Surmounting Obstacles for Academic Resilience: A Dynamic Portal for Supporting an Alliance of Students with Disabilities

Alexis N. Petri[1,2(✉)], Duy H. Ho[1,3], Ye Wang[1,3,4], and Yugyung Lee[1,3]

[1] University of Missouri-Kansas City, Kansas City, MO, USA
[2] NSF's Eddie Bernice Johnson INCLUDES Initiative TAPDINTO-STEM Alliance, Auburn University, Auburn, AL, USA
petria@umkc.edu
[3] UMKC School of Science and Engineering, Kansas City, MO, USA
[4] UMKC School of Humanities and Social Sciences, Kansas City, MO, USA
https://www.tapdintostem,https://www.soar-ai.com

Abstract. Description of the development and use of a dynamic portal for supporting an alliance of colleges and universities focused on supporting students with disabilities and transitioning to careers in science and technology. Called SOAR, the portal is designed to support separate institutes achieve collective impact through shared measures. Significant aspects of SOAR are the user-driven design with three different communication roles, dynamic generation of survey forms, the ability to schedule surveys, collecting data through the surveys, and data presentation through dynamic chart generation. SOAR utilizes and advances the best practices of Universal Access and is central to the alliance's ability to empower individuals with disabilities to live their best lives. One of the most interesting features is the ability for different institutes to customize their forms and collect campus-relevant data that can be changed and the application of machine learning to produce the dynamic chart generation. SOAR allows the alliance to meet individual campus needs and the reporting and evaluation needs of the National Science Foundation.

Keywords: User-friendly Survey App · Inclusive User Experience · Tailored User Interface · Interactive Data Visualization

1 Introduction

In a knowledge-based, technology-driven economy, earning a college degree is requisite for many people, whether for career or personal achievement, and benefits both the individual and society [9]. People with disabilities also have career

This material is based upon work supported by the National Science Foundation under Grant No. 2119902.
Authors would like to acknowledge Ahmed Alanazi for his contribution to the mobile app development and Nichole Stahly for her work on form design and as NSF INCLUDES backbone associate.

M. Antona and C. Stephanidis (Eds.): HCII 2023, LNCS 14021, pp. 1–21, 2023.
https://doi.org/10.1007/978-3-031-35897-5_26

or personal achievement goals that require college. According to the U.S. Bureau of Labor Statistics, 27.7% of individuals with disabilities have a bachelor's degree and are employed, compared with 73.2% of those without disabilities. Additionally, individuals with disabilities were more likely to work part-time (29.0%), work in service occupations (18.2%), or work in transportation and material moving occupations (14.6%), compared with those without a disability (16%) [12]. Success for individuals and for communities requires trajectories of opportunity [6]. In the U.S., we are failing to make available trajectories of opportunity for people with disabilities.

Education, at all levels, strives to be evidence-based. In the U.S., what institutions of higher education, and associated scholarship, know the least about are students with disabilities. One reason may be because reporting outcomes data for students with disabilities is not typically required. Achievement gaps must be closed, but it is difficult to ascertain the achievement gaps for students with disabilities. Students with disabilities often face barriers that hinder their persistence [19]. Increasing numbers of researchers are focused on broadening participation [7], especially since the National Science Foundation launched the Eddie Bernice Johnson INCLUDES Initiative, and yet, disability is rarely a data point for making evidence-based decisions about student outcomes.

Working to achieve collective impact for one of the most significantly underrepresented groups in STEM education and employment, researchers at the University of Missouri-Kansas City have designed a dynamic data portal to support efforts to better understand the successes and barriers of students with disabilities. Named Surmounting Obstacles for Academic Resilience, and referred to as SOAR, this accessible portal was developed as the shared measurement system for the Eddie Bernice Johnson INCLUDES Initiative: The Alliance of Persons with Disabilities for Inclusion, Networking, and Transition Opportunities in STEM (TAPDINTO-STEM), which seeks to increase the number of postsecondary degrees completed by students with disabilities. Shared measurement is one of the five core elements to achieving a powerful collective impact, the commitment of a group of important actors from different sectors to a common agenda for solving a specific social problem, in addition to having a common agenda, mutually reinforcing activities, continuous communication, and a backbone that oversees the work of the collective group [17]. In this nationally collaborative alliance composed of 31 higher education institutions, UMKC serves as one of six regional hubs and the backbone for TAPDINTO-STEM, while SOAR collects individual and institutional data alliance-wide and brings it together in a way that faculty and advocates can use and understand.

While the first step is to develop a functional data collection system that utilizes and advances the best practices of Universal Access, the researchers also understand that the eventual research must tackle the whole system. This research converges around systematic data collection, student policies and practices, and the current societal understanding of empowering individuals with disabilities to realize their best lives. SOAR engages with students, faculty, and institutional data at four points: connection, entry, progress, and completion. Having specific evidence will allow institutions, faculty, and researchers to better consider how their structures, systems, policies, and personnel support or impede student success. If institutions can identify, understand, and help students successfully navigate these points, then they have an increased opportunity to promote protective factors for momentum outcomes and work to remove or lessen the occurrence and the severity of their impact to avoid losses.

In this paper, we explore how the portal supports programming focused on students with disabilities through architectures and tools for universal access, design for all best practices, and interaction personalization. Through the design of SOAR, researchers strive to achieve the following aspirations:

- To make the human-computer interface intelligent instead of static.
- Incorporate an individual workflow based on role, dissemination of data collection, and backed by an integrated database.
- Use an integrated database to connect different sources of data.
- Dynamically generates forms with an autofill option that knows where the user left off.
- Dynamically generates charts that can be automated with machine-learning-led identification of important parameters.
- Data schema design can be part of the research.
- Part of the interaction personalization minimizes questions and maximizes program personnel's understanding of learning outcomes based on the data collection.
- Try to minimize questions and maximize learning outcomes from data collection.

2 Related Work

There is an urgent need to introduce more flexible data management systems into higher education, which are traditionally dominated by relational databases (RDBMS) [16]. The urgency comes from a few sources. First, there has been plenty of research on warehoused data for enterprise solutions, but studies on data warehouses for higher education have lagged behind [16]. Data warehouse research is often bounded by the specific usage setting, for example, telecommunication business [14], agriculture [20], supply chain [24], and academic research [3]. There are largely two streams of data warehouse research in higher education. Tria, Lefons, and Tangorra [11] were among the earlier studies that introduced enterprise data warehouse solutions to higher education. Additionally, e-learning has motivated the design and implementation of data dashboards that collect heterogeneous data and visual analytics for quality assurance [15]. In general, there has not been sufficient research on the use of data warehouses in higher education settings, even though data-driven decision-making is urgently needed for institutes to implement changes to better serve all students.

