***Request for Proposal (RFP)***

Company and Employment Lineage Data and Visualization (CELDV)

***Phase 1 (of 2)***

**RFP Issue Date: Tues 8/29/2023**

**Response Due Date/Time: *5:00pm Friday ~~9/29/20~~*~~23~~ 10/6/2023**

**Table of Contents**

* *The Concept*

The concept for the web application *Company and Employment Lineage Data and Visualization (CELDV)* Capstone Project came through the DCSTEM Dean from a friend of UA Little Rock. It will support the data maintenance and visualization of a *genealogy* of employers, educational institutions, and employees in the State of Arkansas.

The next sections provide an overview of CELDV, high-level business requirements, enumeration of the actors, and an abstract view of the data for the prototype web application is to be based. The content and structure of responses to this RFP are specified in Section 6.

* *A Capstone Project in Two Parts*

CELDV is a web application software project that is larger than can be completed by a team of 4 individuals in one semester (for a three-hour course). The next Section 3 provides an overview of the entire project. The remainder of this RFP contains the requirements for Phase 1 of the project to be completed by teams in the Fall 2023 semester.

* *The Big Picture – Phases 1 and 2*

Imagine a data maintenance and visualization web application that provides a detailed picture of individual employment histories and the history of companies and other employers in a given geographical area or industry type. The web application could support:

* **Employment History**. For each individual represented in CELDV, the ability to query and visualize that person’s employment history. The visualization will depict the employee with links to their employers and job titles present and past.
* **Employer’s Evolution.** Companies1 spin off other companies, companies merge, and companies go out of business. The ability to query and visualize the genealogy of companies can provide otherwise hard to discover information on the evolution of employers in a given geographical area or industry type.
* **Certification Granting Institutions (*Institutions*).** The prototype will support the maintenance of data for institutions including universities, two-year colleges, high schools, and entities providing industry recognized certifications. CELDV will support the association of Employees with the institutions and certifications earned by the employee.
* **Visualization.** The visualizations generated by CELDV will literally be directed graphs (digraphs), i.e., consist of nodes and arcs (arrows) between nodes. Abstractly, we can think of CELDV data as represented by a large digraph in which nodes represent employees, employers, and institutions and the arcs represent relations between these entities. The visualizations will depict specified subgraphs of the CELDV data. For example, users could ask CELDV to visualize:
* the genealogy of an employer including company mergers and the spinning off of one or more companies a specified company.
* an individual’s employment history with arrows from each of the employers of that individual to the node representing that employee. Hovering over each these arc will show the dates of employment.
* an individual’s educational background by depicting all the institutions from which that individuals has earned degrees or certificates.
* **Analysis of Employment Patterns.** How many employees left company A and started working for company B in a specified time period? CELDV will support this type of analysis of

1 We will use *employer* to refer to companies, non-profits, government entities (local, state, and federal) – any entity that hires paid employees. *Company* is used in contexts in which company would be awkward, like *companies merge.*

workforce employment patters.

* Solution Summary

This is a request for proposal (RFP) from a prospective client to specify, design, and develop Phase 1 of the described prototype web application. The content and structure of responses to this RFP are specified in Section 7 of this RFP.

The functional features of CELDV (both Phase I and Phase 2) are described in the following informal epic descriptions.

