Underground Cable Packing Web Tool

SDMAY22-19 (Team 19)
http://sdmay22-19.sd.ece.iastate.edu/

Team Member - Leadership Role

Alexander Young	- DevOps and System Engineer
Brevin Wapp	- Scrum Master
Haadi Majeed	- Quality Assurance Engineer
Matthew Hoskins	- Team Lead
Nate Tucker	- Tech. Lead
Tom Sun	- User Experience and Requirements
Quinten Sorice	- Client Point of Contact

Clients and Advisor

Client:

- Mathew Wymore
- ISU’s Electrical Power Research Center (EPRC)
- Alliant Energy (Perspective User)

Advisor: Mathew Wymore
Teaching Assistant (TA) Advisor: Jacob Conn

Project Overview

Generalized Problem Statement

- Companies with the need to distribute cabling, looking to better withstand environmental disasters and other potentially damaging occurrences, are shifting to underground cabling

- An expanded web tool version of an existing executable program with the addition of new features and improved primary functionality such as: enhanced algorithm, mobile support, and ease of use.

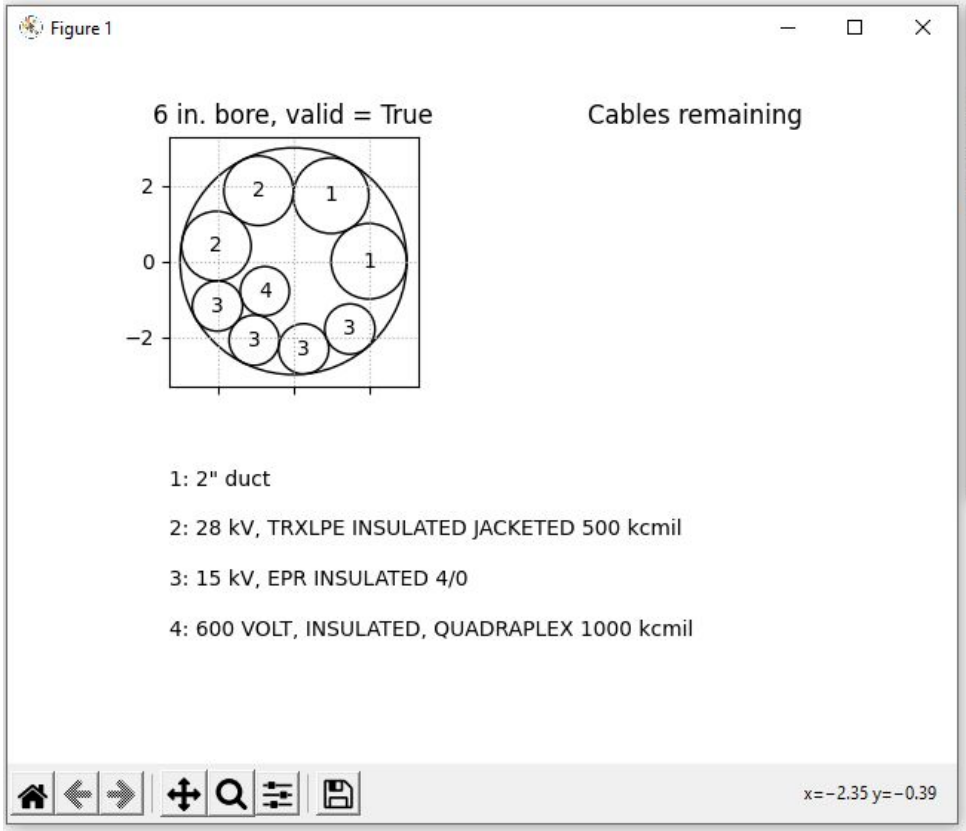
- This web tool will also allow for more readily available functionality being an application available on EPRC's website for immediate use.

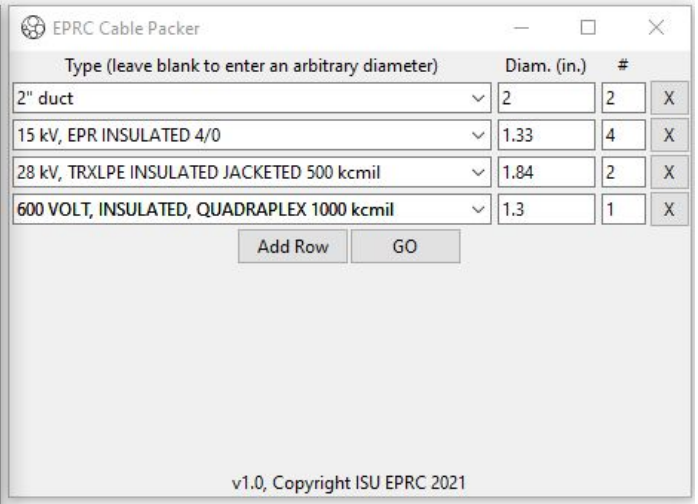
Use Cases

- Contractor bill calculation for underground cabling
- Minimum bore size calculation and visualization
- Underground cabling project material calculation
- Easier specific utility creation for cabling project
- Circle with-in a circle area calculation

