Alumni Network Database

Final Report

Student Name: Karim Iqbal

Course: Information Management

Semester: Spring 2024

Introduction

During my graduate studies at The University of Texas at Dallas (UTD), particularly through my role as a student worker in the Office of Development and Alumni Relations, I observed firsthand the limitations of existing systems like Raiser's Edge in meeting the specific needs of our alumni. Raiser's Edge, a widely utilized Customer Relationship Management (CRM) platform by Blackbaud, serves Nonprofit Organizations (NPOs) and Higher Education Institutions (HEIs) effectively in managing records and fundraising data. However, it lacks targeted functionalities for supporting alumni in their professional development and networking endeavors. This gap prompted the initiation of my project, "Alumni Network," also referred to as "Alumni Connect," which aims to forge a system tailored specifically for enhancing career services management for alumni at UTD.

Alumni Network is designed to bridge the gap left by Raiser's Edge, focusing particularly on facilitating career advancement through its robust features such as job postings, mentorship connections, and notifications of professional events that are specifically tailored to the needs of alumni. The primary objective of this system is to create a platform that not only supports career growth but also integrates seamlessly with the CRM practices that are familiar to university management, staff, and alumni. This integration is achieved through the application of data management techniques, which are intended to deliver comprehensive services that significantly enhance alumni engagement and foster enduring connections with their alma mater.

This introduction serves as a prelude to the discussion of the project that follows, which will cover the design and implementation of the Alumni Network system. The subsequent

sections of this report will delve into the database design, system implementation, challenges encountered during the development process, and the testing and evaluation phases. Through this exploration, the report aims to illustrate not only the technical aspects of the system but also the practical benefits it offers in enhancing the professional and personal development of alumni post-graduation.

Project Design

As I embarked on the journey of designing the Alumni Network system for UTD, I drew upon my observations and experiences working with Raiser's Edge at the Office of Development and Alumni Relations. This hands-on exposure was instrumental in understanding the nuances of a sophisticated CRM system and its database design, particularly how entities and keys are structured and managed. Inspired by this model, the Alumni Network was meticulously crafted to specifically support and enhance alumni career services, with a focus on adopting similar terminologies and functionalities that are familiar within Raiser's Edge.

The database schema of the Alumni Network comprises five principal entities: Alumni, Events, JobPostings, Mentorship, and Users. Each entity is designed to fulfill distinct roles within the system, contributing collectively to a comprehensive service offering for the alumni:

- Alumni: This fundamental entity has detailed personal and professional information about each alumnus, such as constituentid, name, graduationyear, degree, currentposition, organization, industry, and location. The careful collection and structuring of this data establishes the core of the networking platform.

- Events: This entity is tailored to record various alumni-centric events like reunions, galas, and professional seminars. Attributes such as actionid, eventname, description, location, datetime, organizedby, and register are chosen to facilitate the organization of these events and enhance alumni participation.
- JobPostings: This entity stores information about job opportunities specifically curated for alumni. Fields including postingid, constituentid (linking to Alumni), company, title, description, location, posteddate, apply, and lastdatetoapply are utilized to ensure that alumni have efficient access to relevant job opportunities.
- Mentorship: This entity is included for supporting structured mentorship programs, which are vital for ongoing professional development. It maintains details of mentorship relationships through attributes like mentorshipid, mentorid, menteeid, startyear, endyear, and feedback.
- Users: Managing user authentication and role delineation, this entity ensures the security and functionality of the system. It includes fields such as username, password, and role, critical for securing user interactions and defining access permissions within the system.