* **LoginRegistration.** All CELDV will support login and registration. Public users can register for login credentials as a CELDVUser who can visualize and analyze CELDV data. An AdminUser can also maintain CELDV data and can create new AdminUser accounts2. Registration of public users will include the verification of email addresses prior to establishing login credentials. An admin user can set up another admin user, including entering the email address and a password. No email verification is required in this case.
* **Maintain CELDV Data.** Admin users will be able to maintain (add, edit, and, remove – when appropriate3) the following data:
* **Employers.** Admin users will be able to maintain employers with attributes including company name, industry type, date of incorporation, and, if applicable, date of dissolution.
* **Employees.** Admin users will also be able to maintain employee information with attributes including name, phone number, and email address.
* **Institutions**. Admin users will be able to maintain information on institutions that award industry-recognized credentials (degrees and certificates). The attributes of institutions will include name, city, state, and credentials the institution is authorized to award including all degrees and certificates.
* **Record of Employment**. CELDV will support adding an employment record representing that an employee worked for a company with a specified job title, and with a start date and an end date. These will actually be the attributes of a relation (arc) from an employer to an employee.
* **Earned Credentials.** CELDV will support the recording that an employee received a credential from a given institution. CELDV must represent the credential granted and the date. These will be attributes of the relation (arc) between an institution and an employee.
* **Employer Lineage.** CELDV must support the representation of a company being spun off from an existing company (or companies) or that companies are merged, e.g., when one company buys another. CELDV is to record the date of the merger or spin off.
* **Visualize CELDV**. The visualizations of CELDV data are described at the top of this Section 3. Each use case (UC) in this epic will follow a similar pattern:
* Select an entity (employer, employee, or institution) that will serve as the root of the subdigraph to be generated.
* Specify the relation to be depicted between the root entity and the other entities to be included.
* Generate and render the graph.

Examples of nodes and relations between nodes are depicted in . Figure 1.

* **Analysis.** All CELDV registered users will be be able to execute the queries such as:

2 Any instance of the CELDV web application must be initially installed with a database containing an initial AdminUser.

3 Not all removals make sense. E.g., one would not remove a company for which employees have been entered.

* For a given company, the number of employees on a given date.
* For a given company and for each certification held by employees on a given date, the number and percentage of employees with each certification.
* A list of companies by type as of a specified date.

The client seeks suggestions for additional analysis capabilities the respondent believes will be of value and can be implemented by the deadline.



Figure 1. Legend for ECLDV Visualizations

* Prototype Limitations

The client understands that the initial implementation of CELDV will necessarily be a prototype. While all the required functionality is to be demonstrated, the following requirements necessary in a production version of the application may not be included:

* **Software Cybersecurity.** Respondents to this RFP need not plan to incorporate coding practices for cybersecurity or the use of security analysis tools in the prototype implementation.
* **System Performance.** The prototype CELDV will support a few concurrent users, but will not be required to meet any other performance requirements.
* User Types and Actors
* CELDV User Types

Registered users of CELDV will be of one of two types.

* **AdminUser.** Admin users can maintain CELDV data as well as utilize all the CELDV features available to CELDV users. AdminUsers can input (register) login accounts for an additional AdminUser.
* **CELDVUser.** CELDV users can use the visualization and analysis features of CELDV.
* CELDV Actors
* **PublicUser.** A PublicUser does not have login credentials for CELDV. To access CELDV they will need to register for login credentials (username, password).
* **AdminUser.** An AdminUser can maintain CELDV data.
* **CELDVUser.** A CELDVUser can access the analysis and visualization features of the web application.
* **Additional Actors.** Responding teams may find they will need additional actors when writing functional requirements, e.g., the email client might be the actor for the UC for receiving a post from a link in an email sent as a PublicUser is attempting to register for a CELDV login account.
* **CELDV.** There may be some use cases that are triggered within the application. In such cases, the application itself is an actor.
* Overview of CELDV Data

This section contains an informal view of CELDV data. The attributes provided in the description may not be complete and only attributes required for an accurate abstract view of the data are included.

**RegisteredUsers4.** A registeredUser5 contains the information for a user with login credentials on CELDV. The type RegisteredUser will have the following attributes:

* **userName.** CELDV will use email addresses for user names.
* **passWord.**
* **userType.**

**Employers.** This type represents the collection of employers represented in CELDV. The attributes of the type Employer will include:

* **employerName**
* **incorporationDate.** Date employer began doing business.
* **dissolutionDate.** Date on which employer ceased doing business (under employerName). This can be the date on which one company spins-off another company. Possibly null.

