DEUTSCHE BUNDESBANK
EUROSYSTEM

The Data Stewardship Module

Workflows and Technical Requirements

Technical Report 2021-01

This paper is the outcome of external consultancy and support services for the INEXDA network in order to develop a data access/data stewardship software solution. It describes an existing solution developed by the Coleridge Initiative providing a detailed set of specifications that may serve as a best-practice example for the Deutsche Bundesbank and INEXDA.

Graham Henke (Coleridge Initiative)
Daniela Hochfellner (Coleridge Initiative)

Disclaimer: The views expressed here do not necessarily reflect the opinion of the Deutsche Bundesbank, the INEXDA network, or the Eurosystem.

Research Data and Service Centre

COLERIDGE INITIATIVE

The Data Stewardship Module

Workflows and Technical Requirements

Graham Henke, Daniela Hochfellner

Document type
Technical Report prepared for Deutsche Bundesbank

Citation details
Henke, G; Hochfellner, D (2020). The Data Stewardship Module - Workflows and Technical Requirements. Technical Report. Coleridge Initiative.

Date of publication
October 26, 2020

Acknowledgements
We gratefully acknowledge the financial support of the Deutsche Bundesbank, Schmidt Futures, the Alfred P. Sloan Foundation, the Overdeck Family Foundation, the Bill and Melinda Gates Foundation, and the US Department of Agriculture. Furthermore, we thank Amy O'Hara, George Putnam, and Bob Goerge for their valuable comments.

Glossary

ADA Applied Data Analytics.

ADRF Administrative Data Research Facility.

CI Coleridge Initiative.

DUA Data Use Agreement.

IRB Internal Review Board.

IT Information Technology.

MFA Multi-Factor Authentication.

MOU Memorandum of Understanding.

NDA Non Disclosure Agreement.

PI Principal Investigator.

TOU Terms Of Use.

Contents

1

Overview

The Coleridge Initiative (CI) has developed a Data Stewardship web application designed to automate the most common administrative workflows associated with the access to and use of confidential micro-data by approved analysts. In practical terms, the application provides a structured workflow for the management of (i) dataset policies, (ii) data access requests and approval workflows, and (iii) user-generated metadata. It also generates reports on project activity and dataset usage. The goal of this report is to document the implementation of that workflow in the context of its deployment with the Administrative Data Research Facility (ADRF). It includes a description of the features and technical requirements as well as a description of the underlying data model.

1.1 User Groups

The following chapters describe the common workflows and the features of the application organized by user group. It is designed with three user groups in mind.

- Data Users. This group includes government analysts, as well as academic or institutional researchers approved by the Data Stewards. Lead Data Users can add Data Users to their projects.
- Data Stewards. This group includes individuals charged with approving projects and agreements as well as monitoring project work.
- ADRF Administrators. This group is charged with facilitating access to and use of approved projects and data.

2

Workflows

2.1 Workflow Overview

The Data Stewardship application provides a way for Data Stewards to manage the access of Data Users in a responsible and transparent way for all parties involved, monitor the work of Data Users, and create reports for stakeholders. This chapter describes the workflows of the three user groups: Data Users, Data Stewards and ADRF administrators. Figure 1 provides an overview of the workflows of each. The high level workflow depicted in Figure 1 is broken out into workflows for each user group in the following sections.

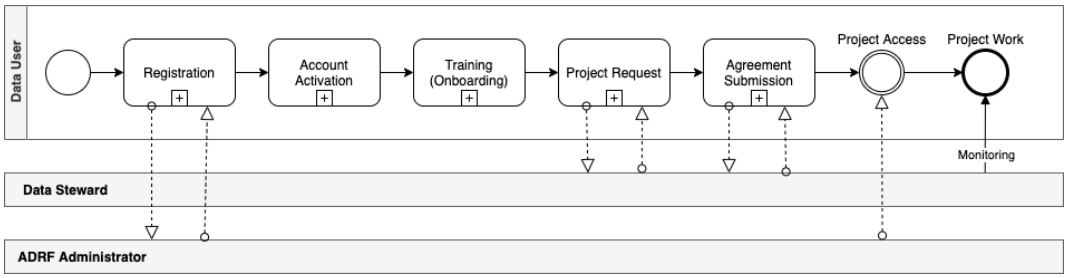

Figure 1: Workflow Overview

The primary purpose of the application is to facilitate the workflow outlined in Figure 1. Once a Data User has been registered, activated, and completed onboarding, they can request access to data by submitting a project request. The request form is filled out by the user and submitted via the application. The application then notifies the person(s) who is(are) responsible for the data requested and allows them to log into the application and see the request there waiting for approval. Once approved, both Data Steward and Data User can work together in an agreement center where both parties can upload and sign the agreements needed for data access. After the agreements are processed a research project is created specifically for the requested datasets in the ADRF. The monitoring aspect provides usage statistics back to the Data Steward.

2.2 Data User Workflows

The Data User must complete a series of actions which ultimately results in the provisioning of a secure project workspace with access to the datasets approved for their project. The following sections detail each step of this process.

2.2.1 Registration

Every person who interacts with the application requires a personal login and authentication. Before being able to use the application for project requests, every person involved needs to register in the application. Registration is done via a simple form shown in Figure 2 where users provide their name, email, institution, and a sponsoring organization (if this is sponsored research).

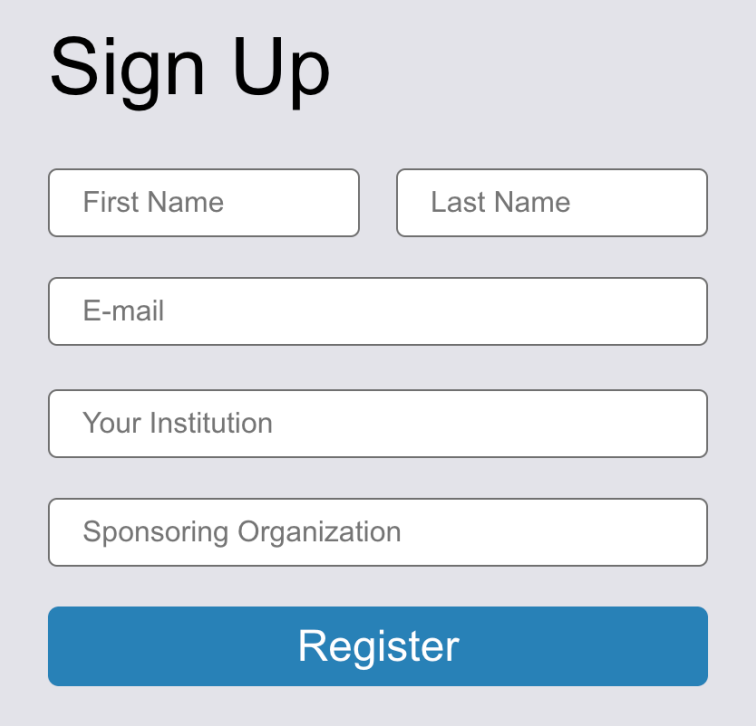

Figure 2: Application Registration

After clicking the submit button, the administrator receives a notification from the system that a new user has signed up and needs to be verified. The administrator is responsible for user verification; that verification is not currently part of the app and could be as simple as sending the user an email. The main purpose of the verification is to ensure that the user has a reason to be working in the ADRF. After verification, the administrator then approves the user. This process is shown in detail in Figure 3

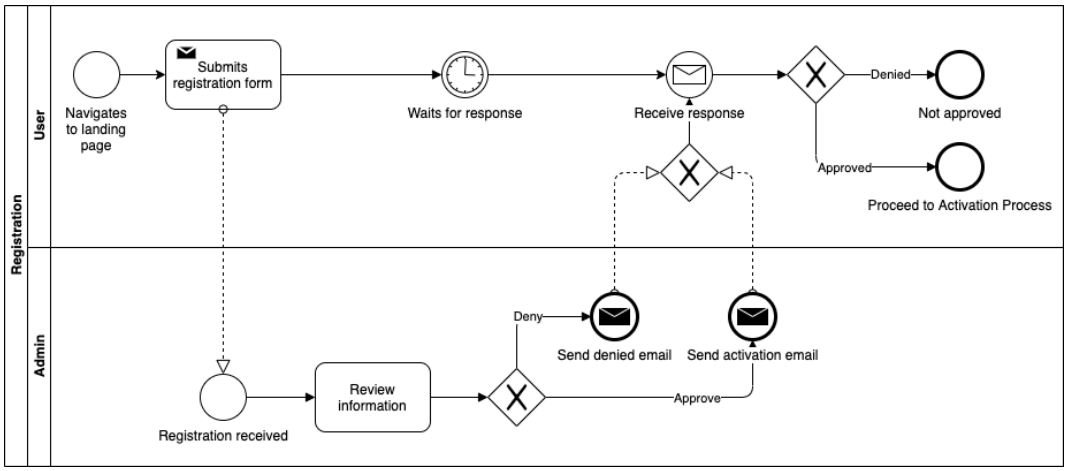

Figure 3: Application Registration Workflow

2.2.2 Account Activation

Once the user has been registered, they can activate their account in the ADRF. Approved users receive an email with an activation link. When the link is clicked, the user is taken to a page to set up Multi-Factor Authentication MFA. The user is shown a QR code which must be scanned using a mobile application such as DuoMobile. The MFA application will generate a random 6-digit code every 30 seconds which the user will use to authenticate. After MFA is set up, the user will also set a password for their account. The process is shown in detail in Figure 4.

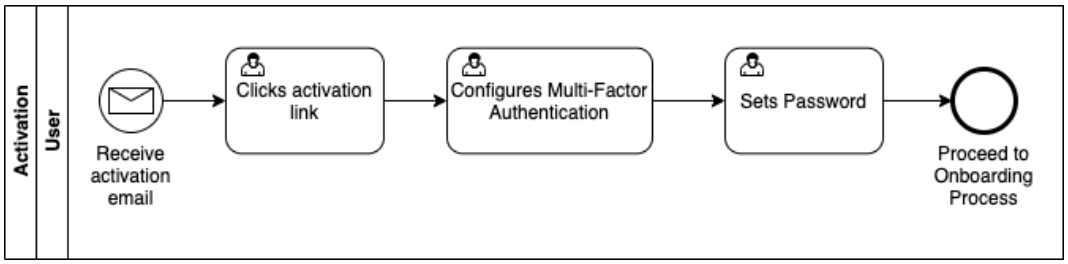

Figure 4: Account Activation Workflow

Once the credentials have been set, the user is directed to log in to the application where they must complete the onboarding process.

2.2.3 Training (Onboarding)

The user can now log into the application and see the onboarding page. The current onboarding consists of two required steps: (i) reading and agreeing to the Terms of Use, and (ii) watching the Security Awareness Training videos and completing a short quiz as proof of completion. Figure 5 illustrates the process.

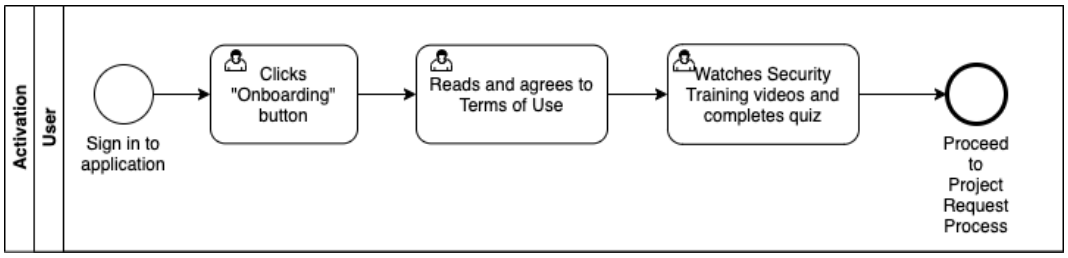

Figure 5: Training (Onboarding) Workflow

After completing the onboarding steps, the user receives full access to the application.

2.2.4 Project Request

The Data User can initiate a project request. The detailed workflow is outlined in Figure 6. When submitting a project request, the Data User lands on a request page and fills out a form with all the information required by the partnering agencies to approve the process. When the Data User submits the project request form, the Data Steward will be notified and will see a new request on their home page in the application.

The Data Steward can either approve or reject a project, and the Data User is notified of the decision. If more information is required, the Data Steward can request it. In this case the Data User is notified and can change the project request form and re-submit it. The history of the interaction of the Steward and the User in the web app is saved.

If the project is rejected, the Data Steward can optionally to provide a reason for rejection. If the project is approved, both the Data User and the Data Steward get access to the Agreement module to finalize the approval.

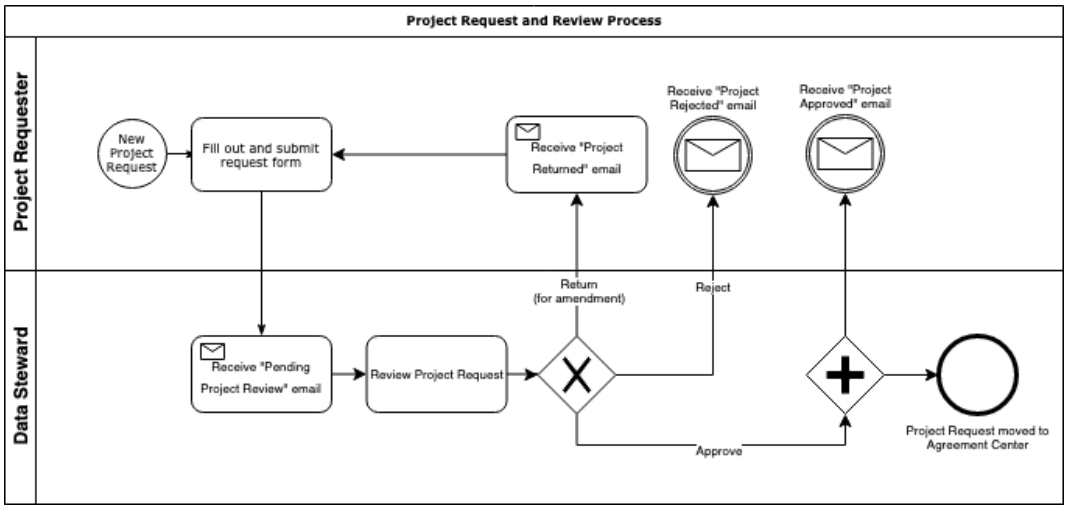

Figure 6: Project Request Workflow

2.2.5 Agreement Submission

The Agreements module functions as support to sign and submit data use agreements. The Data Steward initiates this phase by uploading all the agreements which are necessary for data access.

The Data User then will receive an email from the system notifying them that agreements (MOU and NDA) are waiting to be signed. The Data User can then log into the app, download the agreements, sign them, and upload the signed agreements back into the application (future versions of the applications will replace the download/upload with e-signature). In case negotiation is necessary, the Data User can upload a revised version of the agreement with requested changes. The Data Steward can then take the revised version to their legal counsel and discuss edits. The negotiation process can continue until there is a final version that is signed by all parties. Once the fully executed agreement is uploaded the data access process is completed. The iterative process is displayed in Figure 7.

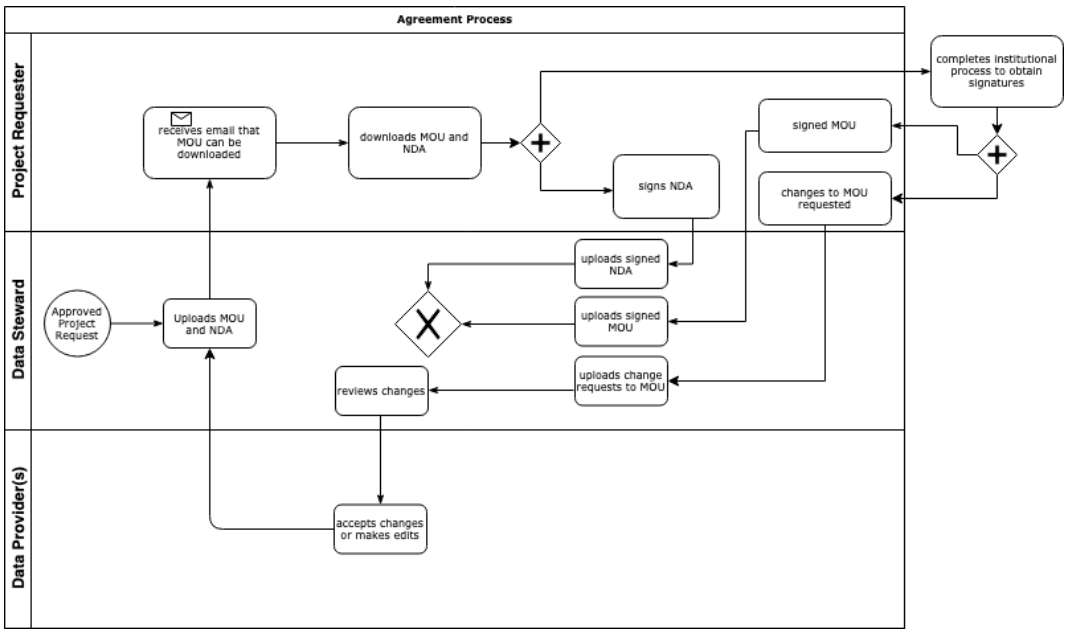

Figure 7: Agreement Submission Workflow

2.2.6 Project Access

The final step requires the ADRF Administrator to provide the Data User access to the project workspace, which includes access to the approved datafiles Figure 8.

If the project does not already exist, the ADRF Administrator creates the project workspace, which will provide access to the data requested as part of the project. Once the project workspace has been created, the

Data Administator will add the Data User to the project and send them the necessary instructions to access the workspace.

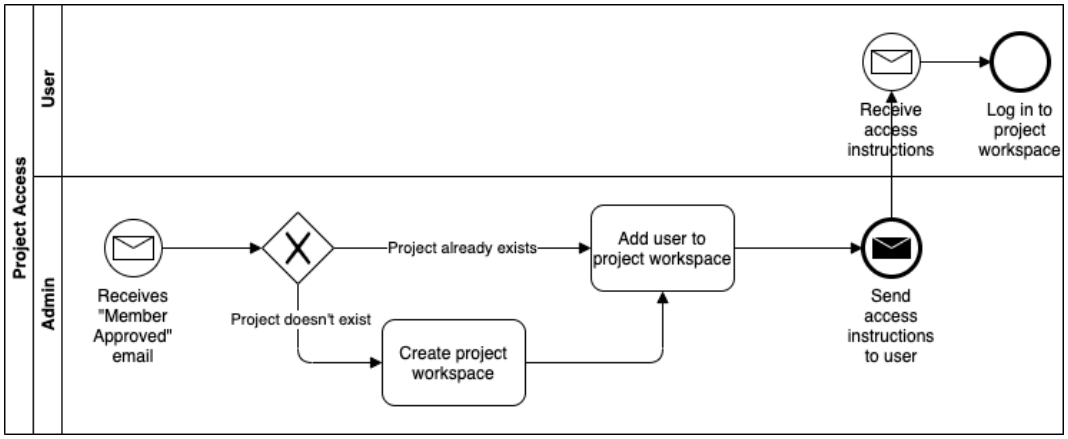

Figure 8: Project Access Workflow

2.3 Data Steward Workflows

The Data Steward, who is primarily responsible for reviewing project requests and monitoring dataset activity, is responsible for administering a number of distinct workflows in the application.

2.3.1 Metadata Creation Workflow

In order for Data Users to be able to work with data, they must be able to see an inventory, or catalog, of available datasets so that they know what is available. This catalog is known as the Data Explorer, and it is the responsibility of the Data Stewards to maintain the metadata for their respective datasets in this catalog.

The diagram in Figure 11 details the process of creating a metadata entry in the application.

