Microgrid Data Platform Enhancements

sdmay23-37

Client: Mathew Wymore

Alexander Haack - Software Engineer

Carson Love - Cyber Security Engineer

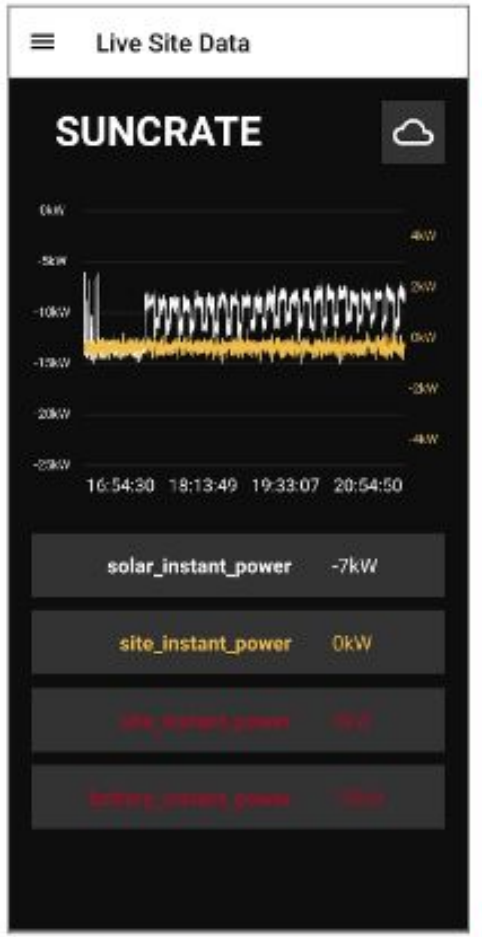
David Harmon - Software Engineer

Harvey Forchu - Computer Engineer

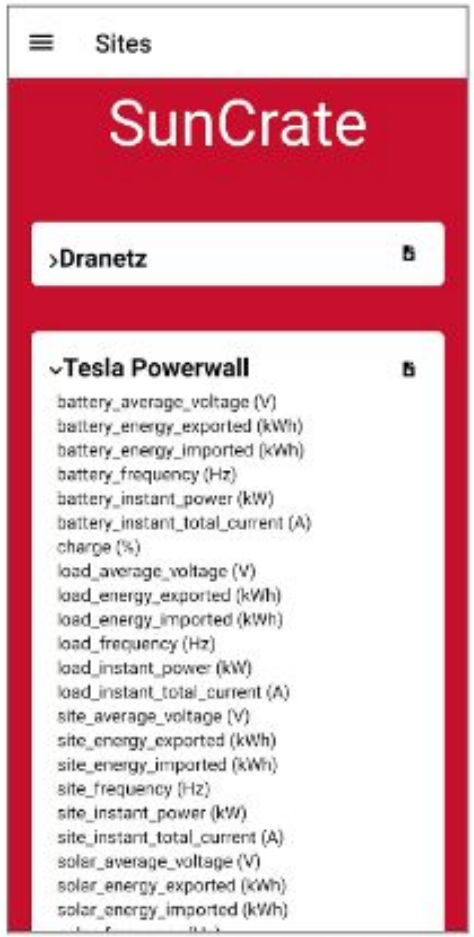
Kenyon Fergen - Cyber Security/Software Engineer

Project Vision

- Objective
  - Gather, Store and Present Microgrid Data
- Interested Parties
  - Microgrid Administrators
  - Researchers
  - General Public
- Purpose
  - Remotely monitor microgrid
  - Research how the solar crate/microgrid is performing
  - Garner excitement for energy production and usage



