GridAI

DESIGN DOCUMENT

sdmay23-38
Ravikumar, Gelli
Advisers
Josh Clinton - Frontend Lead
Tanay Parikh - Data Lead
Ryan Herren - Machine Learning Lead
Elvis Kimara - Voice Assistant Lead
Rolf Anderson - CI/CD & Testing Lead
sdmay23-38@iastate.edu
sdmay23-38.sd.ece.iastate.edu

Revised: December 2, 2022

Executive Summary

Development Standards & Practices Used

Practices Used

- Agile methodology
- Git Version Control
- GitLab CI/CD
- Remote VM testbeds
- Cloud integration

Engineering Standards

- IEEE Data Standards
- HTTP
- IEEE P2840™ - Standard for Responsible AI Licensing
- IEEE P2841™ - Framework and Process for Deep Learning Evaluation
- ISO/IEC 21778:2017 - JSON
- React Architectural Standards

Summary of Requirements

1 Functional Requirements

1.a Program must run locally and be deployable.

1.b Display all necessary power grid data in an interactive grid, such as the amount of power each node uses and how each node is connected.

1.c Capable of running all parts of the program from our speech recognition software.

1.d Utilize machine learning to predict and display future power grid anomalies.

1.e Highlight specific problem areas to help field technicians, such as current power outages.

1.f Allow for both broad and specific scaling of the displayed grid so that users can look at the power grid at both a micro and macro level.

2 Aesthetic Requirements

2.a Display all the grid information on a map clearly and concisely so that the data is easy to read and understand.

2.b Display all the grid information on different types of maps, such as terrain maps, overhead maps, and satellite maps.

3 Security Requirements

3.a Limits accessibility to only those authorized, such as the local government and authorized employees.

3.b Securely receive data without leaks for displaying data about future anomalies, current issues, and current power levels.

Applicable Courses from Iowa State University Curriculum

COM S 227: Object-oriented Programming

COM S 228: Intro to Data Structures

COM S 309: Software Development Practices

COM S 363: Intro to Database Management Systems

COM S 409: Software Requirements Engineering

COM S 572: Principles of Artificial Intelligence

COM S 574: Intro to Machine Learning

DS 201: Introduction to Data Science

DS 201: Data Acquisition and Exploratory Analysis

DS 301: Applied Data Modeling and Predictive Analytics

DS 303: Concepts and Applications of Machine Learning

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum to complete this project.

- Component-Based JavaScript
- Frontend Integration
- Machine Learning
- Reinforcement Learning
- Data Pipelines

Table of Contents

1 Team

1.1 Team Members

Rolf Anderson

Josh Clinton

Ryan Herren

Tanay Parikh

Elvis Kamara

1.2 Required Skill Sets for Your Project

Advanced Programming Techniques

Database Administration

Cloud Computing

Machine Learning/Statistics

1.3 Skill Sets covered by the Team

Machine Learning/Statistics - Ryan, Elvis

Cloud Computing/DevOps - Ryan, Tanay

Front End Development - Josh, Rolf, Elvis

Back End Development - Josh, Rolf, Tanay

1.4 Project Management Style Adopted by the team

Agile Scrum. Weekly scrum meetings during development cycles (sprints) to make progress iteratively. Sprints are three weeks long, using the first two weeks for development and the final week for testing and verification. A sprint retrospective will happen at the first meeting of each new sprint, along with a sprint planning meeting for the new sprint. Meetings happen twice a week, on Mondays and

Wednesdays, to provide time for updates on progress and to allow for feedback to be given throughout the group.

1.5 Initial Project Management Roles

Product Owner - Ravikumar Gelli

Scrum Master - Ryan

Development Team - Rolf, Josh, Tanay, Elvis

2 Introduction

2.1 Problem Statement

The power grid is growing more complex and adding more infrastructure each day. With this growth comes a drastic need for power companies to increase monitoring and gain insight into the health of the grid. To improve our insights and allow for predictive analytics to keep the power grid functioning, we will build a platform that intakes and analyzes power grid data to provide insights into failures to the responsible maintenance teams. Additionally, to prevent further outages, we will build machine learning applications to evaluate current power grid conditions and predict when potential outages or anomalies may occur to decrease response time for responders and keep the grid functioning to its fullest capacity.

2.2 Intended Users and Uses

Energy Companies

The primary intended users for our project are energy companies. The energy sector relies on robust operations and uninterrupted service to customers. As our project provides visibility and analytics of power grids, their specific operators are our most important clients. As grid complexity increases yearly, it is increasingly important for these companies to utilize every tool available to proactively address emerging threats.

i. Headquarters

The power companies will need to monitor the grid on a large scale to make important decisions and fix issues before they happen.

ii. Repair and Maintenance Technicians

Our project will benefit field technicians by giving them visibility into specific issues and confirmation when corrected.

Government Organizations

Local, regional, and national governments have a vested interest in maintaining power grid uptime. Issues and resulting downtime can be major headaches for governments at every level. While not directly responsible for power grids and their upkeep, recent history has shown that incompetent power companies can fail to responsibly manage their domain. Thus, it would be prudent to provide governments with the tools to keep energy companies accountable.

*Note: Due to the sensitive nature of the data we collect and display, we limit who can access our product. Only the power companies, and potentially the government with jurisdiction, will have access. No other users are allowed.

2.3 Requirements & Constraints

List all requirements for your project. Separate your requirements by type, including functional requirements (specification), resource requirements, physical requirements, aesthetic requirements, experiential user requirements, economic/market requirements, environmental requirements, UI requirements, and any others relevant to your project. When a requirement is also a quantitative constraint, separate it into a list of constraints or annotate it at the end of the requirement as "(constraint)." Ensure your requirements are realistic, specific, reflective, in support of user needs, and comprehensive.