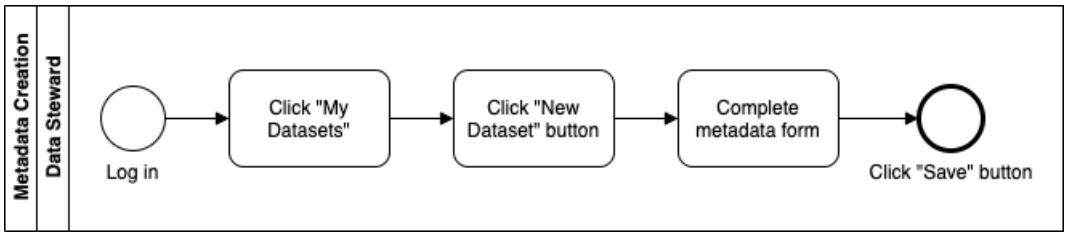

Figure 9: Metadata Creation Workflow

2.3.2 Data Steward Nomination Workflow

The Data Steward Nomination workflow is used when a Data Steward needs to transfer the responsibility of their dataset(s) to another user of the application. This feature is useful if the Data Steward is changing roles within the organization or leaving the organization.

Figure 10 illustrates this process. In the diagram, Data Steward A is the "nominating" steward, or the one who is transferring responsibility of their datasets, where as Data Steward B is the "nominated" steward, or the one who is receiving responsbility of datasets.

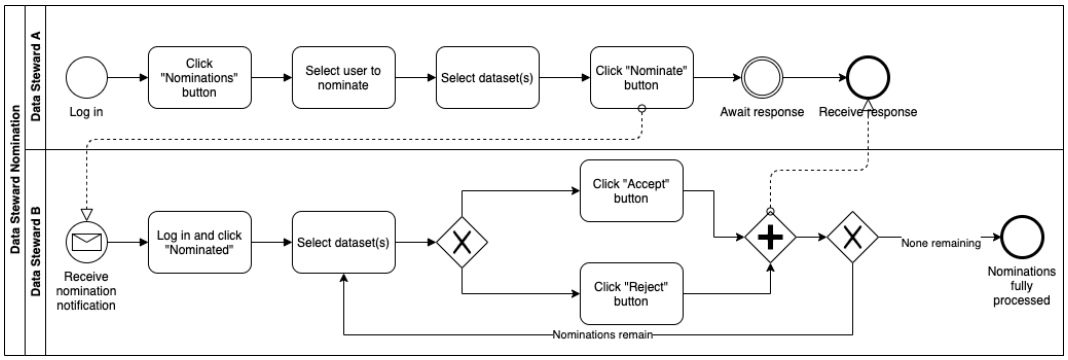

Figure 10: Nomination Workflow

2.4 ADRF Administrator Workflows

The ADRF Administrator is primarily responsible for approving and creating user accounts, creating project workspaces, providing user access to project workspaces, and general application maintenance.

This section describe distinct workflows that the ADRF Administrator performs in the application.

2.4.1 Metadata Creation Workflow

While it is generally the responsibility of the Data Steward to create and maintain metadata, the ADRF Administrator also has access to this functionality.

The diagram in Figure 11 details the process of creating a metadata entry in the application.

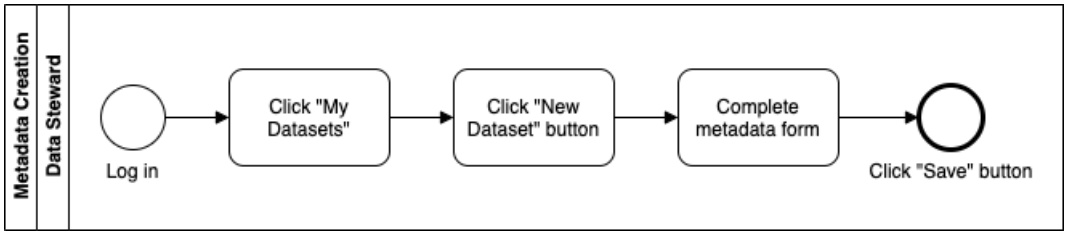

Figure 11: Metadata Creation Workflow

2.4.2 Institutional Input Workflow

An ADRF Administrator has the ability to add additional input fields to the project request form based on requirements from a given institution. This feature is described in more detail in section 3.2.6, while the workflow is shown below in Figure 12.

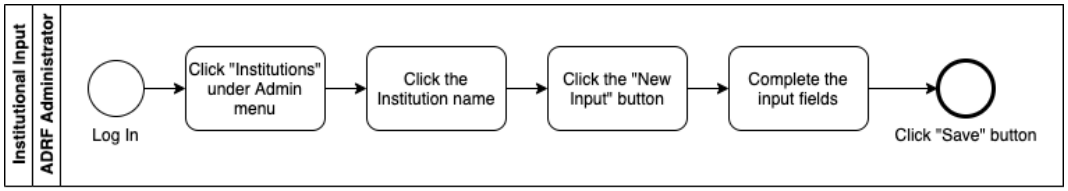

Figure 12: Metadata Creation Workflow

3

Application Features

3.1 Features Overview

The features contained in the application, like the workflows, largely correspond to the specific group to which the end user belongs. As such, this chapter is divided into Data User features, Data Steward features, and ADRF Administrator features, with a section at the end which describes general features that are applicable to more than one group.

3.2 Data User Features

These features are primarily for the Data User role of the application, though the may be used by or require some interaction with other roles. For example, the Project Request Page requires interaction with the Data Steward role, but since a project request is initiated by a Data User, it is documented here.

3.2.1 Onboarding Page

The Onboarding page in the application is actually the first and only page a new Data User will see the first time they log in to the application. The is because before the user is allowed to take any actions in the application, such as requesting a project, they are required to agree to the Terms of Use, complete Security Awareness Training, and affiliate themselves with their institution(s).

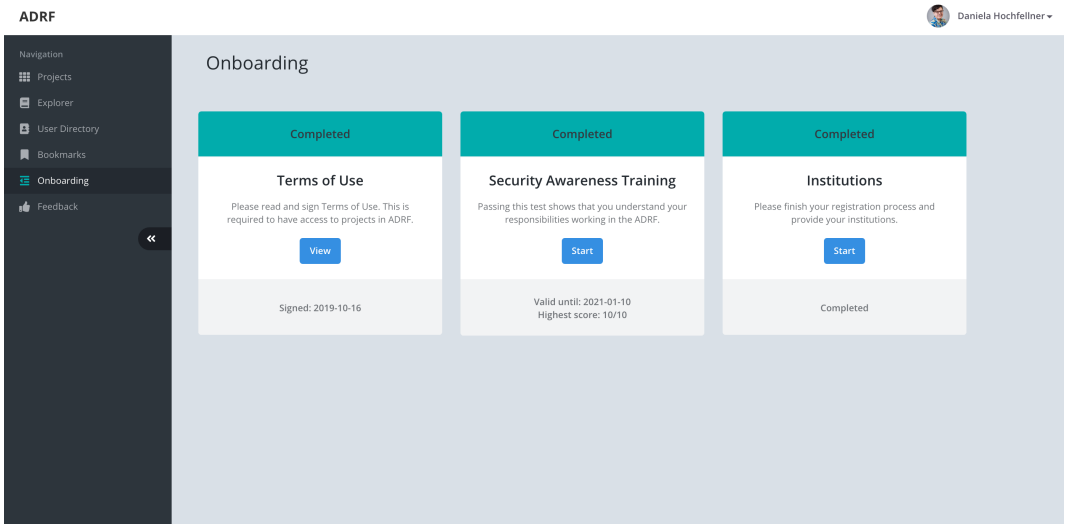

Figure 13: Onboarding Page

Terms of Use

The Terms of Use page contains rules of appropriate use and behavior that all users must agree to in order to be granted permission to work with any of the data that is referenced in the application. At the bottom of

this page, the user must select a checkbox stating that they agree and click the submit button. The date at which they accepted is recorded in the application.

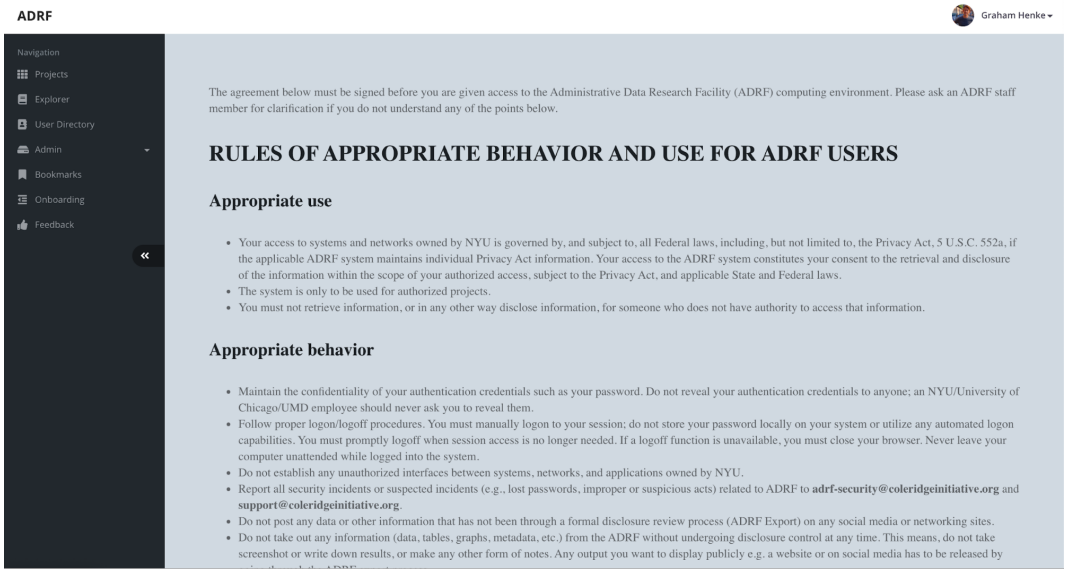

Figure 14: Terms of Use Page

The Terms of Use text is editable by application admins. This option is accessed in the Admin menu under Terms of Use. When the Terms of Use are updated, all users will be required to accept the new Terms of Use once they sign in to the application.

Security Awareness Training

The Security Awareness Training consists of a series of training videos and an accompanying quiz that each user must complete annually. When the user successfully completes the quiz, the date of completion and score are recorded in the application.

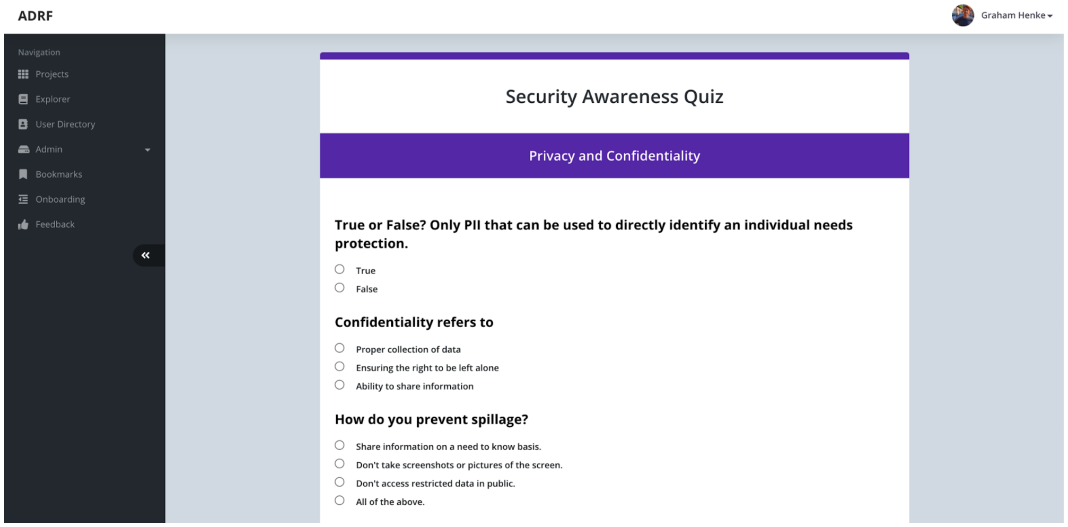

Figure 15: Security Awareness Training Quiz

After one year has passed, the user must again complete the training in order to use the other functionality of the application.

Institutions

The Institutions section of the Onboarding is where a user must set their institutional affiliations. This information is required in order for Data Stewards to determine whether to grant access approval for users to the datasets which they are requesting.

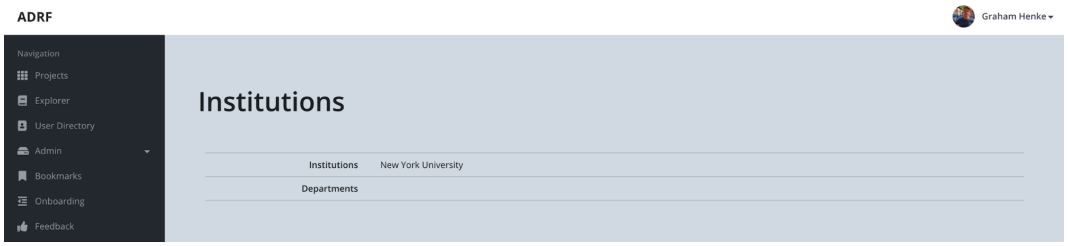

Figure 16: Institutions Page

3.2.2 Biographic Page

The Biographic page of the web application stores all user-relevant data in an interface that can be edited by the user directly. Figure 17 illustrates the page setup. The page is pre-filled with the information the user provides when registering for the app. During the account registration, the user will input personal data that is necessary for the ADRF workflows. Thus the Biographic page needs to have a field for each user attribute. Once a year the application sends out an email to all registered users to update their biographic page.

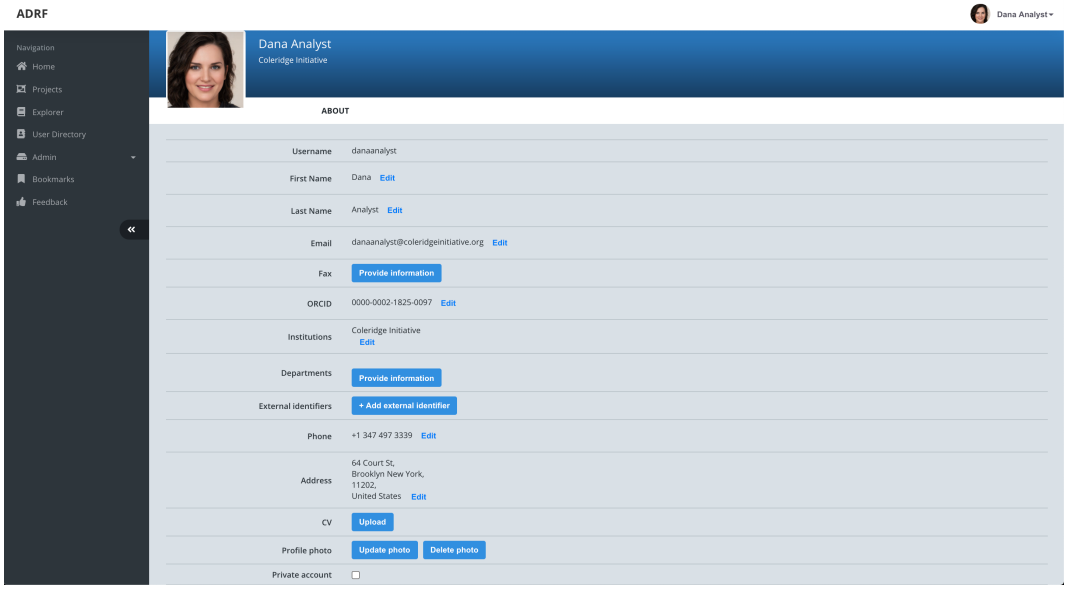

Figure 17: Biographic Page

All users, researchers, administrators and data stewards have a biographic page. Mandatory information is name, email and contact information. This is collected to then automate the information in the agreement center.

Each user can decide on their biographic page if they want to be searchable in the user registry. If they want to be private, there will be a hash generated that the user can forward to collaborators so they can find them in the application. The name of the user is pre-filled and taken from the registration form. The user cannot change their name on their own, as this information is tied to the unique user id. Instead the application has a button next to the user name that allows to send a user change request to the administrator.

The information on the institutional affiliation of the Data User is taken from the registration form. However, the Data User can change that information when an institutional change occurs. The edit button next to the field will allow the Data User to change their affiliation or to choose a secondary affiliation or department. The interface will be updated, and the former affiliation will be stored in the application's institution database. Users can pick secondary institutions and select departments.

The information on the job title and email of the Data User is also taken from the registration form. However, the Data User can change that information when a job change occurs using the edit button next to the field will allow the user to change the affiliation. The email information is the one associated with the account creation, and is pre-filled on the biographic page. The user is allowed to change the email by clicking on the edit button. If applicable the user can enter a Fax number.

The ORCID ID field allows the user to add an external identifier. The ORCID ID can then be used in the data explorer to display publications of the user. The external identifiers field allows users to add another external identifier.

The user can also add their phone number to the project. The address field is mandatory and holds the address of the user. The user will be asked to update the field yearly. Users can also upload a CV as a pdf, which is needed for some project requests. The user will be asked yearly if they would like to update their CV. All versions will be saved in the application. When a user requests a project the CV will be loaded from this page. Each user can also (voluntarily) upload a profile picture.

3.2.3 Data Explorer Page

The data explorer page is where all Data Users can find more information on all datasets that are available and information about the datasets.This page is available for all users of the application. It serves as a data inventory. Researchers can use it to look up data that they can request, data stewards can use to look for research outputs associated with their data. The information displayed in the data explorer is collected upon ingestion of the each dataset in the ADRF. During ingestion, the Data Steward fills out a metadata form. This information is shown in the overview tab when a specific data source is selected. More information on that specific dataset, such as published papers and the names of dataset experts are fed into the application through the rich context API. The following screenshots show the different available views within the data explorer.

Data Tab

The main page of the data explorer as displayed in Figure 18 shows a list of all datasets that are catalogued. The user can filter the search by data that are ADRF hosted, which means accessible in the ADRF, and data that belongs to institutions that have their data outside of the ADRF but use the web application to grant access to their data. In addition, there is a category filter which allows the user to narrow down the search according to predefined topics. The categories are defined by the keywords that agencies provide us during the data ingestion process. If the user is looking for a specific dataset they can use the search bar and start typing the dataset ID (is generated during data ingest) or name. The search bar auto completes the text and brings up the specific dataset. The dataset list displays for each dataset the most important information (collected from the metadata form): Name of data, ID, Name of data owner, name of data steward, and a short description of what information can be found in the dataset. The user is able to click on each of the dataset previews. This will bring them to a new page listing more detailed information about the data.