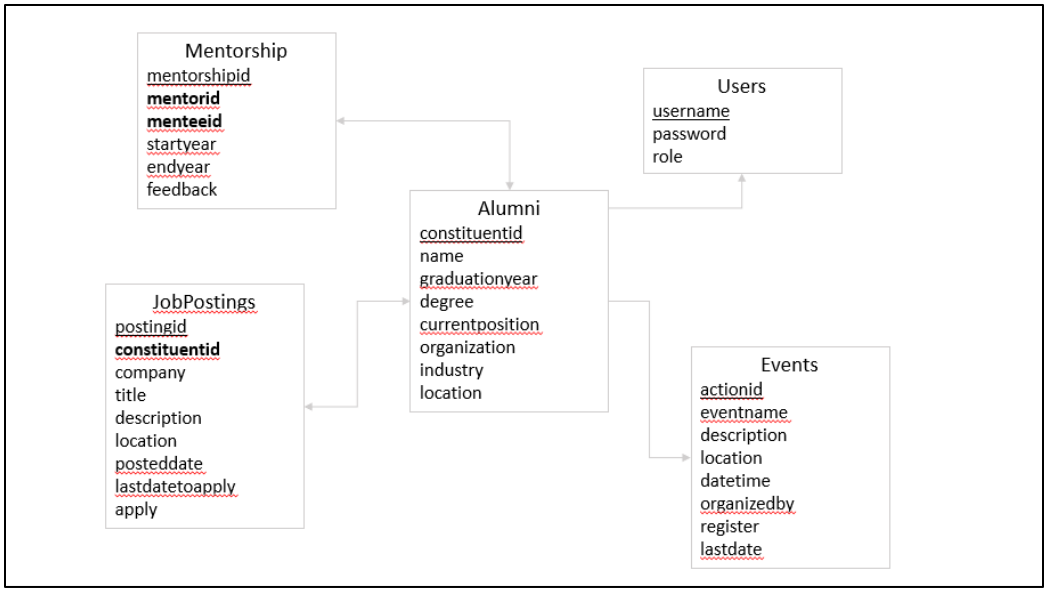

Figure 1: Database Schema of Alumni Network

Integral to the system's design is the use of key relationships between these entities, primarily with foreign keys which enhance data integrity and enable robust relational database functionalities. For instance, the link from JobPostings to Alumni through constituentid ensures job postings are correctly associated with the alumni who posted them. Similarly, mentorid and menteeid in the Mentorship table, both referencing constituentid in the Alumni table, increases the effectiveness of database by identifying unique constituent accordingly.

These relationships are not only pivotal for the functionality of the system but also support complex queries and reports that are crucial for effective alumni management and engagement. Auto-incrementing keys like alumni_constituentid_seq, events_actionid_seq, etc., simplify data management and maintain integrity across the system, ensuring each new entry is uniquely and consistently identified. This design approach mirrors the structured and methodical nature of database systems like Raiser's Edge, thereby allowing

me to apply learned principles directly to the practical challenges of building the Alumni Network.

Project Implementation

The journey of implementing the Alumni Network design was both a rigorous and enlightening experience, characterized by a step-by-step approach to setting up the PostgreSQL database. This setup began with a structured organization of tables and sequences, which was crucial in defining the underlying data structure to support all intended functionalities of the application.

The definition of each table within the database schema was crucial in accommodating the specific data requirements of various system functions. For instance, the Alumni table was created with fields such as constituentid, name, and graduationyear, ensuring that both personal and professional details of the alumni were comprehensively captured. Similar attention was applied in setting up the Events, Job Postings, Mentorship, and User accounts tables, each tailored to facilitate their respective roles within the network. The process of establishing sequences for auto-incrementing primary keys, like alumni_constituentid_seq for Alumni and events_actionid_seq for Events, ensured that each new record received a unique identifier, streamlining data management and maintaining data integrity.

Recognizing the importance of security and system integrity, particularly influenced by my familiarity with Raiser's Edge, I ensured that user authentication was a pivotal aspect of the system architecture from the outset. This was integral not just for aligning with industry standards but also for managing user interactions securely within the system.

Contrary to initial batch-loading techniques, the database employs a dynamic data entry method through the COPY command, structured specifically for real-time data insertion during user interactions. Each time data is entered or updated —whether adding new alumni details, posting job opportunities, or organizing events—the COPY command is triggered to update the respective tables immediately. This approach allows the database to reflect real-time changes and ensures that the system remains up to date with the latest information.

