CONNECTED

to RESEARCH

Personalized Information Portal for Research Scientists
Software Design Document

Sponsors:

Carolyn Cramer, **Nick Cramer**, and **Geoffrey Elliott**

of **Pacific Northwest National Laboratory**

Presented by Team 29, *The Majestic Turtles:*

David Winkler, **Benny Zhao**, and **Christopher Nguyen**

Contents

1. INTRODUCTION

1.1 Purpose

This software design document describes the functionality and application design of the funding component of the Connected to Research web application. The web application is intended for those who need funding opportunities for research. Researchers shall use the funding component of the Connected to Research web app to discover new funding opportunities. This web application shall make researchers' search for funding simpler.

This software is being designed for researchers at the Pacific Northwest National Laboratory, but the final web application is intended to be used by researchers in any fields, anywhere.

1.2 Scope

This software design document shall provide details on three parts of the web application. It shall be composed of the front-end, the database management system, and the back-end database. It shall include details on how the front end interacts with the back-end database through the use of the database management system.

Researchers shall have the ability to interact with the web application. They shall be able to search for funding opportunities that pertains to them based on their interests and preferences. They shall also be able to view their funding opportunities as well as save them within the database.

1.3 Context

Connected to Research is a web application that aims to connect researchers to information relevant to their interests. This includes other researchers, publications, and funding opportunities.

The funding component of Connected to Research is just a portion of the final web application. It serves to interact with the user, where the majority of them will be researchers. The component itself constantly updates itself with more up-to-date funding opportunities and then stores the information into the database. The system then narrows the search based on researchers' input of their interests and preferences. The system will store funding opportunities of different types, and present researchers with results relevant to their interests and preferences. This is how the system aims to simplify their search for funding.

Users shall be able to save the funding opportunities they have already viewed as to optimize search efficiency with respect to the researchers' time. This information can later be used for analysis, planning and/or improvement of the software design.

The database will act as the model in a model-view-controller (MVC) setting, where it stores the necessary information and updates on a scheduled basis. Whereas the front-end portion will serve as the view, displaying relevant information pertaining to the researcher's criteria. Lastly, the back-end is the controller, manipulating the incoming data the user inputs into the system and storing it into the database.

1.4 Summary

This document refers to the funding component of Connected to Research. It includes the application design which shall detail the architectural design of the three parts of the web application. These three parts include the front end, the database management system, and the back-end database. The web application will be designed similar to that of a Model-View-Controller (MVC).

This document shall also include the data design of the database which includes all of the database tables and database attributes. Along with the details of the database tables and database attributes.

This document shall also include the design viewpoints of the application design.

2. DEFINITIONS

Back-end

The databases and data processing components of the system. It launches programs in response to the front-end's requests and operations.

Component

Component refers to a self-contained portion of the Connected to Research web App. It provides separate functionality that is related to (but not dependent on) other components of the application.

Controller

Represents the classes connecting the model and the view, and is used to communicate between classes in the model and view.

Database

An organized collection of data. It is the collection of schemas, tables, queries, reports, views, and other objects.

Database Attributes

A single data item related to a database object.

Database Management System

Software that handles the storage, retrieval, and updating of data in a computer system.

Database Table

A collection of related data held in a structured format within a database.

FedBizOpps.gov

A funding aggregator created and maintained by the US government. It contains all federal procurement opportunities

Front-end

The view or display of information to the user for interacting with the system.

Funding Component

Component of Connected to Research web app that connects researchers to funding opportunities.

Grants.gov

A unified website for interaction between grant applicants and the U.S. Federal agencies that manage grant funds.

Model

The representation of the logical structure of data in a software application. This object model does not contain any information about the user interface.

MVC

A software architectural pattern which separates an application into three main components: the model, the view, and the controller.

PHP

A server-side scripting language designed for web development, but also used as a general-purpose programming language.

View

A collection of classes representing the elements in the user interface. Everything that the user can see and respond to on the screen is part of the view.

Web App

In computing, a web application or web app is a client-server software application in which the client (or user interface) runs in a web browser.

