12-8-2020

# School District Boundaries Map

Nick Huffman
*Western Michigan University*, nickhffm@gmail.com

# Level Data District Boundary Maps

## WMU Senior Design Fall 2020

Nick Huffman, Corey Kline

# Table Of Contents

# Abstract

The purpose of this project is to provide a school district boundary mapping feature to a product sold by Level Data called SDVS, which is a plugin used by districts inside of PowerSchool. Using primarily the features offered by Mapbox, We have implemented a React component that is capable of plotting useful data points related to a student and their school district on a map. The tool is designed to be used primarily by school administrators to determine whether or not a student lives within their district boundaries.  The application uses a dataset that is provided by the NCES to determine the correct enrollment boundaries for the districts, and the districts' own data from Powerschool for everything else.

# Problem Statement

## Need

The primary need for this project came directly from school districts that are clients of Level Data.  School district boundaries are notorious for being oddly shaped, so districts often have a hard time determining which district a student should be attending school in. If a student is not within a district's boundaries, they will likely not receive funding from the state for that student.

A secondary need for this project is for districts to be able to determine what school within a district a student should attend. When new students are enrolled, or when a student moves, it is important to make sure they are able to attend the correct school. A visualization showing where students live with regards to the schools is useful in this regard.

## Objectives

The goal of this project is to provide a map component for SDVS that has the school boundary datasets available. This application should display useful, contextual information with minimal user interaction. The dataset used by the map needs to be easy to update, as the NCES provides new data every few years.

## Definitions

Powerschool: A Student information System used by School Districts
SDVS: State Data Validation Suite is a Powerschool plugin offered by Level Data
NCES: National Center for Education Statistics
GeoJson: A standard format that represents geographical features
API: Interface that defines interactions with a service
Address Validation: Checking the validity of an address and getting the coordinates
React: A component based frontend Javascript library
Material Design: A design standard developed by Google

# Problem Analysis and Research

## Choosing a Mapping Service

The first problem we faced was choosing the best mapping service that would fit our needs. Most mapping services charge based on how many queries are made. Our client expected anywhere between 50,000 and 500,000 queries per month. They did not want to pay any more than $10,000 per month for the service. Preferably, they wanted us to choose the cheapest option. With these requirements, we looked at three different mapping services: Google, Here, and Mapbox.

Google is the most commonly used mapping service that we came across. Their maps are highly versatile and allow for drawing boundaries, validating addresses, and drawing various markers on the map. The biggest downside with their mapping service is the price point. This is the most expensive option that we researched. Initially, we began using Google as our mapping service using the free version. We later switched due to cheaper options existing.

Here maps is another versatile option that we researched. This mapping service has all the same features as Google but at a cheaper price point. One concern that we had was the performance of this mapping service, as on one of our computers, it ran rather slowly. Additionally, we worried about the longevity of the service, because it is not as big of a company as Google.

In the end, the option that we chose was Mapbox. This is a service that is used by many well-known applications such as Snapchat, Facebook, and the Weather Channel. It is another versatile option, like Here and Google. One big factor that contributed towards the selection of Mapbox is their Tiling Services. Mapbox allows uploads of GeoJSON data that can be added to Mapbox styles and queried via their API. Additionally, our client began using Mapbox with one of their other services. Therefore, in order to be consistent and not have to add any new services, we selected Mapbox.

## Storing District Boundary Data

Determining the most efficient way to store district boundary data proved to be a challenge. Originally, the plan was to create a spatial database that would be queried by the front end and return the data as GeoJSON features. The data is downloaded from the NCES official website as a shape file. We would then upload this data into our spatial database. We began implementing this until our client decided that they did not want to host the database. That is when we had to change things up and find the best way to store this data without hosting it ourselves. That is when we found the Mapbox Tiling service.

The Mapbox Tiling service offers a way to upload GeoJSON data into a spatial database that stores features in chunks. There is a limit to the size of the files that can be uploaded, so the data must be split up. Then, the tile query API can be used to query all files based on coordinates, and the properties of the boundary are returned. Using the properties, a Mapbox style can be queried in order to get the polygon's coordinates that are used to render the boundary.