Second, one of the needed transformational changes is to address the lack of diversity in STEM education and the tech industry [2,18]. These changes require a systematic approach that involves multiple institutions and recognizes various perspectives of stakeholders. In terms of data warehouse design, the decision-making and strategies in such a setting cannot become data-driven without a DW [16]. Existing relational databases used by organizations may have different data schemas. Integrating these existing relational databases into one "data warehouse" must address the complexity of the global and local perspectives of stakeholders. "A schema is a description of a piece of the information system" [4]. Data schema integration for local and global queries is a challenging task due to the various perspectives of the stakeholders [5].

In multi-institutional settings, there is very limited research on satisfying the needs of stakeholders from multiple institutions/organizations. Specifically, to what extent and to whom should be involved in the designing process of a data warehouse is a complicated question to answer. Additionally, given the different hierarchies within organizations, how do we create various management roles and access levels that can serve the needs of cross-institution and single-institution planning? Current research on ontology matching investigates element-level and structure-level semantic and syntactic matchings, by employing language-based, string-based, and graph-based techniques, to name a few [21]. With so many questions remaining unanswered, research is needed to find the best methods for integrating data schema from multiple higher educational institutions.

The third challenge is automation. The growing volume and complexity of data have posed challenges to the traditional approach to constructing data warehouses [23]. The tech industry (e.g. Google, Facebook, Twitter, Amazon, etc.) plays the leading role in "big data" solutions, motivated by large amounts of data from Web platforms [23]. As a result, "Not only SQL" (NoSQL) data management systems have attracted a lot of attention due to their flexibility of handling data [23]. NoSQL treats data as key-value dictionary pairs, and each record does not have a prior schema [8]. Due to their scalability and flexibility, NoSQL databases have become the mainstream of big data technologies [13].

In the educational setting, large amounts of data are accumulated from multiple relational databases year after year. Accordingly, implementing "big data" solutions to higher education requires automation. Phipps and Davis [1,22], as one of the earlier studies, presented a user-driven approach to automating the conceptual schema design and evaluation for data warehouses. They pointed out that the initial determination of measures, facts, or events was the most difficult and often needed to be done manually. The automation could help designers to conceptualize data warehouse schema, and users should be consulted on their querying needs. In recent years, the potential for utilizing word embedding and machine learning in automatic ontology matching has also attracted attention [21]. With the rising popularity of NoSQL databases, Bouaziz, Babli, and Gargouri [8] proposed an algorithm to extract data schema from NoSQL records, to assist designers in data warehouse schema design.

Accordingly, three research questions guide this study:

- RQ1: What is the process of designing data warehouse solutions to reflect stakeholders' perspectives from multiple higher ed institutions?

- RQ2: What are the best strategies for automating data warehouse design in such a setting?
- RQ3: How can the data warehouse be designed to support decision-making in such a setting?

3 The SOAR Portal

3.1 User-Driven Design

How to implement user-driven design has been a challenge in data warehouse design due to the huge communication gaps between users and system designers [5]. In this project, communicating with participating stakeholders was crucial to the success of the data warehouse design. Thus, the user-driven design was mandatory. To ensure success, three different communication roles were defined. The first role involved human-to-human communication, directly communicating with the stakeholders. This person was familiar with higher education management. The responsibility of this role was to listen to the stakeholders, collect information on existing relational databases, and seek clarification for the data warehouse design team. The second role handled content-to-content communication. The person was an interaction expert [10] who understood the human and the machine sides of data warehouse design. The responsibilities were to manually examine and compare the data schemas of the existing relational databases by consulting stakeholders' inputs, and to create an integrated data dictionary and a data schema that could be parsed by machines. The third role dealt with content-to-machine communication. This person was an expert in computer science. The responsibilities were to implement the automation of data warehouse design.

3.2 Design of SOAR Portal

The SOAR portal provides an easy-to-use human and machine interface that makes it simple for students, student mentors, teachers, staff, hub lead/campus lead, and hub lead administrators to communicate and share data. Their responsibilities and authority would be explained in terms of SOAR's two main parts: 1) collecting data through a series of surveys and 2) analyzing and showing the data in the form of a dashboard with several charts.

The customized interfaces contain various features and services based on the individual's tasks and responsibilities. The most interesting aspect of this system is that tasks (survey forms) are always being made and sent to users as online or mobile applications, depending on their responsibilities. Through the individualized interfaces of Web and mobile applications, data is collected and stored in serverless distributed databases for dynamic analysis and visualizations shared through intelligent dashboards. The frontend will dynamically link to the backend, including real-time data storage and machine learning, using a cloud-based Jupyter notebook. Thus, this is an end-to-end system that includes data collection via online or mobile applications, data analysis, and analytics via populating real-time databases with data. User interfaces are always being changed by adding new forms, changing forms that are already there, and moving forms around. Users can be deleted from or added to our systems via our applications.

Dynamic Generation for Survey Form Application. SOAR's forms look like Google Forms, which are designed to support inquiry by giving users different ways to ask questions. The dynamic form generation application is made so that a survey form can be used to make a mobile or web application automatically. Automatic application generation will be generated dynamically with multiple capabilities, such as a speech interface, encrypting sensitive information (e.g., student names, GPAs, etc.), survey scheduling and publication, anonymous nickname generation, and automatic validation. This allows the alliance to meet the needs of individual campuses and programs and their students and faculty, while still meeting the reporting and evaluation needs of the National Science Foundation.

A form is composed of a set of different question types:

- Multiple Choice: One answer per question can be selected from an available set of items.
- Check-boxes: Multiple answers can be selected from an available set of items, including "other" for short answers.
- Drop-Down Menu: The respondent selects the appropriate response from a drop-down menu. With this option, the respondent can upload a file from Google Drive in the form of an answer sheet to answer a question.
- Scale Ranging: Respondents answer the question using a scale from 0 to 10.
- Multiple Choice Grid: The system gives the respondent choices from which to choose one answer per row. This option allows individuals to select one or more replies per row.
- Start Date: This is set to determine when a particular survey is available.
- End Date: This is set to determine when a particular survey is closed.