2.4 Engineering Standards

What Engineering standards are likely to apply to your project? Some standards might be built into your requirements (Use 802.11 ac wifi standard) and many others might fall out of design. For each standard listed, also provide a brief justification.

- IEEE Data Standards
    - Our project and included services will use various IEEE data standards. As it is a project centered on data storage and analysis, it is imperative that each piece of data uses the most relevant formatting standard.
    - i. int
    - ii. string
    - iii. float64

- HTTP
    - HTTP requests are integral to data transfer in our project. Our APIs utilize HTTP methods such as GET and POST to transfer data between services and from external sources.
- IEEE P2840™ - Standard for Responsible AI Licensing
    - Our project and included services will use various IEEE data for responsible AI licensing Standardized definitions for referring to components, features, and other elements of AI software, source code, and services.
- IEEE P2841™ - Framework and Process for Deep Learning Evaluation
    - This document defines best practices for developing and implementing deep learning algorithms and defines a framework and criteria for evaluating algorithm reliability and quality of the resulting software systems.

- ISO/IEC 21778:2017 - JSON
    - When transferring data between services and from external sources, our project heavily relies on data marshaling in JSON. This almost universal industry standard facilitates painless integration with external APIs and data sources.
- React Architectural Standards
    - Only include one React component per file; Favour functionless components; Do not use mixins; No unneeded comments;

3 Project Plan

3.1 Project Management/Tracking Procedures

We'll be using Agile for development. We aim to progress weekly on our already determined plan for the year and present this to our client every Monday. With such a goal, we believe agile development would work best since we'd assign specific/minor tasks within a timeframe, just like it's done in Agile.

We will use Gitlab to track our progress and assign tasks to our team members. We'll use Discord to communicate with teammates and conduct online team meetings.

3.2 Task Decomposition

Epics:

General:

- ➔ Explore project codebase from last year.
- ➔ Set up development environments.
  - ◆ Set up 5 VMs for developers.
  - ◆ Run project services locally.

Data:

- ➔ Integration of .dss files into the data upload system.
- ➔ Configure data pipelines for ML.
- ➔ Update databases to reflect ML changes.

Machine Learning:

- ➔ Verify sources.
- ➔ Build data acquisition and ingestion pipelines.
- ➔ Clean data.
- ➔ Create a model.
  - ◆ Train model.
  - ◆ Re-evaluate the model as needed.

Frontend:

- ➔ Implement Machine Learning Algorithms and Voice Assistance

- Communicate with backend
- Design and Implement UI

➔ Redesign UI System
- Design new look
- Implement new look

➔ Task: Implement Filesystem
- Design the page
- Link with UI
- Push Data to Backend
- Retrieve Data from Backend

Voice Assistant

➔ Deploy and analyze code.
- Remove deprecated methods.

➔ Implement new voice commands.

➔ Manage dependencies.
- Update or remove deprecated dependencies.
- Review dependencies for security flaws.

➔ Integrate with frontend.

CI/CD:

➔ Remove all hardcoded links in code, and replace them with Gitlab CI/CD variables.
- Replace hardcoded links in the frontend.

➔ Update pipelines to deploy to reflect new environments.

Testing:

➔ Achieve full testing code coverage.
- Full voice asst. code coverage.
  - Write voice asst. unit tests.
- Full data platform code coverage.
  - Write data platform unit tests.
- Full frontend code coverage.
  - Write frontend unit tests.
- Full machine learning code coverage.
  - Write machine learning unit tests.

➔ Complete integration testing.

➔ Complete system testing.

➔ Complete regression testing.

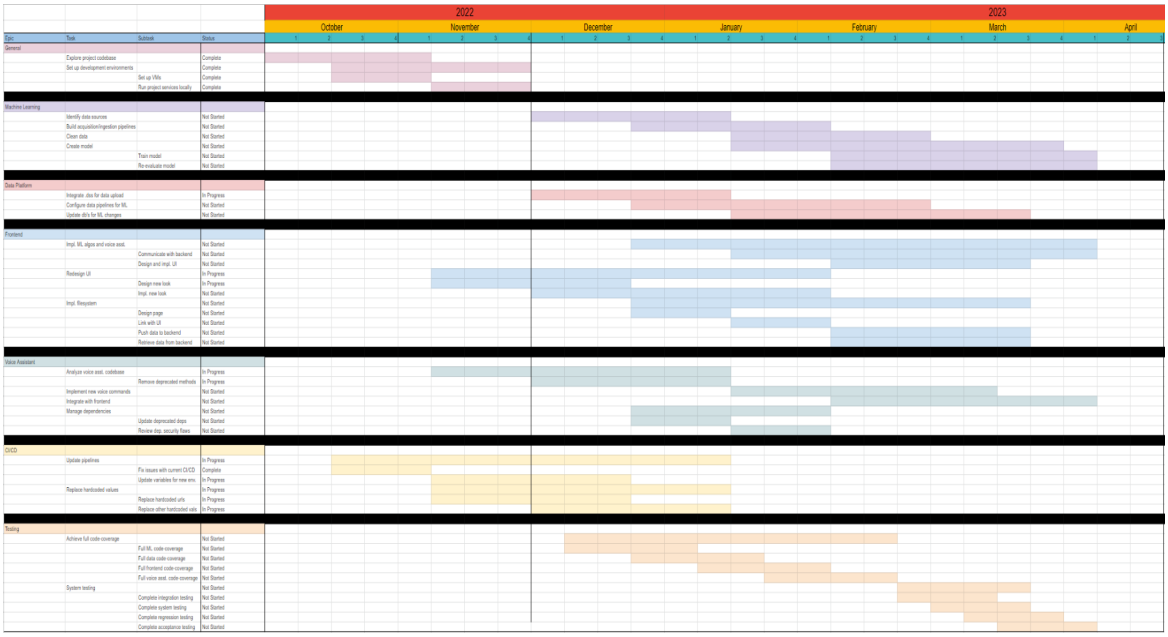
➔ Complete acceptance testing.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