# Requirements

The project requirements as they stand now at the completion of the project:

- Display a student and student data on a map
- Display a district boundary on the map calculated using one of the following locations
  - Location of the district office
  - Location of the student's current school
  - Location of the student's address
- Display the neighboring district boundaries on the map
- Show all the schools in the district
- Calculate the distance between the student and each school
- Calculate which schools are closest to the student
- Integrate the map into SDVS
- Create a process to update the stored boundary data when census information is updated

# Standards and Constraints

## Applicable Standards

### GeoJson

GeoJson is a standard format for dealing with Geographic data in web applications.  We used this data specification throughout our application and when dealing with the Mapbox API.

### Shapefile

Large amounts of geographic data are stored by governmental organizations within shapefiles. All NCES data we dealt with comes from shapefiles.

### React

We designed our front end application in accordance with React standards.  Using this we created a responsive application that does not unnecessarily render components.

### Material Design

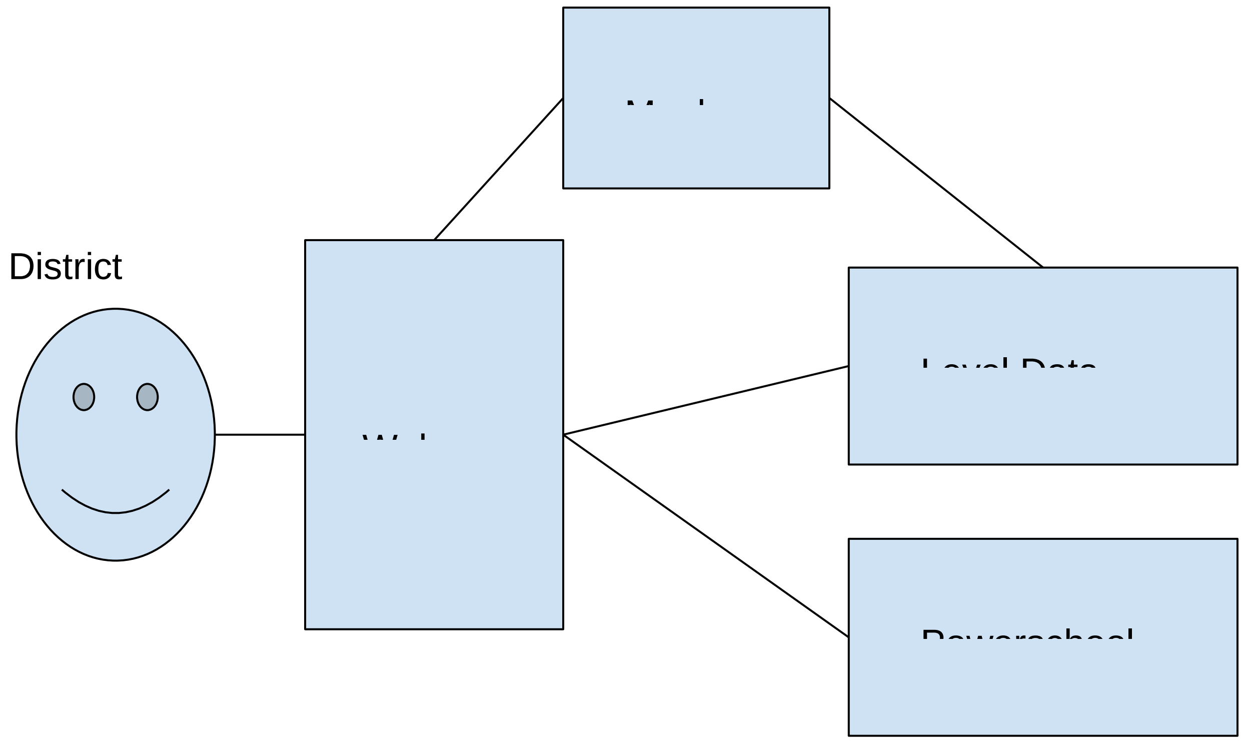
This standard was implemented using the Material UI library for React. This standard implements universal grid-based layouts, animations, padding and margins.

## Constraints

Pricing of a maps service was the main constraint of our project.  Google maps is the most feature rich service, but it comes at a larger cost.  Ultimately we decided to use a service called Mapbox.

