Team No. 10 Project Final Design

Team Members:
Bisher Anadani
Johnathon Bliss
Kaiser Mittenburg
Kevin Ray
Vivek Tallavajhala

Project Name

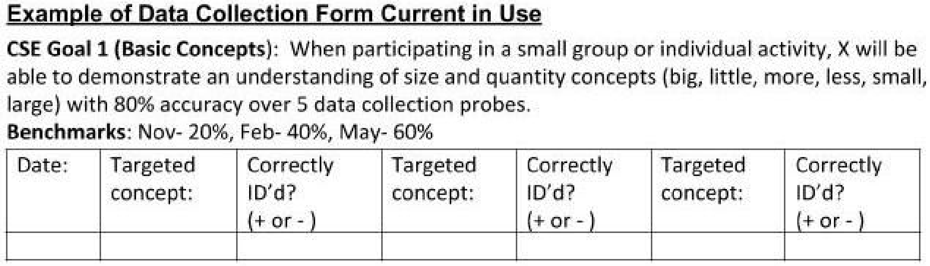
Individualized Education Tracker

Project Synopsis

Custom web form maker that allows teachers to create and customize data collection forms for the purpose of tracking Individual Education Plans.

Project Description

This project is being undertaken at the behest of a special education teacher in the Lawrence school district. The goal is to provide special education teachers with the means of collecting data for Individual Education Plans across multiple platforms and users. With their current resources, the teachers have to do all of the data collection with pen and paper and then store and collate the data sheets for later processing. Some have tried to use google forms for this data collection but it has not been flexible enough for their needs.



Example of Data Collection Form Current in Use

CSE Goal 1 (Basic Concepts): When participating in a small group or individual activity, X will be able to demonstrate an understanding of size and quantity concepts (big, little, more, less, small, large) with 80% accuracy over 5 data collection probes.

Benchmarks: Nov- 20%, Feb- 40%, May- 60%

Date:	Targeted concept:	Correctly ID'd? (+ or -)	Targeted concept:	Correctly ID'd? (+ or -)	Targeted concept:	Correctly ID'd? (+ or -)

The end result will be to provide a custom form maker web application that will allow teachers to build custom data collection forms that will fit the varying needs of each child's Individual Education Plan. This will also require individual accounts and login and the ability to share forms between accounts so that whichever school personnel is assigned to the data collection will have access to the form. The project will also include a back-end database and service to store the form data for later retrieval and use. If there is time and sufficient data, we also plan to implement a means of manipulating, analyzing, and displaying the gathered data for tracking trends.

Project Milestones

First Semester	
Task	**Completion Date**
Gather front-end and back-end requirements	10/21
Map out user-interface	11/23
Define UI and database	2/1

Second Semester	
Task	**Expected Completion Date**
Project Final Description	2/10
Finish Final Implementations and Combine Front and Back Ends	4/12
Finish Testing and Deployment	4/24
Final Video Completed	4/28
Quad Chart Completed	5/1

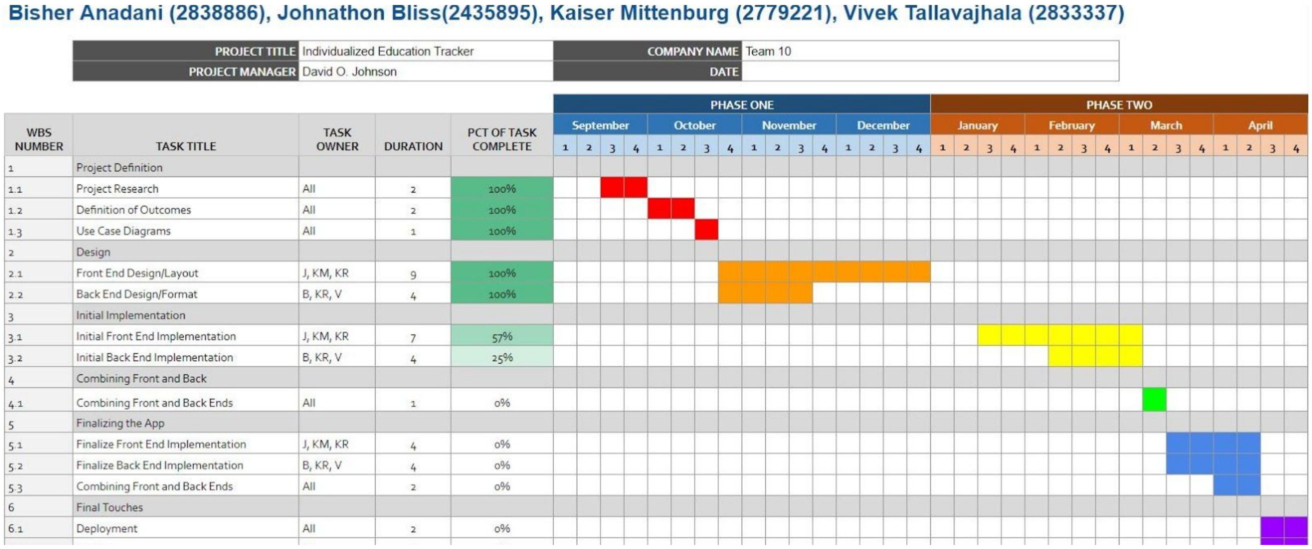
Project Budget:

- Storage database
    - $0 for MongoDB on KU servers
- Website Hosting
    - $0 for KU servers

Work Plan:

- Vivek Tallavajhala - BackEnd Developer
- Bisher Anadani - BackEnd Developer
- Kaiser Mittenburg - FrontEnd Developer
- Johnathon Bliss - FrontEnd Developer
- Kevin Ray - Full Stack Developer

Gantt Chart:



Bisher Anadani (2838886), Johnathon Bliss(2435895), Kaiser Mittenburg (2779221), Vivek Tallavajhala (2833337)

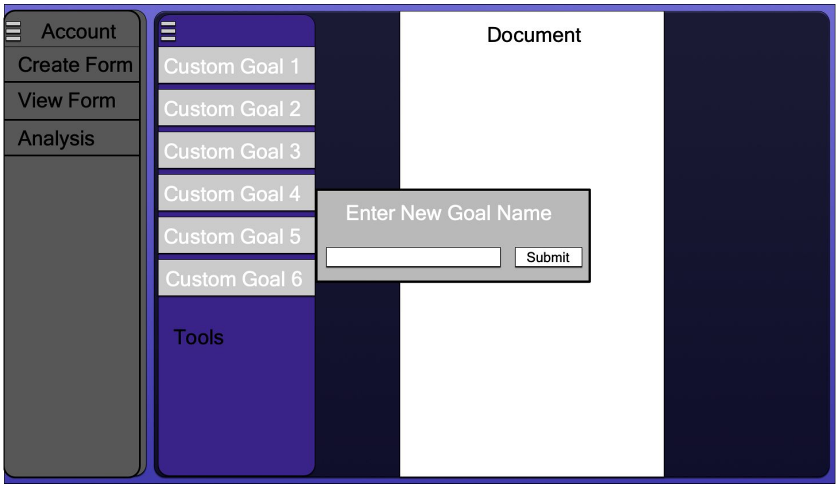
PROJECT TITLE	Individualized Education Tracker	COMPANY NAME	Team 10
PROJECT MANAGER	David O. Johnson	DATE	

					PHASE ONE																PHASE TWO															
					September				October				November				December				January				February				March				April			
WBS NUMBER	TASK TITLE	TASK OWNER	DURATION	PCT OF TASK COMPLETE	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Project Definition																																			
1.1	Project Research	All	2	100%			🟥	🟥																												
1.2	Definition of Outcomes	All	2	100%					🟥	🟥																										
1.3	Use Case Diagrams	All	1	100%							🟥																									
2	Design																																			
2.1	Front End Design/Layout	J, KM, KR	9	100%								🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧																
2.2	Back End Design/Format	B, KR, V	4	100%								🟧	🟧	🟧	🟧																					
3	Initial Implementation																																			
3.1	Initial Front End Implementation	J, KM, KR	7	57%																			🟨	🟨	🟨	🟨	🟨	🟨	🟨							
3.2	Initial Back End Implementation	B, KR, V	4	25%																						🟨	🟨	🟨	🟨							
4	Combining Front and Back																																			
4.1	Combining Front and Back Ends	All	1	0%																										🟩						
5	Finalizing the App																																			
5.1	Finalize Front End Implementation	J, KM, KR	4	0%																											🟦	🟦	🟦	🟦		
5.2	Finalize Back End Implementation	B, KR, V	4	0%																											🟦	🟦	🟦	🟦		
5.3	Combining Front and Back Ends	All	2	0%																													🟦	🟦		
6	Final Touches																																			
6.1	Deployment	All	2	0%																															🟪	🟪