4 The CELDV documentation uses an identifier convention in which collection types are plural, e.g., RegisteredUsers, and elements in the collection are singular, e.g., RegisteredUser.

5 Another identifier convention used in CELDV documentation is to capitalized type names, e.g., RegisteredUser, and lower case versions of the identifier for instances, i.e., objects, of that type, e.g., registeredUser. Of course, multiple identifiers for a given type may be required, e.g., registeredUser, newRegisteredUser, loggedInRegisteredUser, etc.

* **employees.** This attribute of Employers is a collection of references6 to instances of the type EmploysEmployee (described below).

**Institutions.** This type represents the collection of all degree and certificate-granting institutions represented in CELDV. The attributes of Institution will include:

* **institutionName**.
* **foundedDate**
* **institutionCity**
* **institutionState**

**Employees.** This type represents the collection of all employees represented in CELDV. The attributes of Employee will include:

* **employeeLast**
* **employeeFirst**
* **employeeSSN**
* **employers.** A collection of references to instances of EmploysEmployee. Employers represents the work history of an individual.
* **certificationsEarned.** This attribute contains references to instances of AwardedCertification (described below) and represents all the degrees and professional certifications earned by the employee.

**FormsNewCompany** This type represents a relation between two companies in which one company is formed from another company. This might occur when Company A spins-off one or more other companies, say, Company B and Company C. There will be one instance of FormsNewCompany to represent that Company A was a predecessor to Company B and another instance to represent that Company A is a predecessor of Company C.

FormsNewCompany is also use to represent the case in which a Company A is the result of the merger of companies B and C. There will be an instance of FormNewComany to represent that Company B is a predecessor of Company A and another instance to represent that Company C was a predecessor of Company A.

The simple case for the use of an instance of FormsNewCompany is when a company is rebranded to doing business under another legal name, e.g., Twitter rebranded as X.

**EmploysEmployee7.** This type represents that an employer employed an employee (and, consequently, that an employee worked for an employer) with a given job title. This type representing the relation between and employer and an employee will include the following attributes:

* **startDate**.
* **endDate**
* **jobTitle.**

Note that an employee receiving a promotion (new job title) will be represented by a new instance of EmploysEmployee.

**AwardedCertification.** Instances of this type represents that an institution awarded a degree or certificate to an employee. The attributes of the type AwardedCertification will include:

* **certificationAwarded.** The specific degree or certificate awarded.
* **awardedDate.** The date on which the certification was awarded.

6 A reference to an instance, borrowing C/C++ terminology, is a pointer to an object. Note that there can be multiple references to the same object.

7 Note this is a case in which the relation between two object cannot simply be handled via references because the relation itself necesarily has attributes, e.g., startDate.

* *Informal Functional Requirements – Phase 1*

Phase 1 of the CELDV project will implement the portion of CELDV related to CELDV login accounts and employers. Phase 1 will not deal with Institutions or Employees.

***However, of the project requirements for Phase 1 include documenting access to the source code all additional artifacts need to build and deploy the Phase 1 solution. This is to include documentation on how to deploy the application to the hosting environment. (Phase 2 teams can then set up their own instance of the hosting environment – or opt for a different hosting environment.***

The Phase 1 epics includ Epics to be implemented in Phase 1 are:

**Epic 1. RegistrationAndLogin**

All the UCs for registering for CELDV login accounts go here and the UCs related to logging into the application.

**Epic 2. MaintainEmployers.** UCs in this epic will include those needed for maintenance of the data emloyers, including

**UC 2.1. AddEmployer.**

**UC 2.2. EditEmployer**

**UC 2.3. RemoveEmployer.** Avoid dangling reference

**UC 2.4. EstablishCompanyLineage.** Establishes that a company, say, ParentCompany was used in the formation of another company, say, childCompany

**Epic 3. VizualizeCompanyLineage.** Teams responding to this RFP are encouraged to suggest creative UI/UX solutions to visualizing CELDV data. Clearly, with even a moderately sized set of Employers, and company lineage relations, it is not feasible to display all the employers and company lineage relations. Some of the specific challenges include;

* Since there is no restriction on the number of companies that can be represented in CELDV,
* How will the user indicate what portion of the data is to be visualized? Can the user start with one employer or one employee.?
* How does the user extend what is currently being visualized? Note that this will, in general, require that some of what was visualized is no longer visible.