# System Design

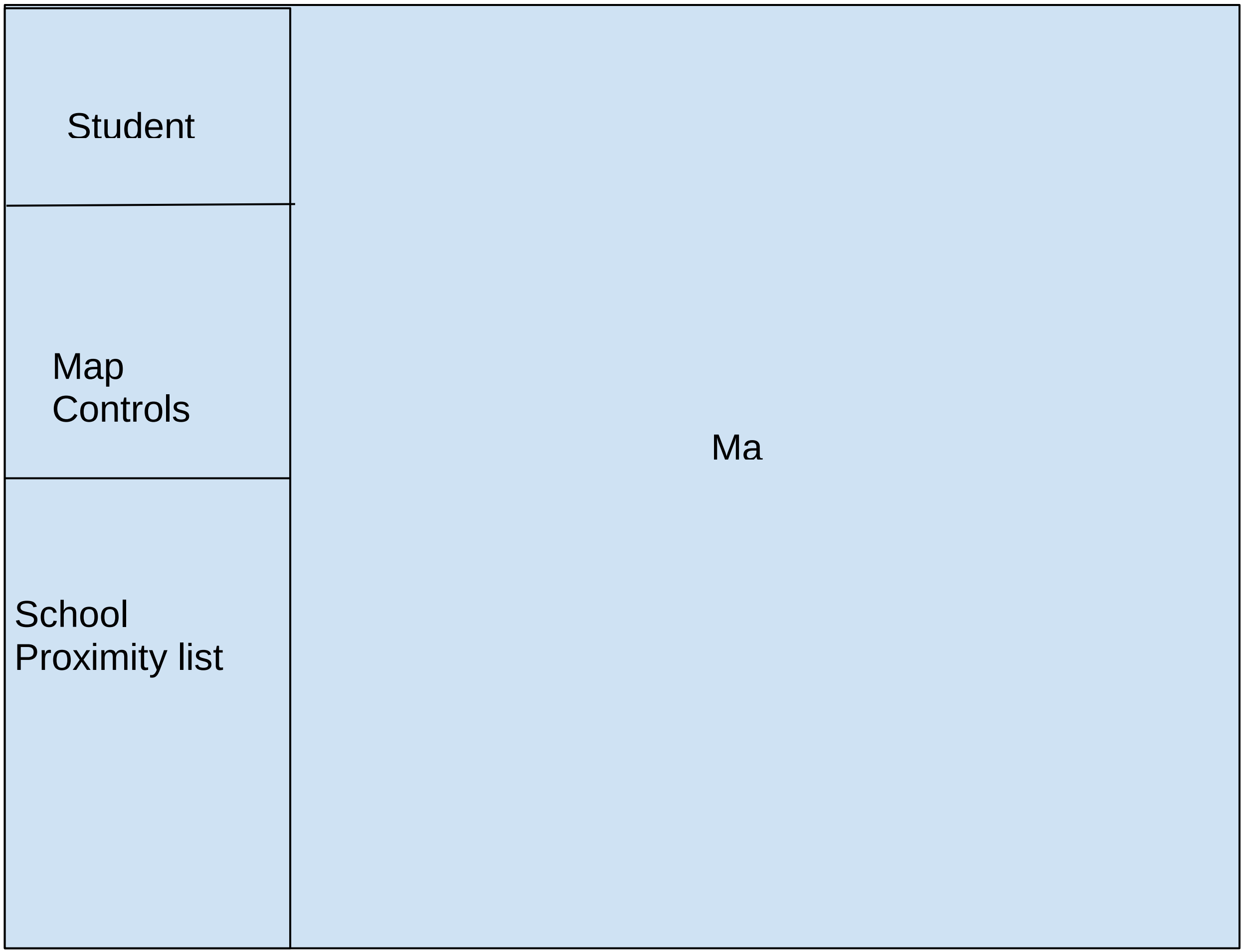
## Application as a Whole

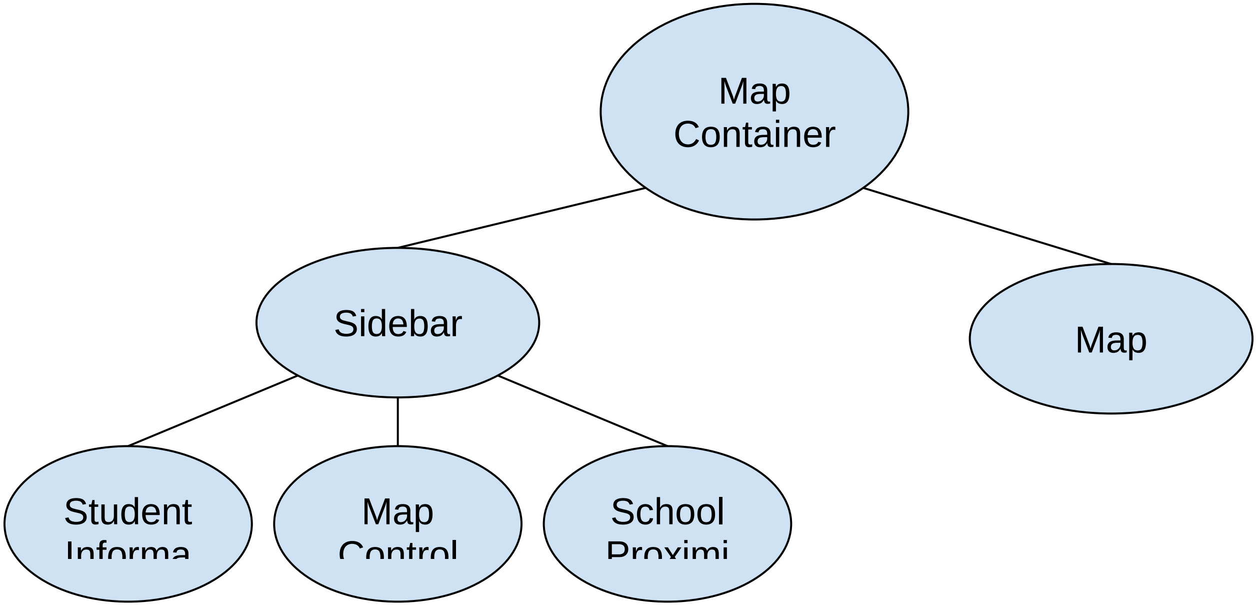


While our Application functions mostly as a normal webpage, however the authentication requests to Mapbox are proxied through Level Data Servers.  This functionality was provided by Level Data to account for security concerns with the API key.

## Web Page Design

The webpage user interface was designed to resemble the interface of other SDVS services. Material design standards were used to keep the features consistent.


Student
Map Controls
School Proximity list
Ma


Map Container
Sidebar
Map
Student Informa
Map Control
School Proximi

# Ethical Considerations

With this project, the most important ethical considerations were dealing with scope and accessibility.  School district boundaries have been a point of contention in the US, as many historical boundaries have been set through racial segregation.  It is worth considering adding features to the map that highlight these sorts of phenomena.  Ultimately, it was not in the scope of the application to make identifying these cases an explicit feature.  The map does however provide useful visualizations to see multiple boundaries at once, and I hope this can help administrators realize any inequities.  For accessibility, there is a question of who this application should be available for.  Personally, if I were developing this outside of Level Data, this would be made available publicly and be fully open source.  Since this was being developed as part of a product set, we did not have much of an option.

# Testing

## Unit Testing

The repository this project lives in makes use of the Jest Framework for JavaScript Unit testing. These tests are run whenever the javascript is compiled by webpack and reports errors in the dev console.  A deployment of the repo will fail if the unit tests fail.