TIMEFRAME	MILESTONE	Metrics	Evaluation Criteria
10/22 to 12/22	Explore Project Codebase	N/A	Team members understand the codebase.
12/22 to 03/23	Voice Asst. MVP	Voice asst. coverage of available functions.	100% coverage.
10/22 to 01/23	Data platform completed	OpenDSS Files	Stores and processes all values in OpenDSS files.
10/22 to 03/23	Enhance Front End	Relevant QGIS Features	50% relative parity.
11/22 to 04/23	Complete and Integrate ML	Anomaly and Prediction	Able to detect 75% of anomalies and predictions are at least 80% accurate.

3.4 Project Timeline/Schedule

See Appendix 8.4.1 for larger image



3.5 Risks And Risk Management/Mitigation

TASK	RISK	RISK	MITIGATION PLAN
Integrating Voice Assistant	Having Deprecated software preventing it	0.2	Research current voice assistant libraries and technologies to find one that is still being actively supported
Fully implementing the Voice Assistant	Having one person solely in charge of this task	0.51	Have people check on one another so that people meet expectations
Build Reinforcement Learning Algorithm	The model may not be effective in preventing anomalies and reducing risk for the grid	0.75	Consult consistently with the client and with field experts to overcome any obstacles and to reevaluate if it is decided that the current method will not be effective

Deployment	Cybersecurity attack	0.5	Implement information security features and methods. Follow best practices in deployment and testing.

3.6 Personnel Effort Requirements

Estimated effort requirements of tasks are listed below, separated by epic.

3.6.1 Effort Requirements for General Epic

Task	Estimated Hours
Explore project codebase	40
Set up development environments	20

3.6.2 Effort Requirements for Machine Learning Epic

Task	Estimated Hours
Identify data sources	40
Build acquisition/ingestion pipelines	40
Clean data	20
Create model	120

3.6.3 Effort Requirements for Data Platform Epic

Task	Estimated Hours
Integrate .dss for data upload	20
Configure data pipelines for ML	40
Update DBs for ML changes	40

3.6.4 Effort Requirements for Frontend Epic

Task	Estimated Hours
Implement ML algos and voice asst.	50
Redesign UI	20
Implement filesystem	40

3.6.5 Effort Requirements for Voice Assistant Epic

Task	Estimated Hours
Analyze voice asst. codebase	20
Implement new voice commands	60
Integrate with frontend	60
Manage dependencies	10

3.6.6 Effort Requirements for CI/CD Epic

Task	Estimated Hours
Update pipelines	10
Replace hardcoded values	10

3.6.7 Effort Requirements for Testing Epic

Task	Estimated Hours
ML unit testing	15
Data platform unit testing	15
Frontend unit testing	15
Voice asst. unit testing	15
Integration testing	20
System testing	15
Regression testing	15

Acceptance testing	20

3.6.8 Total Effort Requirements by Epic

Epic	Total Estimated Hours
General	60
Machine Learning	220
Data Platform	100
Frontend	110
Voice Assistant	150
CI/CD	20
Testing	130

3.7 Other Resource Requirements

Due to our project being completely software-based, we only require a few resources. To host and deploy our project, we will need a Google Cloud Platform (GCP) account and credits. We will also require Iowa State's self-hosted version control system, Gitlab. For development and testing purposes, we will also require example data to simulate what contexts our system would experience in production.

4 Design

4.1 Design Context

4.1.1 Broader Context

We are designing our product for use by local governments in combination with utility/energy companies.

The community using our product is limited to people with access to power grid information who are responsible for performing maintenance on the power grid to prevent/restore outages.

The communities affected by our design are endless. Nearly everyone in the United States relies on electricity provided to them by the power grid, so any impact our product can make on ensuring that the power grid stays healthy will impact millions of people.

Our project addresses the need for more oversight and visibility into the health of the power grid, which is nearly nonexistent right now. The more data you have to see how the grid is doing, the better you can address the issues and prevent them in the future.

The list below outlines relevant considerations related to our project in the following areas:

Area	Description	Examples
Public health, safety, and welfare	There are innumerable public safety benefits to ensuring normal power grid operation. While our project would not solve problems directly, early detection of developing issues in the grid could be the difference between a normal winter storm and the infamous 2021 Texas power crisis.	Hospitals and the patients within depend on the many systems functioning correctly. While they have on-premises generators, these are only stop-gap solutions to short-term outages. Also, summer air conditioning units are critical for public safety, especially for the elderly. Heat waves can be deadly, and badly-timed grid disruptions in the past have shown this.

Global, cultural, and social	Our project reflects the values and practices of the cultural groups very well. It's not limited to nations, workplaces, etc., because it touches everything. Our project would predict power outages and stop them from happening, thus giving societies the freedom to carry on with their existing norms and practices.	During the Chinese New Year celebrations, our project will help ensure abrupt power outages shan't occur, thus allowing for this cultural tradition to go on.
Environmental	This project will decrease energy usage from nonrenewable sources. Catching anomalies early can reduce the need for fossil fuels to fill energy gaps caused by them.	For example, power disruptions may lead to businesses and organizations activating inefficient diesel power generators hosted on-premises. Disruptions may also affect the ability of private solar power generators to provide excess power to the grid, negating any positive effects of these systems.
Economic	This project will save energy companies and local governments money by preventing power outages and other grid issues and increasing the speed at which issues can be resolved. Our project could cause energy companies to downsize if it is successful enough Because they will need fewer technicians.	For example, when the power grid goes down, local coffee shops will not be able to process their credit card transactions. This is just a small example, but it shows how many places are impacted economically by outages of the power grid.