Project Visualization

Project Concept Sketch





(Images from previous executable program as a source of project concept)

Project Requirements

Major Functional Requirements

- Must be a web application
- Processing time targets
  (assuming a test case of one dozen cables/ducts or less)
  - < 20 seconds wait for results on page **(constraint)**
  - < 10 minutes for user emailed results asynchronously **(constraint)**
- Frontend must run on common browsers
- Backend must run on target infrastructure (ETG server)
- Capable of sharing recent results via URL
- Accurate results with the following guidelines
  - All cables must fit within outer diameter
  - Which will not fit in a size smaller

Other Requirements

Qualitative Aesthetics Requirements

- User Interface (UI) needs Electrical Power Research Center (EPRC) branding
- Output: Cables packed to the Center of the Duct

UI Requirements

- Clean and Intuitive UI
- UI is Functional and Usable on Mobile Browsers

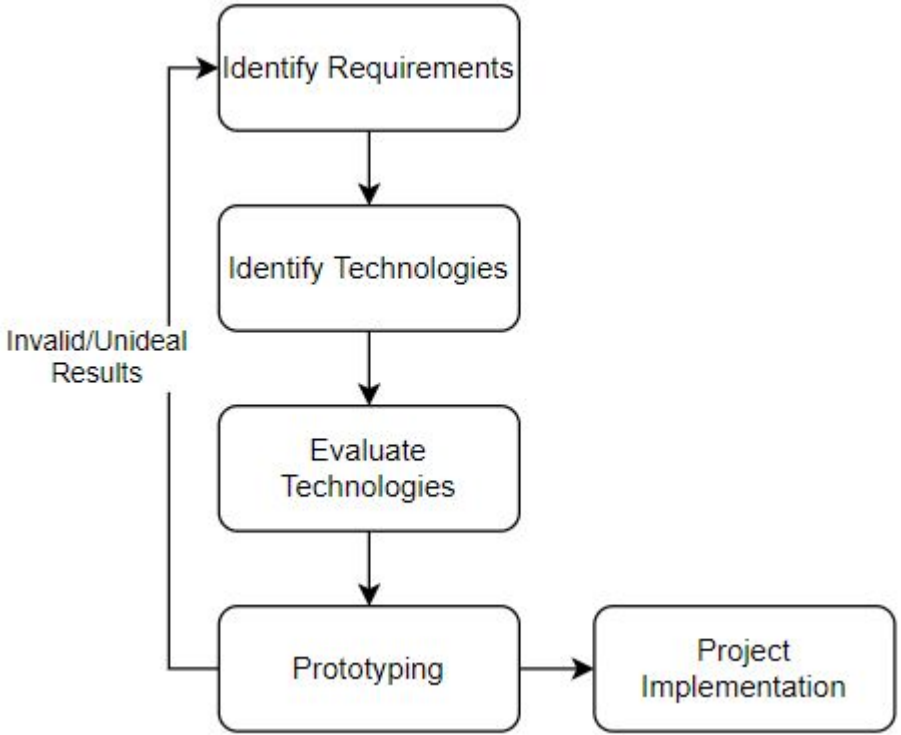
Software Requirements

- Using a Standard Web-Stack with Documentation **(constraint)**

Conceptual Design Diagram

Design Iteration and Approach

- Analysis of requirements
- Major task identification
- Minor and sub task identification
- Identify/decide on technology to perform tasks
- Prototyping stage of design ideas
- Project implementation
- Task testing upon completion
- Acceptance testing upon project completion



Project System Design

Technology, Frameworks, and Libraries

Web Tool Project

- React, version 17.0.2: Frontend development
- Golang, version 1.17: Backend/algorithm development
- PostgreSQL, version 14.1: Database management
- Prettier, version 9.0.0: Software standardization
- GitLab, version 14.4.2: Software development and sharing tool
- JSON, version 2020-12: Data-interchange formatting
- Google Drive, online version: Project document creation and editing

Existing Program

- Python, version 3.9.5: Primary programming language
- Matplotlib, version 3.4.2: Visualization library
- NumPy, version 1.20: Mathematics library