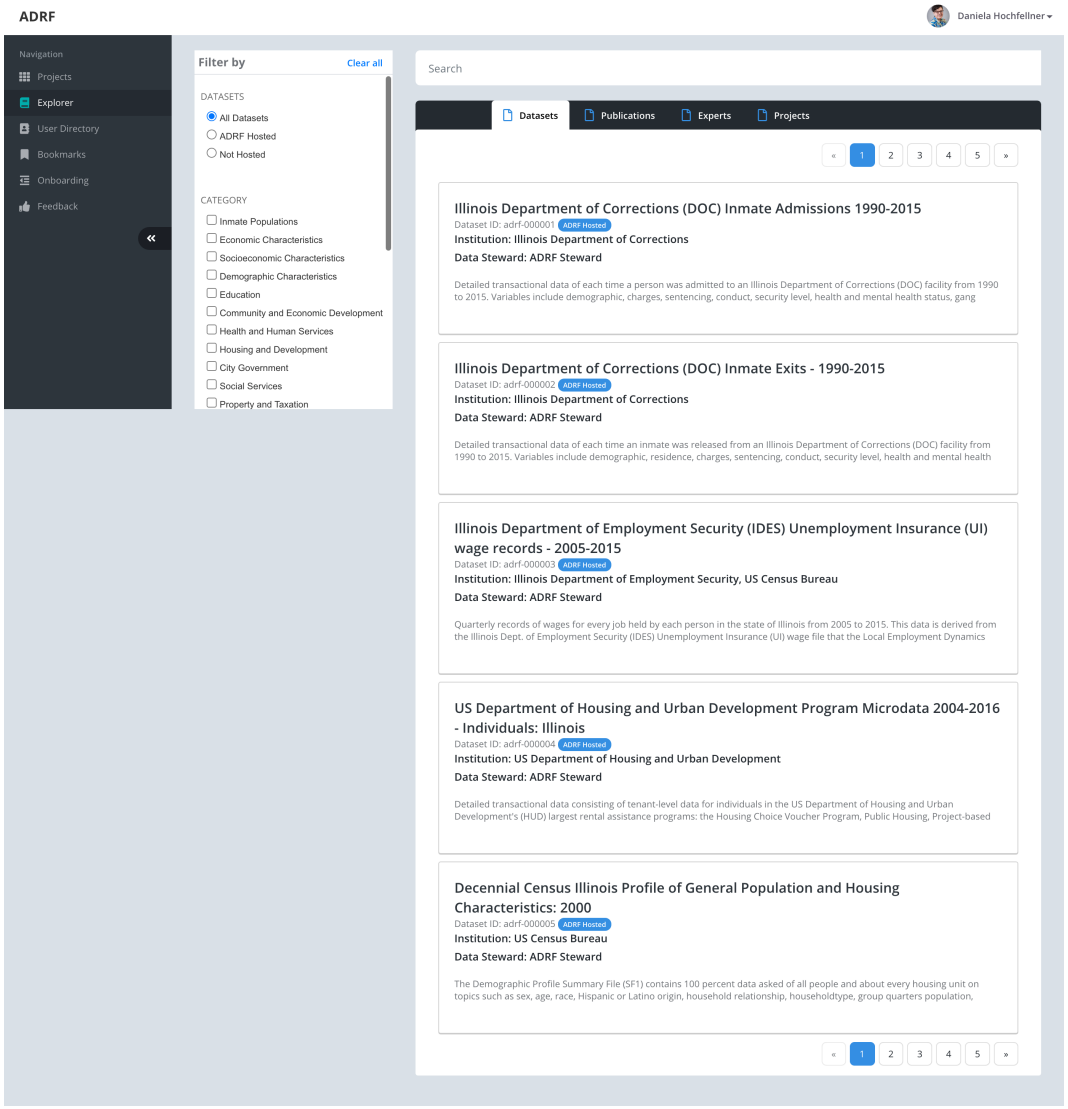

Figure 18: Data Explorer - Data Tab

Figure 19 shows the metadata that is provided for each data in the ADRF explorer. This includes a short description of the data, an indicator if the data is restricted or not, the format, some tags and the covered time period. In addition there is a setof descriptive metadata, such as the ID, the name of the responsible data steward and providing agency. In a second panel on the bottom the application displays information on publications that used the selected data. This information is ingested from the rich content knowledge graph, a project that CI is working on to advance metadata.

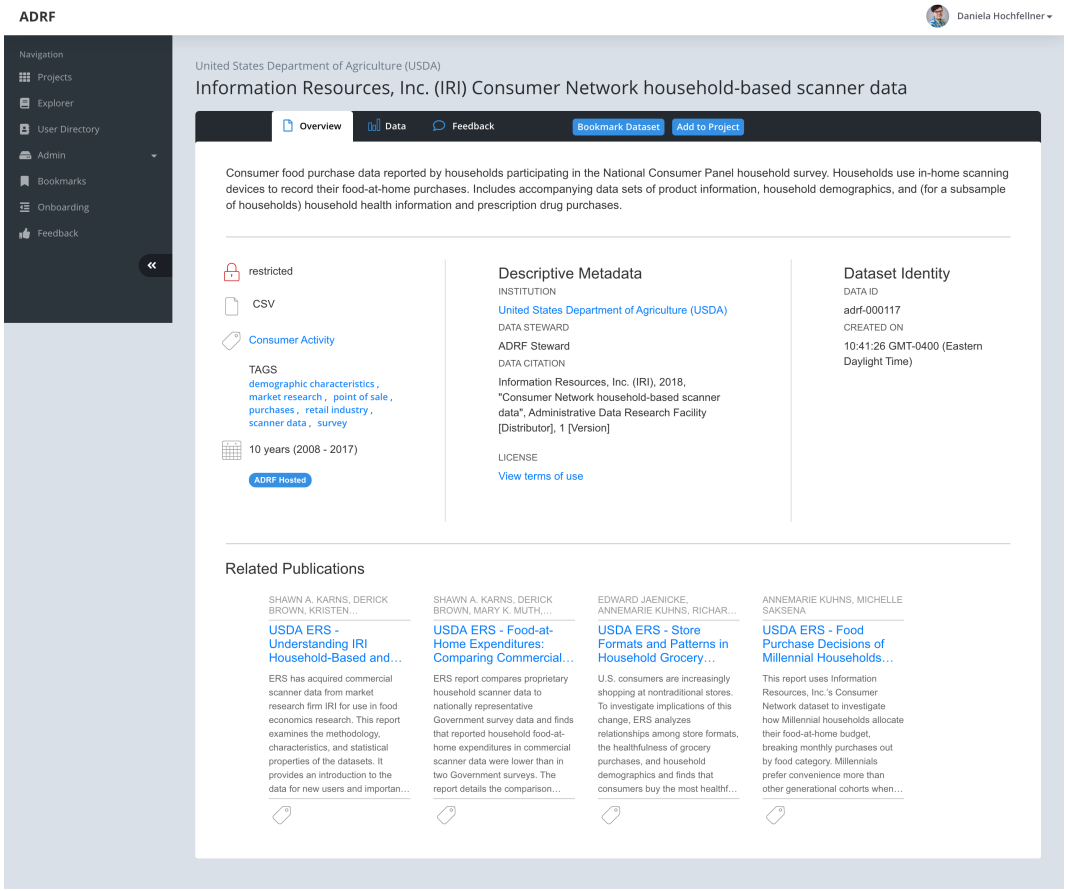

Figure 19: Data Explorer - Data Details: Overview

The next tab as displayed in Figure 20 shows column level metadata if these can be displayed in the explorer. There are some dataset where column level metadata is classified confidential. In this case the data tab will be empty. In general the column level meta data provide all the variable names, a short description of what that variable means or contains, or how it is being generated, in addition to the type of the variable,

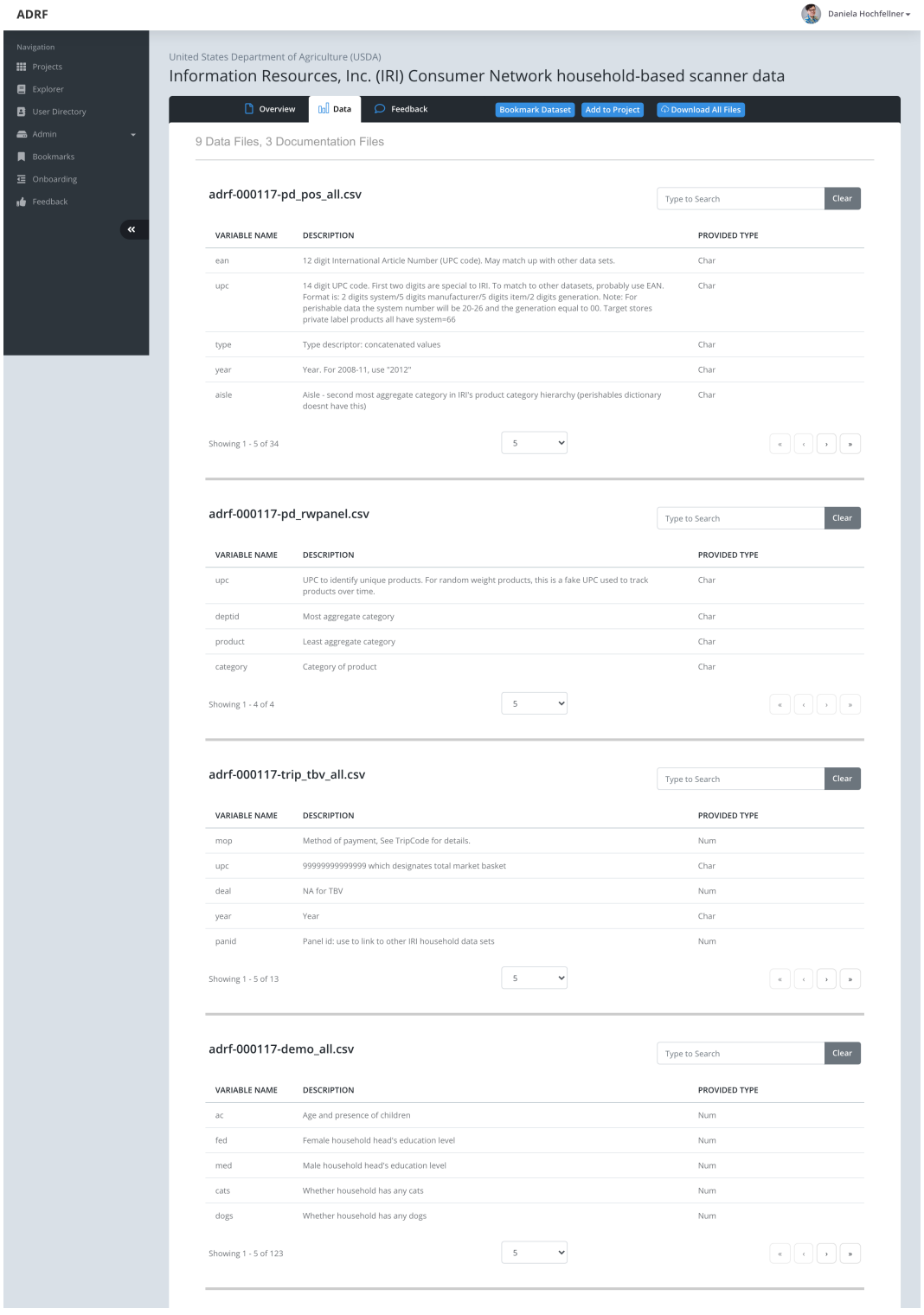

Figure 20: Data Explorer - Data Details: Data

The Feedback tab outlined in Figure 21 is there for users who are working with that dataset to provide feedback. For example if a user finds quality issues with one of the provided variables they can leave feedback for that dataset. This will help the agancy to improve their data products and other users to understand the data better.

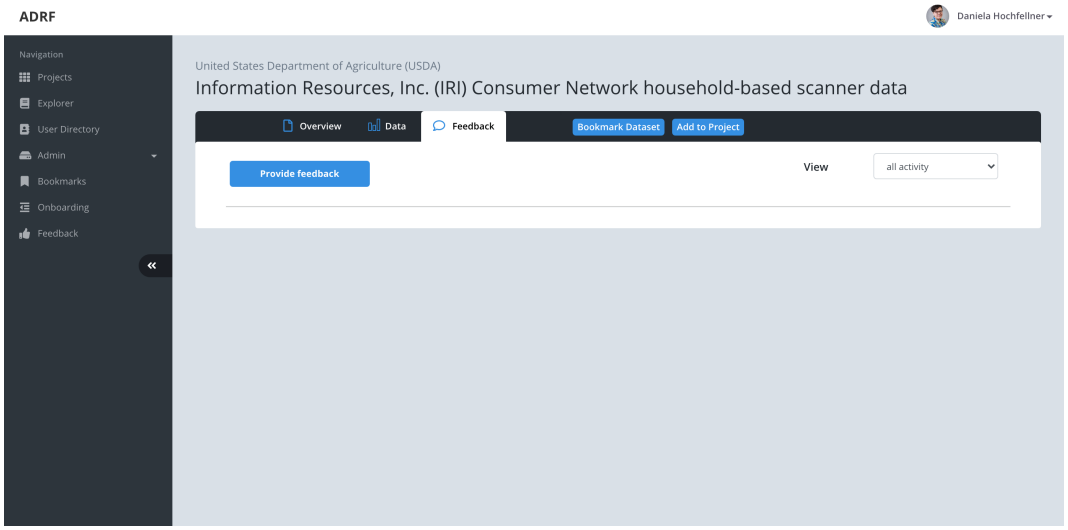

Figure 21: Data Explorer - Data Details: Feedback

Publication Tab

The user can switch to the publication tab and now sees all publications that are cataloged in the ADRF. The list of publications is retrieved by using the rich context API to read in the information. There is a similar search bar set up where the user can filter by author name, topics and keywords. There is also a search bar that allows the user to input text and find a specific article. The standard information shown on the publication preview is the title, the article ID (as defined in the rich context API) and the abstract. Clicking on the preview will bring the user to a full view of the selected article.

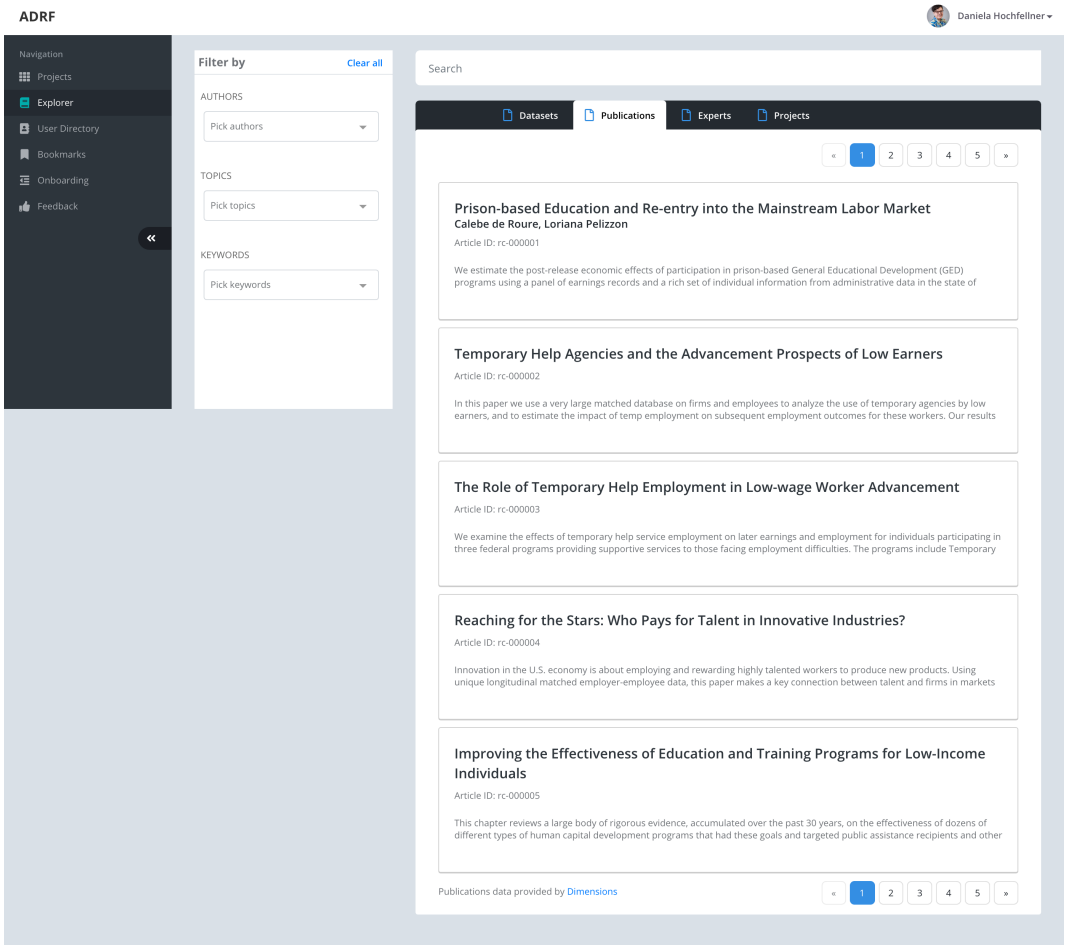

Figure 22: Data Explorer - Publication Tab

The full view, outlined in Figure 23, shows the Title, the full abstract, all author names and further information on the publication such as citations and the source URL (if available). In addition, publication metadata is provided, such as the doi, topics and keywords (if available). A unique display for each of the articles is the related dataset panel. This panel shows datasets that have been used in the publication. When clicking on the listed dataset the user will be taken to the data overview page in the explorer.

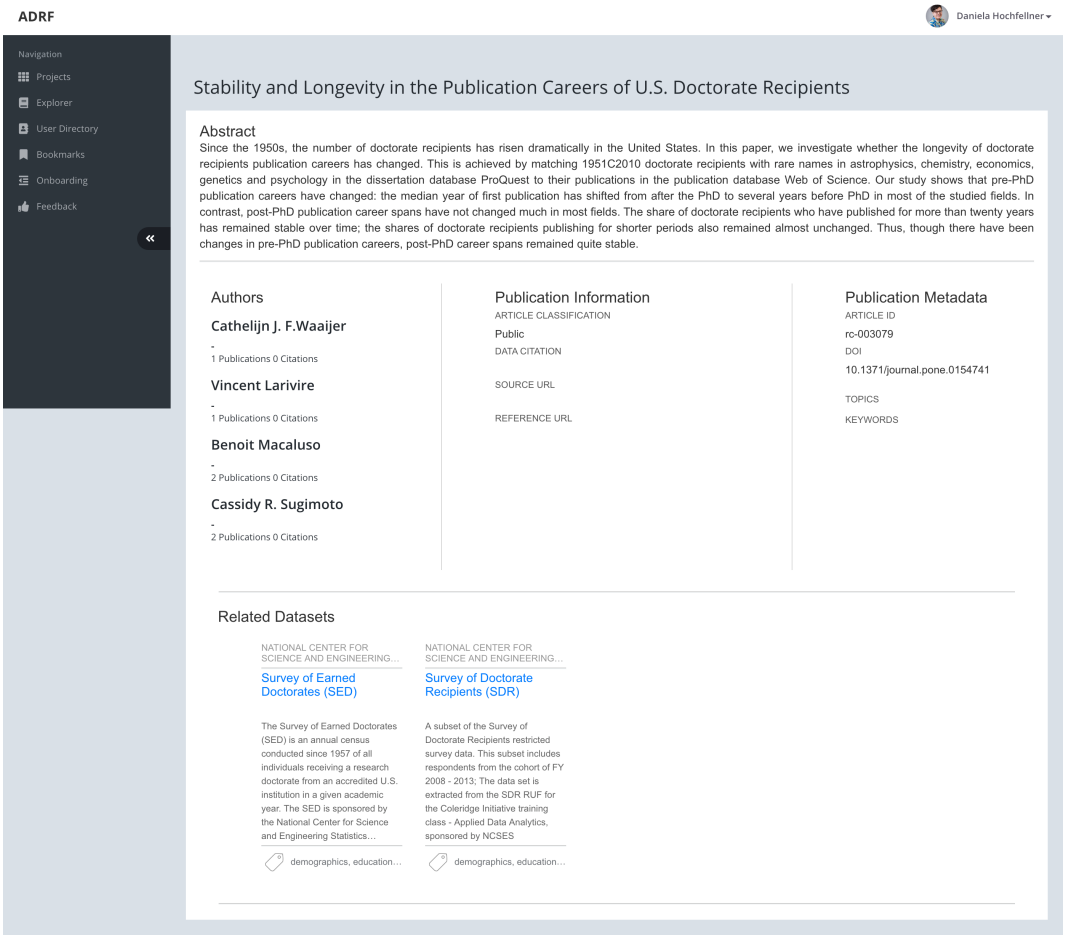

Figure 23: Data Explorer - Publication Details

Expert Tab

The expert tab lists people who might be able to provide subject matter expertise on specific topics and/or data. Any person who is publishing academic papers or policy briefs is considered an expert. The expert tab is filled through the rich context API by querying all authors. The user is able to filter by institution which will help to look up a specific person of a certain institution.