4.1.2 Prior Work/Solutions

OpenDSS is currently a software used to provide insights into the power grid, but it is only available as a desktop application and is a very bulky program. It allows you to see the grid based on existing datasets, toggle through different types of electricity(single phase vs. three phase), and filter by types of users.

The previous work is a great start, but it's hard to access and inefficient. It is only available as a desktop application, so it can't be accessed remotely via any web browser, which is a goal of our project. A website that implements that kind of functionality would be a great asset to all utility companies and local governments. Second, since it's not web-based, it requires a great amount of processing power and local machine usage to run. A cloud-based web application takes all of the load off of the user and puts it on the backend, making it easier to use for the end user.

There's work being done by Camus Energy, along with the Pacific Northwest National Laboratory and Kit Carson Electric Cooperative, that are building a machine learning model that will fix gaps in its grid data. They have received over $750,000 in funds from the US Department of Energy (Camus, Feb. 14, 2022).

PROS

- Our project is sponsored by Iowa State University, and we have a powerful server (one of the best in the US) meant just for us. We also have great resources like Virtual Machines with 4TB space, GCP, an experienced client/professor, and many others.
- We have a solid team of senior software engineering students compared to other projects that might just have one programmer who only works in his/her free time
- We are building on a two-year-old working repository and thus not starting from scratch.

CONS

- We are students with limited expertise in this field, so other teams without such a problem are way ahead.
- Our motivation is slightly above or below meeting a class requirement. This might not be as high as someone motivated to do this to get a job, start a business, or get income for his family. Thus, they have more reason to put in more effort than we do.
- We have a short timeframe to work on this project. As a senior design project, we have a year and after that are finished. This is a con, as other (non senior design) teams could have many years to complete a project.

4.1.3 Technical Complexity

1. Our design consists of multiple components/subsystems. We use APIs, hosts like firebase, GCP, environments in GitLab, webhooks for the google voice assistant, react framework for the frontend and neural networks for machine learning, and neo4j for the influx database.
2. Our project contains many challenging requirements, including creating our own machine learning models. We will use neural networks instead of the old linear learning machine learning models. Another challenging requirement of our project will be using GEOMap to display different layers in the map using the nodeID from the database.

4.2 Design Exploration

4.2.1 Design Decisions

1. Neural network ML model. This choice gives us the performance and accuracy necessary for our anomaly detection and the ability to use different actions in response to anomaly detection in order to create a relevant and working Reinforcement Learning algorithm.
2. Voice assistant. This feature will enable quicker and more natural interaction with our project for our users. With the ability to interact with your phone with your voice, it allows utility workers to check on the status of the grid or specific substations hands-free.
3. Use GEOMap to display node data to the frontend. This allows us to display data in different layers on a grid. This will allow users to filter unnecessary data.

4.2.2 Ideation

When deciding on the type of machine learning model to use, we came up with five significant ideas.

1. Using neural networks instead of linear learning
2. Using our own code instead of integrating old code from the previous team
3. Setting up our own environment on GitLab instead of using the ones made by the previous team
4. We considered using the AI agent offered by Facebook to colleges for research purposes.
5. We considered switching some components to AWS to handle logins to use our website/frontend so that people can use/access our AI agent.

4.2.3 Decision-Making and Trade-Off

When deciding on a paradigm to use for machine learning, we decided to use a neural network rather than a linear learning model. There are many different options of machine learning models to implement when trying to create a reinforcement learning application, but when making a decision, we prioritized performance and accuracy over simplicity and ease of implementation. We decided to pursue a neural network, which gives better performance and accuracy. Furthermore, as we are in this class to learn, we chose the option with more potential for growth and relevance in today's tech scene.

4.3 Proposed Design

4.3.1 Overview

We have three major components in the current design: the backend, the frontend, and the database. The database is where all the data will be stored, and it has its own request handler to handle all the requests. The backend is where all the logic of the application is performed. For example, all of the machine learning models and APIs live here, and it also has its own request handler, so it can send and receive requests to and from the frontend and the database. The frontend is where all the visuals are processed it has its own request handler, so it can request data from the database and display it.

4.3.2 Detailed Design and Visuals

Our project will be a system consisting of a React web app frontend and several backend services connected through an API, as seen in Fig. 1. The frontend will allow users (power company personnel, government monitoring agents, etc.) to view the current state of the system, as seen in Fig. 2. Users will be able to examine individual nodes including their past and predicted future values. Furthermore, the frontend will alert users to current or predicted future anomalies which may indicate dangers to the power grid's health. Users will have the option of setting filters on the alerts received. For example, a user may limit their alerts received to a specific state or municipality. To increase ease of use, users can create specific profiles to store their information, such as location or role, and their selected filters. An additional feature of the frontend will be the implementation of a voice assistant, allowing more natural and flexible interaction with the system. The voice assistant will be integrated with the web app frontend. The backend will consist of several microservices deployed to Google Cloud Platform (GCP). Influx and Neo4j will be used for backend data storage. The backend data storage will utilize time-series and graph-based databases to efficiently store and serve recent data with accurate geographical positions on the frontend views. The frontend will use GCP Firebase for data storage specific to the frontend, such as profile information. Our system will leverage Tensorflow on Python through two machine learning (ML) services designed to predict future values and detect anomalies respectively. The frontend and backend services communicate with each other through a RESTful API implemented in Go (also known as Golang). External data sources, shown as the transmitter nodes in Fig. 1, will push data to the system through this API. This is how data in the system at large is updated.