Page for viewing live data from a site



Page for exporting data from a site

sddec21-21 Mobile App

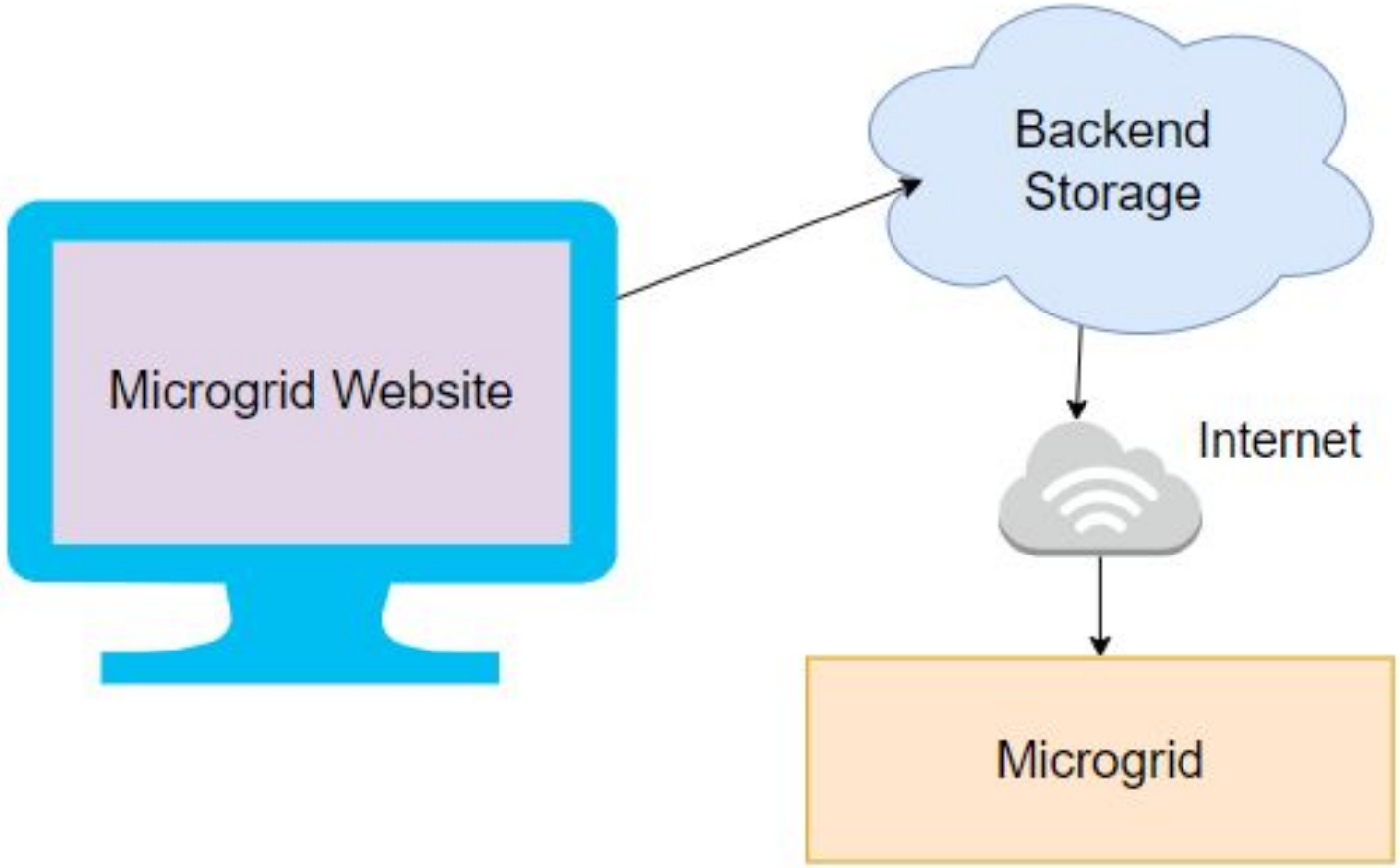
Requirements

- Display historical data from microgrids.
- Restrict microgrid viewing access to authorized users.
- Securely transmit data from the microgrid to the database and webserver.
- Collect and store data from microgrid sensors.
- Allow new microgrids to be added to the system.
- Have functionality parity with the existing mobile app.