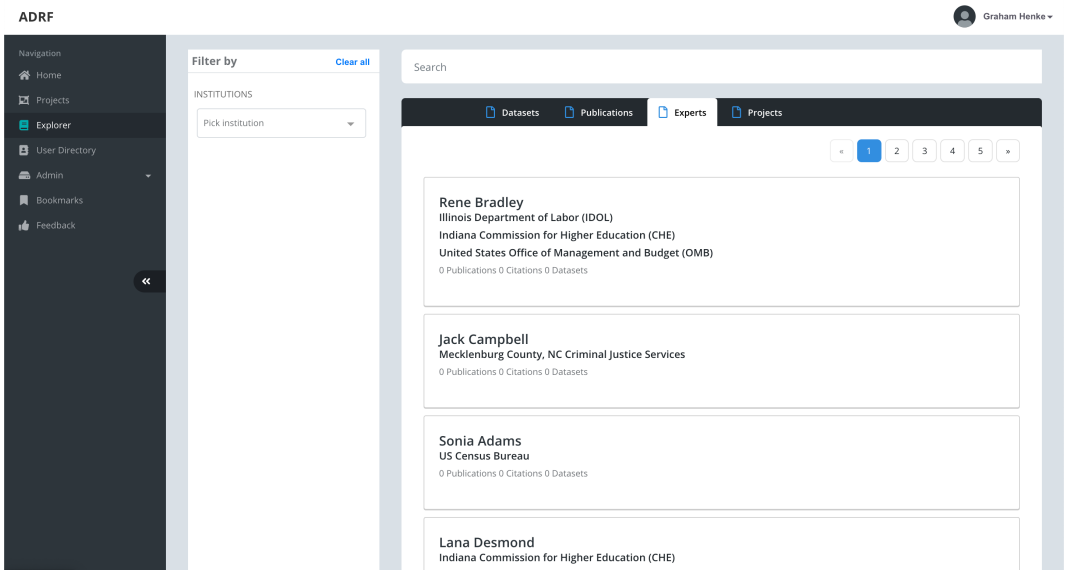

Figure 24: Data Explorer - Expert Tab

Project Tab

This tab is specifically designed for the Applied Data Analytics (ADA) Classes organized by the Coleridge Initiative. If a specific dataset in the ADRF explorer is used in any of the classes, the projects tab will show the title and short description of the class projects. The ADA classes typically have teams working on small research projects during classes. At the end of the classes each group submits a project release form. The title and description of the projects displayed on the projects tab is ingested from these forms.

3.2.4 Bookmarks Page

The Bookmarks page serves as a list of datasets that the user has saved, or "bookmarked", that they may want to easily return to at a future date. This may include datasets that they want to include in a project request in the future.

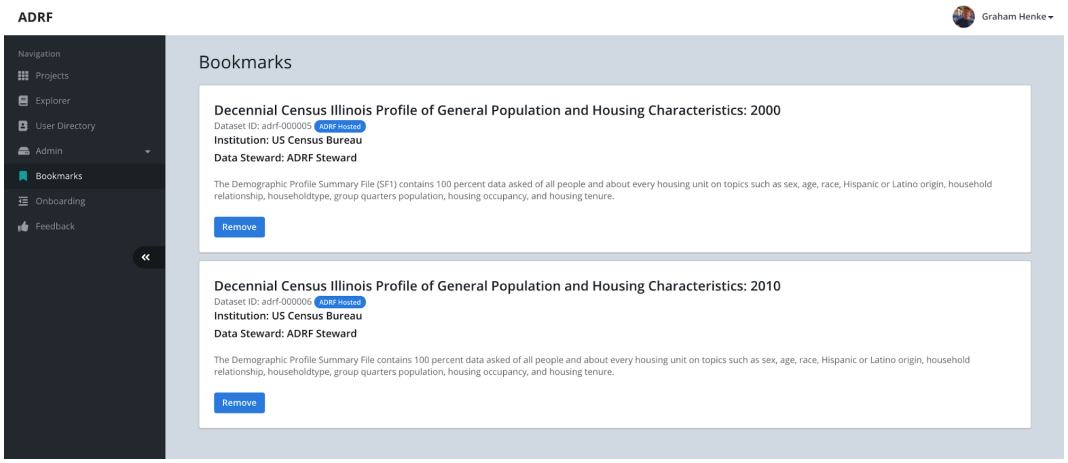

Figure 25: Bookmarks Page

Adding Bookmarks

When a user browses datasets in the Data Explorer and click on the name of the dataset in order to see the detail page, there will be a button at the top of the page which says "Bookmark Dataset". Clicking this button will save the dataset in the Bookmarks section.

The Bookmarks functionality in the current application only support datasets at this time, though in the future it may be desirable to add bookmarks for other entities, such as publications or authors.

3.2.5 Projects Page

The Projects page is where all users, regardless of their role, can see a list of all projects they are associated with. The projects will be grouped based on the following criteria.

- Active Projects: These are projects that the user has requested and that are currently active.
- Pending Approval: These are projects that the user has requested and are awaiting approval.
- Member Projects: These are projects that the user is a member of, but that someone else has requested.
- Data Steward Projects (Data Steward only): These are projects that use datasets associated with a given Data Steward.
- Admin Projects (ADRF Administrator only): These are projects which are at a stage which requires an adminstrative action, like activation of the project in the ADRF.

From this screen, the user can see the status of a their projects and click to see the project details.

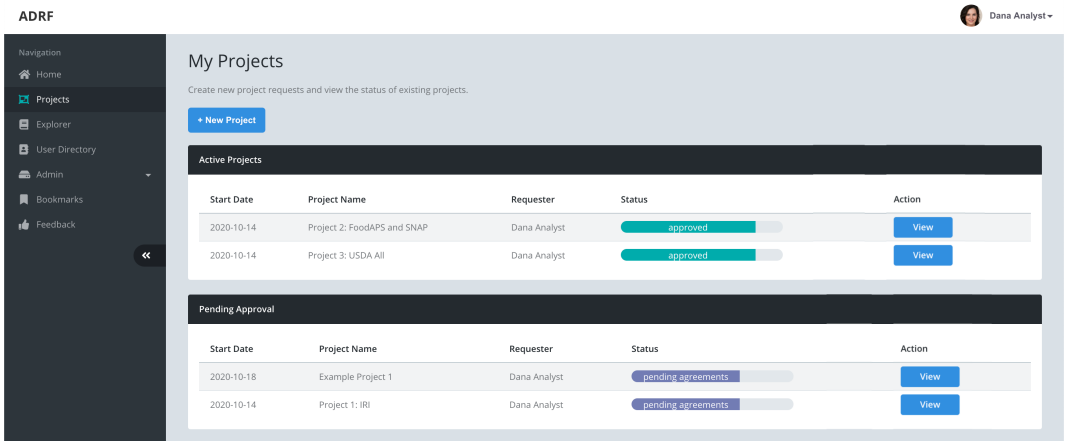

Figure 26: Projects Page

3.2.6 Project Request Page

The Project Request page is where all Data Users will initiate a request to work with datasets as part of a research project. Any Data User who has completed required onboarding can initiate a project request. A project request contains several groups of information, including a project form requesting information such as title, dates, and standard questions. Requests are also composed of project members, datasets, and project agreements. In the example application, these groups are separated into tabs.

Overview Tab

The Overview tab is where basic project information is collected. The standard information that a prospective Data User must provide are the following:

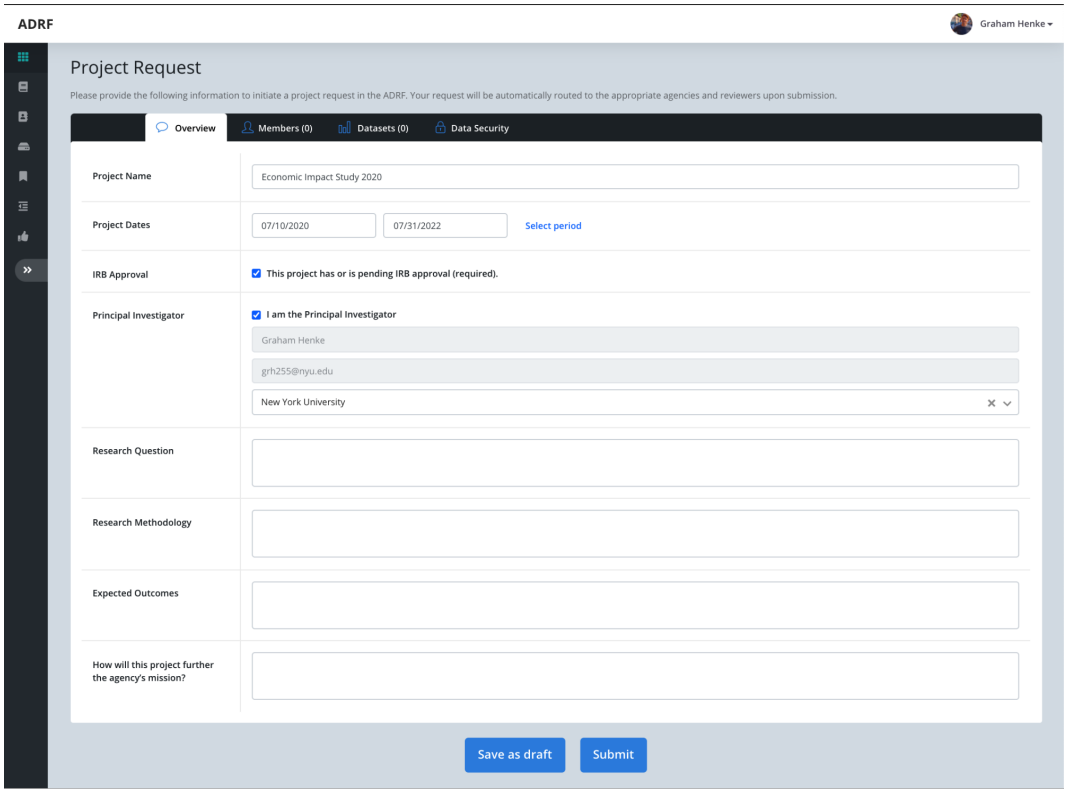

Figure 27: Overview Tab

- Project name
- Project dates (or project period)
- IRB Approval
- Principal Investigator (PI)
- Research Question
- Research Methodology
- Expected Outcomes
- How will the project further the agency's mission

Beyond this basic information, additional inputs may be required by Data Steward institutions whose data is being requested in the project. More details on this are provided in the Institutional Input Tab section.

Members Tab

The Members Tab is where a Data User can choose which other Data Users to add to the research project. Members can be added using the following three methods.

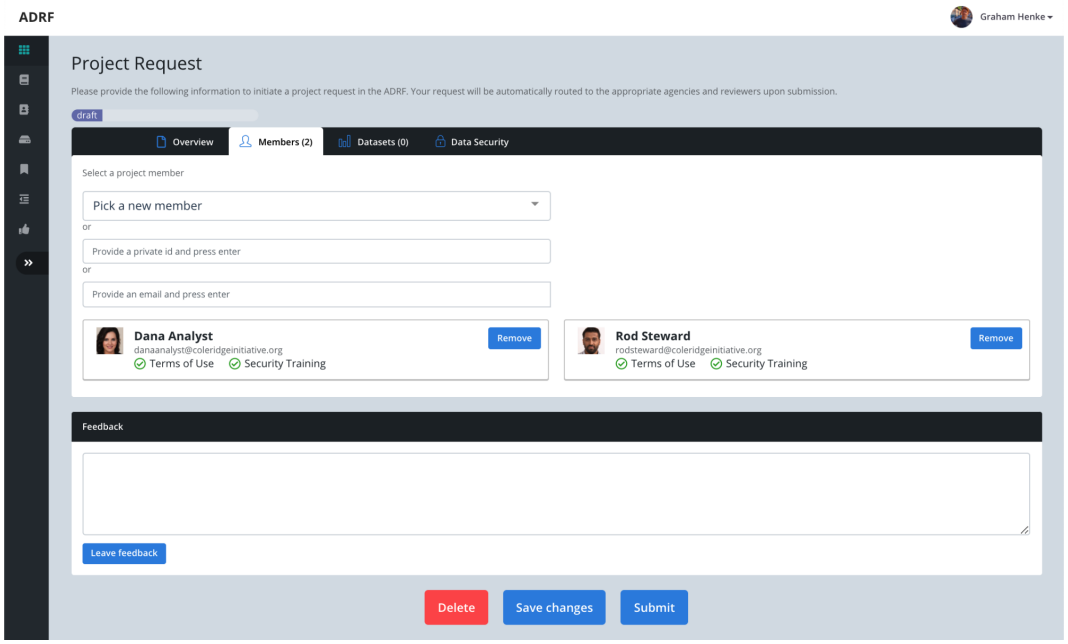

Figure 28: Members Tab

1. Select from a drop down list
2. Provide the private ID for a user
3. Provide an email address

When providing an email address, if the user is already in the system, they will be added to the project. If there is no user associated with that email address, the project requester will be prompted to send the user an invitation to register for the application. A placeholder entry will be shown on the project request in the meantime. Once an invited user has registered and completed onboarding, their full information will be shown on the project request.

Datasets Tab

The Datasets Tab is where a Data User can add datasets to the project request. This is primarily done using a drop-down list. Datasets can also be added to existing project requests from the Data Explorer. This functionality is detailed in the Data Explorer section.

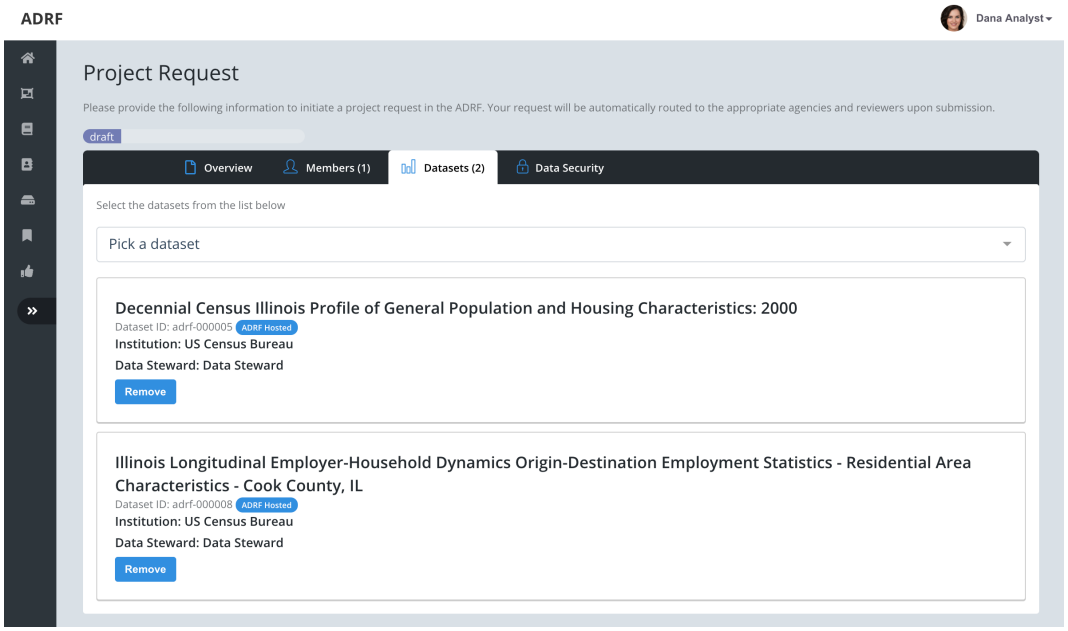

Figure 29: Datasets Tab

Data Security Tab

The Data Security Tab is to provide a reminder of the security and data disclosure policies to the Data Users. This is shown as static text. A future implementation could require inputs from the user, such as confirming a checkbox or signing additional documents; however, in the current applications, these agreements are made during the onboarding process.

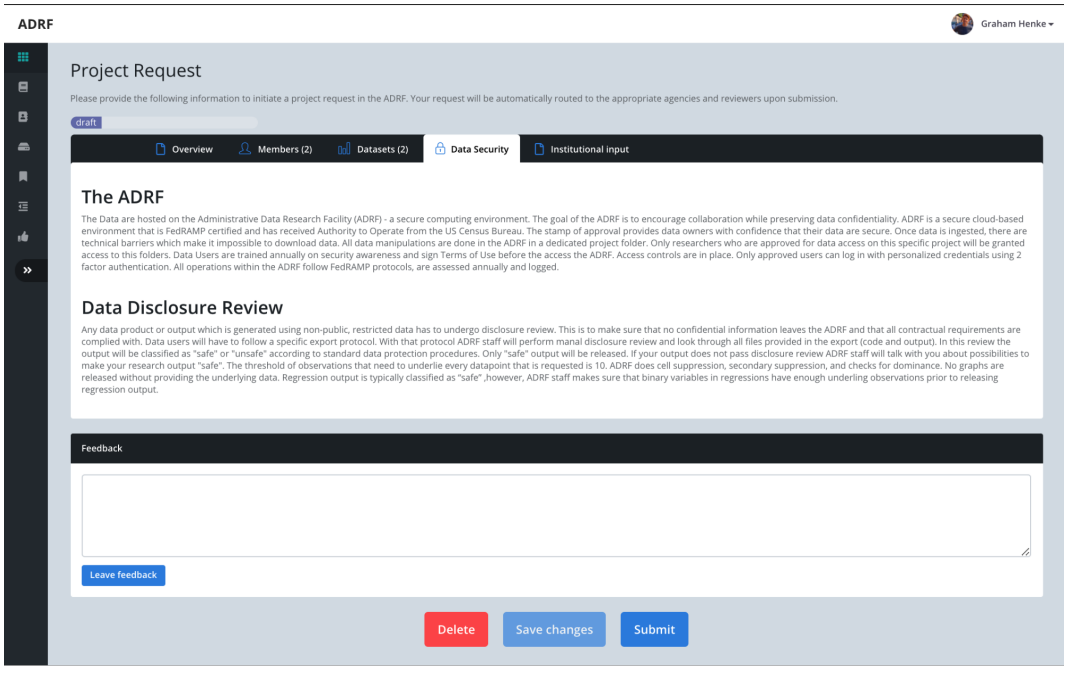

Figure 30: Data Security Tab

Agreements Tab

The Agreements Tab is where project agreements between Data Stewards and Data Users are submitted.

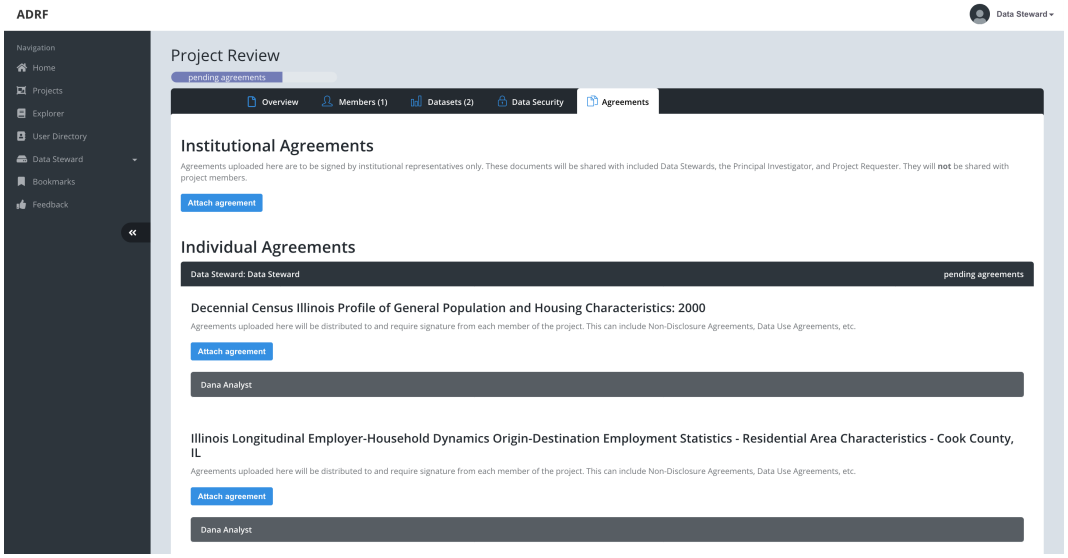

Figure 31: Agreements Tab

The Agreements Tab will display differently and provide different functionality based on the user role which is accessing it.