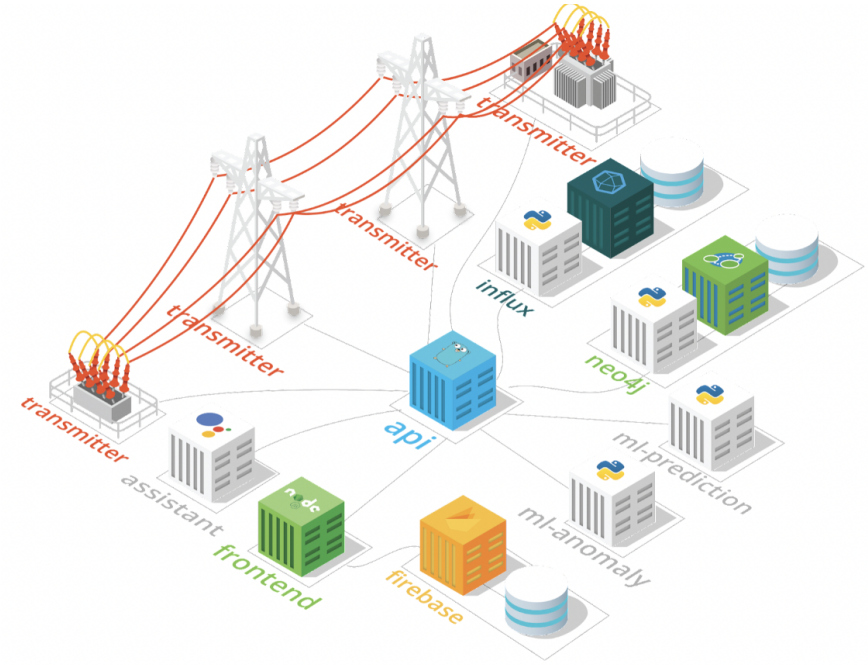


Fig. 1, Design Components Overview

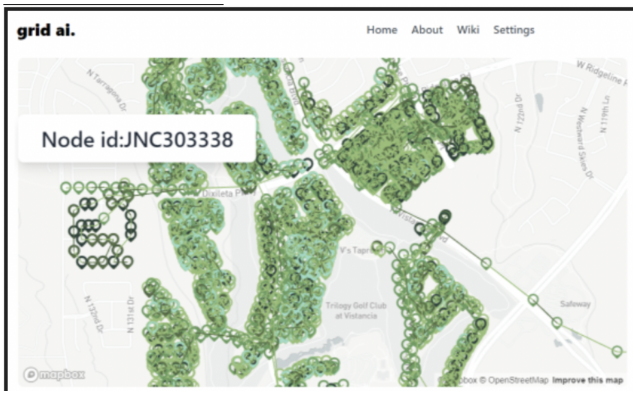


Fig. 2, UI Basic View

4.3.3 Functionality

Describe how your design is intended to operate in its user and/or real-world context. What would a user do? How would the device/system/etc. respond? This description can be supplemented by a visual, such as a timeline, storyboard, or sketch.

In a real-world context, a grid operator/inspector would use our design in the form of a website that they'd be able to log on to on a tablet or smartphone. They would be able to see different types of data both visually through the help of graphs, charts, and map representations, plus through texts or written format. The user would be able to use their voice to get information through a google voice assistant that uses machine learning to predict and provide appropriate responses/data.

4.3.4 Areas of Concern and Development

Our team's current design should satisfy the requirements and meet user needs. I believe we will be able to achieve all the goals that we have set for this project.

The area of concern would be that we have to make our machine learning models to make the reinforcement learning applications work, as there are no machine learning models in place to build on top of them. This will take the PowerGym library and other pre-established open-source libraries.

Our current plans include making a robust design for the whole project and to learn as much as we can on the machine learning aspect of things so we can implement machine learning models in the near future

4.4 Technology Considerations

Our design will purely be software-based, so most or all of our technologies shall be on the software side of things. That said, we'll implement neural networks for the Machine learning model. Regarding the voice assistant, we'll use Google's virtual assistant action center and webhooks to pass API and expected responses. This part shall be implemented in python. Neo4j shall be used to represent our influx database visually, plus react for the front end too! We also have a top-of-the-line server.

4.5 Design Analysis

We have a two-year-old repository for our design — courtesy of the two teams that worked on GridAI before us. In terms of what we've done so far, we've made weekly slides and presentations to our client to discuss/plan various aspects of the project.

We have presented what we know regarding how to improve the project and on deeper understandings of the project (aided with visuals) and many other important topics. We've also worked on class documentation like Design context, a proposed design, and many others, plus a lightning talk we did in October. From a technical standpoint, we have migrated to our own virtual machines, accessed our own GCP accounts, and gotten our firebase, GitLab, and its environment up and running. We are actively studying the code base and learning the skillsets we need to make bigger strides in the project. Our proposed design works, but it is incomplete since it misses major components. We plan on getting our react website running and integrating it with the voice assistant, plus also making new Machine Learning Neural networks.

5 Testing

5.1 Unit Testing

We will be testing all of the endpoints in all of our components, as well as integration testing all the components with the API to see if the system components communicate well with each other. For this, we will use tools like Pytest and jest.