The design of the database also catered to the dynamic handling of updates and queries, essential for interactive user experiences within the application. For example, linking the Job Postings table to the Alumni table through the constituentid foreign key was crucial. This linkage not only facilitated the retrieval and display of job postings relevant to specific alumni but also enhanced the overall functionality and user experience of the application. The extensive use of foreign keys throughout the database was vital in maintaining data accuracy and enabling complex queries that support core functionalities, such as matching alumni with relevant job opportunities and mentorship programs.

This phase of project implementation laid a solid foundation for the Alumni Network, ensuring that the database was robust enough to support comprehensive data management and dynamic interactions within the application. The strategic application of PostgreSQL features, such as sequences for auto-incrementing keys and the COPY command for efficient data insertion, played critical roles in optimizing both the setup and the initial population of the database. These collective efforts ensured that the system was not only functional but also equipped to effectively serve the alumni community.

Application Development

As I ventured into developing the Alumni Network Shiny application, the challenge was not only to create a functional interface but also to ensure it was robust, secure, and intuitive, mirroring the complexities and capabilities of systems like Raiser's Edge. My experience at UTD greatly influenced my approach to incorporating advanced features such as user authentication and interactive data management.

The application's interface was crafted using the Shiny framework, which facilitated the creation of a dynamic and responsive user experience. Starting with the Login module, I implemented secure input fields for username and password, which were the gatekeepers to the system. Successful authentication dynamically determines, and grants access based on user roles, safeguarding sensitive information like job details and alumni contacts from unauthorized access. This level of security was imperative, reflecting my understanding of similar safeguards within Raiser's Edge.

```
# Define UI for the application
# Define UI for the application
ui <- fluidPage(
  useShinyjs(),
  navbarPage(
    theme = shinytheme("flatly"),
    "Alumni Network",
    id = "navbar_tabs",
    tabPanel("Login",
             id="login",
             fluidRow(
               column(2,
                      textInput("username", "Username"),
                      passwordInput("password", "Password"),
                      actionButton("submit_login", "Submit", class = "btn btn-primary")
               )
             )
    ),
    tabPanel("Alumni Listing",
             dataTableOutput("alumni_table")
    ),
```

Figure 2: Screengrab from a section of Shiny App Code

Each module within the application was designed with a specific focus:

- Alumni Listing and Alumni Create: These modules are vital for maintaining an active and updated alumni database, allowing admin to view detailed profiles and both admin and users add new alumni information respectively.
- Job Postings and Create Job Posting: These modules provide a platform for alumni and admin to post and view job opportunities, facilitating professional growth within the network.
- Events Listing and Create Event: These modules allow users to stay informed about new events, enhancing connectivity and interaction among alumni.
- Mentorship Listing and Create Mentorship: These features support the establishment and management of mentor-mentee relationships, crucial for ongoing professional development within the alumni community.

On the backend, the application's server logic was meticulously developed to interact seamlessly with the PostgreSQL database, managing real-time data operations. Using the RPostgres package, the server handles data retrieval, and updates efficiently, ensuring the user interface remains up-to-date and reflective of the database's current state. Reactive outputs are employed to refresh user views instantly when data changes, such as when new job postings are added, or events are updated.

The system's capability to manage user sessions was another critical aspect, enhancing security and personalized user experience by controlling the visibility and accessibility of various application panels based on the user's login status and role. This setup not only

reinforced the system's security but also ensured that each user interaction was smooth and tailored.

Each step, from designing the user interface to integrating complex server-side logic, was a piece of a larger puzzle of building a comprehensive system. The focus on user authentication and secure data handling was particularly influenced by the best practices observed in Raiser's Edge, ensuring that Alumni Network was not just functional but also secure and professionally aligned with industry standards.