- Data User: A Data User can only download template agreements, upload signed agreements for themself, and view their own agreements. Data Users can not see other user's agreements and can also not view the institution-level project agreements. The lead data user has a greater number of privileges since they are the one responsible for the project administration. A lead data user has full access in terms of uploading and downloading agreements, both for themselves and on behalf of project members. This means the lead data user can upload template agreements to be made available to the project members, uploaded signed agreements on behalf of project members, as well as view project agreements that the project members themselves have submitted. A lead data user can also upload and view institution-level agreements.
- Data Steward: A Data Steward has the same level of privileges as a lead data user. In addition, the Data Steward is the only one that can give final approval for the project. This is done once the Data Steward has confirmed that any required institutional-level agreements have been received.
- ADRF Administrator: Once a Data Steward has approved a project, the ADRF Administrator can view agreements that have been submitted. This is so the ADRF Administrator can confirm receipt of the agreements and give individual members final approval to access the dataset listed on the project request.

Institutional Input Tab

The Institutional Input tab allows for additional questions to be added by Data Stewards that are not covered on the form in the Overview tab. Data Stewards are able to specify these questions in the Administrative panel of the application (Figure 32). These fields will be available to the Data Steward in the Institutional Input tab. (Figure 33). The workflow for this feature is described in subsection 2.4.2.

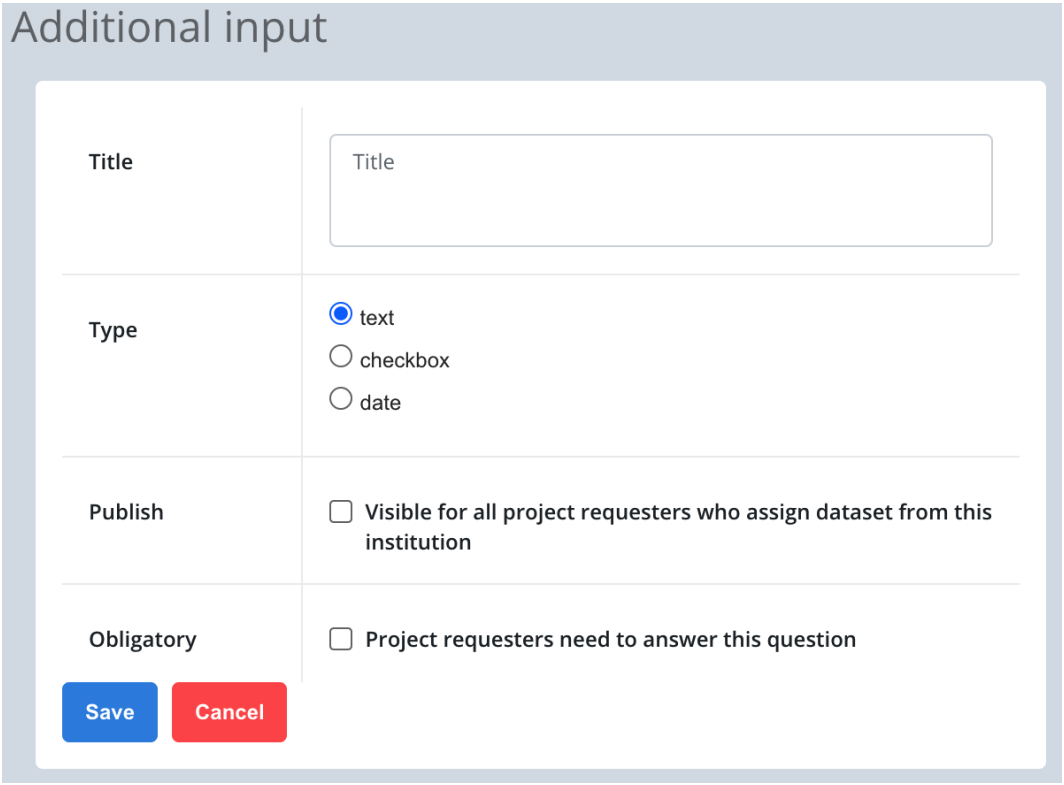

Figure 32: Institutional Input Fields

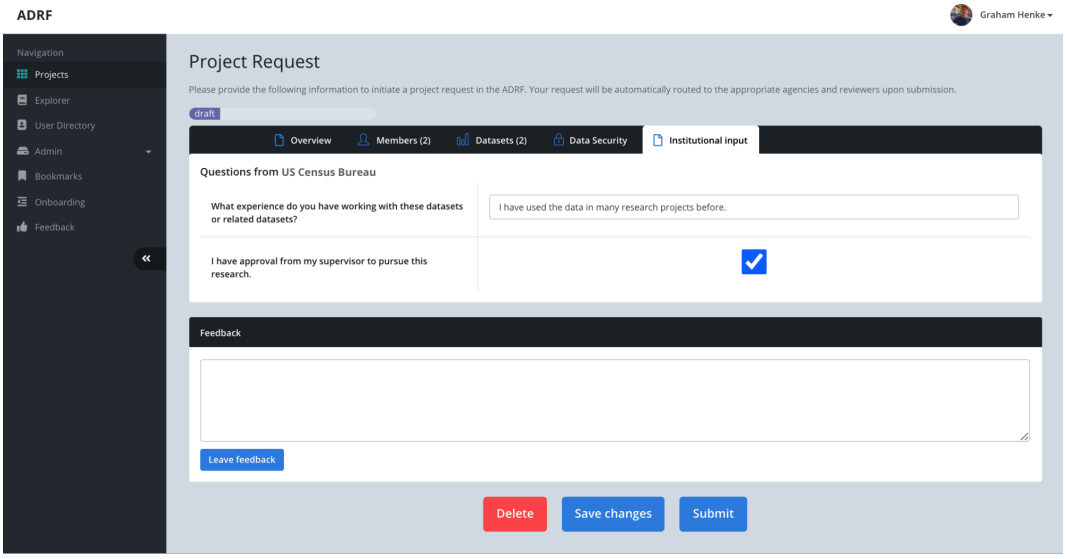

Figure 33: Institutional Input Tab

3.3 Data Steward Features

These features are specifically for the Data Steward role of the web application. The Data Steward has privileged app access, but also has the responsibility to input information.

3.3.1 Dashboard Page

The Data Steward Dashboard page gives a Data Steward a high-level view of the activity involving datasets which they are responsible for.

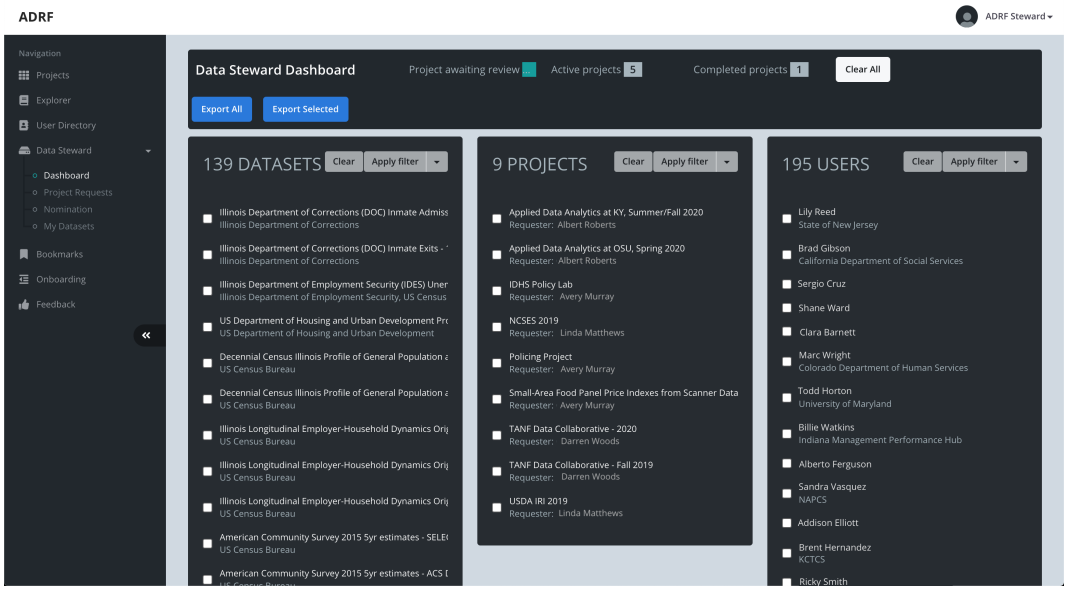

Figure 34: Data Steward Dashboard Page

The dashboard has three columns which contain the following information:

- Datasets: A list of datasets which the data steward has responsibility over, meaning they receive and decide approval of projects which contain these datasets.
- Projects: A list of projects which have used, or are currently using, datasets belonging to the data steward.
- Users: A list of users which have access to the data steward's datasets via projects.

By default, the columns display all entities that are related in some way. Each column contains filtering options which allow the user to see only relationship of selected entities. For example, selecting the first entity in the "Projects" column and then clicking "Apply filter" would then refresh the Datasets column to only show datasets which are part of the selected projects and the Users column would only show members of that project.

At the top of the dashboard, there are counts of the following:

- Projects awaiting review
- Active projects
- Completed projects

Finally, there are two options for exporting information from the dashboard for reporting purposes:

- Export All: Exports all information available in the dashboard to an Excel file.
- Export Selected: Exports only the selected information when a filtered view is applied.

3.3.2 Usage Metrics Page

The Usage Metrics provides the Data Steward with more granular information on how their projects and data are being used. This page give daily aggregated statistics at the individual level of workspace, CPU, memory, database, and file system utilization.

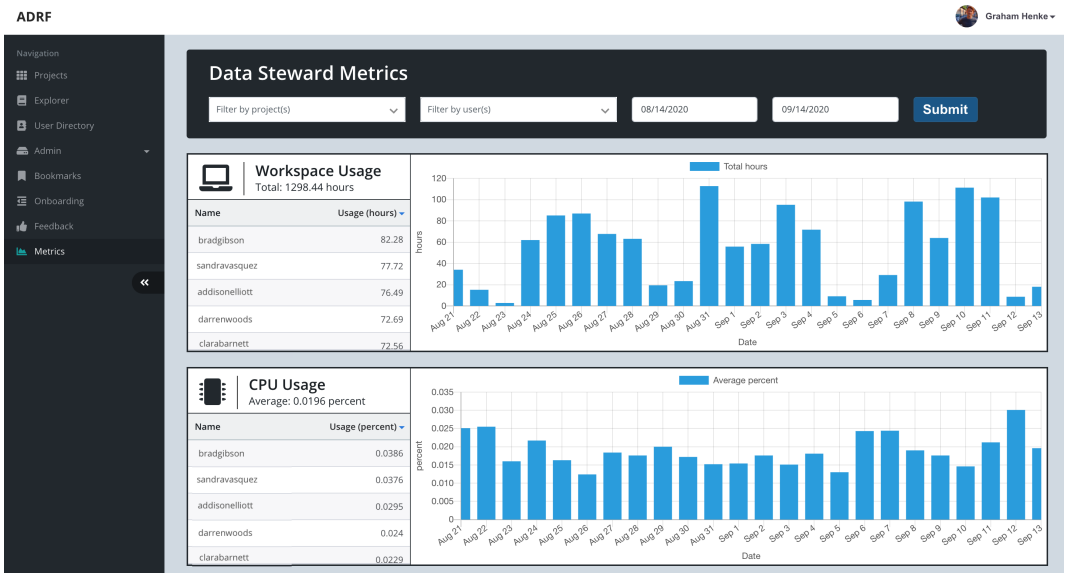

Figure 35: Data Steward Project Requests Page

3.3.3 Project Requests Page

The Project Requests page is the same style of view as shown on the Projects Page (section 3.2.5, Figure 26), expect this view is filter to only show projects that are pending an action from the data steward.

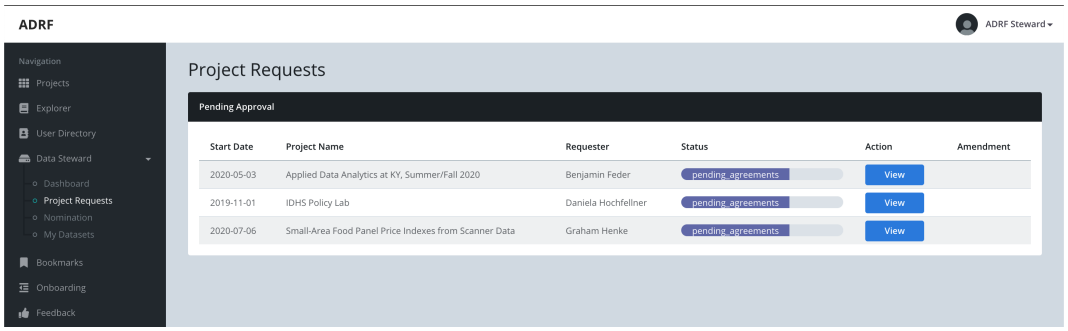

Figure 36: Data Steward Project Requests Page

3.3.4 Nomination Page

The Nomination page is where a data steward can transfer the responsibility of their dataset(s) to another user of the application. This feature is useful if the data steward is changing roles within the organization or leaving the organization. In these cases, the steward will need to "hand-off" their datasets to another member.

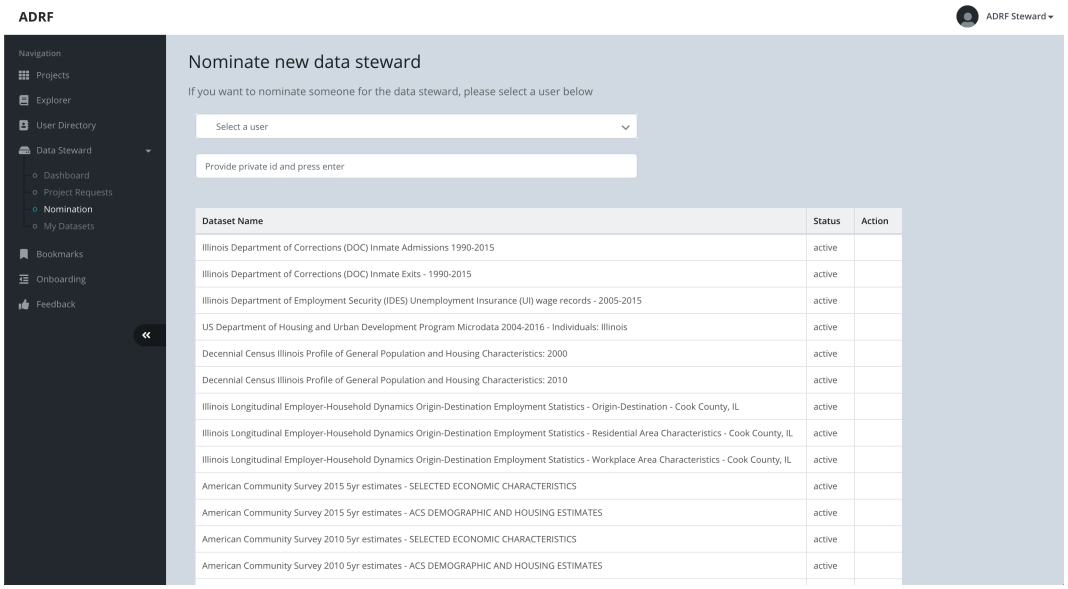

Figure 37: Data Steward Nomination Page

Using the nomination page, the data steward can select a member from the dropdown list, or enter the private id of a user. Then the steward must select the datasets they wish to transfer, then click the "Nominate" button at the bottom.

At this point, the nominated user will get a notification and have an extra menu when they log in to view and accept/reject these nominations. The original Data Steward can also revoke a nomination if they made an error or need to nominate a different user instead.

3.3.5 My Datasets Page

The My Datasets page is an inventory of the datasets which the data steward is responsible for. Additionally, the Data Steward can click the "Edit" button if they need to make changes to the metadata, such as update fields or add data dictionaries.

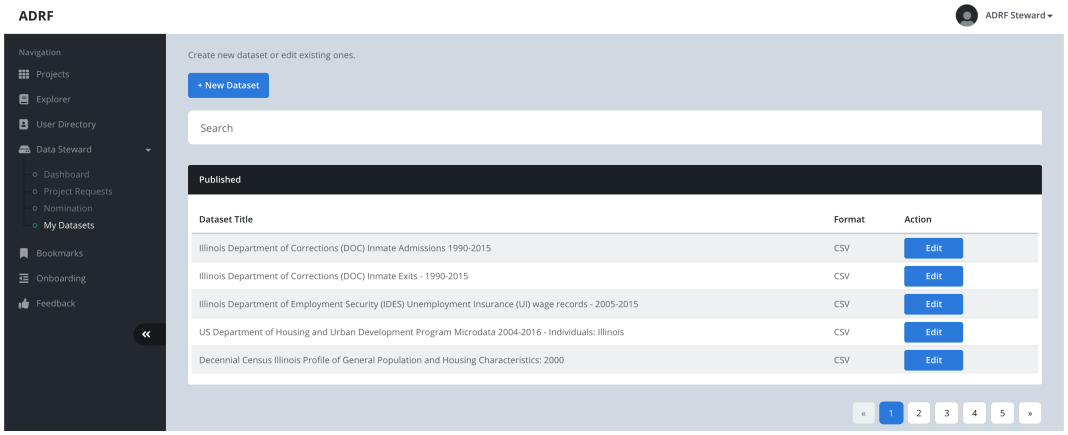

Figure 38: Data Steward Datasets Page

3.4 ADRF Administrator Features

These features are specifically for the ADRF Administrator role of the web application. It will be used to input all agency specific information and fields and customize the application. The ADRF Administrator has privileged app access, but also has the responsibility to input information.

3.4.1 News Input Page

The homepage of the application shows a news box whenever the user logs in. This box can be customized by each agency using the application. The news input page in the administrator panel can be used to enter or change the information that is shown on the homepage of the user.