XML

Extensible Markup Language. XML is a markup language that is designed to allow structured data to be stored in files. Many websites publish data in XML formatted files. XML can be *well-formed* if it conforms to the XML standard. XML can be *valid* if it conforms to a schema.

3. APPLICATION DESIGN

3.1 Introduction

The funding component of the Connected to Research web app will be made up of three interrelated design entities. There will be a front-end presented to users, a database for storing all information, and a back-end to populate the database and connect it to the front-end. The web app will allow users to register for an account. At some point, users will enter their areas of interest and the web app will record them. Their interests will be recorded in the database under a table related to user interests. The user interests will be used to match funding opportunities to users. The user and interest tagging system is not the primary focus of the funding component. However, they will need to be implemented to support the implementation and testing of the funding component.

The front-end of the component refers to the visible user interface that users will interact with. The front end will include a search and browse page. The search page will allow users to find any and all funding opportunities related to their search terms. The browse page will present users with opportunities selected by the component based on their interests.

The database refers to the data storage mechanism used for the user, interest, and funding data. The database will be populated by the back-end scripts that will pull data from funding information sources. For the funding component, the database will contain data relating to funding sources, user information, and interests. Other components will maintain separate tables in the database.

The back-end refers to the scripts necessary to implement features in the front-end and database. The front-end and database will need functionality implemented in the back-end. Funding opportunities will be periodically updated from online sources. New opportunities will be added, and interests will be associated to those opportunities. Users whose interests match those associated with a funding opportunity will be notified of the opportunity.

3.2 Front-End

The design of the front-end will consist of using the Bootstrap web framework as a design foundation. Where the framework already contains pre-built components for better efficiency of web development and usability for the user/developer.

In terms of designing the layout for the user interface, there is already a proto-type design created by our sponsor. But our main focus is on the completion of the functionality of the funding component. Therefore, the front-end portion will be the lowest priority. Furthermore, the approach to a comfortable user interface will be simple enough for researchers to utilize, thus eliminating time required to learn the user interface or searching for capabilities on finding funding.

The layout design for the funding component will consist of three main columns. The first column (left-most column) contains filters for **agencies**, **funding**, and **sort**. The capability of this column is to let the user (researcher) be able to filter funding opportunity based on which **agency** it is originated from, the amount of **funding** available, and **sorting** by either date the funding started, time length or duration of funding, and PNNL projects associated with funding.

The second column (center column) will contain the funding opportunities based on filters and interests of the user. Displayed within each funding opportunity is the title/name of the funding, the agency originated from, and the current amount of PNNL projects associated with the particular agency. It also includes the PNNL contact information, concept paper due date, invitation for full proposal date, and the proposal due date. Lastly a description of the funding opportunity will be within the same bootstrap container.

In the third column (right-most column) will be research events and deadlines previously, currently, and upcoming. The user will be able to click on the hyperlinks of the deadline to navigate to that funding opportunity.

Within the user's account, there will be an option for the user to input their interest. Where the interest will be used for narrowing the search of funding opportunities that pertain to the user. The interest table will store a list of strings that will correspond to a list of strings in the funding opportunity table.

Each funding opportunity has an associated category. For example, Grants.gov has CFDA numbers and funding opportunity categories. Meanwhile, FedBizOpps.gov has NAICS codes and classification codes. Both of these correspond to different interested parties. We need some way to map these to more general interests, like 'Biology'. To do this, we will make use of an intermediate table, called Interests. The Interests table will have a Category, which is the general interest. It will have a String field, which is a list of strings of NAICS and classification codes, CFDA numbers, and any other descriptors that may come along with new sources.

3.3 Database

The database will be composed of tables to store the funding opportunities. There will be a table for each source of funding defined, because they may contain different fields that will need to be kept track of. For example, FedBizOpps.gov and Grants.gov feature different classification codes for funding opportunities, such as NAICS codes and CFDA numbers, which may overlap and need to be kept distinct. To avoid having unnecessary fields duplicated in each funding source table, there will be a parent table from which the source table are derived. The parent table will include the shared fields between the funding sources, such as opportunity number, title, post date, and response date.

