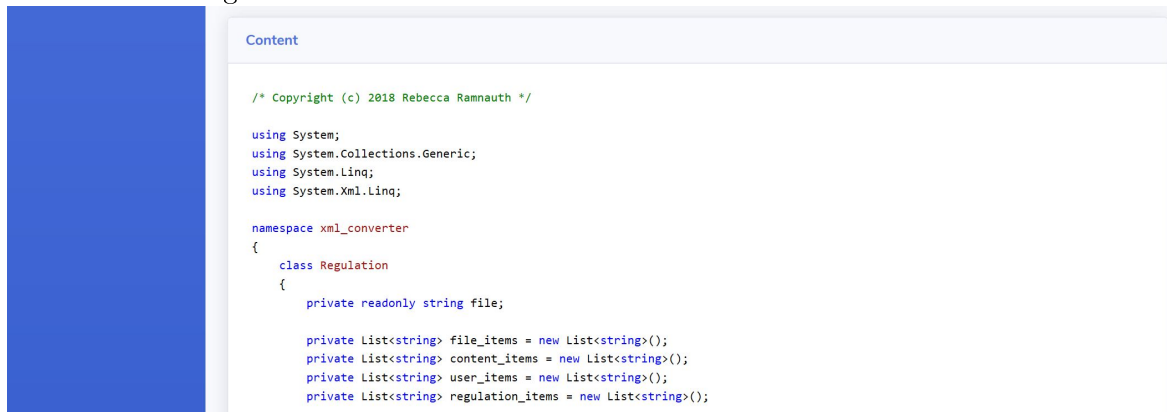


Figure 7: File View of LexisNexis XML Parser Source Content



Parser source code. For each file in the directory, there are the options to view, download, or delete the given file.

Similar to other file views, an authorized user can view a file on the same page below the ASP.NET gridview.

5.3 Dashboards

A dashboard is a custom page of set modules that summarize content throughout views as illustrated in figure 9.

Figure 8: File View of LexisNexis XML Parser Logs

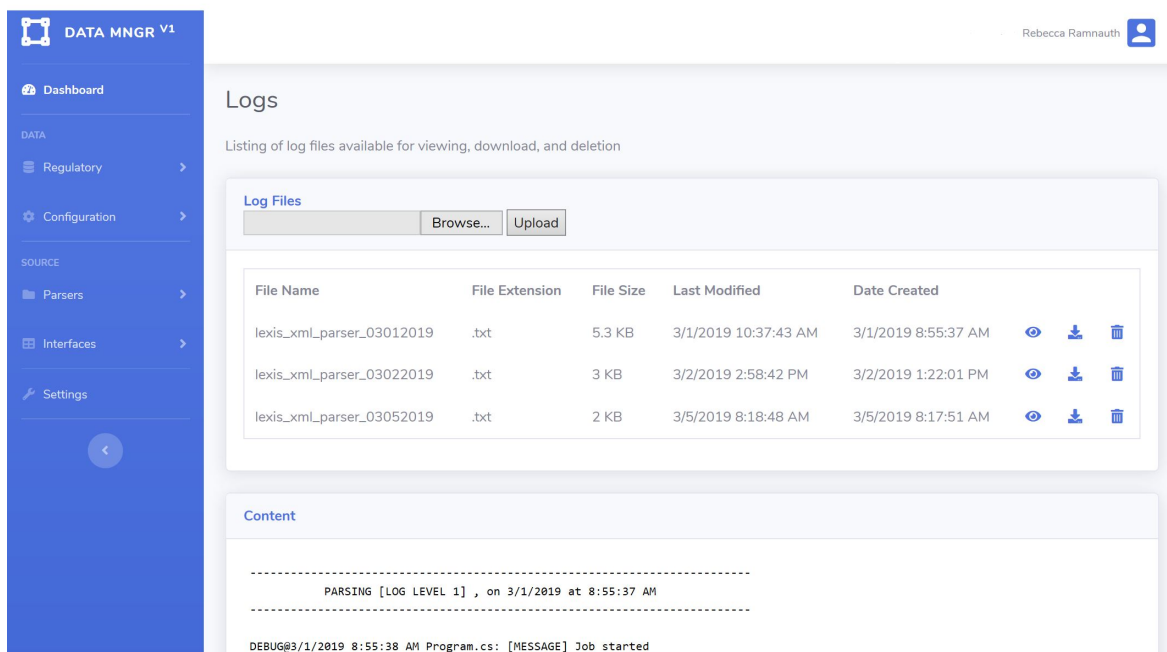
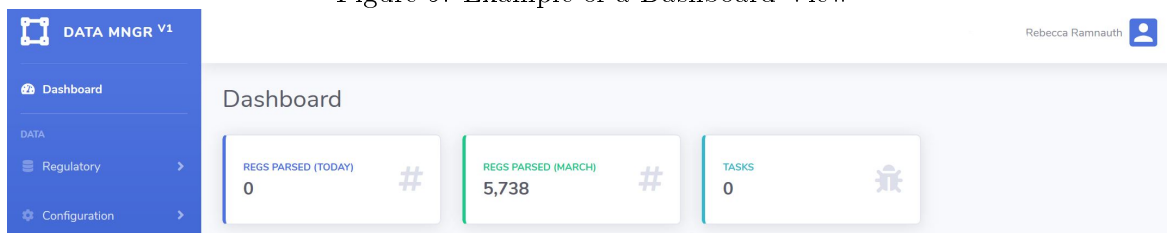