Survey Forms. The SOAR portal has a number of different survey forms that are made and sent to users based on their roles and requirements for NSF's Eddie Bernice Johnson INCLUDES Initiative data collection. Survey questionnaires are designed for the following users:

- Hub or Campus Lead: This group is administrators. They are given the highest level of accessibility. This form collects institutional data, such as resources, disability services, etc.
- Faculty or Staff: This group includes all faculty and staff who have participated in the project. This form collects faculty/staff's activities and interactions with students.
- Student Mentor: This group includes participating students, who, meanwhile, serve as mentors for other students. This form collects the demographics, academic performances, participation, and mentoring status.

– Student: This group includes all student participants. This form collects demographics, academic performances, and participation.

Survey Scheduling. The survey forms will be arranged based on the beginning and end dates. Before the start date, the form will not be posted, and after the end date, the form will no longer be accessible. In addition, form prerequisites are defined. For instance, the demographic form is a prerequisite for the performance form. So, it will be decided that the student performance form will be filled out after the demographics form.

Data Collection Through Survey Generation. The SOAR portal supports specific users as follows:

– Admin can design, generate, and deploy survey forms with minimal effort.
– Hub or Campus Lead can design, create, and schedule survey forms. The Admin reviews and publishes the forms.
– Faculty and staff are required to complete surveys in the order they are sent to them.
– Student mentors are required to respond to the surveys in the order they are deployed. Student mentors are required to fill out 10 forms. The sole distinction between students and student mentors is that student mentors also act as mentors for students. Thus, the forms for students and student mentors are nearly identical.
– Students have to fill out the surveys in the order that they are given to them. Students are responsible for completing 10 surveys.

Data Presentation Through Dynamic Chart Generation. The dashboard for the SOAR portal is innovative because the charts that are shown are made automatically by (1) figuring out which parameters in the data are most important and putting them in order of importance using machine learning (ML) algorithms; (2) finding a link between parameters; (3) computing the links as a correlation coefficient matrix by using ML; and (4) making individual charts based on the characteristics that are strongly linked. The goal of the intelligent dashboard is to make the work of domain experts easier. Of course, domain experts can also build charts by hand to find connections in the data that were not known before. This automation will be set up as soon as there are sufficient data points from which meaningful information can be elicited.

We also look at the charts suggested by domain experts to get around the limitations of the data-driven dashboard system. If domain experts' charts were suggested instead of machine learning's automated charts, the missing charts would be added to the dashboard. Consequently, the intelligent dashboard system relies on human and computer cooperation.

3.3 The SOAR Portal's Frontend Component

Front-end design considerations for SOAR include (1) User friendliness and accessibility; (2) Customized interfaces; (3) Task completion with minimal effort and time; and (4) Assistance for users with disabilities.

User friendliness: As an example of being user-friendly, scheduled surveys are well-organized based on what users need to do first and when each survey begins and ends. SOAR divides the forms into two categories - completed and unfinished - so the user can easily figure out what comes next. It's easy for Lead Hub Administrators to create, update, copy, and send surveys to all roles or institutions, or just to some of them.

Customized interfaces: The user interface will differ depending on the user's identification. For instance, if the user is a student who has already finished three forms and has two more to do, the user interface will change. This interface will show unfinished forms in the order they need to be finished, along with start and end dates and requirements. In addition, the user can update completed forms using the auto-fill options of prior responses.

User efficiency: Users can update completed forms using the auto-fill options from prior responses so that they are not answering the same questions multiple times.

Supporting people with disabilities: Users with disabilities are supported throughout SOAR. Each form is reviewed for universal design for evaluation so that the questions are clear for a broad array of respondents. Users have multiple ways to interact with SOAR. Users may complete the forms in various ways: They may complete forms using speech-to-text where SOAR reads aloud the forms and the respondents speak the answers, or SOAR reads aloud the forms, and the respondent types the answers, or the respondent reads the forms and types the answers. The determination is fluid, so respondents do not need to decide that their forms will always be presented and answered in one way.

The interfaces for the personalized services for various individuals are illustrated with screenshots.

Student Interface: The student interface of the SOAR mobile app, shown in Fig. 1, includes five forms that require completion. When users access the login screen (Fig. 1(a)), those who haven't registered can proceed to the registration screen (Fig. 1(b)). After successfully logging in, the main screen is displayed (Fig. 1(c)). By selecting the "Forms" button, a list of available forms is presented (Fig. 1(d)), allowing the user to choose a specific form to complete (Fig. 1(e)). The form can be completed by typing or using voice commands, which can be activated by holding down the microphone icon. The list of voice commands includes:

Selecting options, e.g., saying "My gender is male" to select the male option Typing text, e.g., saying "My first name is John" to type "John" into the text field. Resetting a field, e.g., saying "reset my first name" to empty the first name text field. Turning on/off the speak out feature by pressing its corresponding button. This feature speaks outs the selected part of the application, but only if the user's phone is not in silent mode. Finally, the user can submit the form by pressing the submit button. After submission, the user can view the remaining forms or applications to be submitted.

When it is turned on, the app will announce which part of the application the user has selected, for example, the first name field. When it is turned off, the user will no longer hear the speech from the app. Note that to use this feature, the user's smartphone must not be in silent mode. Once the form is complete, the user can submit it by pressing the submit button. After submitting the form, the user can see a list of the remaining forms or applications that need to be submitted.

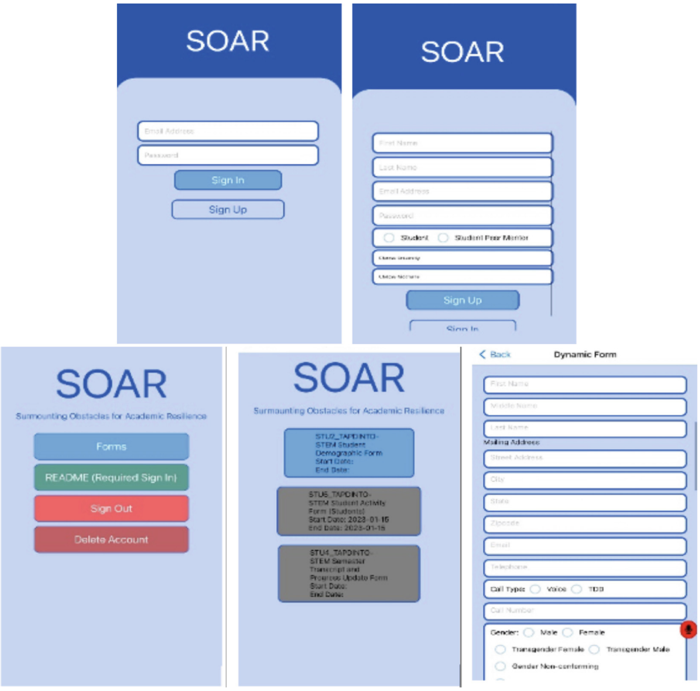

Fig. 1. SOAR Student's Mobile App: (a) login screen (top left) (b) registration screen (top right) (c) main screen (bottom left) (d) list of available forms (bottom middle) (e) filling out the form by typing or using voice command (bottom right)

Student Peer Mentor and Faculty/Staff Interface: Since student peer mentors and faculty/staff have equivalent system rights compared to students (meaning they can only complete assigned surveys), the interfaces are similar in terms of layout and design, except for the individual sequence of forms specified by Hub Lead Admins that need to be completed by each role respectively.