*Solution Technical Requirements*

The successful RFP response must satisfy the following technical requirements.

* **Browser.** The CELDV prototype need only function in a single browser. Google Chrome is a reasonable choice since UA Little Rock is a “Google Shop”. However, if there are compelling reasons for doing so, another commonly used browser may be selected.
* **Responsive.** A responsive design for the web application is a “very-nice-to-have” but not a “must have”. A responsive design accommodates varying display sizes, e.g., on a laptop screen and a smart phone.
* **Deployment and Accessibility.** The CELDV web application must be accessible through the targeted browser by any public user, i.e., the web application must be deployed to a standard web-accessible hosting environment.
* **Database.** The website must utilize a database to store the information required by the CELDV solution.
* **Concurrent Users.** Your solution must be capable of supporting at least 5 concurrent users. This might be handled, e.g., via a session ID generated and checked by the server-side application.

The following technical features are preferred by the client but are not essential for meeting requirements for the prototype <>

*Mandated Response Content*

Responses to this RFP must include the follow sections and include the specified information for each section. An MS Word template for the RFP responses will be provided during the second week of the semester.

6.1 Overview - Kylota Stewart

Include a summary of your team’s proposal. What are the highlights of your solution? What do you think might distinguish your team’s response? Provide the client with clear and compelling reasons to read the remainder of the proposal. The client may well weigh the content of this section more than might generally be expected.

6.2 Actors and User Types - Juan Perez

The mandated User Types are specified in the RFP. Be sure to define any additional actors (that will appear in use case functional requirements specifications). Is EmailService an actor in the CELDV solution?

6.3 Abstract Data Model (ADM) - Juan Perez

An approach to providing a specification of the data pertinent to a CELDV design and implementation will be presented in class. The approach provides an abstract representation of the data that suppresses implementation details. The ADM will be used to write the functional requirements. Much of the ADM will be developed in class.

6.4 Functional Requirements – Phase 1 - Kylota Stewart

This section will include functional requirements specifications for your CELDV design and implementation. All functional requirements will be expressed using a modified and structured format. The approach to these model-based use case specifications will be presented in class and a use case template will be included in the RFP template. The following subsections represent the five epics presented in this RFP.

6.4.1 RegistrationAndLogin

Individual UC functional requirements related to Login and Registration go here.

6.4.2 Maintain CELDV Data

Individual UC functional requirements related to entering and maintaining CELDV Data go here.

6.4.3 Visualize CELDV Data

Individual UC functional requirements related to the visualization of CELDV Data go here.

6.5 User Interface - Emily Thomas

The proposal must inform the client of the proposed look and feel of the solution’s UI and at least four sample pages (or wireframe representations of the planned UI). Preference will be given to proposals that describe any UI standards for the proposed solution, e.g., standard behavior of all Save buttons.

6.6 High-level Design (Architecture) - Minh Dang

Responses must include diagrams depicting the architecture of your proposed solution. The architecture diagrams must represent all the components of your proposed solution and the distinct interfaces between components. The high-level architecture does not contain depictions of the implementation of the requirements themselves. Example high-level architecture diagrams will be presented in class.

6.7 Implementing Technologies - Emily Thomas

Clearly specify the technologies to be included in the implementation of CELDV. This section is to include an inventory of all the technologies contained within the solution as well as of the tools used to implement the solution.

6.7.1 Client-side Technologies

Specify the technology to be used in the implementation of the client-side component of your CELDV solution. Include all supporting technologies, e.g., JavaScript, supporting UI functionality, HTML 5 for web page content, etc. Be sure to include any JavaScript libraries you team will utilize, e.g., React.