Testing and Evaluation

Throughout the development of the Alumni Network, I engaged in an extensive process of functional testing, an essential step in ensuring that all aspects of the application performed as intended. This phase was particularly instructive as it allowed me to apply theoretical knowledge in practical scenarios and observe the real-time effects of interactions within the system.

The testing phase involved a combination of automated and manual testing strategies to thoroughly vet the system across multiple scenarios. One of the first tests conducted was to verify the user authentication process. It was crucial that the system accurately authenticated user credentials and appropriately restricted access based on predefined roles. Further tests included data insertion checks within various modules of the application. For instance, I manually entered new alumni data in the 'Create Alumni' module and followed up by populating the 'Create Job Posting' module with job opportunities linked to these alumni entries. Each insertion was a test case to validate not only the integrity of the

data entry processes but also the relational aspects of the database, such as foreign key linkages.

As part of my learning process, I frequently inserted new events through the 'Create Event' module and set up mentorship pairings in the 'Create Mentorship' section, each time monitoring how these entries were handled and reflected in the database. This hands-on approach helped ensure that the underlying database dynamically updated with every new entry or modification, providing a real-time validation of the system's responsiveness and data handling capabilities.

Engaging with the initial user feedback was a critical component of the development cycle. Early users of the application, which were mainly some of my family and friends, provided insights that were instrumental in refining the application. For example, feedback on navigating between different modules led to enhancements in the interface to facilitate smoother transitions and more intuitive interaction pathways. Users also suggested improvements in the clarity of error messages and confirmation prompts, which I implemented promptly. Each iteration was followed by a re-evaluation phase where these adjustments were tested to confirm their effectiveness and ensure they met user expectations.

The testing and evaluation phase was not just about ensuring the application worked correctly but also about learning from each test case, understanding user needs, and continuously striving to enhance the system. This phase underscored the importance of meticulous testing and user feedback in developing a reliable and effective application,

providing me with invaluable insights into practical aspects of system development and user interaction design.

Challenges and Solutions

During the development of the Alumni Network, I encountered several technical challenges that enhanced my learning in database management and application development. Each challenge provided a unique learning opportunity that deepened my understanding of real-world system implementation.

A significant challenge was integrating the Shiny application with the PostgreSQL database while ensuring data consistency and security. This integration required meticulous configuration, which taught me about the balance between system functionality and security. This experience was pivotal in understanding the complexities of backend processing and its implications for user-facing functionalities.

Additionally, the project taught me the importance of a well-structured database schema. I learned how to effectively manage table relationships and enforce data integrity through foreign keys and constraints. This was crucial for maintaining the system's reliability and preventing data anomalies.

The benefits of modular development and testing became evident throughout the project. This approach facilitated easier updates and iterative improvements based on user feedback, which significantly influenced the refinement of the user interface. Making the interface more intuitive and user-friendly enhanced overall user engagement and satisfaction. Reflecting on these challenges, the project was a profoundly educational

experience. These lessons will guide my future projects, enhancing both technical robustness and user experience.

Conclusion

The development of the Alumni Network for my Information Management course provided valuable insights into designing a database system aimed at enhancing alumni engagement and career support. Although this system has not been deployed beyond the academic setting, it represents a theoretical exercise in how such a platform could serve the UTD community and other HEIs.

The prototype designed through this project centralized access to job postings, mentorship opportunities, and professional events, showcasing the potential benefits of such a system. Future enhancements could include mobile optimization to improve accessibility and integration with platforms like LinkedIn to offer personalized career support.

Alumni Network project laid a foundational framework for a system that, if implemented, could significantly benefit alumni interaction and career development. This experience has been instrumental in my growth as a graduate student, understanding the complexities of database systems in supporting community engagement.

Appendices

- Live Alumni Network Shiny App: Visit the application designed to enhance alumni engagement through job postings, mentorship, and event management (For ease of access, Username: Admin, Password: 123).
- GitHub Repository: View the source code here.
- Few screengrabs of the application are shared below.