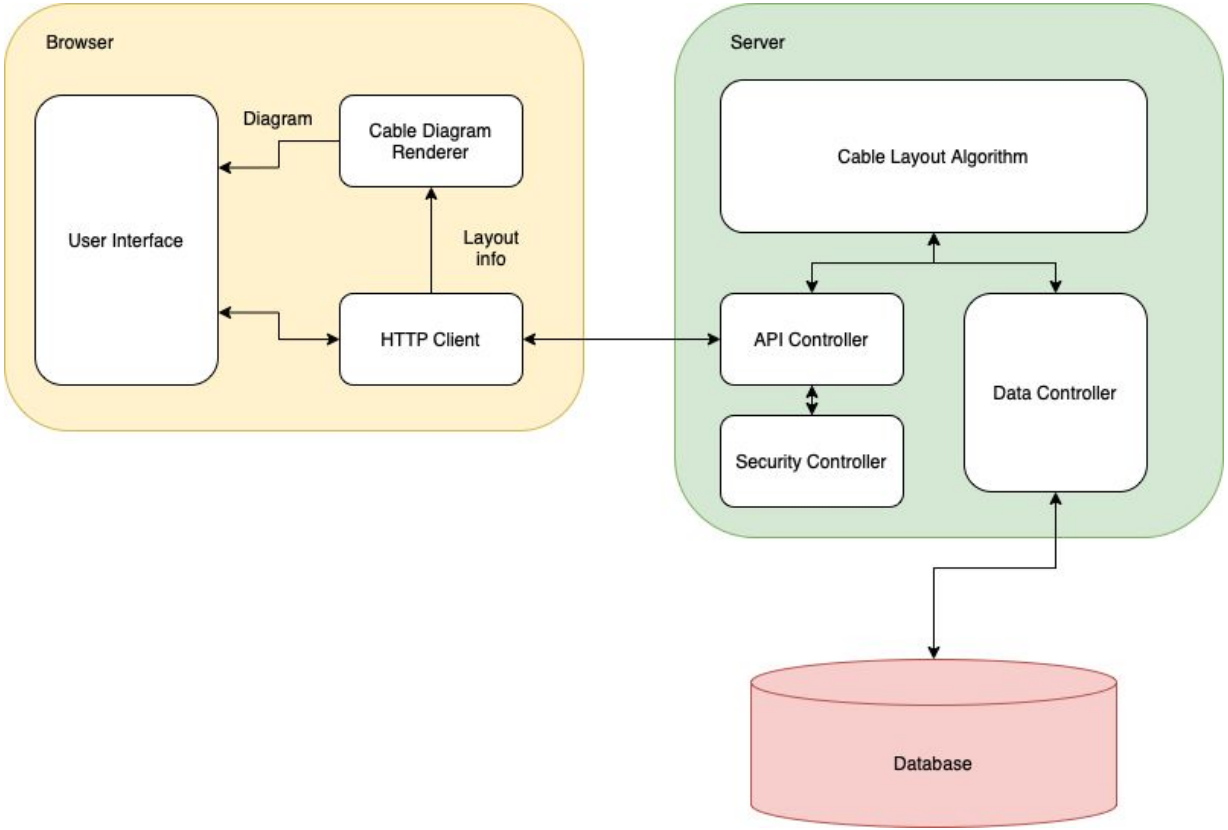
Software Engineering Standards

- IEEE 1016
  - Create and maintain software design document
- IEEE 830
  - Create Software Requirements Specification (SRS)
- ISO/IEC/IEEE 29119
  - Software tests defined, operated, and documented properly

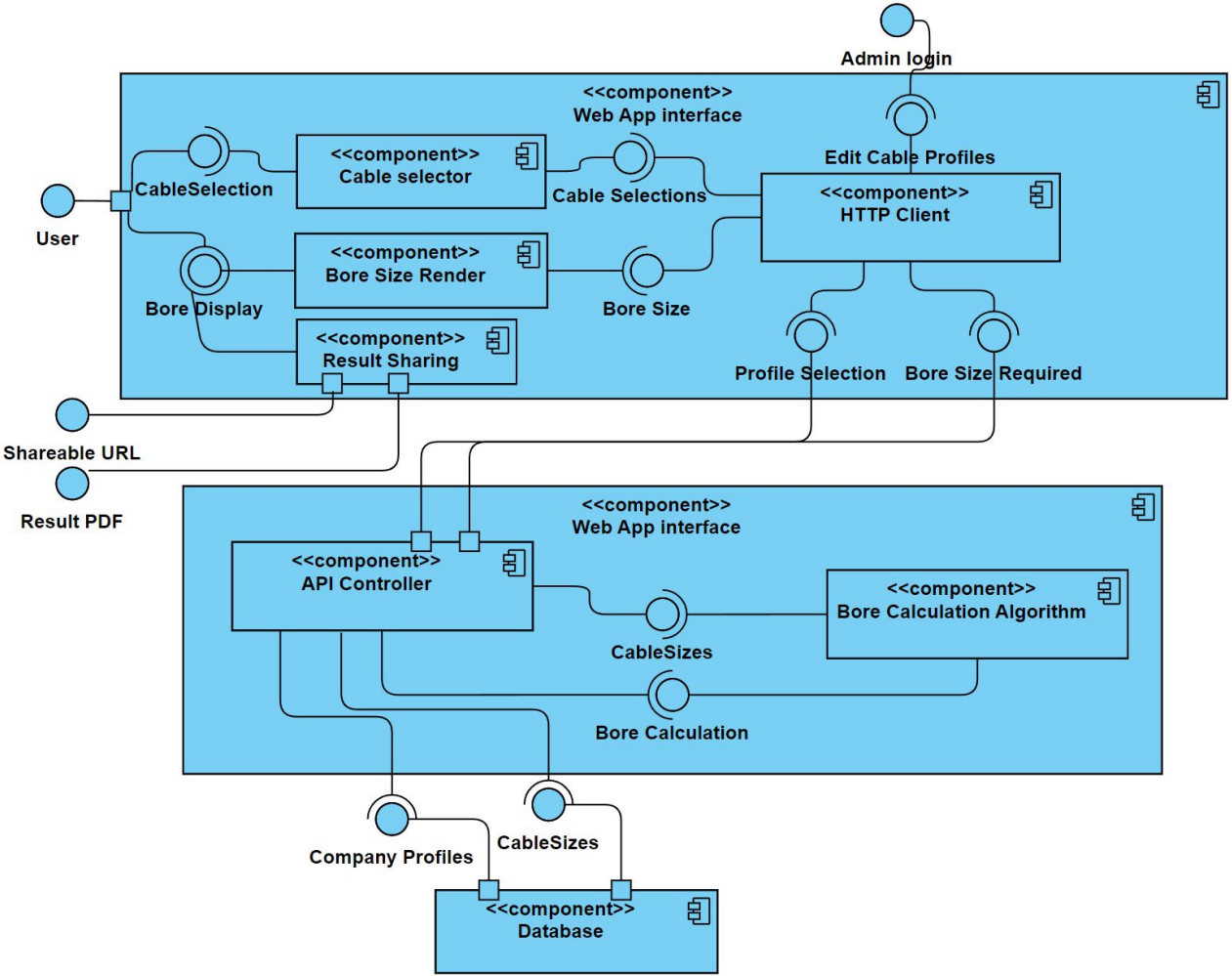
Used Abbreviations:
Institute of Electrical and Electronics Engineers (IEEE), International Organization for Standardization (ISO), International Electrotechnical Commision (IEC)