Hub Lead/Campus Lead Interface: Hub Lead/Campus Leads have the ability to create, edit, clone, and delete surveys. However, they cannot publish

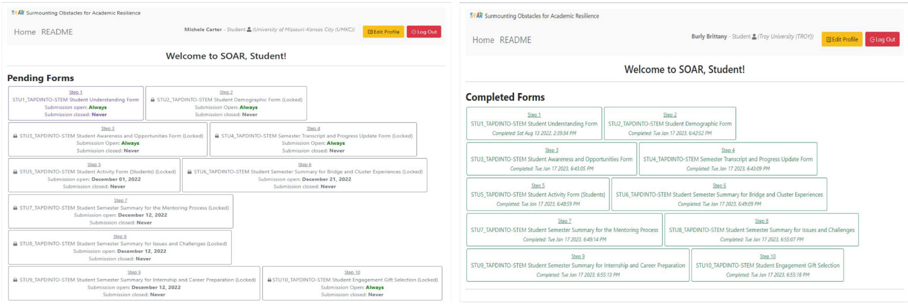

Fig. 2. Student Web Interface Dashboard: (a) Initial State (left) (b) All Surveys Completed (right)

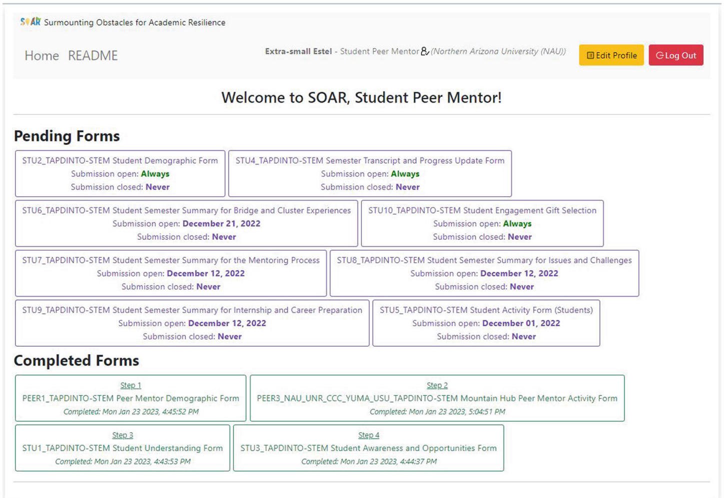

Fig. 3. Student Peer Mentor Web Interface

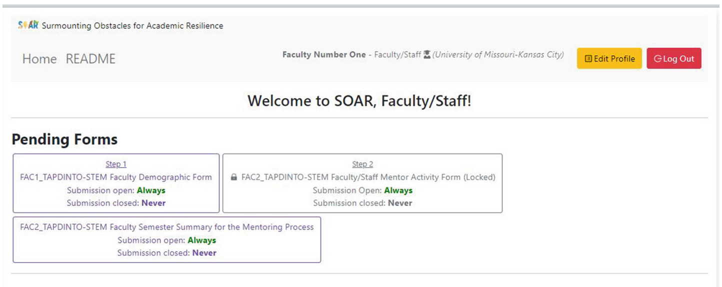

Fig. 4. Faculty/Staff Interface

the newly created forms. Figures 2, 3, and 4 display web interfaces for students, student mentors, and faculty staff. Surveys undergo review and approval by Hub Lead Admins before being published. Figure 5 shows the Campus/Hub Lead Web Interface. Figure 6 illustrates the survey creation process, including specifying prerequisites and publishing the survey. Figure 7 compares form control options for Campus/Hub Leads and Hub Lead Admins, highlighting their respective responsibilities.

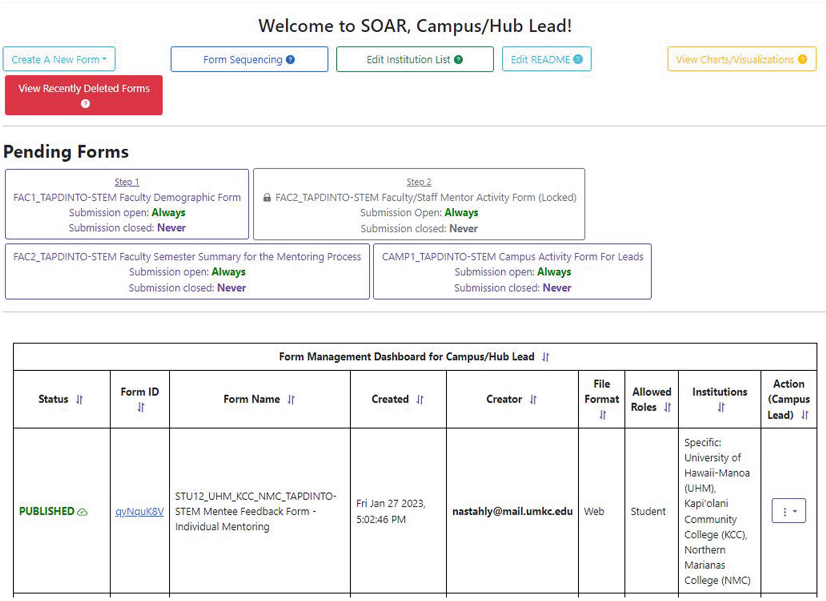

Fig. 5. Campus/Hub Lead Web Interface

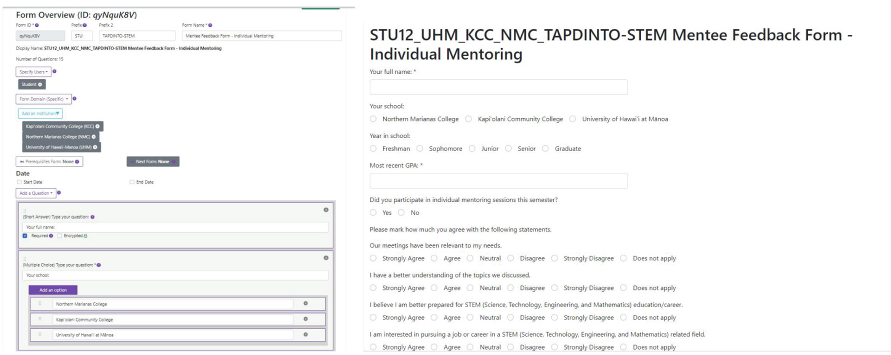

Fig. 6. Survey creation process (a) Specify all prerequisite requirements, access restrictions, and questions. (b) Publish and Render Survey

Fig. 7. Campus/Hub Lead vs Hub Lead Admin's form control options. Campus/Hub Leads can create, edit, and delete forms, while survey status is managed by Hub Lead Admins only. Hub Lead Admins can also download form responses in CSV format.