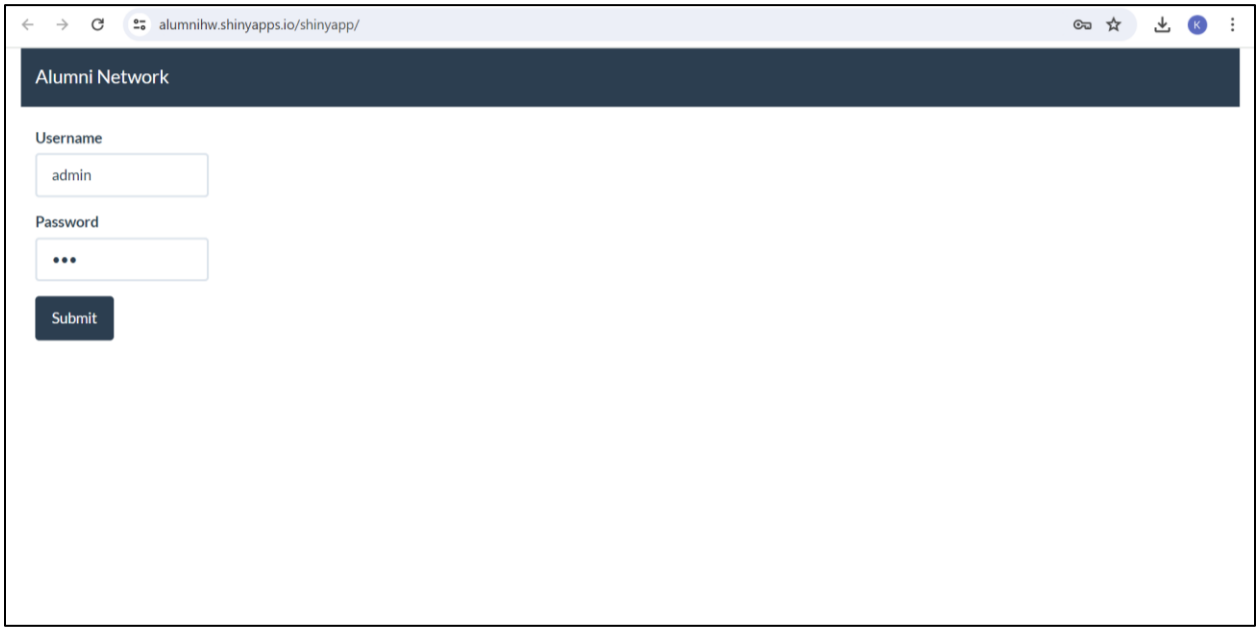

Figure 3: Login Page

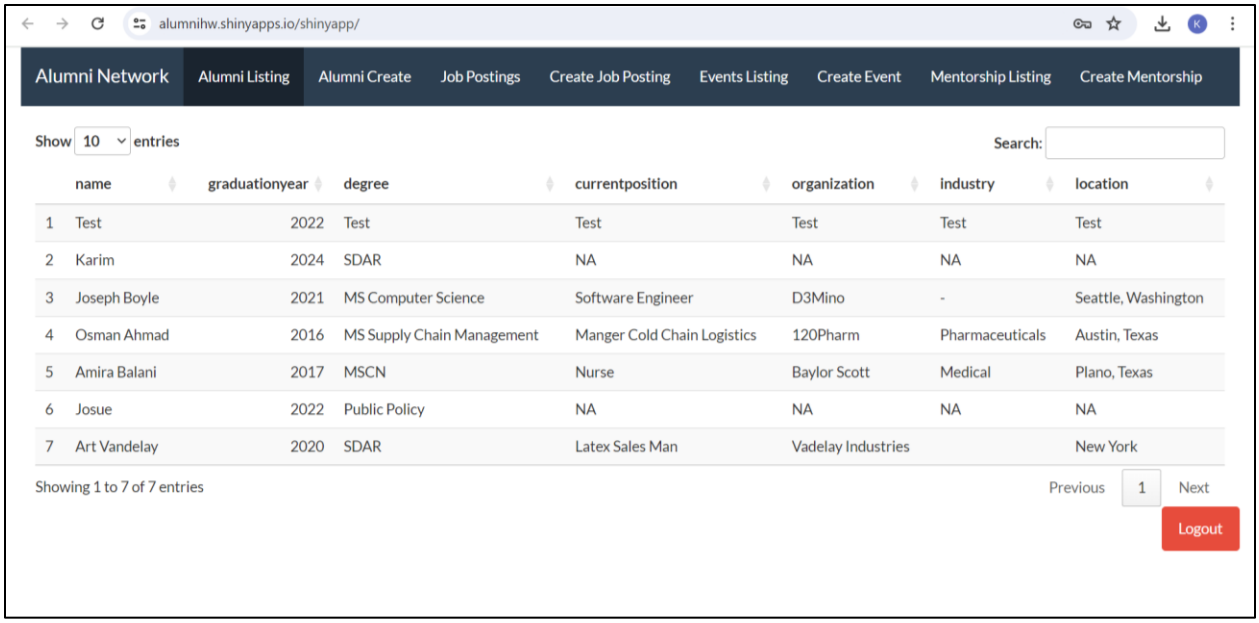