Software Architecture

- Browser
- Server
- Database



Component Diagram



Design Focuses

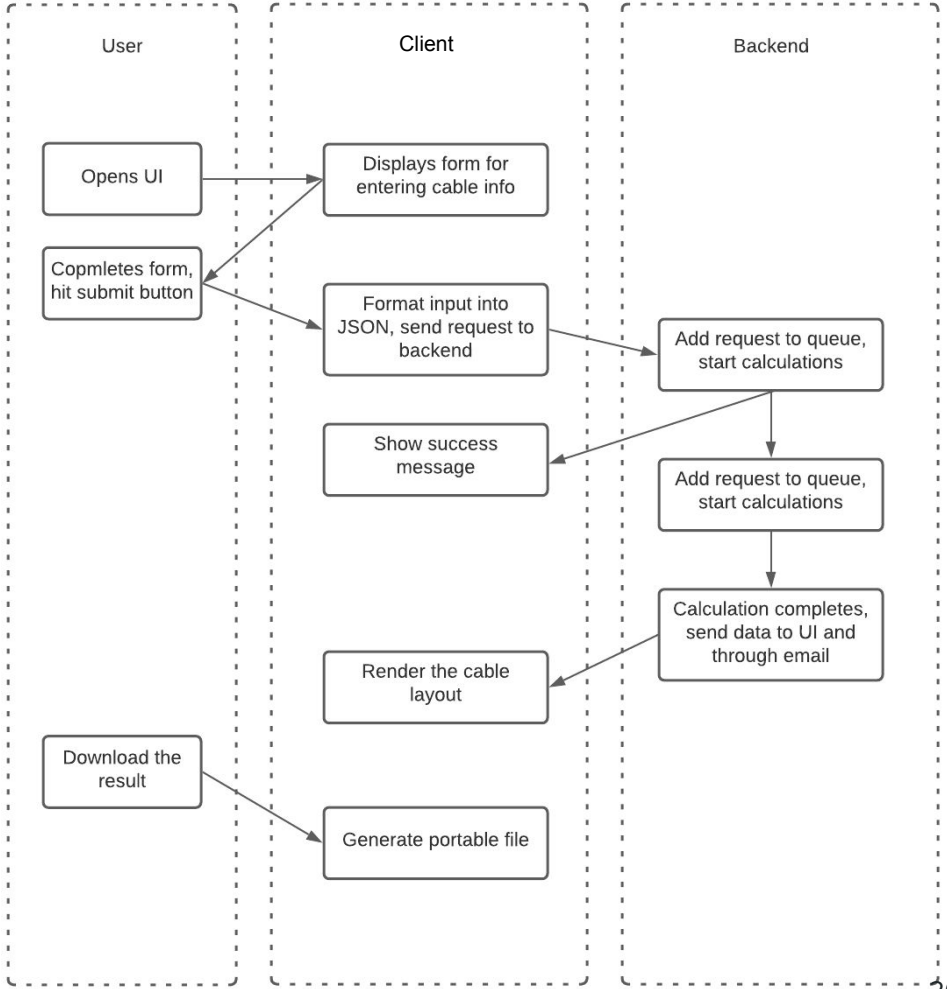
In order to fulfill the project requirements:

- Frontend UI takes in user input
- Backend algorithm receives input and generates result
- Result send via JSON back to frontend
- Frontend renders received JSON into final result
- Special URL created for final results viewing
- Backend sends results to specified email (if applicable)