5.2 Interface Testing

There are a lot of components in our project that will require a great amount of testing in regard to the interfaces. First, most of our project runs through an API that takes data from the power grid and integrates it into our environments. We will have to test that interface to ensure that the data is being ingested correctly. Next, once the data is in our environments, we will need to test the interface between the databases and our Google Cloud Platform environment, which is where most of the computation will happen. GCP controls the hosting of all aspects of our project, so it will be important to ensure that it interfaces correctly with the databases. Our methods of interface testing will be workflow and performance tests. We will be testing the workflow to make sure that it is able to ingest live data and support our real-time analytics, which will be part of the functionality of the site. Second, we will have to test the performance of our infrastructure to make sure that it can efficiently integrate high volumes of data

5.3 Integration Testing

In the system, all services communicate through a centralized API. This is very useful as it allows us to manipulate and monitor data flow and interactions between services. We will leverage this to perform integration testing on our system. There will be several action flows and control paths to test that involve two or more systems. Examples include user login/authentication, uploading and managing data, and defining filters and alerts, among other macro-actions consistent with user behavior. We will use Cypress to perform integration testing.

5.4 System Testing

Using a blend of unit testing, integration testing, and regression testing. We will ensure that each system component is functional and communicate with other components successfully. The system will be in a microservice configuration, so we will test that each microservice functions correctly and communicates successfully. Because we will be combining unit, integration, and regression testing, we will combine PyTest, Cypress, and CI/CD pipelines to complete system testing.

5.5 Regression Testing

Having a CI/CD pipeline ensures system integrity after every push. This will ensure the critical features in all parts of the systems do not break when adding or developing new features. We also will be making sure the database does not break. This will be done by utilizing GitLab CI/CD.

5.6 Acceptance Testing

We will report the overall system performance from both a functional and non-functional perspective and check if all the design requirements are met or not. We will execute several real-life scenarios from the perspective of an end-user. We will evaluate the software's performance and how well they comply with the requirements and also how well it works from the user end.

5.7 Security Testing

Security testing is very important for our project. We are handling a lot of sensitive data from the power grid that should not be publicly accessible by anyone outside of local governments and utility companies. To comprehensively assess the security of our platform, we will need to thoroughly test the API to make sure that data access is strictly limited to only known users and to make sure that there is no way to intercept our data as it’s being transmitted from the power grid, through the

API, to our databases. Additionally, we will have to perform vulnerability scans on our user interfaces to make sure that anything accessible on the site is protected and that there are no vulnerabilities on the frontend that can allow people to access databases without proper rights.

5.8 Results

Below is a table outlining the type of test, the process used for the test, and the expected outcome. These tests will help ensure compliance with the project requirements as they each test different aspects of these project requirements. Therefore, we will be aiming to pass as many tests as possible and clearly communicate to the client where our project has not passed our desired test

Testing	Process (How to ensure compliance)	Results
Unit	Jest	Prove that internal functions and systems are running
Interface	Postman	Test API Endpoints
Integration	Cypress	Ensure that multiple components of the systems work as expected when combined to produce the desired result.
System	Cypress	Have complete end-to-end testing done on the complete software to ensure each system works as expected.
Regression	CI/CD	Ensure existing features/functionality remain as new functionalities are added.

Acceptance	Manual QA	Ensure the software meets the requirements of the clients or users. Through constant communication with the client, this is easily accomplished.
Security	Manual QA	Attempt to break a software's security checks to gain access to confidential data. This is crucial for our project because many consumers rely on energy

6 Implementation

We have been focusing on readiness and getting up to speed with where the previous team left off. We have emphasized getting our environments running, properly distributing tasks, and learning the relevant skills for new technologies, such as machine learning and voice learning. Next semester, we'll continue where we left off, with each of us leading the initiative in our allocated roles. We really hope to get a lot more completed and have a minimum viable product midway through the semester. We want to have a functional frontend, a working database, a good reinforcement learning agent, and a reliable voice assistant.

7 Professional Responsibility

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012.

7.1 Areas of Responsibility

Area	Definition	NSPE	IEEE
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	I.6 to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	I.3 to avoid unlawful conduct in professional activities, and to reject bribery in all its forms; I.4 to avoid unlawful conduct in professional activities, and to reject bribery in all its forms;
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	1.5 to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others;
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	I.1 to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment;
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	II.5 ...and to credit properly the contributions of others;

Sustainability	Protect environment and natural resources locally and globally.		I.1 ...to strive to comply with ethical design and sustainable development practices... and to disclose promptly factors that might endanger the public or the environment;
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	II. To treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others.

7.2 Project Specific Professional Responsibility Areas

Area	Importance in Our Project	Why this Importance Value was Assigned	Performance in Our Project	Why this Performance Value was Assigned
Work Competence	Medium	While we do have experience with software engineering systems and processes, none of our team has experience with machine learning principles. Thus it will be important to achieve competence in that regard.	Low	While we have learned a lot about our respective technologies, we need to build further competencies in machine learning to create an elegant system.
Financial Responsibility	Medium	While we are not a for-profit organization, we need to manage resources efficiently to deliver a minimum viable product by the project's end.	High	We've been effective at managing resources.
Communication Honesty	High	Our project requires close contact with our client to understand the project and its relevant technologies.	High	Our team has had excellent communication with each other and our client. We meet multiple times weekly and use Discord daily to discuss work progress.

Health, Safety, Well-Being	High	Our project deals directly with maintaining the health and safety of the public.	N/A	We have not deployed a production system yet, it is impossible to judge.
Property Ownership	High	The information ingested by our system will be sensitive, it is imperative that we implement strong data security.	Medium	We have not yet included strong information security into our system; however, we plan on implementing it further into the process..
Sustainability	High	Our project could improve sustainability for power grids.	N/A	We have not deployed a production system yet, it is impossible to judge.
Social Responsibility	High	As the public will rely on our project to help maintain power grids, we must be responsible with our design and only ship a robust system.	N/A	We have not deployed a production system yet, it is impossible to judge.