UX Requirements

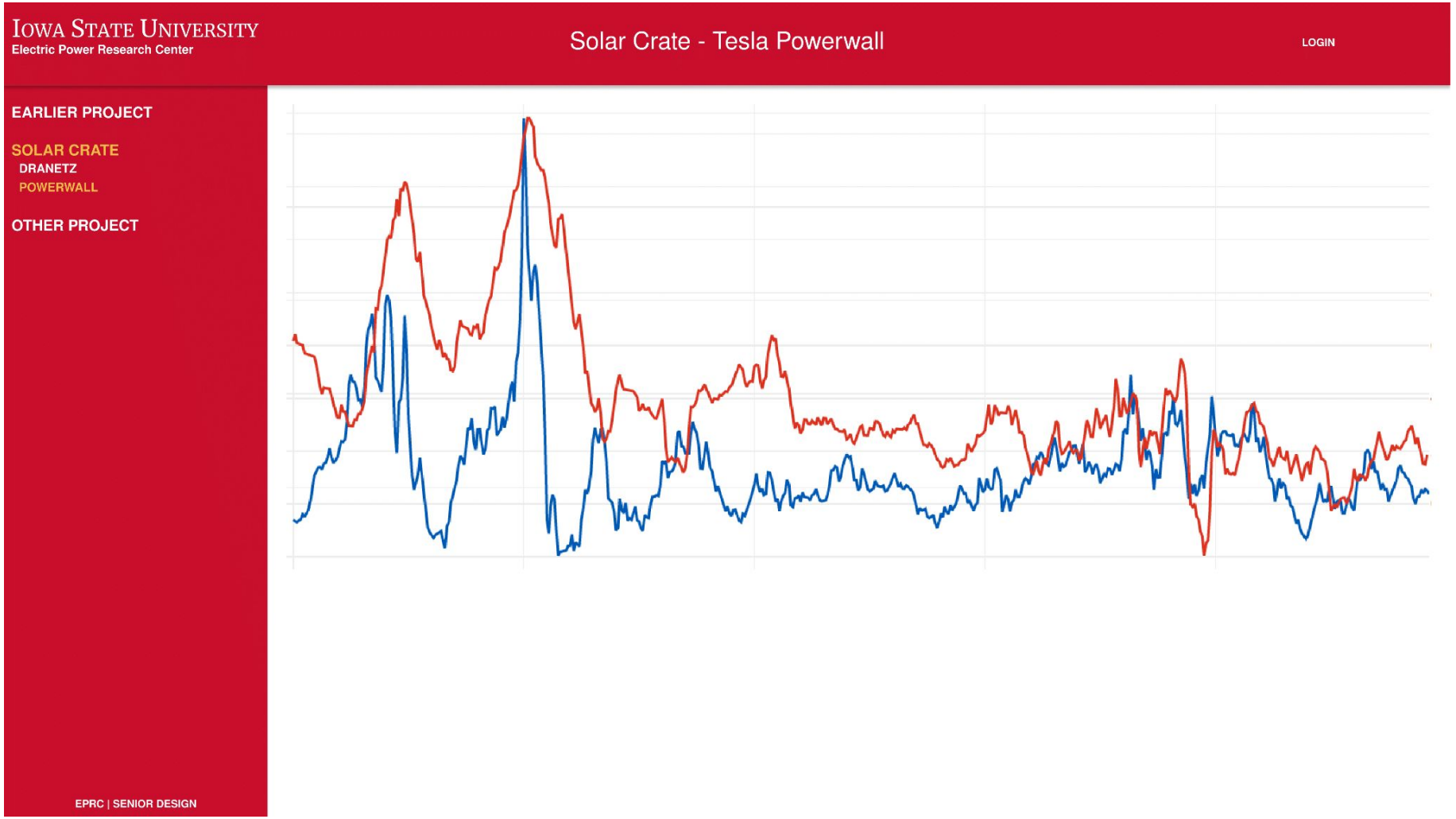
- Display the graphs from the micro grid in an easy and digestible format appropriate to its audience.
- Display simplified data for public users.
- Provide a more advanced breakdown for researchers.
- Navigate between different graphs and different components of the microgrid.