## Integration Testing

This component was developed to work standalone, as it was not quite decided how it would be used by Level Data until a few months prior to the conclusion of this project.  We have been testing integrating this feature with Powerschool and the latest changes to the Level Data SDVS product.

## Peer review

In order to ensure the code is up to standards, it is reviewed by 2 other Level Data engineers, including the lead web developer.

## User Acceptance Testing

We had weekly rounds of user acceptance testing and were able to get very rapid feedback as a result.  This almost always yielded excellent feedback from the client that allowed us to implement an intuitive, feature-rich interface.

# Results

## Realization of requirements

Most of the final requirements for the project were met. We made an aesthetically pleasing map with all of the functional abilities that were requested by the client. When you open the map, you can clearly see where the student is located, along with the schools in the district, and the nearby district boundaries. There is a sidebar that allows the toggle of certain features on the map, and the selection of what schools to show.

The only requirement that was not fully met is the integration with SDVS. This is currently in progress and will be completed before the map is deployed for customer use. One reason this was not able to be fulfilled was due to changing requirements over the duration of the project. Also, our client had a change in staffing, so we had to transition between multiple developers who had different ideas for the project.

## Realization of Standards and Constraints

All standards and constraints that were specified were followed. This was done so that when the map was no longer standalone, it could be seamlessly integrated into the existing SDVS project. No new dependencies, standards, or constraints had to be added.

## Testing results

When running the map standalone, there were no testing failures at runtime. After integrating with the SDVS project, all pre-existing unit tests passed as well.