Figure 4: Admin Access

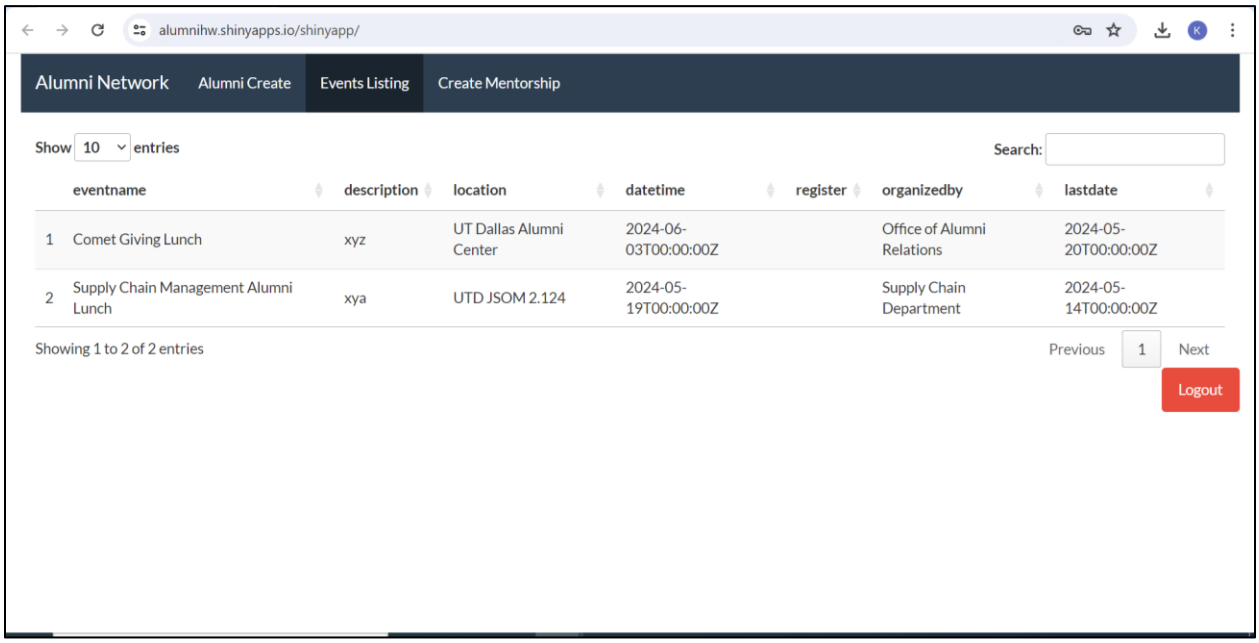

Figure 5: User Access