Figure 9: Example of a Dashboard View



6 Use Cases & Data Flow

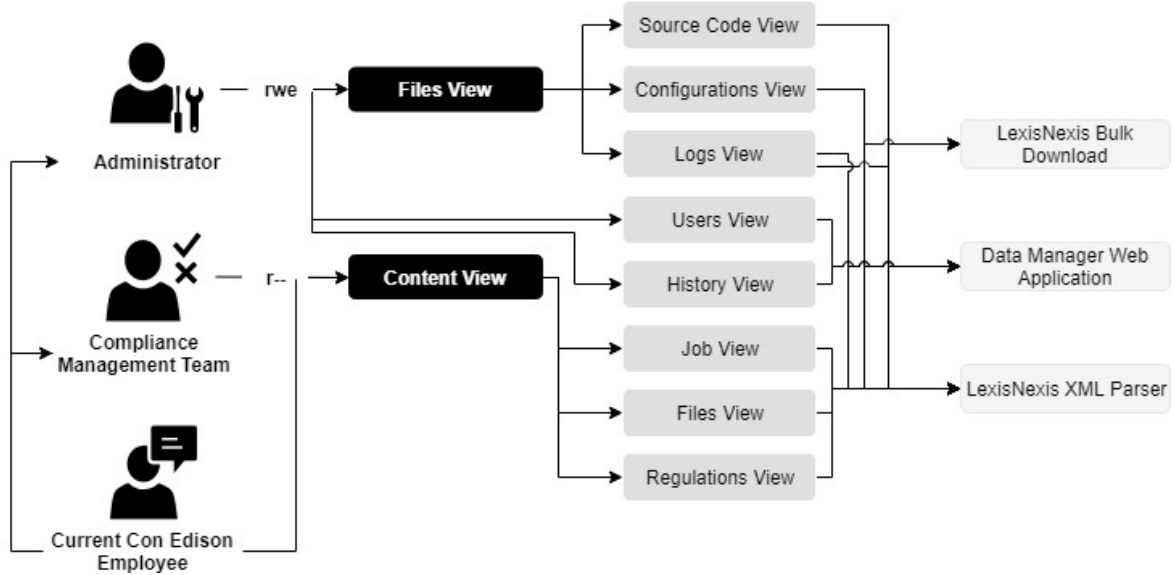
The following figure 10 illustrates the three primary user roles (i.e., Administrator, Compliance Management Team member, and guests). To highlight the distinction between a files and content view, only administrators can read (r), write (w), or execute (e) on these views. The *read* permission allows users to simply access the resource, whereas *write* allows authorized users to edit or insert rows in that view, and *execute* permits users to update or commit their changes to that view and, thus, to the underlying data table.

Furthermore, these high-level view types are broken down by purpose such as viewing the logs of a related project or reviewing the users of Data Manager. In addition, these more specific views are related to example projects, for instance, the LexisNexis XML Parser.

User roles build upon each other. For example, it is implied that a Compliance Management Team member will already have the permissions of a current Con Edison employee; a

Data Manager administrator will have the permissions of both a CM member and a guest user in addition to the administrator's elevated permissions.

Figure 10: Roles, Permissions, and Use Cases



The data from each view ultimately flow from Data Manager to the Con Edison staging database.