Program modules broken down into specific functionality

API and Interface Diagram

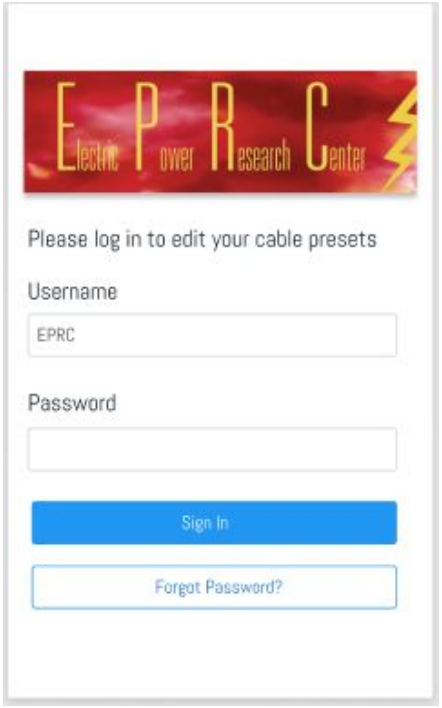
- User
  - Utilize UI for input
  - Obtain results
- API
  - Data transference and rendering
- Backend
  - Take request input
  - Run algorithm
  - Return

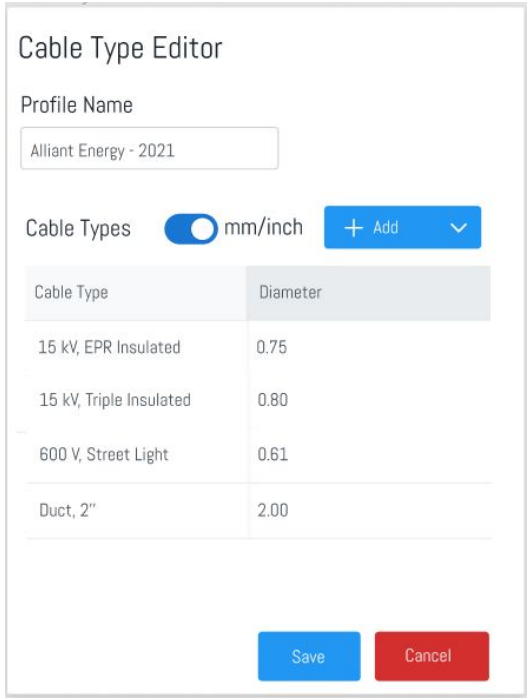


User Interface (UI) Design

Mock-up (1):

- Login panel to access type editor
- Overlay panel to edit predefined cable list sets
- Each company/user can set up a profile (authenticated)

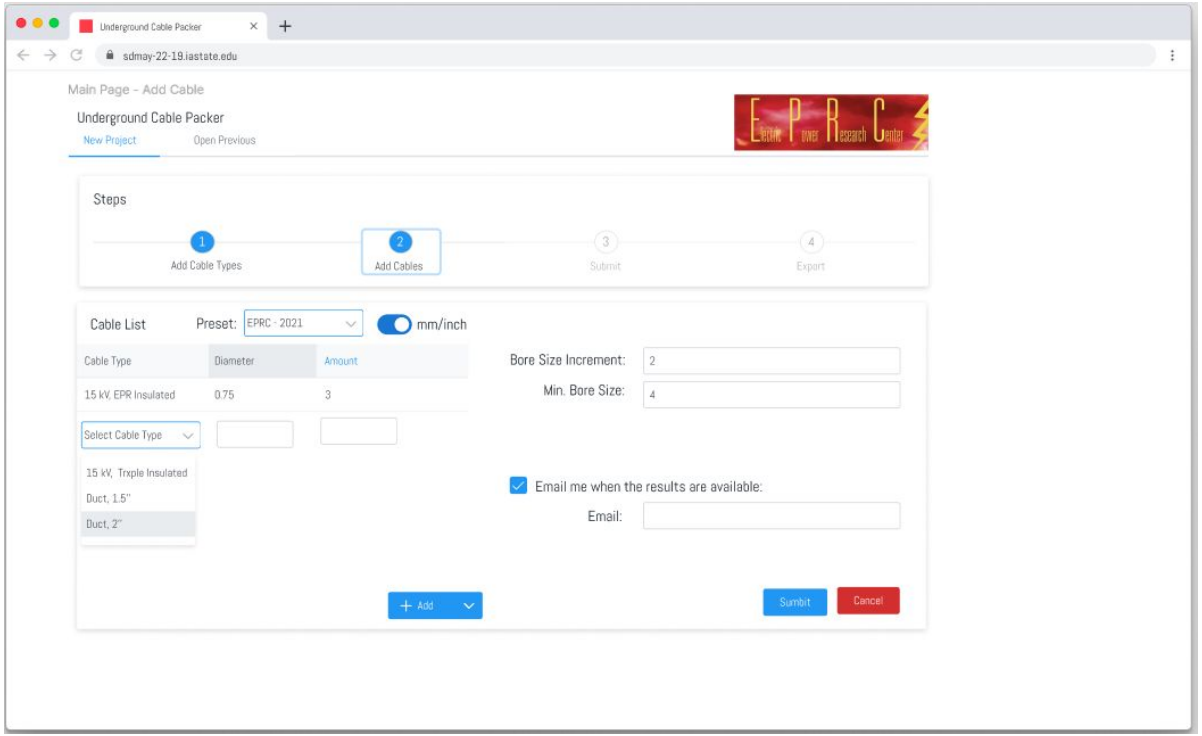




User Interface (UI) Design

Mock-up (2):