# Future Work

Both members of this project (Nick Huffman and Corey Kline) have been employed by Level Data to work full time after graduation, so providing support will be an easy matter if it is determined to be a priority.  Nick is working on the Customer Experience team, where he largely works with School Districts to maintain and update their data integration services through Level Data.  Corey Is working on the engineering team, and is more likely to be brought onto a project of this type.

The highest priority is to get this service deployed into production, which requires finishing integrating the component within the larger SDVS project.

In terms of new features to be added, there may be some possibility to display individual school enrollment zones within each district, but this data is currently not reported at a national scale.

# Conclusion

With the completion of the District Boundary Map project, we created a tool that can be used by districts that maps useful data points, such as district boundaries, school and student locations. In order to accomplish this, we had to follow specific constraints and requirements. With the completion of the project, the constraints and requirements were fulfilled. We retrieved data from the NCES, along with PowerSchool, to provide useful, accurate, up-to-date information. This information is displayed in a contextual way that requires little interaction. On the map, the student’s location is displayed, alongside student data such as their address and grade level. Additionally, the specific district boundary is selected based on district office and school locations, and displayed in relation to the student. An option is provided to show neighboring district boundaries. Schools inside of the district are displayed on the map as well, and can be filtered based on grade level and whether the student is currently enrolled at that school. The distance between the students and schools are calculated and displayed on the sidebar. This map will be further integrated into PowerSchool alongside Level Data’s SDVS. A process was created to update the boundary data due to census and NCES updates. Unit testing was implemented through the Jest Framework, and is run on compilation of webpack. Any failures are then reported to the console. Peer review and user acceptance testing became integral parts of the development process, as weekly meetings lead to constant feedback. Because of the constant feedback, we were able to implement plenty of intuitive features in a way that was desirable to the client. In the end, the full integration with PowerSchool was not entirely completed. This was due to changes in requirements over the course of the project, along with a change in staff overseeing the project. Some future work will need to be completed in order to complete the integration. After this is completed. Level Data and their clients will have a fully functioning district boundary tool integrated into their SDVS services that will be used by school districts across the country.

# References

Facebook Inc. (2020). *React – A JavaScript library for building user interfaces*. React. https://reactjs.org/

Mapbox. (2020). *Documentation*. https://docs.mapbox.com/

Material-UI. (2020). *Material-UI: A popular React UI framework*. https://material-ui.com/

*Powerschool*. (2020). Powerschool. https://www.powerschool.com/

U.S. Department of Education. (2020, December 1). *National Center for Education Statistics (NCES) Home Page, a part of the U.S. Department of Education*. National Center for Education Statistics. https://nces.ed.gov/

# Appendices

## Project Management Plan

# Minimum Product Release

## Milestone 1 - Standalone map viewer react app 10/23

- [DB2-1] Working dev environment
- [DB2-8] District boundary shown on map
- [DB2-3] Show schools within the district boundary
- [DB2-2] View students and student data on map with hover details

## Milestone 2 - Standalone map styling and functionality 11/6

- [DB2-9] Create a button that opens a modal with the map
- [DB2-10] Create a hovering legend
- [DB2-11] Match the style of the map to that of RTR reports
- [DB2-12] Single control that only toggles grade levels displayed
- [DB2-13] Use Turf JS bounding box used to calculate min and max zoom levels

## Milestone 3 - District Boundary data available to the application

- [DB2-14] Determine whether Mapbox tilesets can be used or if a backend needs to be hosted (11/6)
- App is able to fetch US District boundary information at any location
- If possible, have available individual school zones
- Filter visible boundaries based on grade ranges of the boundary

## Milestone 4 - Integrate with SDVS

- [DB2-5] Show district boundary for current PowerSchool district
    - Figure out which district based on PowerSchool

# Progress Reports

## 10/11/2020

### Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

We forked the client's repository and started a new branch to create the development environment. Then, we got the client's API keys for Mapbox. Then, we documented a process in which shape files are converted to geojson files and then split up into smaller files that can be uploaded as Mapbox tile sets. With the geojson files uploaded, we created multiple layers and customized them so that the map on the development environment shows district boundaries. We then started digging more into the documentation and experimented with adding coordinate points and icons to the map.

## Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

This past week we met with the client a couple times in order to discuss how we would go about setting up the dev environment and how we should go about creating our endpoint in the project that we forked. He went over with us some of the libraries, dependencies, and webpack development. Our client told us that we are on the right path. We did not demonstrate a prototype.

## Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

We are currently working on showing a point on the map in relation to its district's boundaries. Our planned date of completion for this part is Oct. 23. We are currently on schedule for this part.

## Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

One issue is displaying coordinates for schools and students on the map. Another issue is showing individual district boundaries rather than all at once. Another issue is displaying the boundaries at all levels of zoom.

## Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

One problem that seems to be recurring is showing boundary detail on the map without costing a fortune. There are a lot of different features in Mapbox, but there are a lot that cost extra. We need to spend extra time going through the documentation to make sure we stay on track.

I foresee more issues popping up as we get deeper into implementing Mapbox. So far, we have been able to go to a developer at Level Data with questions and issues. He has worked with Mapbox before, so he has been a very helpful resource.

# 10/19/2020

## Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

We added a basic React Map component, a UI with material design, different layers that correspond to the UI, and mock data stored in the UI. We explored options with tilesets and datasets in Mapbox and how we can use different APIs to query this stored data. We created our react components as functional components and began converting them to class components in order to have better state management.

## Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

We met with Level Data on Friday. First, we went over our current progress and the accuracy of the currently mapped out boundaries. Then we touched on project scope and minimum requirements. Also, we talked about setting up a document with story descriptions and due dates that align with our Jira board to better organize our projects.

## Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

*We are currently working on displaying district boundaries in relation to a student. The original due date was October 23. We are behind on schedule.*

## Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

We are currently working on displaying a district boundary, schools, and students as separate layers on the UI with mock data. This will then lead into building a backend with real data that can be plugged into the UI.

## Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

The biggest problem is that the Mapbox tilequery API does not return coordinate data. It only returns a single point within a boundary. The solution is that we will have to create a backend database that stores boundary data. To address this, we will have to set aside time to setup the backend.

# 10/26/2020

## Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

First, we worked on getting the data for school enrollment zones. Then, we added a few more boundaries to the mock data. Then we worked on fixing the refresh bug that re-renders the map every time a layer is selected or de-selected. Additionally, we have been working with Mapbox support to try to resolve our previous issue with querying tilesets.

## Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

Yes, we discussed the progress of the UI, and what to do about a backend. They like the direction we are going in but gave a lot of suggestions for improvements that we can make to the UI.

## Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

We are currently working on improving the UI so that it displays boundaries, schools, and students. This will be completed by November 12. We are currently on schedule.

## Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

Currently, we are still trying to resolve the re-rendering issue. Also, we are trying to refactor and clean up the project. Another issue is working with Mapbox support and their Tilequery API.

## Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

There have been no new problems that have arisen in the past week. We are still working to figure out how we will be querying boundaries, and whether we will need to create a spacial database.

# 11/02/2020

## Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

We have tweaked the UI to better match the client's feedback and have added more advanced controls to interface with the data. We have been in contact with Mapbox support on whether or not they can host our data in an actionable way

## Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

We met with our client last thursday, they were very pleased with the results. We generally have all the features in place, we just need to make sure the data gets hosted.

## Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

We are working on getting the data hosted and polishing off the UI. I would say we are on schedule, but any more setbacks would put us in trouble. Our goal for this is next friday Nov. 13th.

## Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

Boundary data hosted and available to the application.

## Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

We have had problems with getting the map data hosted. As a result we may have to spin up our own spatial database, although Level Data would like to avoid this if possible.

# 11/09/2020

## Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

This week, we have been focusing on getting the backend of our application working. We now have it querying district data based on student coordinates. Additionally, we have been a shape file to geojson converter that will be used to convert future census data into district boundary geojson files that can be easily uploaded into Mapbox.

## Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

Yes, we met with our client. They had a lot of suggestions of UI specifics and liked the direction we are going in.

## Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

Currently, we are working on getting things seemed together. We got queries working, and we have a UI, we just need both to work in cohesion. Also, we need to finalize a query neighboring boundaries algorithm.

## Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

Currently we are working on a geojson coverter. Also, we are working on refactoring the structure of our project to match how we are handling the backend.

## Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

No new problems have arisen, thankfully. It looks like we are on schedule and will have a full demo within weeks.

# 11/17/2020

## Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

Various UI improvements. We have added a few features as well, such as the ability to sort school districts based on their distance from the student and highlight them on the map. There was also some code restructuring done to better reflect project standards.

## Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

We did not meet with our client in the last week as a few people were unable to make the scheduled time

## Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

We are working on Setting up the Final demo and implementation in the production environment.

## Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

Displaying neighboring district boundaries alongside the current boundary.

## Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

We do not currently have any major problems

# Development Costs

## Institution Costs

None

## Sponsor Costs

None

## Team Costs

None