Hub Lead Admin Interface: Hub Lead Admins possess higher authority and access additional options compared to other roles. This includes Fig. 8 for the Established Form Sequence for Students, Student Peer Mentors, and Faculty/Staff. Figure 9 shows additional rights that Campus/Hub Leads and Hub Lead Admins share, such as updating institutions, editing README pages, recovering deleted surveys, and modifying account type. Figure 10 depicts the pending form workflow, Fig. 11 is the Hub Lead Admin interface, and Fig. 12 displays a sample user roster exclusively for Hub Lead Admins, containing name, email, institution, and account type.

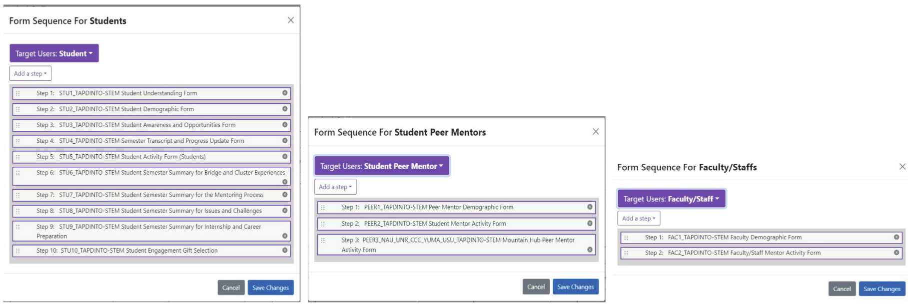

Fig. 8. Established Form Sequence for Individual Roles: (a) Students (b) Student Peer Mentors (c) Faculty/Staff

SOAR Frontend Implementation: The frontend of SOAR was created using the two frameworks listed below. First, the Web application is the main SOAR frontend, and all user roles can use all of its features. The frontend provides each of the five main types of users (i.e. student, student peer mentor, faculty/staff, hub lead/campus lead, and hub lead administrator) a dedicated interface. We build the SOAR web application via React (a component-based Javascript frontend library) and deployed to the public domain through Firebase Hosting and Vercel via a private GitHub repository. The established development and deployment workflow through GitHub branches is seamless because stable and nightly

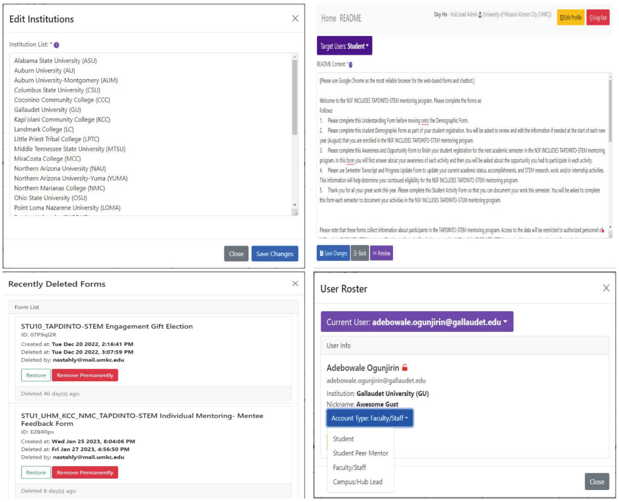

Fig. 9. Additional rights shared by Campus/Hub Leads and Hub Lead Admins. (a) Update list of institutions (top left) (b) Edit individual README pages for each account type (top right) (c) Recover recently deleted surveys (bottom left) (d) Modify account type and role (bottom right).

Status	Form ID	Form Name
AWAITING APPROVAL	ABMSoOGH	Sample Campus Lead Form
UNPUBLISHED	ABMSoOGH	Sample Campus Lead Form
PUBLISHED	ABMSoOGH	Sample Campus Lead Form

Fig. 10. Pending Form Workflow after Creation: Awaiting Approval from Hub Lead Admin (top), Rejected by Hub Lead Admin and Unpublished (middle), Accepted and Published to Users (bottom).