6.7.2 Client-server Interaction

Your response must clearly identify the technologies to be employed for each component interfaces identified in the high-level design – in this case, the interface from the client to the server. For example, will your solution leverage Restful web service calls using, e.g., Ajax?

6.7.3 Visualization Library

A key feature of CELDV is the ability to visualize CELDV Data. Specify the graphics library to be utilized.

6.7.4 Server-side Technologies

What technologies will be included in the server-side portion of your implementation?

6.7.5 Deployment and Hosting Environment

Describe the process and tools your team will use to deploy your solution. What web application hosting service will you team utilize? Be sure to include any details about the deployment process required to successfully deploy to your selected hosting service.

6.8 Software Development Methodology

The client is requiring that teams use a hybrid software development methodology. Since the RFP response must contain the functional requirements specification and an initial UI design, the start of the project is necessitating a waterfall approach, i.e., all functional requirements are to be completed and approved by the client prior to beginning the design, development, implement, and test (DDIT) phases of the project. DDIT is to be done using an agile methodology with defined sprints.

6.9 Project Management and Project Plan - Minh Dang

Include the project plan you will utilize to meet the final project deadline for presenting and demonstrating your prototype implementation of CELDV.

Your project plan must define your team’s ***sprints***, the one team member *accountable* for the work products produced in a sprint, and the deadline date for each sprint.8. Note the accountable team member will not necessarily do all the work. They are, however, solely accountable for ensuring the milestone is met.

***Please note that after the development, hosting, and source code control (GitHub) have been configured each sprint must result in a deployable version of the application. Testing at the end of a sprint is to be done with the application deployed to the hosting environment.***

Your project plan must be specific and complete. Do not leave out essential tasks such as setting up and testing the deployment and hosting strategy. All the tasks required to deliver the prototype are to be included.

8 The team member *accountable* for a project task is accountable for the success of the task and is usually the decision maker. *Responsible* team members are responsible for doing the actual work.

Recall that a sprint should result in a deployable version of the application9.

6.10 Team Strengths - Jared Williams

Provide a very brief biographical sketch for each team member. Highlight each member’s technical competencies and experiences. *Team strengths* are more than just the capabilities of the individual team members. For you team, is the whole more than the sum of the parts? In this section, convince the client why they should choose your team to implement a prototype CELDV solution?

* *Consulting with Client*

A representative for the client will be available during each class meeting time to answer questions and address concerns. Online consulting sessions with the client can be arranged. The goal for the RFP response is to convince the client that yours is the team that should be selected for the implementation. ***Imagine*** that only one team will win the contract to proceed with the implementation. In actuality, the client will review each RFP response, provide feedback and suggestions, and may require revisions to the response prior to providing approval to proceed with the implementation.

9 The first sprints may involve setting up the development, source code control and code repository, database installation, etc. Obviously, these early sprints will not result in a deployable version of the application.

* *Addendum on 9/26/2023 – Schedule Adjustment*

Given the client was unavailable during the week of Monday September 18/2023, the following adjustments to the project deliverables and schedule have been made:

* **Required Deliverables.** The following project deliverables are required to meet minimum functional requirements:
* LoginRegistration
* Maintain CELDV Data – Employers and FormsNewCompany relation.
* Visualize CELDV – Employers and FormsNewCompany relation.
* **Optional Deliverables.** The following deliverables are to be implemented after the required deliverables are implemented, deployed, and tested.
* Maintain CELDV Data – Employees and EmploysEmployee relation.
* Visualize CELDV – Employees and EmploysEmployee relation. The following adjustments to the project milestones are in effect:
* **RFP Response Due.** Friday October 6, 2023
* **LoginRegistration.** Teams must demo their deployed solution to this use case on or before Tuesday 10/31/2023 **at 9:25am – before the start of class**.
* **Final Demos.** Unchanged. Teams to schedule 2 hour Final Demos during final exam week.