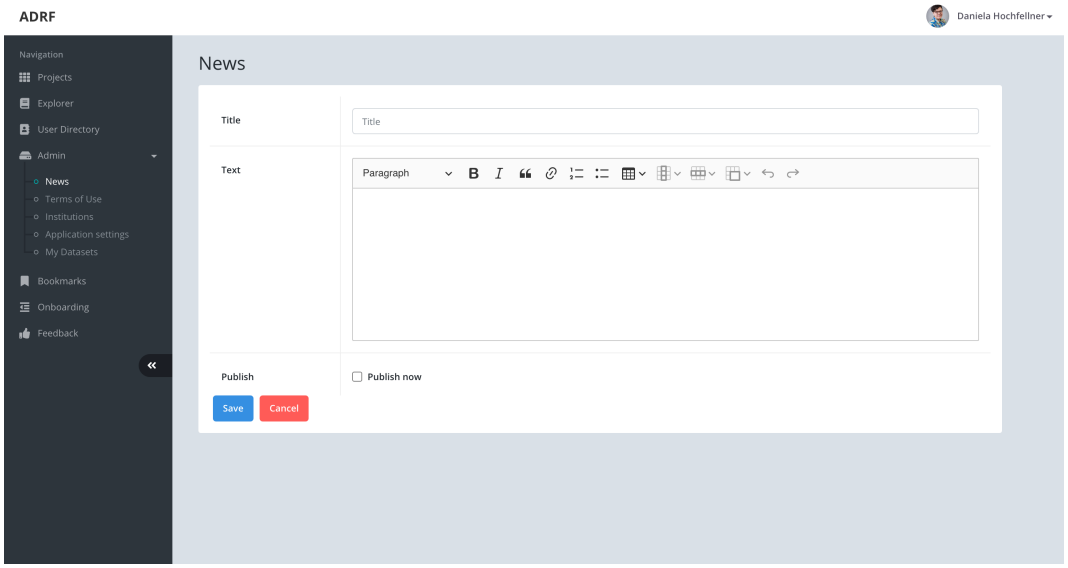

Figure 39: Administrator Page - News Input Page

The administrator can enter a title for the News item they want to post and the actual text by clicking the add news button. The text can be customized, the font size can be changed, it is possible to itemize and add tables. Once the title and text are in the form the administrator can save the news item. If they check the publish now box the text will be populated on the homepage of every users. If they don't want to publish the news item now they can click the save button and the item will be saved in the news input page. This way it is possible to have several news items saved.

3.4.2 Terms of Use Input Page

Every user of the ADRF needs to comply with the terms of use of the ADRF before access to the ADRF is granted. This is a piece of information that is the same for every user, no matter what other data use agreements might be involved in future research projects. The administrator terms of use input panel allows to enter the terms of use and adjust if needed. The text the administrator saves in the provided text field is populated in the onboarding page of the web-application. The text can be formatted and once the save button is clicked it will be visible on the onboarding page. It can happen that the terms of use document changes over time. If so the administrator can revise the text and click the save button again and a new document will be populated on the onboarding page. In addition to that the web-application will also ask every user when they log in the first time after the change of the terms of use to sign the document again. This way every user will have signed the most recent terms of use.

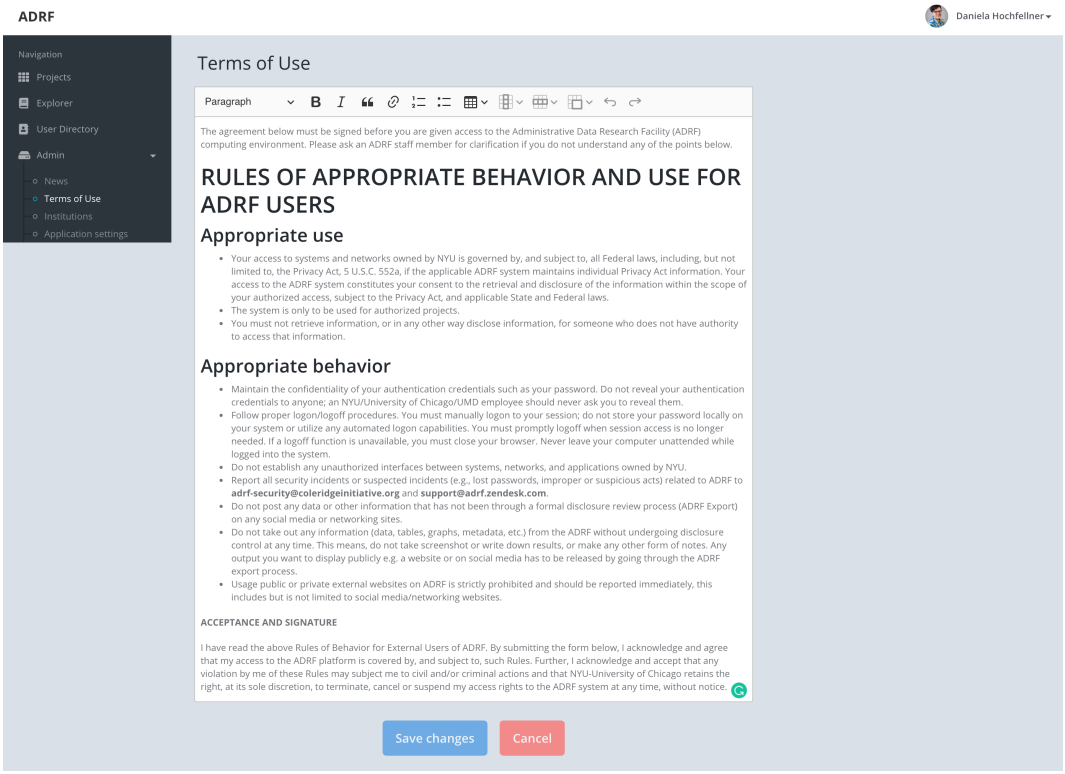

Figure 40: Administrator Page - TOU Input Page

3.4.3 Institution Input Page

The purpose of the institution input page is to have a register of all existing institutions that the users are affiliated with. This is separate from the institution that the user can add on their biographic page. This is a list of institutions that are verified by the administrator. Only the administrator can add institutions to that list. The panel shows institutions that are already in the database and a button that allows the administrator to add a new institution.

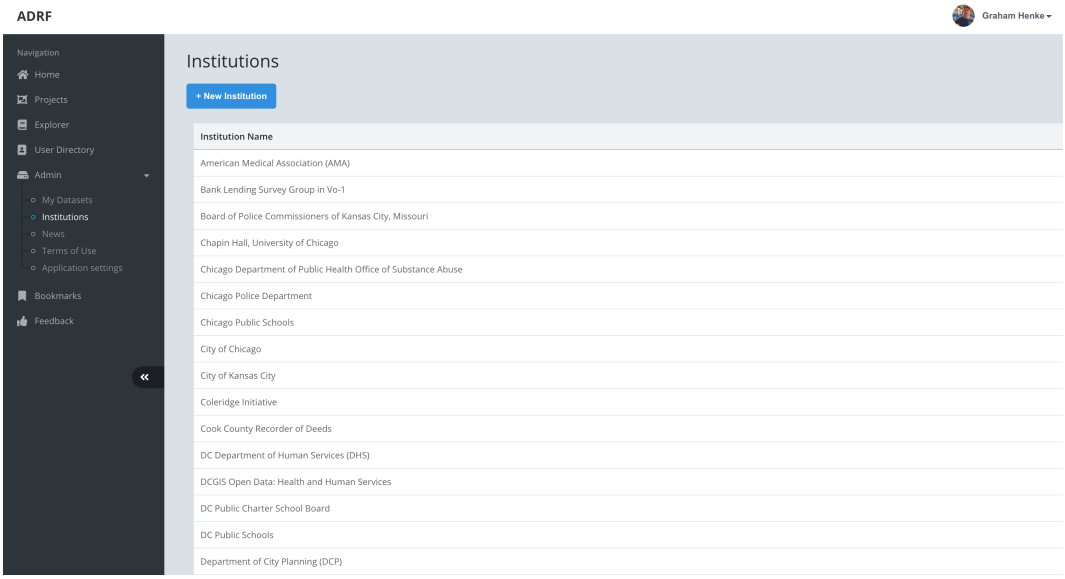

Figure 41: Administrator Page - Institution Input Page

3.4.4 Application Settings Page

The purpose of this page is to make sure that whenever the user interacts with the web-application somebody gets notified. The settings page let's the administrator enter who will be receiving emails from the system for

two events: when a new user signs up, and when the feedback form is filled out. In both cases, the emails listed here will be receiving an update email. It is possible to add new emails, but also edit or delete emails that are displayed.

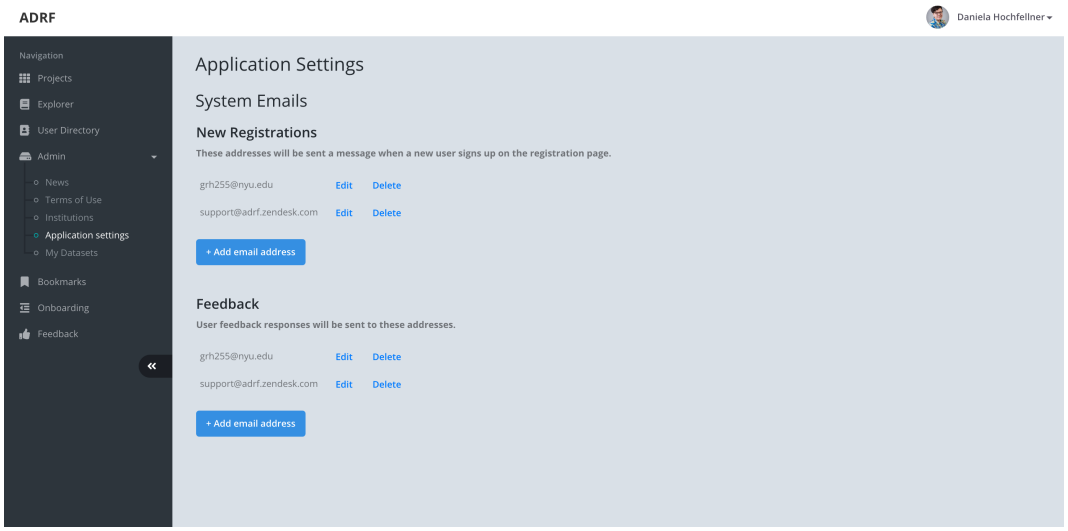

Figure 42: Administrator Page - Application Settings Input Page

3.4.5 My Datasets Page

The ADRF Explorer shows all metadata for datsets ingested into the application. The administrator is able to see all datasets and the corresponding meta data, such as title, format, the data steward and the data provider. The interface can be used by the administrator to either change existing metadata or add metadata once a new dataset is ingested into the ADRF.

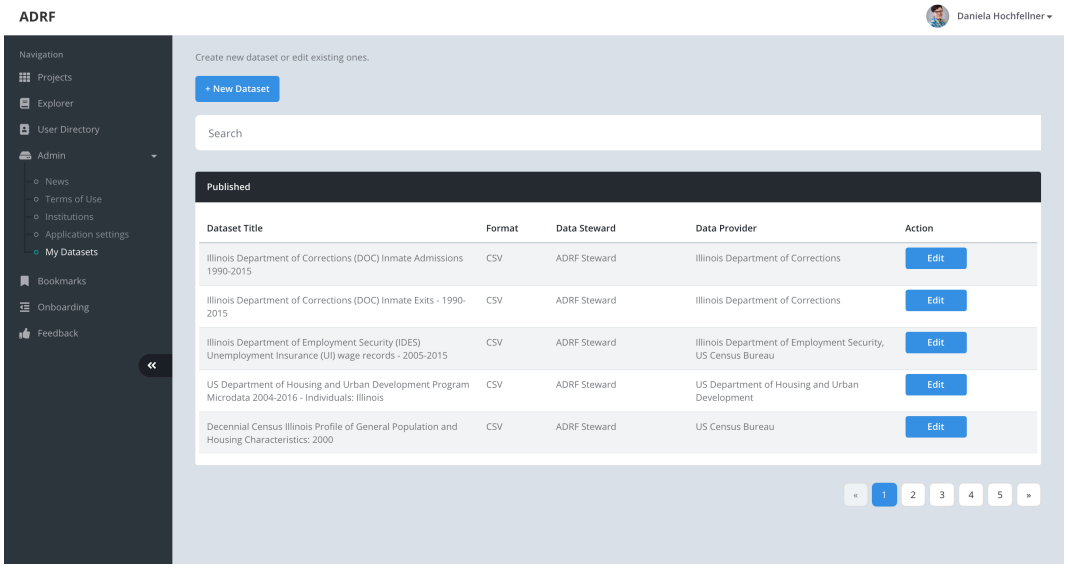

Figure 43: Administrator Page - My Datasets Page

Clicking on the new dataset button will open a form that asks for specific information for the dataset. This is the information that is displayed in the explorer. The administrator is responsible to fill out these forms for each dataset that exists in the ADRF. The form asks for a title, description, citation, format, a category (these categories are predefined) and tags. In addition, the time period and name of data steward and provider are collected.

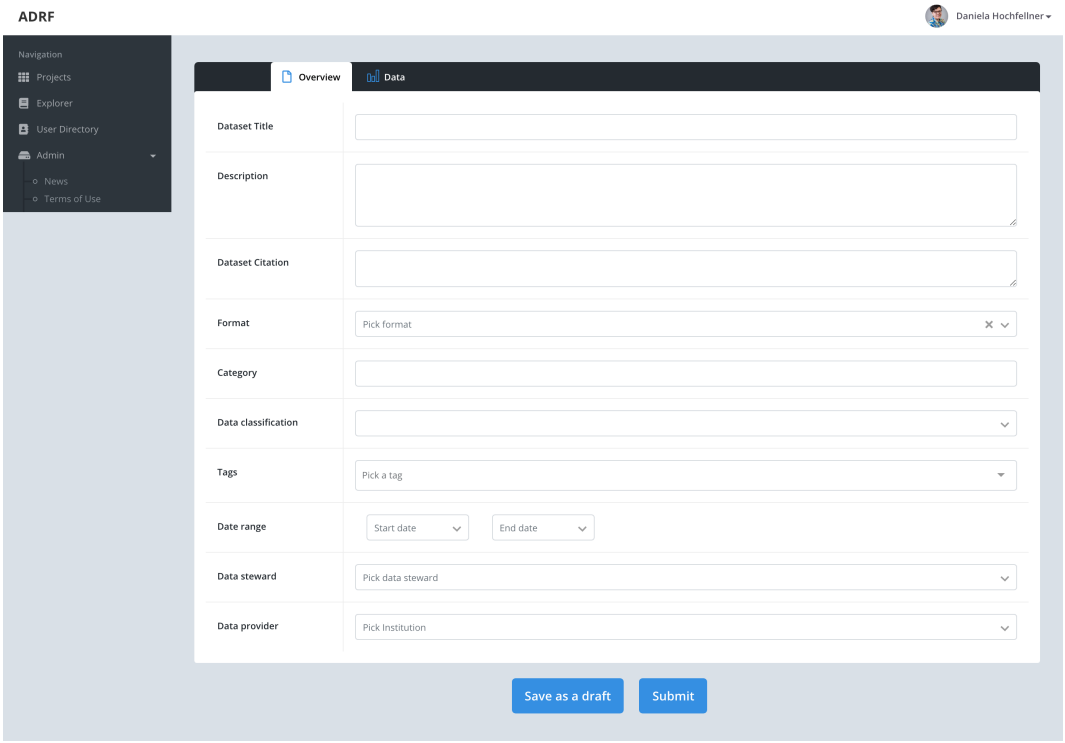

Figure 44: Administrator Page - My Datasets Page: Overview

The data tab can be used to upload the column specifc metadata and add additional files like data manuals.

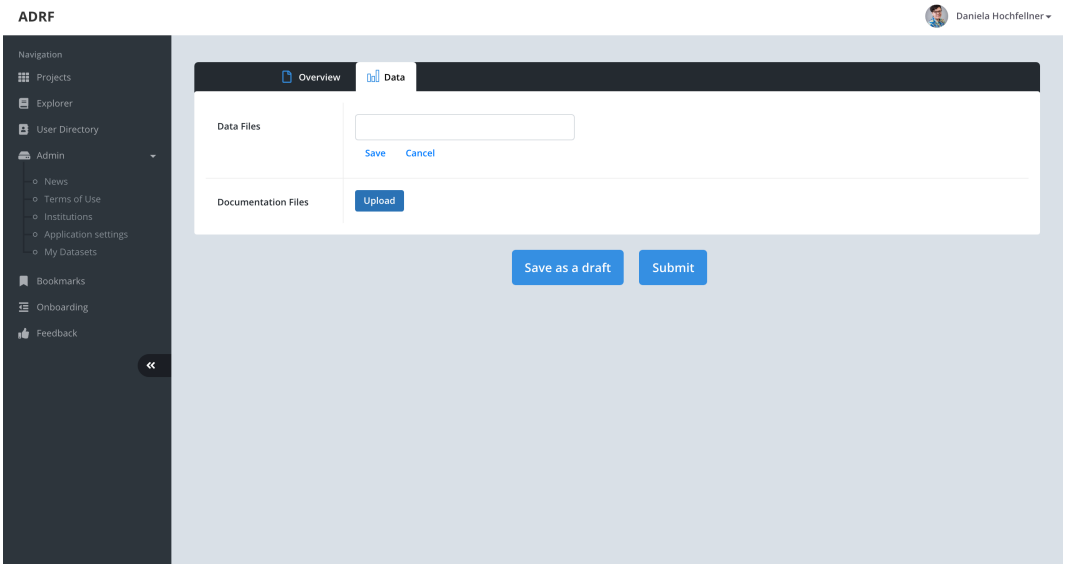

Figure 45: Administrator Page - My Datasets Page: Data

The easiest way to upload column level metadata is adding it in json format as the format that is used to store metadata in the application is json. For example, in the data explorer it will look as outlined below. Each json file includes meta-data of a specific dataset. There is a required minimum information that needs to be provided by the agency.

```
[
{
  "model": "ds.dataset",
  "pk": 80,
  "fields": {
    "temporal_coverage_end": "2012",
```

```
    "files_total": 1,
    "data_classification": "Public",
    "access_actions_required": "No further actions required to access this dataset",
    "geographical_coverage": [
      "United States"
    ],
    "keywords": [
      32,
      33,
      34,
      35
    ],
    "dataset_id": "adrf-000080",
    "category": 11,
    "dataset_version": 1,
    "title": "2012 North American Industry Classification System (NAICS) Definitions - 2 to 6 Digit codes",
    "data_usage_policy": "This dataset is intended for public access and use.",
    "data_steward_organization": "The Coleridge Initiative",
    "data_steward": 1,
    "files": [
      {
        "file_name": "adrf-000080-naics_2012.csv",
        "columns_metadata": {
          "naics_us_title": {
            "provided-type": "text",
            "description": "2012 NAICS Title.",
            "categories": [],
            "categorical": false
          },
          "seq_no": {
            "provided-type": "integer",
            "description": "Sequence Number",
            "categories": [],
            "categorical": false
          },
          "naics_us_code": {
            "provided-type": "text",
            "description": "2012 NAICS Code",
            "categories": [],
            "categorical": false
          }
        }
      }
    ],
    "access_requirements": "No access restrictions on this dataset",
    "description": "The North American Industry Classification System (NAICS) is the standard used by Federal statistical agencies in classifying business establishments for the purpose of collecting, analyzing, and publishing statistical data related to the U.S. business economy.",
    "source_url": "https://www.census.gov/eos/www/naics/2012NAICS/2-digit_2012_Codes.xls",
    "geographical_unit": [
      ""
    ],
    "related_articles": [],
    "institution": 6,
    "dataset_documentation": [
      "naics_2012_Definition_File.pdf"
    ],
```

```
    "dataset_version_date": 1526311715,
    "source_archive": "US Census Bureau",
    "dataset_citation": "United States Office of Management and Budget (OMB), 2012, \"2
        012 North American Industry Classification System (NAICS) Definitions - 2 to 6
        Digit codes\", https://www.census.gov/eos/www/naics/2012NAICS/2-digit_2012
        _Codes.xls, US Census Bureau [Distributor], 1 [Version]",
    "reference_url": "https://www.census.gov/eos/www/naics/",
    "temporal_coverage_start": "2012",
    "file_names": [
      "adrf-000080-naics_2012.csv"
    ],
    "created_at": "2019-04-04T14:41:26-04:00",
    "updated_at": "2019-04-04T14:41:26-04:00"
  }
}
]
```


3.5 General Features

There are some additional features of the application which are not specific to a user group and may be generally useful to multiple user groups. These are described here.