The database will be populated by the back-end using scheduled scripts. The front-end will communicate with the back-end to service user requests for information. The front-end will access a list of funding opportunities related to a user's interests using a back-end service routine. The back-end will use the user's interests, obtained from the user's corresponding table, to access up to five interest entries.

3.4 Back-End

For the back-end, there will be XML files downloaded daily or weekly from funding sources: FedBizOpps.gov and Grants.gov. that will be parsed into the database. A script will be implemented to pull data from these funding sources and into the database. Within this script will be a PHP XML parser that takes in a XML file as input. The PHP XML parser will match all of the elements within the XML file to corresponding attributes within the database. Then it will perform SQL insertion with PostgreSQL's *pg_query()* function to insert the elements' text content to its corresponding database tables.

Both Grants.gov and FedBizOpps.gov publish periodic XML files that we can parse. The XML files from Grants.gov are both well-formed and valid. They also make available a description of the XML files and the data contained in them. We can use this document to map data from the XML file to the database. The XML published by FedBizOpps.gov comes in the form of both weekly and nightly XML dumps. The weekly file is well-formed and valid, but will probably update too infrequently for our purposes. It is also a very large file. The weekly dump as of writing is over 1 GB. The nightly dumps are smaller, but are not well-formed. However, they are still simple to parse. Using these two XML sources, we should be able to pull all the data we need for our database.

4. DATA DESIGN

4.1 Data Description

The ERD diagram below displays the general overview of the data design. Currently, we plan to support two funding sources. These will be FedBizOpps.gov and Grants.gov. With our design, it will be simple to add support for more funding sources. Any source that may need to be supported in the future will support, at minimum, the fields contained in the Funding Opportunity table. This is not an unreasonable assumption, as any opportunity that does not include that data does not have enough data to describe the opportunity adequately.

The tables derived from the general Funding Opportunity table will contain any additional information specific to that funding source that is not contained within the Funding Opportunity table. The Funding Opportunity will contain a field storing the interests associated with a funding opportunity. There may be any number of interests associated with a given opportunity, for example, their may be an interdisciplinary opportunity for biologists and chemists.

There will be a table for storing each user's information. Each user will be able to list up to five interests related to funding opportunities. The interests in the user table will be related to a category in the interests table. For example, a user might list their interest as 'Toxicology'. Then their interest will map to an entry in the Interest table with a 'Toxicologist' category. The String entry will be a list of substring, each of which corresponds to a, interest string in the Funding Opportunity.