Conceptual/Visual Sketch

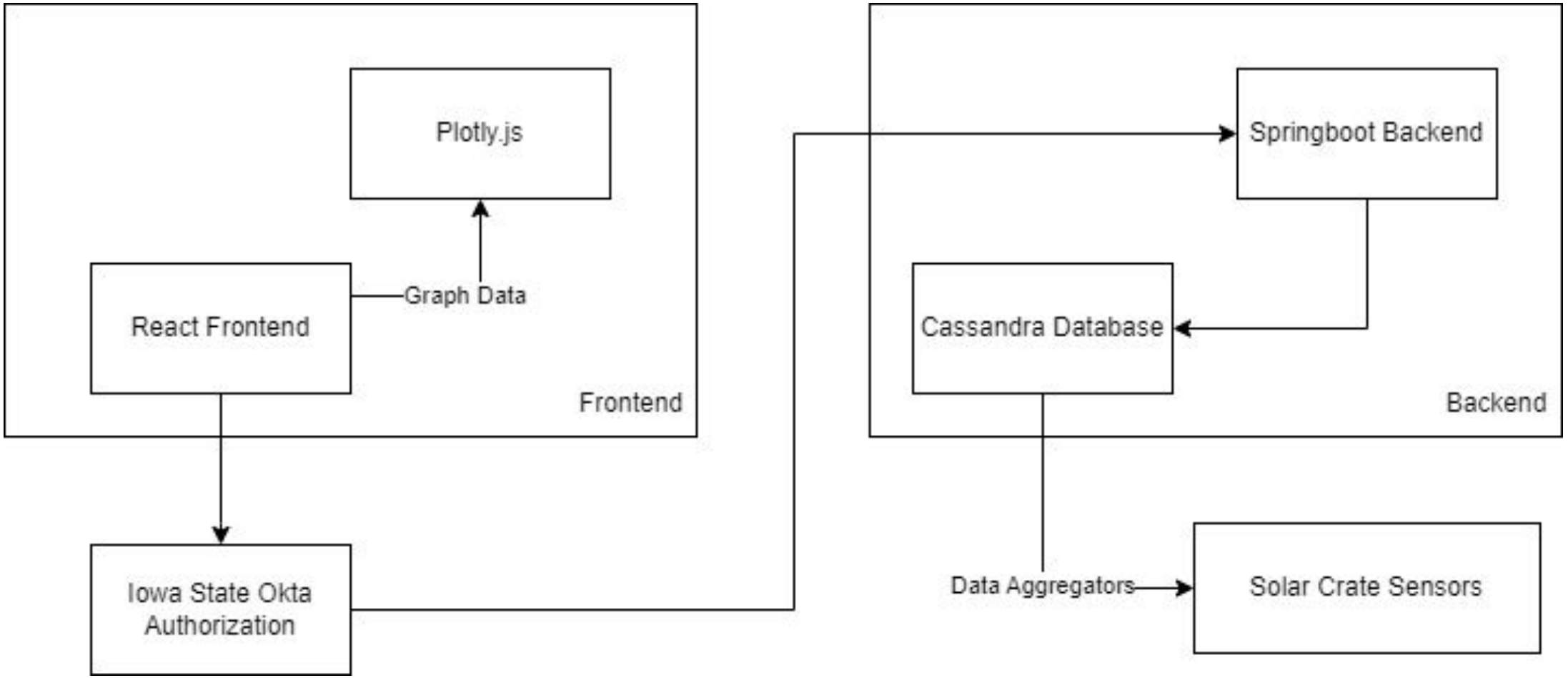


Prototype

- Initial Figma Design
- Graph traces depend on data available for component



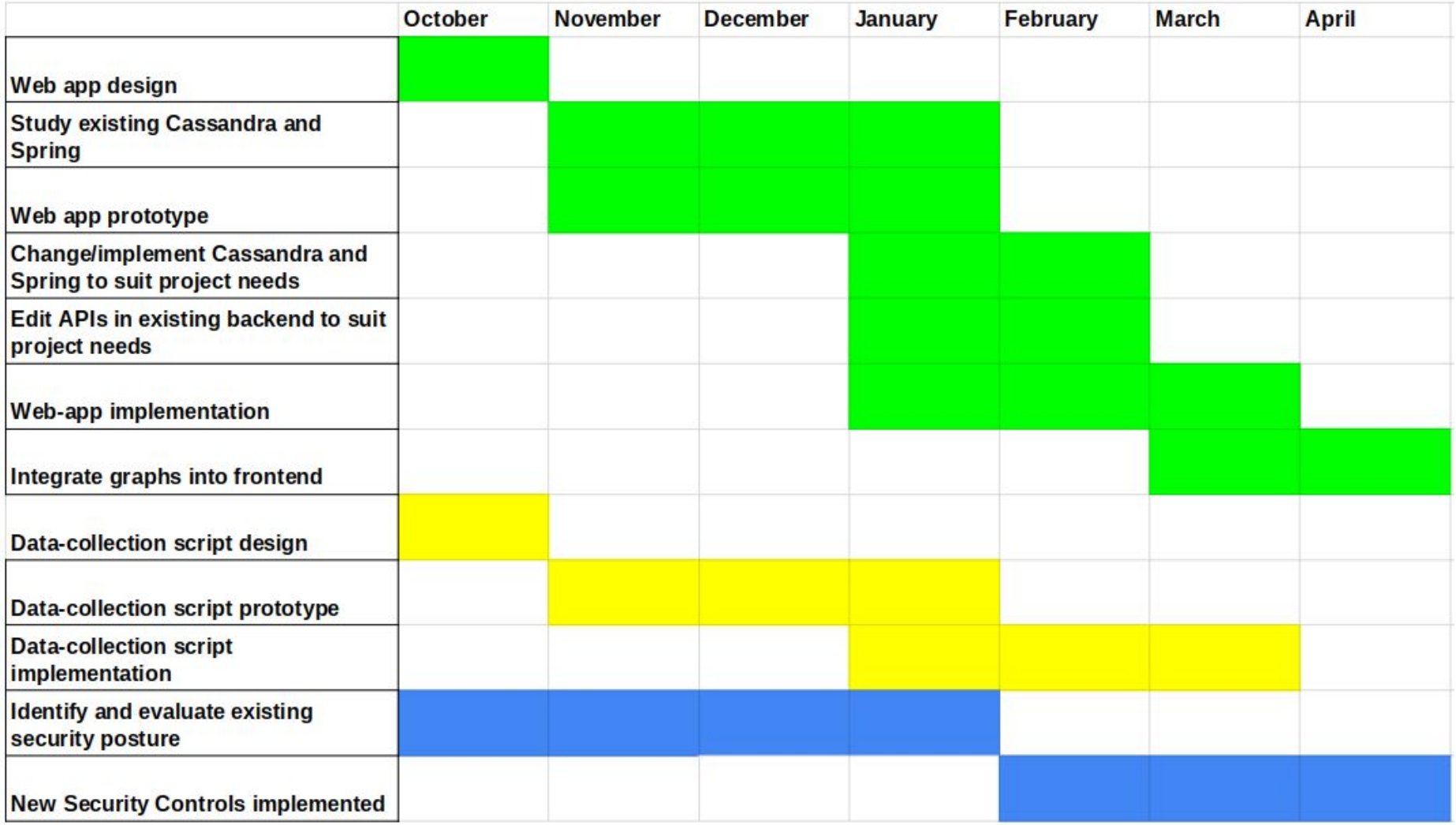
Conceptual Design Diagram



Design Complexity

- Full stack interaction between various components
  - Microgrid/Python data aggregators
  - Cassandra Database/Spring Boot Backend server
  - Web frontend
- Security
  - Authentication and secure data transmission
- Indeterminate list of various sensors/devices
  - Tesla Powerwall, Solar Panels, Dranetz Power Quality Meter

Project Plan – Schedule



Project Plan - Risks

	Probability	*Severity*	*Total Risk*	*Mitigations*
Solar Grid goes offline (breaks)	0-10%	Moderate	**Low**	Use existing data for testing the project.
Solar Grid goes offline (moved/scheduled maintenance)	40-60%	Moderate	**Moderate**	Be aware of planned downtime by the grid and plan around it.
Security Breach (corrupting/manipulating data)	0-10%	High	**Low**	Backup our data to restore any lost/corrupted information.
Code lost to laptop crash	80-100%	Low	**Moderate**	Push code regularly. Work on separate branches to store progress without breaking other people's code.
ETG VMs go down	20-40%	High	**Moderate**	Restart VMs, contact ETG for support if needed.

Test Plan

Testing will be executed via simulating interactions within and between different levels of the stack.

Unit Testing	**Interface/Integration Testing**	**System Testing**
Aggregators	Frontend to Backend API	Aggregation System Testing
Backend API	Backend API to Database	Full-stack System Testing
Frontend/React	Aggregators to Database	

Conclusions

- Fall 2022
  - Completed Web Application Design
  - Created Plan and Timeline for Spring 2023 Semester
- Spring 2023
  - Implement Web Application Design
  - Cybersecurity Assessment


Thank you!