3.5.1 Home Page

This section of the report describes in detail all functional requirements of the Welcome page of the Data Stewardship Tool, which is the page that a user lands on when logging in to the web application. The goal of this page is to provide an overview for a Data User on what their current project status is and if there are any actions that the user has to take on any of the workflows being involved. The left navigation panel can be used to navigate to different pages in the application. Clicking the profile button in the upper right corner will lead the user to their biographic page. Figure 46 illustrates how the welcome page looks like.

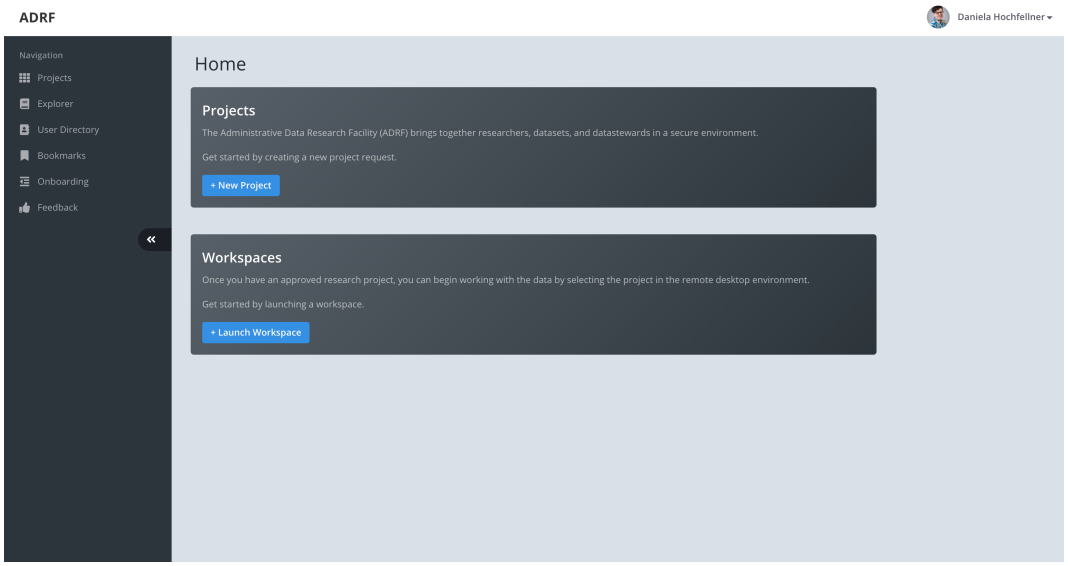

Figure 46: user home page

In addition to these two boxes there is the option to display a news box on the homepage as well. This can be set up by the administrator. For example, system downtimes can be displayed to inform users about upcoming situations that affect them.

3.5.2 User Directory Page

The data explorer page is where all Data Users can find other Data Users that have an active project on the ADRF. The view is slightly different for the administrator. Every Data User can see a list of all other Data

Users whose account is set to not private.

The list consists of multiple tiles: each tile represents a Data User. The tiles list the user name, show a profile picture if the user has uploaded one on their biographic page, as well as the current state of the annual security awareness training and the terms of use.The tile will show a green check mark if the user has completed these elements and a red cross if the user didn't. The Data Administrator and the Data Steward will see an additional row underneath the search bar that provides information about both active and inactive users. The Administrator and data steward can see private users for monitoring and approval purposes.

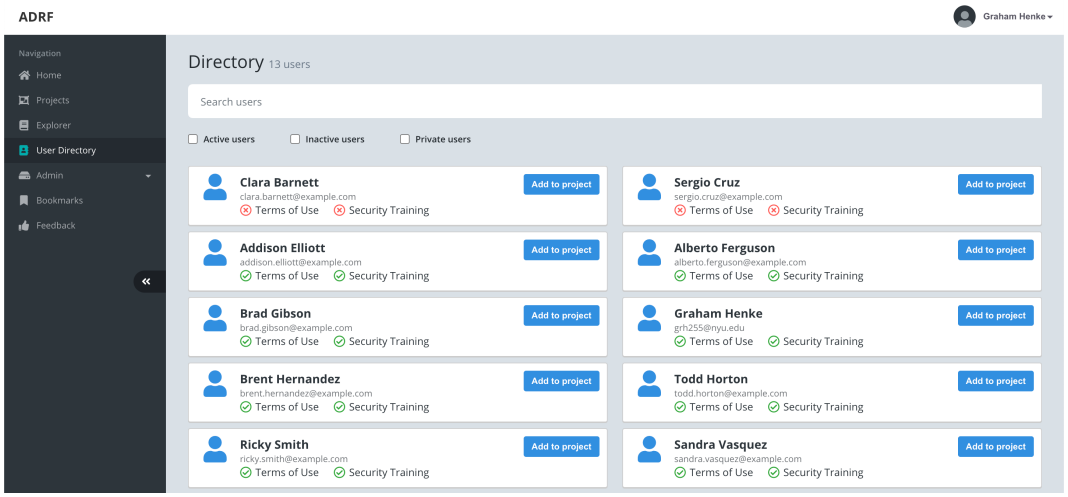

Figure 47: User Directory

The search bar helps to find users by name, and has an autocomplete functionality. Data Users can, for example, use the directory to find their co-authors and then click on the add to project button if they wish to request that the co-author be added to a new or existing project.

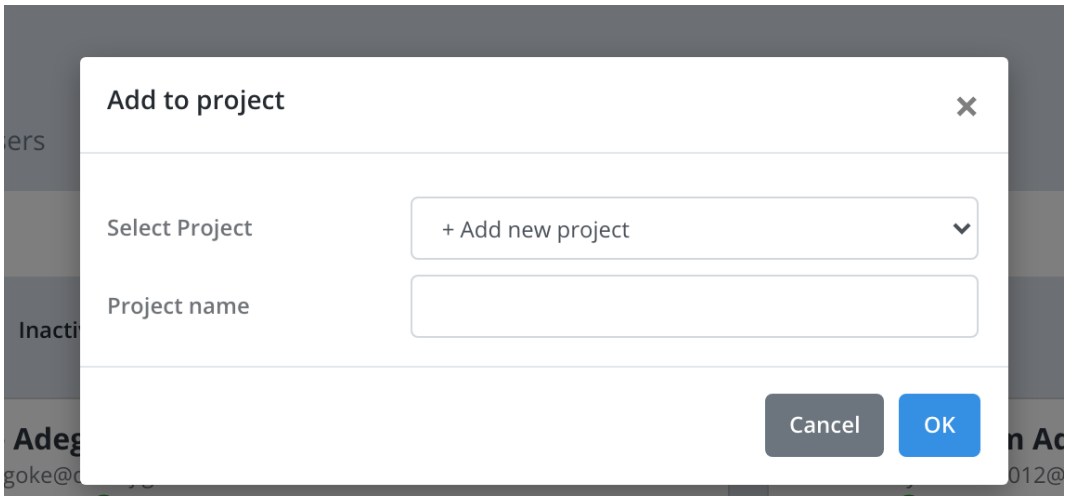

Figure 48: User Directory - Add New User

3.5.3 Feedback Page

The Feedback page allows the application users to communicate back to the administrators and owners of the app. The feedback mechanism also allows for attaching a screenshot in case the user needs to illustrate the issue or comment they have described. Once the comment is submitted, it will be emailed to a list of recipients which are saved in the Application Settings in the Admin menu.

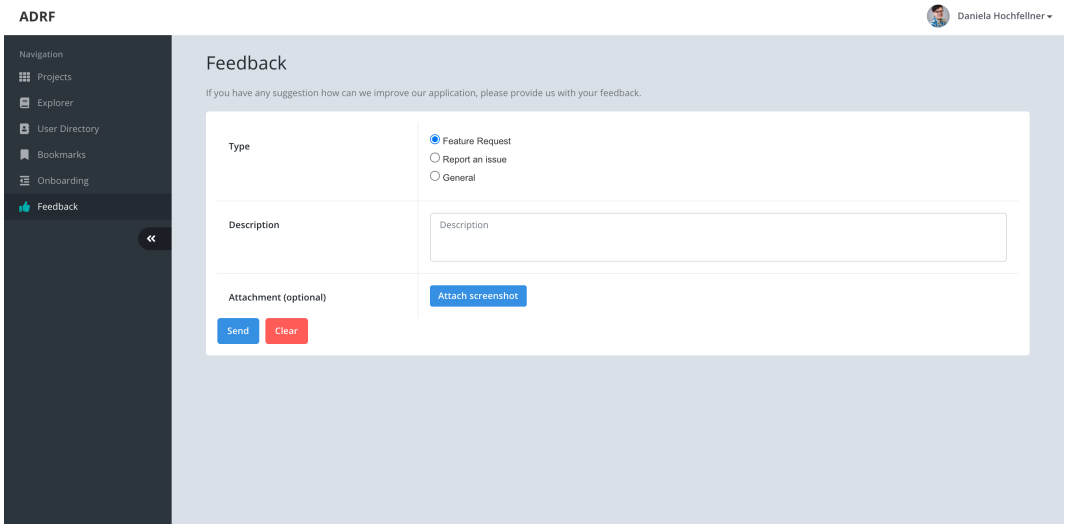

Figure 49: Feedback Page

There are three categories, or types, of feedback which can be provided.

Feature Request

The "Feature Request" type is for users to submit feature requests, which would be additional functionality that they might find useful or an improvement on an existing feature or process.

Report an Issue

The "Report an issue" type should be used if a user identifies an error or bug in the application. This way the message can be passed on to the support team, and then, if necessary, the developers who maintain the application.

General

The "General" type is a catch-all type which allows the user to submit general comments. Perhaps they would like to give a compliment, or have a general suggestion that isn't a specific feature request.

4

Technical Requirements

4.1 Application layers

While the Data Stewardship application is generally thought of as a single application, it is helpful to divide the application into 4 conceptual layers. First, there is the database layer, where all of the application data and relationships are stored. Second, there is the file storage, where items like agreements and CVs are stored. Third, there is the connecting layer, which is the API layer. Finally, there is the front-end layer, where data from the previous layers is presented to the end user.

4.1.1 Database

The database layer is the most fundamental and important layer of the application. This is where all of the application data and relationships are stored, such as information about users, projects, and datasets.

Implementations: While any database technology could theoretically be used, since the data in the application is highly relational, it is recommended to use a Relationship Database Manage System (RDBMS). There are many options in this category. Some well-known and supported choices include MySQL, Oracle, IBM Db2, and PostgreSQL. The example implementation referred to throughout this document uses PostgreSQL, also known as Postgres. Postgres was chosen because it is open-source and has over 30 years of active development, making it a robust, well-supported technology.

4.1.2 File Storage

The file storage layer is used for storing documents and files that are better suited for a traditional file system or object storage, such as PDFs and image files.

Implementations: Possible solutions for file storage include network file system (NFS), cloud storage services from providers such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP), or simply using the file system on the web server running the application. Our example implementation uses Amazon S3 (Simple Storage Service) because it is affordable, reliable, easily accessible, and provides a number of nice built-in features such as file encryption and versioning.

4.1.3 API

The API (application program interface) layer is used to communicate between the backend layers (database and file storage) and the front-end interface(s).

Implementations: As with the other applications, there are several choices when it comes to API architecture and design. The most commonly used is REST (Representational State Transfer) and using this will ensure that just about any developer can work with it.

However, the example implementation chosen for this application is GraphQL, which is a newer technology. This was chosen because the highly relational nature of data lends itself well to GraphQL. For example, retrieving all of the members and datasets of a given project may take several calls with a REST implementation, this can easily be done in a single request with GraphQL.

4.1.4 Front-end

The Front-end layer is the UI (User Interface) layer and what most people think of when they think of the application.

Implementations: Again, there are many options here. Although it's possible to build a native application for Windows, Mac, or Linux, it is recommended to use a web framework to build a web-based front-end. This will ensure that the application can run from any system and be accessed from any location without the need for the user to install any special software on their device.

Popular front-end frameworks and libraries include Angular, React, and Vue. Our example uses Vue as it is a lightweight, easy to learn, and requires less code to build simple applications than something like React.

4.2 System Requirements

The system requirements for running the application will depend largely on the number of users expected to be using the application. There may also be some considerations that will need to be made regarding the computing infrastructure available to host the application, for example, whether it runs on servers that or on-premises vs cloud-based hosting.

The requirements listed below are for the example implementation, which is cloud-based and supports 50 concurrent users. Please consult with your IT department to seek their recommendations on hosting.

4.2.1 Hardware Requirements

The following hardware requirements are sufficient to support a basic installation.

- Two 2.3 GHz Intel Xeon processors
- 2GB RAM
- 10GB file storage
- 20GB database storage

4.2.2 Software Requirements

The software requirements are largely dependent on the chosen implementation. For our example implementation, the core requirements are:

- Linux OS - Ubuntu 16.04 or higher
- Apache or nginx web server
- Python 3.7 or higher
- Django 2.2 or higher

In addition, there are a number of Python packages which are listed in Appendix D. It is also recommended that the software is written to be containerized using Docker. This will allow all software dependencies to be bundled into one image and only require that the host machine has Docker Engine installed, configured, and running.

4.3 Identity Management

Identity is managed outside of the application by an external service. Our example relies on Keycloak, which is an open-source identity and access management application which is self-hosted. However, the application can be easily integrated with any identity provided that support the OAuth2 protocol, or modified to work with other authentication standards.

4.4 Security

Security is critically important for any application, especially one that manages who has access to restricted datasets. There are a number of measures that can be taken to ensure security and these are all used in the example implementation:

- Encrypted SSL web traffic: All requests to the application should only be made and allowed over https protocol.

- Multi-factor authentication: The identity management system used should require multi-factor authentication to safeguard against weak or compromised passwords.
- Token-based access: All request made to the API should require an authorization token be included in the request header to verify that the user is authorized to make the request.
- Encrypted storage: All file storage and database storage storage should be encrypted at rest.

4.5 Versioning

Software versioning is a decision that is left to the software developer, but it is recommended to use a scheme that is described in the Wikipedia page on Software Versioning, where there is a major, minor, and patch number.

It can be assumed that the example implementation described within this documentation is version 1.0.0.

4.6 Logging

There are a number of reasons why keeping a record of activity in the application may be useful, such as for debugging, auditing, security, or analysis. Most common languages like Java, C++, and Python, have built-in support for logging.

Our example application has not yet implemented logging. We plan to add this in the future by using the logging module built into Python.

4.7 Accessibility

It is important that the application is accessible for all users, including those that may have disabilities. As such, as content should conform to the guidelines set forth in the latest Web Content Accessibility Guidelines (WCAG).

To check for compliance, a scanning tool such as WAVE can be used to find common errors.

5

Data Model, Attributes, and Dependencies

5.1 Database Schema

The data model is displayed in Figure 50. It contains all the information that is collected in the application and stored in a Postgres database. The arrows indication the relations between the different tables.

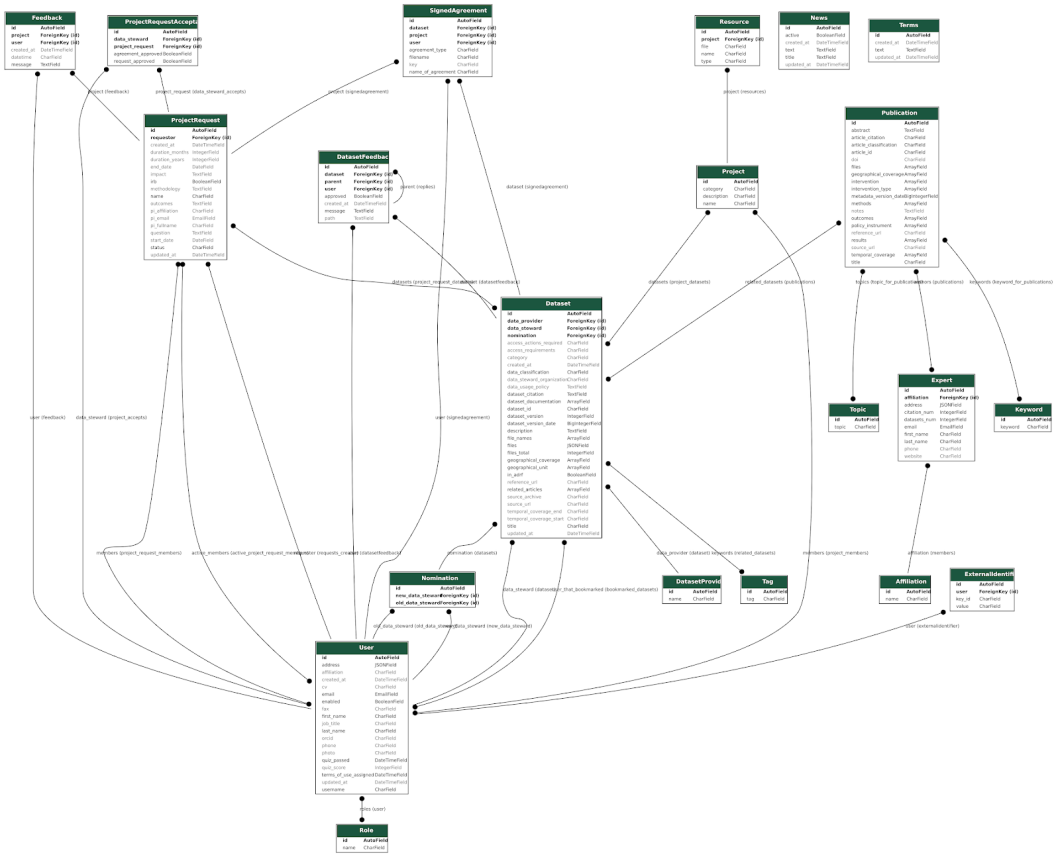

Figure 50: Data Model of the Data Stewardship Application

The database can be queried via an API (GraphQL). All functionalities of the applications and actions are defined by queries.

5.2 Database Tables

5.2.1 ds_amendment

This section provides a more detailed overview on all the tables and their content that can be found in the database of the web application.

