

1. Introduction

- a. **ATTENTION GETTER:** How often do you look at power lines? If at all, it's probably to think that they're ugly and that you wish they could all be buried. It's even less likely is that you might look at the huge metal towers. For some people they need to think about such things. It's also important to remember that the reason we all have power is because of those structures.
- b. My name is Eric Knigge and I worked with Zaya and Lukas on a project to modernize the lookup software for engineering drawings of the big power structures, or as they're more formally known in the industry transmission structures
- c. Zaya and Lukas took this class last year, due to financial and personal reason I didn't officially take the class because I couldn't complete the required number of hours at that time, and am finishing those this quarter
- d. **PREVIEW:** in this presentation we will review the project description, development process, next steps, and most importantly do a demo

2. Project Description

- a. Background
 - i. Context: transmission lines carry high-voltage current over long distances and between critical power infrastructure. You can think of them as the arteries of the electrical grid, or freeways of power
 - ii. They're often lattice metal structures but can just be tall metal or wood poles
 - iii. We received a database of the engineering drawing records, which are used to lookup information about these structures
- b. Existing Conditions
 - i. Microsoft Access is used to connect to the database and run queries
 - ii. Service is functional but is not well used, intuitive, or well-understood by users
 - iii. Using access limits extensibility and possible future features
- c. New System
 - i. Web-based system that doesn't require installing any software
 - ii. Use open-source or low-cost software to minimize operational expenses
 - iii. Intuitive and easy interface so that tool is more widely used
 - iv. Provide administrative interface so that database changes can be made client side instead of by a DBA
- d. Key Decisions

- i. We wanted to select technologies that we were familiar with, are low-cost or free, and could be extended to provide future functionality
 - ii. Since we all had experience with Javascript we decided to use that for both the front and back-end for a unified code base, Python for data wrangling and API tests, and MySQL for the database
 - iii. After a few weeks of work we realized it was best to specialize and that each team member would be more productive as individual contributors instead of having everyone work on everything. We broke up into working on UI, front-end, and back-end/API
- e. Project Manager
 - i. Like any big task we decided to break it up into pieces, starting with the end goal and work backwards from that to establish milestone
 - ii. At a minimum we wanted to provide similar or identical functionality to the existing tool, and provide a path for additional features to be added
 - iii. To do this we created a repository, kanban board, project schedule, and assigned roles to team members
- f. My Tasks
 - i. I served both as an individual contributor and project manager. Originally we had an additional team member who wanted to focus on project management tasks, but they left after finding an internship
 - ii. I was the one who originally sourced the data and devised this project, so I felt responsible and interested in it's inception. As such I helped extract data from the Access system and import it into MySQL.
 - iii. Finally, my primary development focus was on the back-end, developing the documentation and API
- g. Key/Recent Updates
 - i. Since I'm not very familiar with the front-end work, I decided to focus on adding additional features to the back-end. These included the ability to reset passwords, add new users, and updates to user removal process
 - ii. Lastly, I updated documentation and cleaning up the code base and project files
- h. Next Steps
 - i. In the future it would be good to implement the functionality provided by the API into an administrator interface
 - ii. Other improvements would be to add the ability to edit user information and review system logs

- iii. There would also be UI and other small changes based on additional feedback collected from the end users
- i. Demo
 - i. **TRANSITION:** before we get into the software demo let's talk about the project documentation
 - ii. Documentation
 - 1. Overview
 - a. Will cover the highlights, some pages are very sparse and will be omitted from the presentation
 - 2. Getting Started
 - a. Page includes details about software installation and setup
 - b. Allows any user who downloads the repository and required software to run the application
 - 3. Existing Conditions
 - a. Provides context similar to what was previously shared about how the system worked before the new application was created
 - b. Serves as a reference, and was used during the development process to create user stories for essential features
 - 4. Database
 - a. Provides the entity-relationship diagram or ERD and information for creating the MySQL database used to run our application
 - 5. API
 - a. This page documented the API and was essential in clarifying information shared between back and front-end interfaces
 - b. All allowed commands have documentation on what is required, what is returned, and what types of errors can occur
 - iii. Not all API calls were implemented in the front-end but could be at a future date.
- j. Existing System Demo
 - 1. Drawings by line number
 - a. Enter 120
 - 2. Drawings by stencil

- a. EA-L 2/16
- ii. New System Demo
 - 1. Drawings by line number
 - a. Enter 120
 - 2. Drawings by stencil
 - a. EA-L 2/16
 - 3. Admin
 - a. Delete a pole
 - b. Edit a drawing by changing the revision number
 - c. Data visualization, show manufacturer date