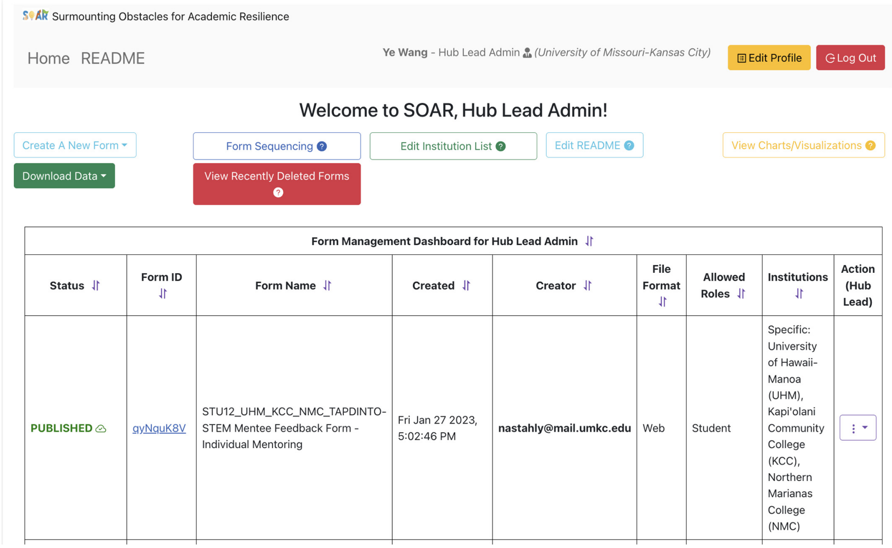

Fig. 11. Hub Lead Admin Interface

FirstName	LastName	email	Institution	Role
[illegible]	[illegible]	[illegible]	San Diego State University (SDSU)	student-mentor
[illegible]	[illegible]	[illegible]	Northern Arizona University (NAU)	student-mentor
[illegible]	[illegible]	[illegible]	University of Hawaii at Manoa	student
[illegible]	[illegible]	[illegible]	University of Missouri-Kansas City (UMKC)	hub-lead-admin
[illegible]	[illegible]	[illegible]	University of Wisconsin-Milwaukee (UWM)	student
[illegible]	[illegible]	[illegible]	University of San Diego (USD)	student
[illegible]	[illegible]	[illegible]	Northern Arizona University (NAU)	student-mentor
[illegible]	[illegible]	[illegible]	University of Missouri-Kansas City (UMKC)	student
[illegible]	[illegible]	[illegible]	Columbus State University (CSU)	faculty
[illegible]	[illegible]	[illegible]	Troy University (TROY)	student
[illegible]	[illegible]	[illegible]	Coconino Community College (CCC)	student-mentor
[illegible]	[illegible]	[illegible]	Auburn University (AU)	faculty
[illegible]	[illegible]	[illegible]	Auburn University (AU)	faculty
[illegible]	[illegible]	[illegible]	Northern Arizona University (NAU)	student
[illegible]	[illegible]	[illegible]	University of Missouri-Kansas City (UMKC)	hub-lead-admin
[illegible]	[illegible]	[illegible]	Wichita State University (WSU)	student
[illegible]	[illegible]	[illegible]	Northern Marianas College (NMC)	faculty
[illegible]	[illegible]	[illegible]	Purdue University (PURDUE)	student
[illegible]	[illegible]	[illegible]	University of Missouri-Kansas City	student
[illegible]	[illegible]	[illegible]	Purdue University (PURDUE)	student
[illegible]	[illegible]	[illegible]	University of Wisconsin-Milwaukee	administrator
[illegible]	[illegible]	[illegible]	University of San Diego (USD)	student
[illegible]	[illegible]	[illegible]	Northern Arizona University (NAU)	campus-lead
[illegible]	[illegible]	[illegible]	Point Loma Nazarene University (LOMA)	student
[illegible]	[illegible]	[illegible]	University of Hawaii-Manoa (UHM)	faculty

Fig. 12. Sample exported user roster from the Google Cloud Build backend, accessible to Hub Lead Admins only. Information includes name, display email, institution, and account type.

(development) versions are managed by separate branches and deployed through separate subdomains. The publicly available soar-ai.com is the main branch and the most stable version. Vercel is responsible for managing Git pushes, gathering packages, building environments, and ultimately ensuring the continuous deployment and integration of the project into our custom domain is successful. For authentication, database connection, and storage, the frontend communicates with our serverless backend through Firebase Authentication, Firestore, and Storage. In terms of visualizations, the data-driven charts are rendered through React ChartJS 2, a React-specific implementation for Chart.js, the most popular chart library. It does not only provide interactive charts but also supports many popular chart types including bar, line, bubble, pie, doughnut, scatter, radar, and polar. However, its limitation is that these charts provide only univariate and bivariate aggregations due to the underlying constraints of frontend-only renders. Other complex tasks such as correlation, regression, and classification involve heavier computational resources and specific data-oriented platforms, requiring a dedicated backend. Therefore, to further gain more powerful insights from the responses from the portal, we created our charts via a Python script that supported data analysis such as Pandas and Plotly, built an interactive web application through Flask, and deployed it through our secure private endpoint services via JupyterDash and Ngrok (more details in Sect. 3.4).

Second, the mobile application is developed from React Native, a software development framework with React principles with native functionalities enabled. The mobile application is tailored for students or student peer mentors who prefer a convenient way to fill out the surveys without having to type a lot of answers. The mobile application shares real-time access to the same Firestore database and, therefore all user and survey information is always up-to-date and in sync with the React web application. Moreover, to leverage the power of speech recognition, we developed our cross-platform mobile application with high compatibility for both Android devices along with iOS phones and tablets (Fig. 1).

3.4 The SOAR System's Backend Component

As illustrated in Fig. 15, the SOAR architecture consists of 3 main components: Serverless backend and database, Python backend with dedicated domain, and multi-platform frontend, in which each component is composed of several smaller modules and services. The functions and communication are also clearly defined within and between each component.

Serverless Backend. The serverless backend is structured on top of Google's Firebase ecosystem, which provides powerful services to create a secure application in terms of scalability, reliability, and maintainability. The Firebase authentication service provides a ready-to-use and secure system to create and manage our users via multiple options such as email, phone number, and social media accounts. The database was built using Firestore, a well-known real-time NoSQL

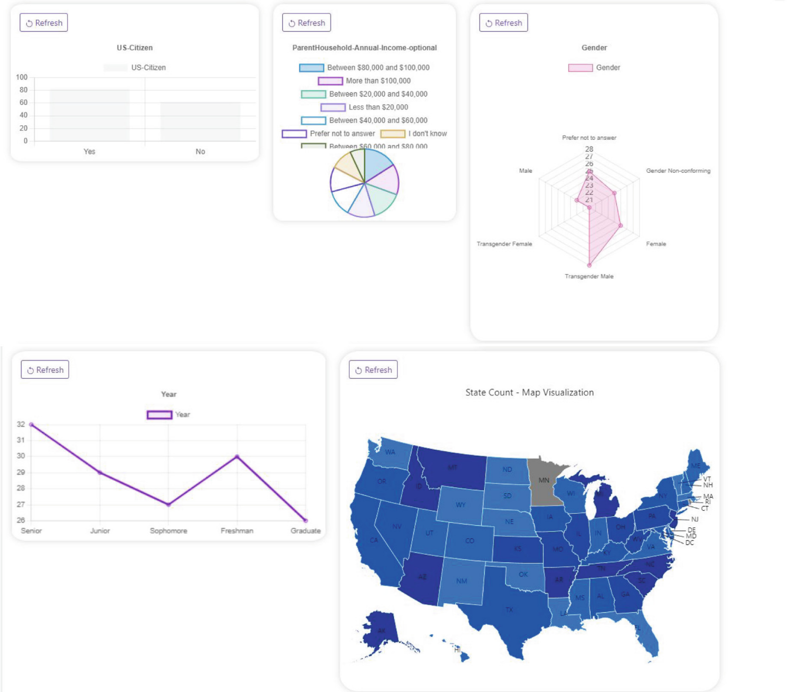

Fig. 13. Visualization Dashboard (Frontend Render) (a) Bar Chart (top left) (b) Pie Chart (top middle) (c) Radar Chart (top right) (d) Line Chart (bottom left) (e) Geographic Map Chart (bottom right).