8 Closing Material

8.1 Discussion

To deem this a successful project, we will analyze the ease of use for a general user and the impact that it has on general operations of local government and utility companies that are in charge of performing maintenance on and analyzing the performance of the power grid. It will ideally have an interactive voice assistant, a robust machine learning platform to power data-driven insights, and an organized, easy-to-use interface to allow the users to easily gather insights.

8.2 Conclusion

The use of machine learning has greatly impacted many industries worldwide, but never has it been utilized to the extent it will be in this project to analyze and defend the power grid. Implementing a friendly user interface with features like voice recognition and real-time location information, this project has the opportunity to make a real difference for everyone in the civilized world. It has the

potential to save lives, protect the environment, defend human infrastructure, and prevent large disasters.

8.3 References

Overcoming Imperfect Grid Data with Machine Learning | Camus Energy. https://www.camus.energy/blog/overcoming-imperfect-grid-data-with-machine-learning. Accessed 21 Oct. 2022.

PowerGym: A Gym-like environment for Volt-Var control in power distribution systems. | Github https://github.com/siemens/powergym. Accessed 21 Nov. 2022.

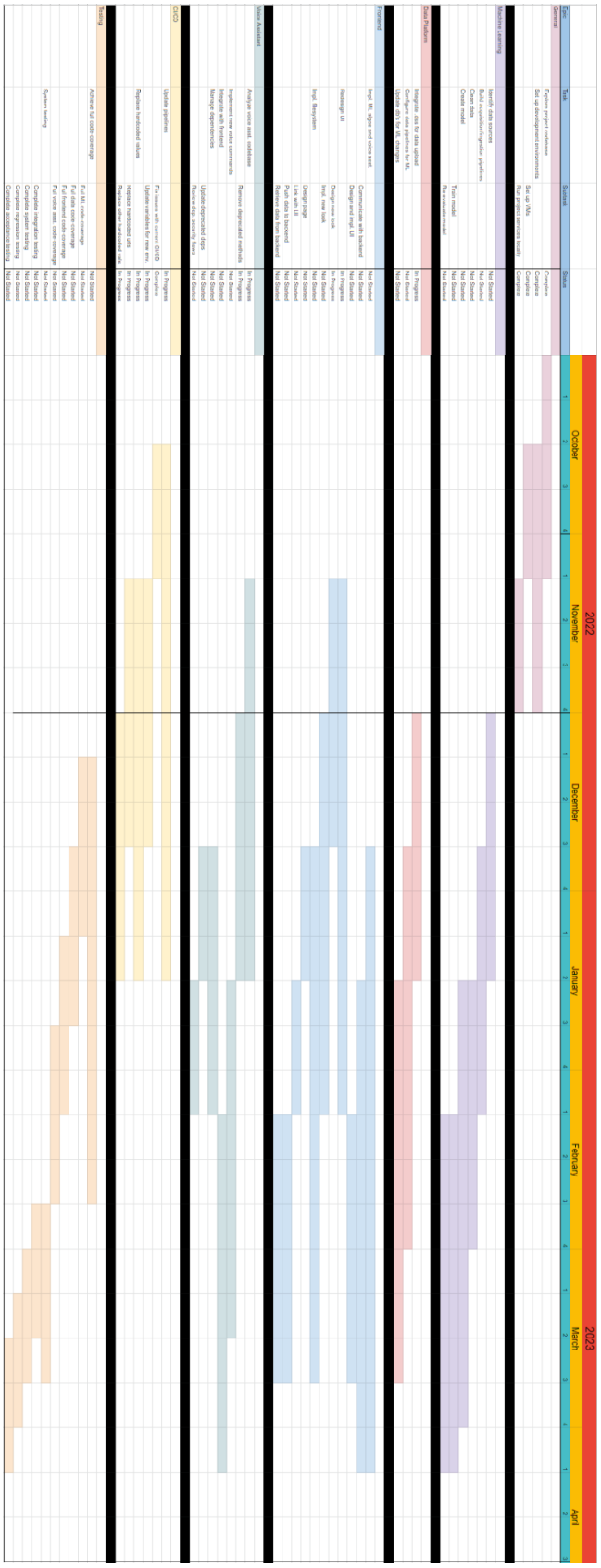
QGIS: A Free and Open Source Geographic Information System | https://www.qgis.org/en/site/. Accessed 21 Nov. 2022.

Ravikumar Gelli | Research Assistant Professor | Iowa State University Department of Electrical and Computer Engineering.

sdmay22-35: GridAI | Site http://sdmay22-35.sd.ece.iastate.edu. Accessed 21 Nov. 2022.

8.4 Appendices

8.4.1 Gannt Chart


Epic
General
Machine Learning
Data Pipelines
Frontend
Node Addition
DevOps
CI/CD
Testing
Task
Explore project codebase
Set up development environments
Set up VMs
Test all services locally
Identify data sources
Build acquisition/ingestion pipelines
Clean data
Create model
Train model
Re-evaluate model
Integrate .dvc for data upload
Configure data pipelines for ML
Update dirs for ML changes
Impl. ML algos and voice asst.
Communicate with backend
Design and impl. UI
Redesign UI
Design mgmt. tools
Impl. mgmt. tools
Impl. filesystem
Design page
Impl. page UI
Push data to backend
Retrieve data from backend
Analyze voice asst. codebase
Implement new voice commands
Integrate with frontend
Manage dependencies
Remove deprecated methods
Update deprecated deps
Remove dev services mocks
Update pipelines
Replace hardcoded values
Fix issues with current CI/CD
Update variables for new env.
Resolve hardcoded urls
Replace other hardcoded urls
Achieve full code coverage
Fix ML code coverage
Fix data code coverage
Full voice asst. code coverage
Full node test. code coverage
System testing
Complete integration testing
Complete system testing
Complete regression testing
Complete acceptance testing
Status
Complete
Complete
Complete
Complete
Not Started
Not Started
Not Started
Not Started
Not Started
In Progress
Not Started
Not Started
Not Started
Not Started
Not Started
Not Started
Not Started
Not Started
Not Started
Not Started
In Progress
Complete
In Progress
In Progress
In Progress
Not Started
Not Started
Not Started
Not Started
Not Started
Not Started
Not Started
Not Started
October
November
December
January
February
March
April
2022
2023