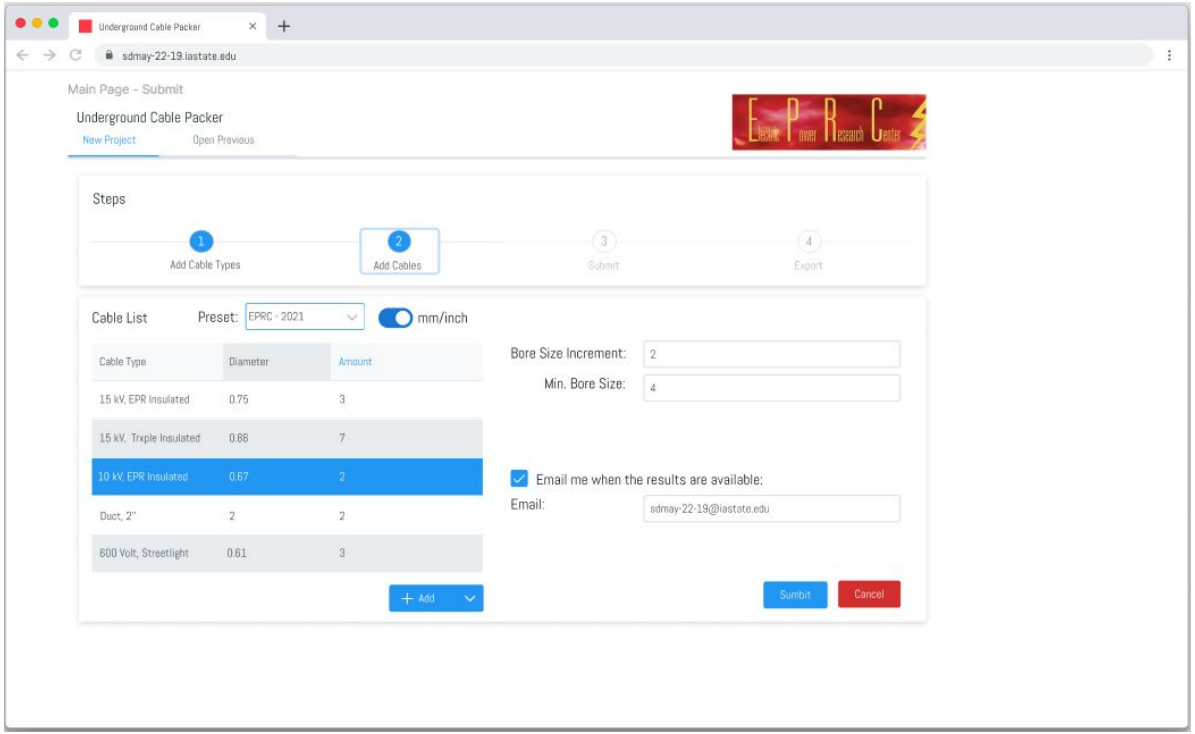
- Add cables to be packed
- Set parameters



User Interface (UI) Design

Mock-up (3):