Project Design

How The Software Works:

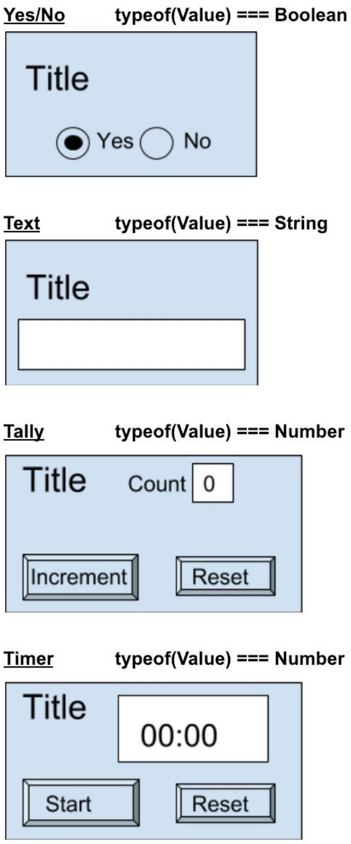
Our software will be used in classrooms during evaluations and must be flexible, catering to desktops, laptops and tablets. Because of this, we have decided to create a web application that can be accessed in a web browser. We have decided to exclude mobile phones as a platform candidate, as there is often not enough "real estate" to work with effectively in regards to screen size, which can cripple usability. We may re-evaluate this once our primary goals and concerns have been addressed.



A major goal we would like to achieve aesthetically is to have our service look visually appealing, and to feel like a web app. We don't want to constantly see large portions (or all) of the screen loading or buffering on simple button clicks, nor have the user feel like they are using a basic html website. Facebook's React javascript library will help us build a beautiful, consistent user interface, that will keep dimensions in tact and bring a polish to our web app that is hard to do with simple html/css alone. Accommodating different screen sizes while keeping identical page layouts and relative dimensions is one of the hardest things to do in a web environment. React will help us accomplish that.

The web app will support a few core workflows such as account administration, form creation, form use, and time permitting, data analysis. A simple navigation column will allow movement between these workflows. A teacher or para educator should be able to manage their collection of students, build forms to track their students, and use those forms. There are many use cases involved in using a form, such as filling it out, updating, printing, emailing and saving to file. We hope to address those, and any others that would be appropriate. Form creation will be accomplished through the use of pop-ups. A button will be provided for creating a new task. That button will trigger a pop-up. From there, the user can enter all of the pertinent information for that task. There is no set standard for IEP forms. Each form is custom tailored to accommodate each student. In order to satisfy the relatively diverse set of goals that can make up an IEP, we will offer a collection of general data recording techniques that will allow for the recording of timed goals, affirmative observation, number of occurrences, accuracy, and more. A strong collection of recording techniques and the ability to create custom forms will give teachers a tool to suit all of their students.

Types of Fields


Yes/No typeof(Value) === Boolean
Title
● Yes ○ No
Text typeof(Value) === String
Title
Tally typeof(Value) === Number
Title
Count 0
Increment
Reset
Timer typeof(Value) === Number
Title
00:00
Start
Reset

For the account administration it will be key to allow the sharing of forms across users. Currently, a teacher creates a physical form but then it is often the paraeducator who uses the form for the data collection. In order to simulate this in our web application, a user who owns a form will need to be able to grant access to the form so that other uses can use it for data collection. This will require user login and credential verification as well storing which users have valid access to which forms.