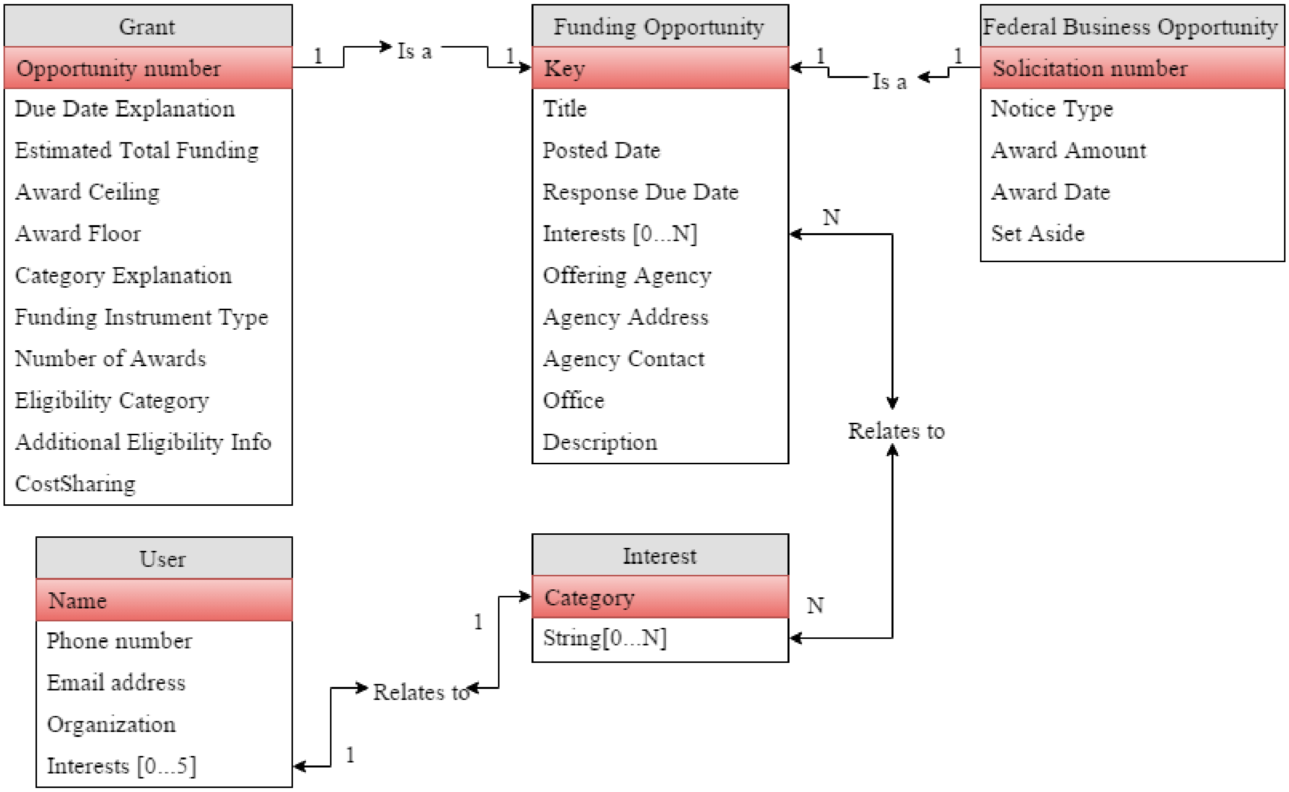

Figure 1 - ERD Diagram: Data Design

4.2 Data Dictionary

Below is a dictionary of the terms used in the data design diagram.

Grant: A funding opportunity from Grants.gov.

Opportunity Number: A unique identifier for each grant offered by Grants.gov.

Due Date Explanation: Further elaboration on the due date.

Estimated Total Funding: An estimate of the total funding available.

Award Ceiling: Maximum funding available for specific funding opportunity.

Award Floor: Minimum funding available for specific funding opportunity.

Category Explanation: Explanation of the funding category.

Funding Instrument Type: The type of funding agreement opportunity uses.

Number of Awards: Total number

Eligibility Category: Category of applicant eligible for this opportunity.

Additional Eligibility Info: Additional requirement criteria for the funding opportunity.

Cost Sharing: Whether or not funding will be shared.

Funding Opportunity: A funding opportunity from FedBizOpps.gov

Key: A unique identification number of the funding opportunity.

Title: The headlines of the funding opportunity.

Posted Date: The date when the funding opportunity is first advertised.

Response Due Date: The deadline date for the funding opportunity.

Offering Agency: The name of the agency proposing the funding opportunity.

Agency Address: The address of the agency proposing the funding opportunity.

Agency Contact: The contact information of the agency proposing the funding opportunity.

Office: The location of the agency proposing the funding opportunity.

Interests: A string of category information which maps to a category in the interests table.

Description: The description of the funding opportunity.

Federal Business Opportunity: A funding opportunity from FedBizOpps.gov

Solicitation Number: A government number that identifies a certain request for quote.

Notice Type: The type of notice that the funding pertains to (award notice, combined synopsis/solicitation).

Award Amount: The amount of available award.

Award Date: The date the award is posted.

Set Aside: The proportion of funds reserved for a particular service.

Interest: Entity that holds all interests of users and connects that with funding opportunity.

Category: Interest category (e.g. biology, toxicology, etc.)

String: A list of strings that might appear in funding opportunity.

User: Entity that holds the user's information.

Name: Name of the user.

Phone Number: Phone Number of the user.

Email Address: Email Address of the user.

Organization: Organization the user belongs to.

Interests: User's research interests.