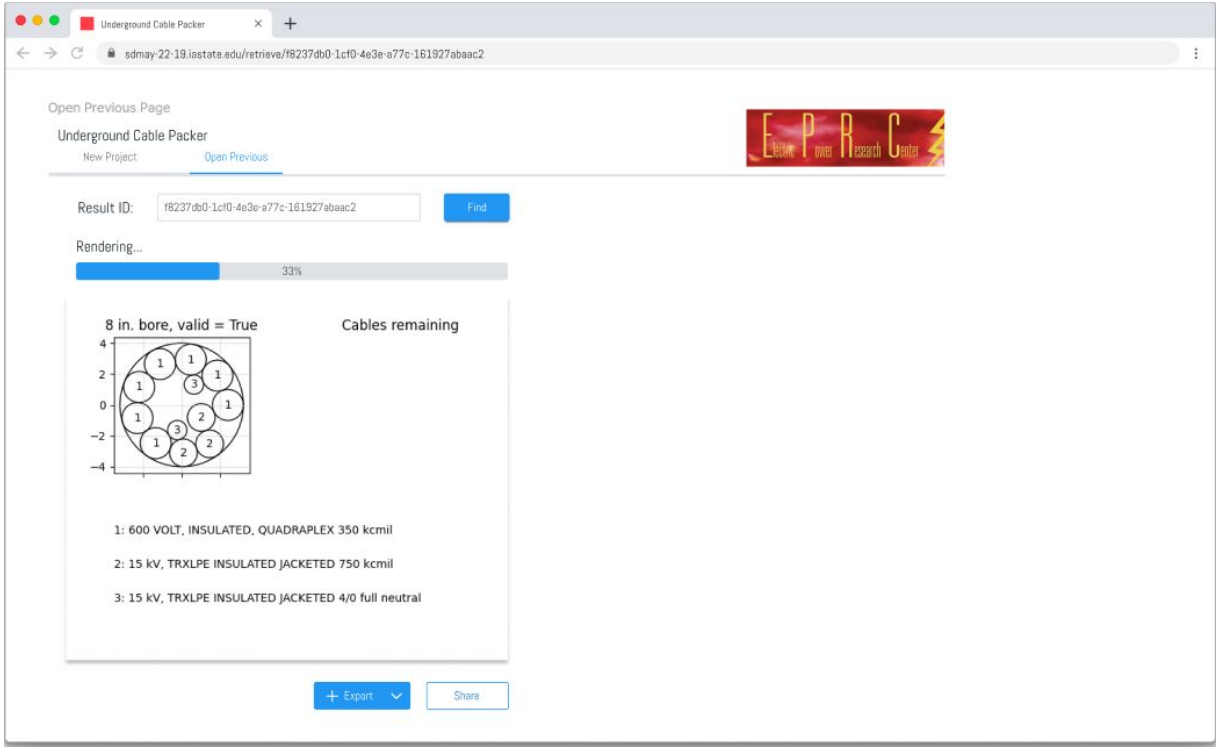
- Completed form



User Interface (UI) Design

Mock-up (4):

- Results



Design Complexity

Complexity

Complexities

- Sharing and saving results
- Accommodating mobile users

Iterations

- Current desktop application functions as version 1
  - Iterations required to add functionality and improve speed
  - Still ongoing
- Client conversations created

Risks and Mitigation

- ISU Engineering Hosted Server
  - Allows close contact and no externally hosted information
- Performance Requirements - Software
  - Performance testing of individual performance
- Scheduling Conflict with External Stakeholders
  - Maintain close contact, plan early
- Security (Data Leak, DoS, etc)
  - Avoid storing sensitive data
  - Internally hosted
  - Use up-to-date dependencies

Project Plan

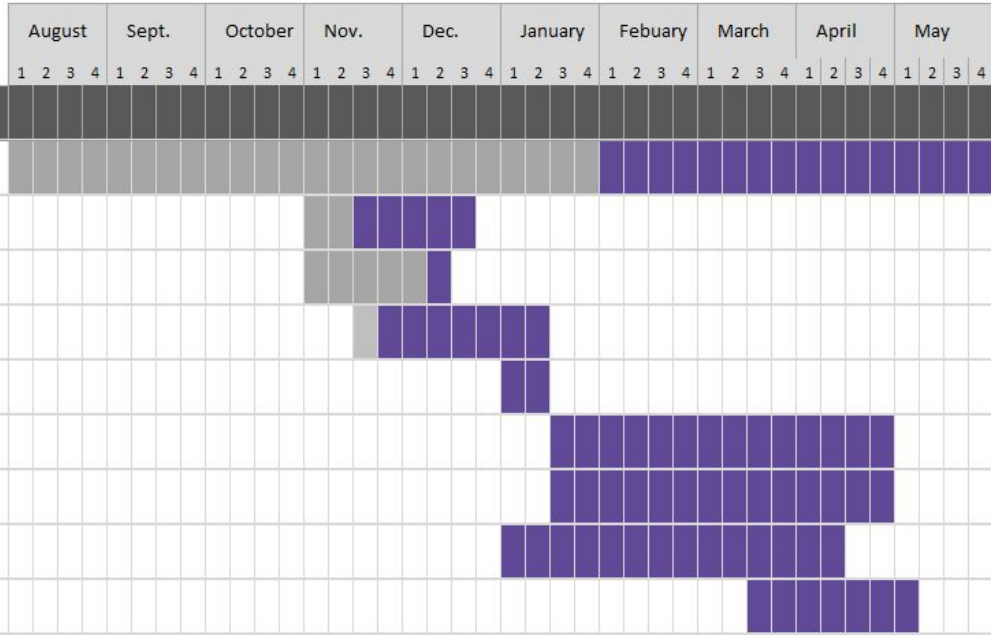
Generalized Project Schedule - by Major Task

Underground Cable Package Management Web Tool - Team 19 (SDMAY22-19)

Project Start: Mon, 8/30/2021

Display Month: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Project Planning & Defining	Full Team	68%	8/30/2021	4/30/2022
DevOps & Tech Setup	Full Team	15%	10/24/2021	12/21/2021
Software Design & Functional Verification	Full Team	81%	10/31/2021	12/27/2021
Redesign Algorithm	Full Team	15%	11/22/2021	1/5/2022
Setup Data Tables	Full Team	0%	1/3/2022	1/17/2022
Backend Construction	Full Team	0%	1/17/2022	3/28/2022
UI (User Interface) Construction	Full Team	0%	1/17/2022	3/28/2022
Development Testing Suite	Full Team	0%	1/3/2022	3/14/2022
UAT (User Acceptance Testing) & Deploymer	Full Team	0%	3/28/2022	4/25/2022



Project Milestones

1. Protocols, Technologies, and Requirements have a team consensus.
2. Git configured with CI/CD, individual work environments are set up.
3. Mockups are verified by client, professor, and TA.
4. Algorithm must produce the correct result within 20 seconds.
5. Frontend and backend can successfully communicate.
6. Application must pass all unit tests and produce expected results.
7. Application must be deployed on the Iowa State server.