Being a data driven application, we will store all of our data in a database. When needed, we will ping our MongoDB database to load or store form information. We have decided to architect our app using a non relational, “NoSQL”, schema in hopes of simplifying our page design and data handling at the same time. Form rendering will rely heavily on what type of data is being recorded in the form, thus, our technique of data handling will heavily impact rendering speed, as well as number of renders. Further rationale for choosing NoSQL is discussed in our design constraints section. If our application is a success, and we see high traffic, we may also implement a intermediary data service to act as a distribution layer, which will increase efficiency. Our frontend could even be deployed on a customer-by-customer basis. The modern school district has a complex shared network, fully capable of hosting their own instance of the frontend, allowing use only on their secure network.

Design Constraints:

The largest project requirement is the need for customizable data collection forms. Since these forms need to be flexible enough to correspond to any child's development goals, the ordering and types of entries can not be known prior to the forms creation. This has resulted in a technical constraint related to how the form data is stored. The usage of a traditional relational database is impractical as this would require a table for each type of entry and a means of forwarding to know which field comes next. This would most likely result in longer load times since the form would have to be rebuilt each time it was loaded. Instead, this has forced us to consider the use of a non-relational database schema in order to store each form as a JSON. This allows each form to be stored as it was created and does not require the rebuilding of the form contents each time it is fetched.Another constraint has emerged from the need for portability of the web application. Since this is an application that could be used across multiple school districts or states, it is unknown what browsers each will use. There are various ways to ensure this portability, but one of the simplest for our team was the use of the React JavaScript library. The React library is used and maintained by Facebook and as a result has functionality to support all major browsers.

There is also a budget constraint on the project. Since the EECS has available server space for web hosting, our budget does not include extra money for using external server space or web hosting. As a result, we have decided to use the provided web hosting to host the entire project. This will allow the front, back, and database to all be hosted on the same server. Not only does this reduce cost, but it also has the potential to increase response time between the back end and the database. Since this hosting will not last indefinitely, if the project continues to see use after the semester, it may become necessary to pursue outside hosting options. The decision on whether to pursue that option will hinge to a great extent on how many active users we have and what options we have for monetizing the application.

Ethical & Intellectual Property Issues

Ethical Issues:

The primary ethical issue our project faces in regards to Data, specifically student data and how we protect it. Since we are creating a Web app for IEPs, we will be collecting data on students that will be stored in a database. The ethical issue we face is in regards to what kind of data we store, and making sure we are not in violation of FERPA. To protect against potentially violating FERPA and avoid crossing ethical boundaries, our plan is to only track data needed for the web app, and keeping student information out of it. This way our database isn't storing confidential information, and if it did, we would have an ethical responsibility to put extra effort in ensuring our databases are safe from cyber security threats. This is still obviously a concern for us, but by limiting the confidential information we store we can better control the ramifications if something negative was to happen.

Additionally we have an ethical responsibility to create a platform that is easy to use and intuitive for teachers to design their IEPs. The purpose of our project is assist teachers with the current, manual process in recording data for students, so to ethically and effectively solve this problem, we need to ensure we design our web app with simplicity of use in mind.

Intellectual Property Issues:

The first IP issue we face is in regards to using React. React is an open-source Javascript framework released to the public by Facebook. Facebook has released a licensing model for all their project, including react, called “BSD + patents”. This BDS license grants us a copyright usage license, but we are only granted that license as long as we never sue Facebook for copyright infringement. In the situation we try to sue Facebook for copyright infringement, our patent rights to react are immediately revoked. Additional intellectual property issues may arise depending on what future open source software we use. We will need to continually be familiar with the copyright policies for those software, and ensure we use them in a way following that policy.

Change Log

- The original project had a drag and drop implementation for the form creation. In testing, this was found to hinder user flow and was replaced with a pop-up to gather the necessary task information.
- No longer using AWS DynamoDB for database requirements. Instead, a MongoDB database will be hosted on the EECS servers alongside the front end.
- Updated Gantt chart to reflect the amount of time spent planning out the application and to reflect the reevaluation of how long it will take to complete the front end as compared to how long it will take to complete the back end.
- Updated the milestones to reflect what was actually completed in the first semester. All parts of the implementation are now in the second semester with a finalized date for when testing and debugging needs to be done in time for presentations. Also included parts for the final project description, quad chart completion, and the video completion.