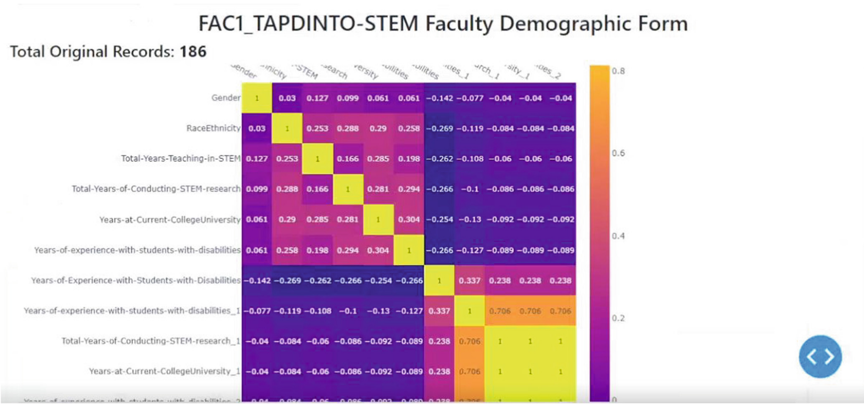

Fig. 14. Advanced Visualization: correlation matrix. These types of advanced visualizations were created from a dedicated Python script, deployed through JupyterDash, and hosted via Ngrok.

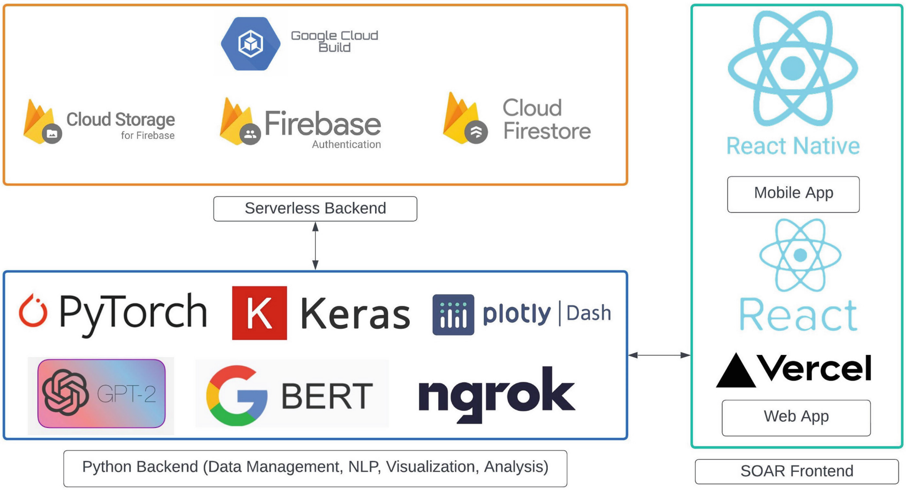

Fig. 15. SOAR Architecture

database, not only because of its flexibility in structuring data and automatic scaling, but also its integration with other Google and third-party services like Authentication, Cloud Messaging, Google Cloud Console, Storage, and Google Analytics written across multiple native SDKs and programming languages such as Javascript, C#, Python, Java, and more. Thus, the information stored in our database can be fetched from multiple platforms and elevated even further with numerous powerful backend tools. Even though the database is designed to work with all kinds of structured, semi-structured, and unstructured data, it also provides a hierarchy for these data structures in the form of documents, collections, and sub-collections to make single-index and multi-index querying easier across the whole database.

In addition, the database includes information about survey questions, schedules, requirements (form sequence, deadline, prerequisites), access restrictions for account types and/or institutions (universities), and real-time updates of survey progress and fulfillment. Aside from primitive and nested objects, we also utilize Storage that allows developers to store user-uploaded files such as transcripts, custom PDF and text documents, and audio files. Additionally, Storage provides ready-to-download URLs, which provide users and administrators a way to download their uploaded files in the database directly from the frontend application. Additionally, since authenticated user information is considered sensitive and should not be retrieved as batches from the React frontend, exporting user data (as requested by the Hub Lead Admin) requires the Python SDK of Firebase instead of the Javascript SDK. Therefore, this feature is written in Python and deployed as an API endpoint through the Google Cloud Build (which containerized the script, the Firebase Admin SDK, and the remaining packages required in the environment in a Docker file) that is available 24/7 as a standalone URL embedded in a button on the web application. Once clicked, the API will invoke a method that fetches, constructs, and aggregates the latest user data, and finally download a file directly to the user's computer in the CSV format.

Python Backend. Even though Firebase provides a well-integrated system to establish, store, and manage data, its ability to perform complex data-related tasks is still limited, whereas data comprehension and analysis tasks are vital to understanding what users would like to convey and improving the quality of the service. Since the SOAR database and authentication system are hosted on the cloud that is seamlessly integrated with various native libraries, we had an opportunity to expand the application further with the second component in the architecture, our dedicated backend that carries out extensive analysis, efficient querying, data insight, interactive visualization, in-depth interpretation, and prediction using powerful state-of-the-art techniques such as natural language processing (NLP), machine learning (ML), and deep learning (DL).

The backend is based on the JupyterLab environment written in Python. For data comprehension, we utilized GPT-2 and Bert models (built on top of PyTorch and Keras). Each model is trained on specialized tasks suitable to their architecture and design. Our chatbot, besides being able to follow the strict order of questions established by the Hub Lead Admin in the survey, is also enhanced by OpenAI's GPT-2 with the ability to generate sentences based on user inputs and engage in a natural conversation with the users. GPT-2, an autoregressive decoder, is extremely good with text generation. On the other hand, tasks regarding understanding the data, such as sentiment analysis and question answering, are performed with Google's BERT model, a deep learning model that aims at learning the words and the surrounding context.

For comprehensive data analysis and analytics, we utilized Firestore data export and constructed tables to facilitate various statistical methods such as filtering, dimensionality reduction, correlation, regression, classification, topic modeling, and clustering. These techniques enabled us to identify significant variables, uncover data relationships, and derive valuable insights. Interactive charts visualizing these connections were created using Plotly. To make these visualizations accessible, we deployed them locally as a web application through JupyterDash and made them publicly available using Ngrok. JupyterDash, built on Flask, serves as a development server, while Ngrok enables secure deployment of the Flask-based JupyterDash web application with HTTPS, requiring minimal code modification. Refer to Figs. 13 and 14 for illustrations. These technologies provide an excellent opportunity to incorporate complex interactive visualizations into the web application to accommodate the evolving backend features. By leveraging their computational power and resources, we achieve decoupling of complexity while seamlessly integrating our data-driven backend with the React frontend. Finalized charts, designed and implemented by administrators, will be containerized using Docker. Depending on their frequency and computational workload, these containerized charts will be either pushed to the cloud using Google Cloud Build or deployed on our dedicated server.