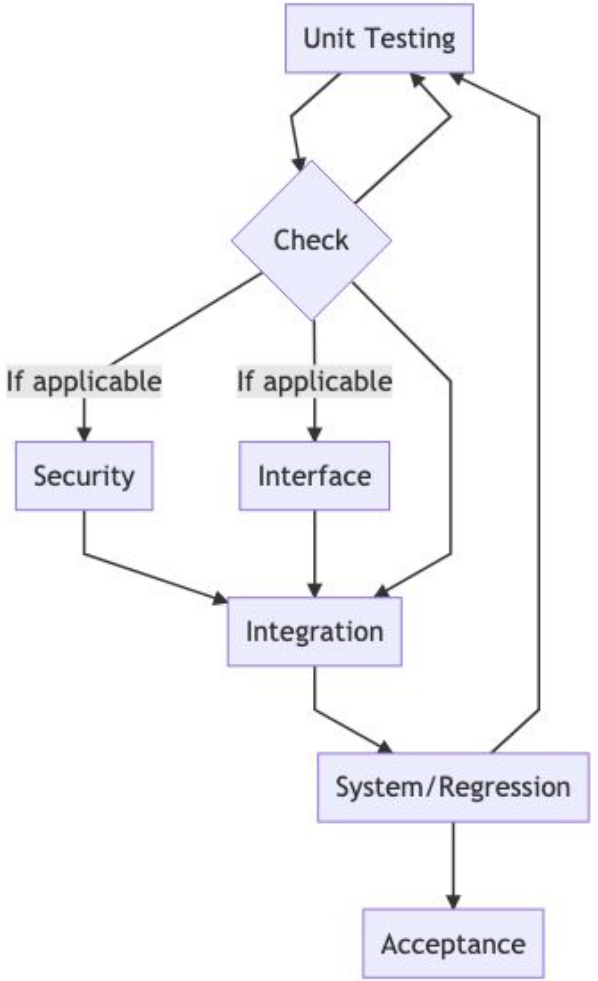
Testing Plan

Testing Process

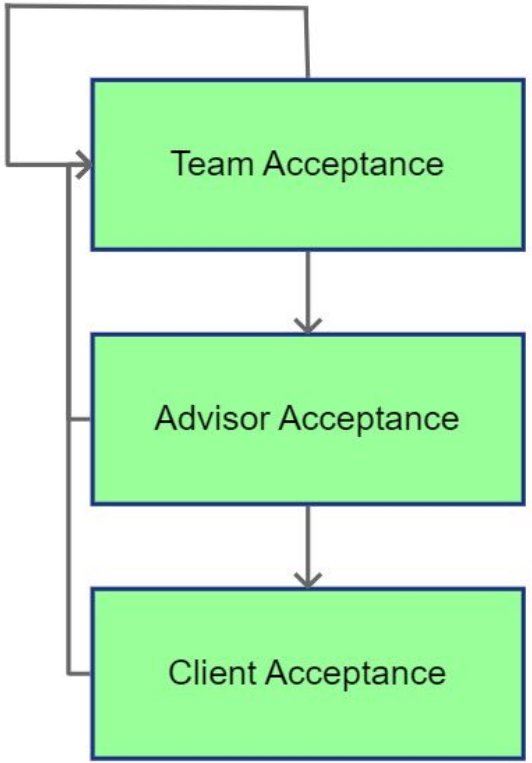
Testing Performed

1. Unit Testing
   a. Jest and Go testing suite
2. Security or Interface (if necessary)
3. Integration
   a. Docker (databases), Selenium (browser env.)
4. System & Regression
   a. Testing after merge
5. Acceptance

(Repeating 1-4 often with each new merge)



Acceptance Testing



The last phase of testing and project development

Three phases:

- Occurs in team first
  - Test full functionality within the team
- Advisor Wymore acceptance testing
  - Have Wymore test full functionality
- Perspective client Alliant Energy
  - Have Alliant Energy Representative test full functionality

Conclusions

Schedule Status

Pre-Implementation Phase:

- Project Planning and Defining [On-Going]
- DevOps & Tech. setup [IN-PROGRESS]
- Project Design and Verification [Final Stage]

Implementation Phase:

- Algorithm Redesign [IN-PROGRESS]

Next Semester (Spring 2022) Schedule

Implementation Phase:

- Create Tables [January Start - End]
- Backend Construction [January Start - March End]
- UI/Frontend Construction [January Start - March End]
- Development Testing Suite [January Start - March End]

End Phase:

- UAT and Deployment [March Start - April End]

Underground Cable Packing Web Tool

SDMAY22-19 (Team 19)
http://sdmay22-19.sd.ece.iastate.edu/