Column	Type	Default	Nullable
id	integer	nextval('ds_amendment_id_seq'::regclass)	NO
status	character varying(255)	None	NO
start_date	date	None	YES
end_date	date	None	YES
duration_years	integer	None	YES
duration_months	integer	None	YES
reason	text	None	NO
project_request_id	integer	None	YES

5.2.2 ds_amendment_datasets

Column	Type	Default	Nullable
id	integer	nextval('ds_amendment_datasets_id_seq'::regclass)	NO
amendment_id	integer	None	NO
dataset_id	integer	None	NO

5.2.3 ds_amendmentacceptance

Column	Type	Default	Nullable
id	integer	nextval('ds_amendmentacceptance_id_seq'::regclass)	NO
amendment_approved	boolean	None	YES
amendment_id	integer	None	YES
data_steward_id	integer	None	YES

5.2.4 ds_amendmentmembership

Column	Type	Default	Nullable
id	integer	nextval('ds_amendmentmembership_id_seq'::regclass)	NO
is_approved	boolean	None	NO
member_id	integer	None	NO
project_request_id	integer	None	NO

5.2.5 ds_category

Column	Type	Default	Nullable
id	integer	nextval('ds_category_id_seq'::regclass)	NO
category	character varying(255)	None	NO

5.2.6 ds_dataset

Column	Type	Default	Nullable
id	integer	nextval('ds_dataset_id_seq'::regclass)	NO
temporal_coverage_start	character varying(10)	None	NO
temporal_coverage_end	character varying(10)	None	NO
files_total	integer	None	YES
data_classification	character varying(255)	None	YES
access_actions_required	character varying(255)	None	NO
geographical_coverage	ARRAY	None	YES
dataset_id	character varying(255)	None	NO
category_id	integer	None	YES
dataset_version	integer	None	YES
title	character varying(255)	None	NO
data_usage_policy	text	None	NO
data_steward_organization	character varying(255)	None	NO
files	jsonb	None	YES
access_requirements	character varying(255)	None	NO
description	text	None	NO
source_url	character varying(255)	None	NO
geographical_unit	ARRAY	None	YES
related_articles	ARRAY	None	YES
dataset_documentation	ARRAY	None	YES
dataset_version_date	bigint	None	NO
source_archive	character varying(255)	None	NO
dataset_citation	text	None	NO
reference_url	character varying(255)	None	NO
file_names	ARRAY	None	YES
created_at	timestamp with time zone	None	NO
updated_at	timestamp with time zone	None	NO
data_steward_id	integer	None	YES
in_adrf	boolean	None	NO
nomination_id	integer	None	YES
institution_id	integer	None	YES
dataset_family	character varying(255)	None	YES
file_format	character varying(255)	None	YES
is_published	boolean	None	NO

5.2.7 ds_dataset_keywords

Column	Type	Default	Nullable
id	integer	nextval('ds_dataset_keywords_id_seq'::regclass)	NO
dataset_id	integer	None	NO
tag_id	integer	None	NO

Column	Type	Default	Nullable

5.2.8 ds_dataset_user_that_bookmarked

Column	Type	Default	Nullable
id	integer	nextval('ds_dataset_user_that_bookmarked_id_seq'::regclass)	NO
dataset_id	integer	None	NO
user_id	integer	None	NO

5.2.9 ds_datasetfeedback

Column	Type	Default	Nullable
id	integer	nextval('ds_datasetfeedback_id_seq'::regclass)	NO
created_at	timestamp with time zone	None	NO
message	text	None	NO
approved	boolean	None	NO
path	text	None	YES
dataset_id	integer	None	YES
parent_id	integer	None	YES
user_id	integer	None	YES

5.2.10 ds_emailshouldnotifyfeedback

Column	Type	Default	Nullable
id	integer	nextval('ds_emailshouldnotifyfeedback_id_seq'::regclass)	NO
email	character varying(255)	None	NO

5.2.11 ds_emailshouldnotifyregistrations

Column	Type	Default	Nullable
id	integer	nextval('ds_emailshouldnotifyregistrations_id_seq'::regclass)	NO
email	character varying(255)	None	NO

5.2.12 ds_expert

Column	Type	Default	Nullable
id	integer	nextval('ds_expert_id_seq'::regclass)	NO
first_name	character varying(255)	None	NO
last_name	character varying(255)	None	NO
email	character varying(254)	None	NO
website	character varying(255)	None	NO

Column	Type	Default	Nullable
address	jsonb	None	YES
phone	character varying(255)	None	NO
datasets_num	integer	None	NO
citation_num	integer	None	NO

5.2.13 ds_expert_departments

Column	Type	Default	Nullable
id	integer	nextval('ds_expert_departments_id_seq'::regclass)	NO
expert_id	integer	None	NO
institution_id	integer	None	NO

5.2.14 ds_expert_institutions

Column	Type	Default	Nullable
id	integer	nextval('ds_expert_institutions_id_seq'::regclass)	NO
expert_id	integer	None	NO
institution_id	integer	None	NO

5.2.15 ds_externalidentifier

Column	Type	Default	Nullable
id	integer	nextval('ds_externalidentifier_id_seq'::regclass)	NO
key_id	character varying(255)	None	NO
value	character varying(255)	None	NO
user_id	integer	None	YES

5.2.16 ds_feedback

Column	Type	Default	Nullable
id	integer	nextval('ds_feedback_id_seq'::regclass)	NO
created_at	timestamp with time zone	None	NO
message	text	None	NO
datetime	character varying(255)	None	YES
project_id	integer	None	YES
user_id	integer	None	YES

5.2.17 ds_input

Column	Type	Default	Nullable
id	integer	nextval('ds_input_id_seq'::regclass)	NO
title	text	None	NO
type	text	None	NO
is_obligatory	boolean	None	NO
is_visible	boolean	None	NO
created_at	timestamp with time zone	None	NO
institution_id	integer	None	YES

5.2.18 ds_institution

Column	Type	Default	Nullable
id	integer	nextval('ds_institution_id_seq'::regclass)	NO
name	character varying(255)	None	NO
email	character varying(254)	None	NO
address	jsonb	None	YES
phone	character varying(255)	None	NO
fax	character varying(255)	None	NO
head_unit_id	integer	None	YES

ds_keyword

Column	Type	Default	Nullable
id	integer	nextval('ds_keyword_id_seq'::regclass)	NO
keyword	character varying(255)	None	NO

5.2.19 ds_news

Column	Type	Default	Nullable
id	integer	nextval('ds_news_id_seq'::regclass)	NO
title	text	None	NO
text	text	None	NO
active	boolean	None	NO
created_at	timestamp with time zone	None	NO
updated_at	timestamp with time zone	None	NO

5.2.20 ds_nomination

Column	Type	Default	Nullable
id	integer	nextval('ds_nomination_id_seq'::regclass)	NO
new_data_steward_id	integer	None	YES

Column	Type	Default	Nullable
old_data_steward_id	integer	None	YES

5.2.21 ds_project

Column	Type	Default	Nullable
id	integer	nextval('ds_project_id_seq'::regclass)	NO
name	character varying(255)	None	NO
category_id	integer	None	YES
description	character varying(2048)	None	NO

5.2.22 ds_project_datasets

Column	Type	Default	Nullable
id	integer	nextval('ds_project_datasets_id_seq'::regclass)	NO
project_id	integer	None	NO
dataset_id	integer	None	NO

5.2.23 ds_project_members

Column	Type	Default	Nullable
id	integer	nextval('ds_project_members_id_seq'::regclass)	NO
project_id	integer	None	NO
user_id	integer	None	NO

5.2.24 ds_projectreqmembership

Column	Type	Default	Nullable
id	integer	nextval('ds_projectreqmembership_id_seq'::regclass)	NO
is_approved	boolean	None	NO
member_id	integer	None	NO
project_request_id	integer	None	NO

5.2.25 ds_projectrequest

Column	Type	Default	Nullable
id	integer	nextval('ds_projectrequest_id_seq'::regclass)	NO
name	character varying(255)	None	NO
status	character varying(255)	None	NO
start_date	date	None	YES

Column	Type	Default	Nullable
end_date	date	None	YES
duration_years	integer	None	YES
duration_months	integer	None	YES
irb	boolean	None	YES
question	text	None	YES
methodology	text	None	YES
outcomes	text	None	YES
impact	text	None	YES
pi_fullname	character varying(255)	None	YES
pi_email	character varying(254)	None	YES
pi_affiliation_id	integer	None	YES
created_at	timestamp with time zone	None	NO
updated_at	timestamp with time zone	None	NO
requester_id	integer	None	YES
abstract	text	None	YES
category_id	integer	None	YES
exported_project_id	integer	None	YES
external_id	character varying(255)	None	NO
external_name	character varying(255)	None	NO

5.2.26 ds_projectrequest_datasets

Column	Type	Default	Nullable
id	integer	nextval('ds_projectrequest_datasets_id_seq'::regclass)	NO
projectrequest_id	integer	None	NO
dataset_id	integer	None	NO

5.2.27 ds_projectrequestacceptance

Column	Type	Default	Nullable
id	integer	nextval('ds_projectrequestacceptance_id_seq'::regclass)	NO
request_approved	boolean	None	YES
agreement_approved	boolean	None	YES
data_steward_id	integer	None	YES
project_request_id	integer	None	YES

5.2.28 ds_projectrequestinput

Column	Type	Default	Nullable
id	integer	nextval('ds_projectrequestinput_id_seq'::regclass)	NO

Column	Type	Default	Nullable
answer	text	None	NO
input_id	integer	None	YES
project_request_id	integer	None	YES

5.2.29 ds_publication

Column	Type	Default	Nullable
id	integer	nextval('ds_publication_id_seq'::regclass)	NO
methods	ARRAY	None	YES
results	ARRAY	None	YES
outcomes	ARRAY	None	YES
geographical_coverage	ARRAY	None	YES
policy_instrument	ARRAY	None	YES
intervention	ARRAY	None	YES
metadata_version_date	bigint	None	NO
title	character varying(255)	None	NO
abstract	text	None	NO
files	ARRAY	None	YES
article_id	character varying(255)	None	NO
doi	character varying(255)	None	YES
intervention_type	ARRAY	None	YES
notes	text	None	NO
temporal_coverage	ARRAY	None	YES
reference_url	character varying(255)	None	NO
article_classification	character varying(255)	None	NO
article_citation	character varying(255)	None	NO
source_url	character varying(255)	None	NO

5.2.30 ds_publication_authors

Column	Type	Default	Nullable
id	integer	nextval('ds_publication_authors_id_seq'::regclass)	NO
publication_id	integer	None	NO
expert_id	integer	None	NO

5.2.31 ds_publication_keywords

Column	Type	Default	Nullable
id	integer	nextval('ds_publication_keywords_id_seq'::regclass)	NO
publication_id	integer	None	NO

Column	Type	Default	Nullable
keyword_id	integer	None	NO

5.2.32 ds_publication_related_datasets

Column	Type	Default	Nullable
id	integer	nextval('ds_publication_related_datasets_id_seq'::regclass)	NO
publication_id	integer	None	NO
dataset_id	integer	None	NO

5.2.33 ds_publication_topics

Column	Type	Default	Nullable
id	integer	nextval('ds_publication_topics_id_seq'::regclass)	NO
publication_id	integer	None	NO
topic_id	integer	None	NO

5.2.34 ds_resource

Column	Type	Default	Nullable
id	integer	nextval('ds_resource_id_seq'::regclass)	NO
name	character varying(255)	None	NO
file	character varying(255)	None	NO
type	character varying(255)	None	NO
project_id	integer	None	YES

5.2.35 ds_role

Column	Type	Default	Nullable
id	integer	nextval('ds_role_id_seq'::regclass)	NO
name	character varying(255)	None	NO

5.2.36 ds_signedagreement

Column	Type	Default	Nullable
id	integer	nextval('ds_signedagreement_id_seq'::regclass)	NO
filename	character varying(255)	None	NO
name_of_agreement	character varying(255)	None	NO
agreement_type	character varying(255)	None	NO
key	character varying(255)	None	YES
project_id	integer	None	YES

Column	Type	Default	Nullable
user_id	integer	None	YES
dataset_id	integer	None	YES

5.2.37 ds_tag

Column	Type	Default	Nullable
id	integer	nextval('ds_tag_id_seq'::regclass)	NO
tag	character varying(255)	None	NO

5.2.38 ds_terms

Column	Type	Default	Nullable
id	integer	nextval('ds_terms_id_seq'::regclass)	NO
text	text	None	NO
created_at	timestamp with time zone	None	NO
updated_at	timestamp with time zone	None	NO

5.2.39 ds_topic

Column	Type	Default	Nullable
id	integer	nextval('ds_topic_id_seq'::regclass)	NO
topic	character varying(255)	None	NO

5.2.40 ds_user

Column	Type	Default	Nullable
id	integer	nextval('ds_user_id_seq'::regclass)	NO
enabled	boolean	None	NO
username	character varying(255)	None	NO
first_name	character varying(255)	None	NO
last_name	character varying(255)	None	NO
orcid	character varying(255)	None	NO
email	character varying(254)	None	NO
job_title	character varying(255)	None	NO
address	jsonb	None	YES
cv	character varying(255)	None	NO
phone	character varying(255)	None	NO
photo	character varying(255)	None	NO
fax	character varying(255)	None	NO
created_at	timestamp with time zone	None	NO

Column	Type	Default	Nullable
updated_at	timestamp with time zone	None	NO
terms_of_use_assigned	timestamp with time zone	None	YES
quiz_passed	timestamp with time zone	None	YES
quiz_score	integer	None	YES
is_private	boolean	None	NO
private_id	uuid	None	NO
is_registered	boolean	None	NO
should_sent_invitation_email	boolean	None	NO

5.2.41 ds_user_departments

Column	Type	Default	Nullable
id	integer	nextval('ds_user_departments_id_seq'::regclass)	NO
user_id	integer	None	NO
institution_id	integer	None	NO

5.2.42 ds_user_institutions

Column	Type	Default	Nullable
id	integer	nextval('ds_user_institutions_id_seq'::regclass)	NO
user_id	integer	None	NO
institution_id	integer	None	NO

5.2.43 ds_user_roles

Column	Type	Default	Nullable
id	integer	nextval('ds_user_roles_id_seq'::regclass)	NO
user_id	integer	None	NO
role_id	integer	None	NO

A

Use Cases

Federal and state government agencies collect data that might be available for research purposes. Getting access is typically hard because agencies do not have the resources to deal with data access requests in a timely manner. Most of the agencies also do not have a very well defined process and applying for data access ends up being a long paper trail. It is hard for agencies to increase their resources because they can only operate within their political mandate, which most of the time does not include providing data access for research. We see agencies using the data stewardship application to have a structured way to process access requests and document these requests along the way. Furthermore, we see researchers using the application to request access and find out more information about available data and information around a dataset.

A.1 Use Case 1 - The Data Providing Agency

An agency typically employs one person who is responsible for managing data access. Oftentimes this is an admin person dealing with user requests to access data. This person however, does not necessarily have the capacity to decide on their own if data access is granted. They might consult with an IT Security person, and legal counsel. However, this person is responsible to facilitate the workflow of a data access request. We call this person data steward. Instead of the agency employee this can also be an external person that the agency nominates to be responsible for facilitating data access of the agency's data in the ADRF. The data steward receiving the access requests in general looks for a clear project description, information on the people who request access, what the outcome of the project is, and if the requested project can be done with existing agreements or if new agreements need to be set up. Data Stewards receive data access requests through the data stewardship application and they determine if data access is approved by clicking the approve button in the application. In a second step the data steward can use the data stewardship application to start the process of getting all legal documents signed, such as MOU or DUA, NDA. The data stewardship module helps the agency to track this entire process so that they always know at which stage different projects are. In addition, by collecting all the data that are generated through this process the data providing agency receives reports on how their data is being used via the application.

A.2 Use Case 2 - The Researcher and Class Participant

The Researcher and class participant is the second use case. Researcher are external people that are using the ADRF to work on their research projects, typically coming from academia or other public institutions. Class participants are typically employees of state and federal agencies that sign up for the Applied Data Analytics Class that CI offers. The data stewardship application helps them to collect all the information that is needed for a data request. Using the data stewardship module these users will be given a structure to follow along to submit the request. They would like to know more about the data sources, and the metadata of the data being requested. In addition, it is important for the researcher to be able to track the data request. When requesting data ideally the researcher would like to have some guidance. For example, when requesting data for a new project it would be really helpful to know if somebody else is working on a similar research question. If so, it would be great to get more information on the projects other people are working on. In addition, the user will benefit from having more information on the specific data sources they requested. The data stewardship application provides this information to the user.

A.3 Use Case 3 - The Administrator

CI has an administrator who is responsible that both data steward and researcher can use the application successfully. For example, agencies might have slightly different workflows that need to be accounted for in the application. Thus, the data stewardship application was designed to be customizable. This means a dedicated administrator can change for example add specific input fields to the project request. For the current projects in the ADRF the administrator is a CI employee, however, this user can be located at the agency too. The administator is only using the application to make changes to the default layout.

B

Definition of Roles

There are different agents working together in the application, each of which has a role that comes with different responsibilities and privileges. All agents have to register in the application in order to use the application. When visiting the application this can be done by clicking the register button. During this step, public users change their status to a registered user. When the user submits the registration form, an ADRF Administrator of the application will receive a notification and can verify the user by making sure the registration is valid and coming from a real person. After verification the user role is updated to verified user and the user can now log into the application and start the onboarding process. Once the onboarding is complete the user can access all features of the application that are relevant to their roles. The specific roles are described in the following.

B.1 User States While Onboarding

For a user to assume any role in the application, they must first go through the sign-up and onboarding process. During this process, the user goes through various "states" which allow some set of functionality in the application. These states are as follows.

- Public User: A user who visits the application, but has not registered.
- Unverified User: Once a user signs up for an account on the application, the user state switches to Unverified User. It is a user that is registered, but whose identity has not yet been verified.
- Verified User: The Administrator receives an email whenever there is a new Unverified User on the application. Upon receiving that notification, the Administrator is responsible to verify the user. Verification of a user typically refers to making sure the at the user is a real person, by getting in touch with the user in person or remotely. Once verified, the Administrator approves basic access to the application. Thus, a verified user is a user whose identity has been verified and is approved for basic access to the application. If the user is to have a privileged role, such as Data Steward, the Administrator can assign the role at this time.
- Active User: A Verified User has to undergo security training and sign the terms of use of the ADRF via the onboarding tool before being able to submit project requests. Once the onboarding is completed, the user's state changes to active.

B.2 Application Roles

The following roles are granted to an individual user and give that user certain privileges across the entire application.

- Data User: A standard user that can submit research project requests.
- Data Steward: A user that is responsible for dataset administration within the application.
- ADRF Administrator: A user that has administrative access to the application.

B.3 Process Specific Roles

The following roles are assumed only with the context of a specific process.

- Project Requester: A user that has requested a project.
- Project Member: A user that is included as a member of a requested project.

C

Privileges of Roles

The following table captures all of the actions that can be taken by users in the baseline functionality of the example application.

Role/State (Who)	Privilege (What)
Public User	Can view landing page with website information Can register for an account
Unverified User	Can be added to a project (Project Member role)
Verified User	Can login with credentials Can submit forgot password request Can view user-specific home page Can view and edit profile details Can view and edit application settings Can search all datasets Can view all details of a dataset Can be added to a project (Project Member role)
Project Requester	Can input required fields in project request form Can add other Verified Registered Users ("Project Members") to project request Can add Datasets to project request Can specify a different Verified Registered User as the project PI Can submit project request Can view status of a submitted project request Can view details of a submitted project request Can edit a submitted project request if not yet approved/denied Can provide feedback or additional information on a project request Can cancel a submitted project request Can upload signed/counter-signed MOUs, NDAs, or other legal agreements on behalf of PI and Project Members Can request to add a member to an existing project Can request to add a dataset to an existing project
Project Member	Can view status and details of a project request Cannot provide feedback or edit project request Can upload signed NDAs or other legal agreements
Data Steward	Will receive notification email of a new project request in which one or more of their datasets are requested Can view projects requests pending review Can view required fields of a project request Can view Project Members (including details) of a project request Can view Datasets (including details and other Data Steward details) of a project request Can provide feedback on a project request Can request additional information from a request submitter Can reject a project request Can approve a project request Can upload blank MOUs or other legal agreements that must be signed by PI and/or Data Owners Can upload blank NDAs or other legal agreements that must be signed by Project Members Can upload signed/counter-signed MOUs, NDAs, or other legal agreements Can verify all legal agreements have been received Can send a final project request to Admin (or system) for project workspace creation Can view Dashboard and Metrics of Active/Completed projects which use their datasets Can nominate another user to become the new Data Steward of select datasets
Admin	Can add news/announcements to the home page Can edit the Terms of Use Can edit the list of Institutions Can edit the Applications settings Can add/edit dataset metadata Can add agency specific requirements Can make changes to a given user's role(s)
System	Will notify users by email of relevant updates to a request or when an action is required Will maintain, version, and make accessible legal agreements to their concerned parties

D

Python Requirements

The following list are Python packages which are in the requirements.txt file for the Data Stewardship backend application.

Django==2.2.16
graphene-django==2.2.0
psycopg2
django-cors-headers
requests
boto3
graphene-file-upload
python-keycloak
python-dateutil
python-decouple
django-prometheus
django-extensions
gunicorn
PyJWT
cryptography