8.5 Team Contract

Team Name: GridAI

Team Members:

1) Joshua Clinton
2) Tanay Parikh
3) Elvis Kimara
4) Rolf Anderson
5) Ryan Herren

Team Procedures

1. **Day, time, and location (face-to-face or virtual) for regular team meetings:**
    a. Monday at 4 - 5 pm, in-person and virtual on alternating weeks.
    b. Wednesdays at 4 - 5 pm during Sprints
2. **Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):**
    a. Discord.
3. **Decision-making policy (e.g., consensus, majority vote):**
    a. We'll attempt a discussion for consensus but a majority vote as last resort.
4. **Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):**
    a. Elvis Kimara shall take the meeting notes. He'll make a google drive and doc for each meeting. This document will be open to all team members for contribution.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team **meetings**:
    a. Expected punctual attendance for all group meetings. Must communicate any absences or tardiness before the meeting.
2. **Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:**
    a. Expected to fulfill agreed-upon responsibilities, for example, learning ML. Each member is expected to ensure their tasks are completed.

   b. Expected to lead by example and be the best software engineer version of yourself that you can be. This means being early, committing, communicating properly, etc.
3. **Expected** level of communication with other team members:
   a. Expected to communicate availability changes, plus those concerning project tasks.
   b. Expected to share ideas, thoughts, and be actively involved in group discussions.
   c. Expected to ask for help when needed (from team or advisor) so we all learn.
4. **Expected level of commitment to team decisions and tasks:**
   a. Expected to live up to commitments made, fulfill tasks, and everything else pertaining to the project.

Leadership

1. **Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):**
   a. We are still figuring out roles and responsibilities. As of now:
      The lead will be in charge of all deadlines and information in their area. Not responsible for all the work in that area.
      i. Communications: Rolf Anderson
      ii. Secretary (minutes handler): Elvis Kimara
      iii. Test Lead: Ryan Herren
      iv. Cloud Lead: Tanay Parikh
      v. Backend Lead including voice assistance: Joshua Clinton
      vi. Frontend Lead: Elvis Kimara
2. **Strategies for supporting and guiding the work of all team members:**
   a. Provide feedback on progress to each team member
   b. Share resources about valuable things learned online, like tutorials, articles, and videos.
3. **Strategies** for recognizing the contributions of all team members:
   a. Share and discuss each member's contributions at each weekly meeting.
   b. Understanding/appreciating different ways of contributing such as through coding, ideas, taking on responsibilities, designing diagrams, and whatnot.

Collaboration and Inclusion

1. **Describe the skills, expertise, and unique perspectives each team member brings to the team.**

1) Joshua Clinton: is a very fast learner with several different internships under his belt, both in person and remote.
2) Tanay Parikh: I took Introduction to cloud computing last semester, so I have some familiarity with managing the cloud and CI/CD and also have some experience coding frontend in react and android studios.
3) Elvis Kimara: I will be taking Introduction to Machine Learning COMS 472 and Introduction to Artificial intelligence COMS 474 this semester. I've been an intern for Iowa DOT for almost a year and a half.
4) Rolf Anderson: Computer engineer with 4 years of professional experience in web development, test automation, systems engineering, and infrastructure engineering. Operates a homelab as a hobby. Unix enthusiast.
5) Ryan Herren: Software Engineer with 1.5 years of industry experience. Primarily spent time in full-stack .NET development, but also spent time in DevOps, ML, and Data Engineering. Completed a Data Science minor and has real-world machine learning experience with a Fortune 500 construction company.

2. **Strategies for encouraging and supporting contributions and ideas from all team members:**
    a. We'll meet up in person when our task demands that.
    b. We can have quick updates on days class ends early
    c. We'll create an environment that encourages the free expression of thoughts and ideas.
3. **Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)**
    a. Verbally share with a teammate or group at large, if you feel the team environment is hindering your ability to contribute in a meaningful way.
    b. Talk to the professor or advisor and discuss how best the situation can be improved.

Goal-Setting, Planning, and Execution

1. **Team goals for this semester:**
    a. Gain knowledge and experience specifically in ML.
    b. Create requirements for the project.
    c. Begin developing and prototyping projects, modifying requirements as necessary.

2. **Strategies for planning and assigning individuals and teamwork:**
    a. We will balance assigning tasks based on experience; we want to play into team members' strengths while giving learning opportunities to those without prior experience.
3. **Strategies for keeping on task:**
    a. We will use weekly meetings to track progress and ensure members are staying on task and putting in the necessary effort.

Consequences for Not Adhering to Team Contract

1. **How will you handle infractions of any of the obligations of this team contract?**
    a. Infractions will be dealt with on a case-to-case basis, with proportional responses and escalating consequences if a pattern emerges.
    b. If late to an online meeting, they must turn their camera on for the rest of the meeting.
2. **What will your team do if the infractions continue?**
    a. Escalate consequences, eventually contacting our advisor if the behavior is severe enough to warrant it.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*
1) Tanay Parikh _____ DATE ___09/21/2022_____
2) Joshua Clinton _____DATE ___09/21/2022_____
3) Rolf Anderson_____ DATE ___09/21/2022_____
4) Elvis Kimara_____ DATE ___09/21/2022_____
5) Ryan Herren_____ DATE ___09/21/2022_____