SOAR Mobile Apps Support Speech Interface: SOAR delivers speech recognition, synthesis, and message translation utilizing robust APIs such as Google Translation API and Speech-To-Text to encourage diversity and inclusion. SOAR does audio transcription by using the Speech-To-Text API, which is one of the most advanced voice recognition services, to turn audio communications from moderators and participants into state-of-the-art captions. This enables multi-modality and privacy protection. The domain-specific deep learning model used in this study was trained with a large dataset and advanced techniques to give the most accurate captions regarding meaning and grammar. Each temporary audio file added to our database is then sent to our cloud-based speech recognition engine, which makes real-time captions for participants and moderators. We provide students and student mentors via SOAR's mobile applications (iOS and Android).

4 Discussion and Future Work

In this paper, we discussed the design and development of a data portal to support a large multi-institution alliance that supports college students with disabilities who are majoring in and transitioning to careers in STEM. SOAR is designed to support shared measures and evidence-based decision making. It connects operations, internal evaluation, external evaluation, and embedded research. Through use, the application will become more focused on the specific project. In the future, we would like to encourage the use of this system for other large collaborations. Having real-time access to data can help large collaborations stay focused and adapt to student needs as they emerge. Additionally, a system such as SOAR provides users with an advanced interface that is accessible, and intelligent, connects data systems from multiple sources, aids implementation, and advances discovery.

References

1. Abai, N.H.Z., Yahaya, J.H., Deraman, A.: User requirement analysis in data warehouse design: a review. Procedia Technol. **11**, 801–806 (2013)
2. Alfred, M.V., Ray, S.M., Johnson, M.A.: Advancing women of color in stem: an imperative for us global competitiveness. Adv. Dev. Hum. Resour. **21**(1), 114–132 (2019)
3. Amorim, R.C., Castro, J.A., Rocha da Silva, J., Ribeiro, C.: A comparison of research data management platforms: architecture, flexible metadata and interoperability. Universal Access Inform. Soc. **16**(4), 851–862 (2017)
4. Batini, C., Di Battista, G., Santucci, G.: Structuring primitives for a dictionary of entity relationship data schemas. IEEE Trans. Software Eng. **19**(4), 344–365 (1993)
5. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. (CSUR) **18**(4), 323–364 (1986)

6. Bierbaum, A.H., Vincent, J.M., Katz, J.P.: Planning for opportunity: linking smart growth to public education and workforce development. In: Handbook on Smart Growth, pp. 207–227. Edward Elgar Publishing (2022)
7. Blaser, B., Ladner, R.E.: Why is data on disability so hard to collect and understand? In: 2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), vol. 1, pp. 1–8. IEEE (2020)
8. Bouaziz, S., Nabli, A., Gargouri, F.: Design a data warehouse schema from document-oriented database. Proc. Comput. Sci. **159**, 221–230 (2019)
9. Cimera, R.E., Thoma, C.A., Whittenburg, H.N., Ruhl, A.N.: Is getting a post-secondary education a good investment for supported employees with intellectual disability and taxpayers? Inclusion **6**(2), 97–109 (2018)
10. Collins, H.M., Evans, R.: The third wave of science studies: Studies of expertise and experience. Soc. Stud. Sci. **32**(2), 235–296 (2002)
11. Di Tria, F., Lefons, E., Tangorra, F.: Research data mart in an academic system. In: 2012 Spring Congress on Engineering and Technology, pp. 1–5. IEEE (2012)
12. DOL: Table 1. employment status of the civilian noninstitutional population by disability status and selected characteristics, 2021 annual averages. In: U.S. Bureau of Labor Statistics Economic News Release (2022). https://www.bls.gov/news.release/disabl.t01.htm
13. Emani, C.K., Cullot, N., Nicolle, C.: Understandable big data: a survey. Comput. Sci. Rev. **17**, 70–81 (2015)
14. Garani, G., Chernov, A., Savvas, I., Butakova, M.: A data warehouse approach for business intelligence. In: 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 70–75. IEEE (2019)
15. Guitart Hormigo, I., Rodríguez, M.E., Baró, X.: Design and implementation of dashboards to support teachers decision-making process in e-assessment systems. In: Baneres, D., Rodríguez, M.E., Guerrero-Roldán, A.E. (eds.) Engineering Data-Driven Adaptive Trust-based e-Assessment Systems. LNDECT, vol. 34, pp. 109–132. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-29326-0_6
16. Hamoud, A.K., Marwah, K.H., Alhilfi, Z., Sabr, R.H.: Implementing data-driven decision support system based on independent educational data mart. Int. J. Elect. Comput. Eng. **11**(6), 5301 (2021)
17. Kania, J., Kramer, M.: Collective impact. Stanford Soc. Innovat. Rev. **9**, 36–41 (2011). https://doi.org/10.48558/5900-KN19
18. McGee, E.O.: Interrogating structural racism in stem higher education. Educ. Res. **49**(9), 633–644 (2020)
19. Moriña, A., Sandoval, M., Carnerero, F.: Higher education inclusivity: When the disability enriches the university. Higher Educ. Res. Developm. **39**(6), 1202–1216 (2020)
20. Ngo, V.M., Le-Khac, N.A., Kechadi, M.T.: Data warehouse and decision support on integrated crop big data. Int. J. Bus. Process. Integr. Manag. **10**(1), 17–28 (2020)
21. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: A literature review. Expert Syst. Appl. **42**(2), 949–971 (2015)
22. Phipps, C., Davis, K.C.: Automating data warehouse conceptual schema design and evaluation. In: DMDW, vol. 2, pp. 23–32. Citeseer (2002)
23. Sellami, A., Nabli, A., Gargouri, F.: Transformation of data warehouse schema to NoSQL graph data base. In: Abraham, A., Cherukuri, A.K., Melin, P., Gandhi, N. (eds.) ISDA 2018 2018. AISC, vol. 941, pp. 410–420. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-16660-1_41

24. Vieira, A.A.C., Pedro, L., Santos, M.Y., Fernandes, J.M., Dias, L.S.: Data requirements elicitation in big data warehousing. In: Themistocleous, M., Rupino da Cunha, P. (eds.) EMCIS 2018. LNBIP, vol. 341, pp. 106–113. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11395-7_10