Limitless Automation Possibilities

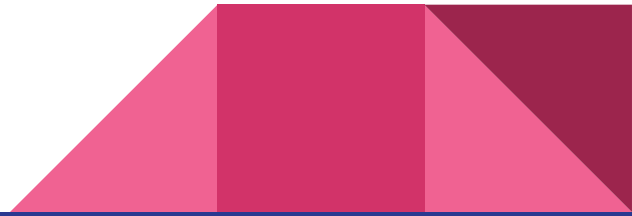
Meet the Team

William Gibson

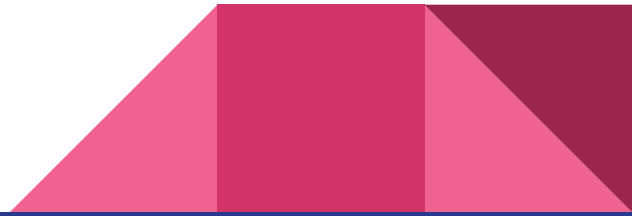
- Part time software engineer at Blue Ridge Automation for ~half a year
- Full time cat dad.
- Experience with webdev and automation engineering
- Worked in quite a few languages
- favorite is javascript flavors
    - typescript, nodejs, react, etc...

Jacob Dye

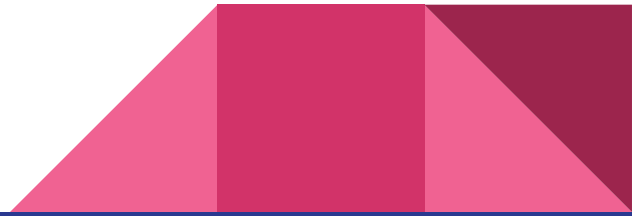
- Big gamer
- Just coding experience with classwork and random Lua
- Excited to learn alot about JS



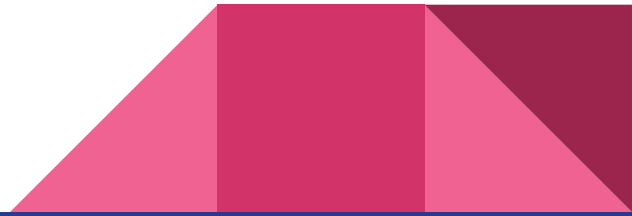
Andrew Dickinson



Alex Hupe

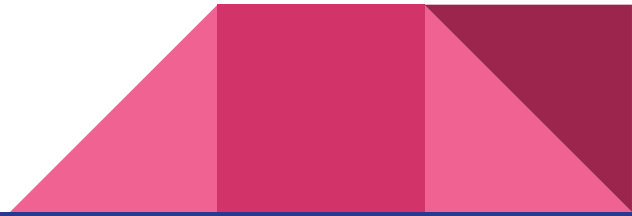


Wesley Kring



Meet the Client

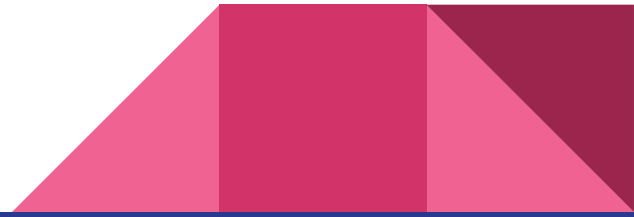
- Blue Ridge Automation
- Program PLC controllers for machinery
- Develop software to help with PLC automation
- Develop HMIs (human machine interfaces) to make engineers’ jobs easier
- Growing company with lofty goals and high potential



Our Task

Business Requirements

- Save users time writing automation programs
- Allow companies current tools to be easier to use with a lower barrier of entry



Use Cases

- **Select a configuration from the database**. Users: Automation Engineers. This is a use case because any sequence is under a certain configuration, so development is impossible without selecting a configuration. The user will first open the application, then select a configuration to edit.
- **Select a type of node to use**. Users: Automation Engineers. This is a use case because there are several types of nodes that a user will be able to drag onto the workspace. These need to be organized so the system is easy to use. The user will select a configuration, then select a type of node to browse.
- **Drag sequences, steps, or control modules into a workspace to create a node**. Users: Automation Engineers. This is a use case because the entire system is based on a user dragging nodes into a workspace, and then connecting them together based on their types. The user will select a configuration, select a type of node to browse, then drag it onto the workspace.
- **Connect existing nodes together with a line.** Users: Automation Engineers. This is a use case because users will need to able to connect nodes together to make a usable configuration. The user will select a configuration, then a type of node, then draw a line from one to the other.
- **Remove nodes from the workspace.** Users: Automation Engineers. This is a use case because users will need to remove a node from the workspace in any normal use of the program. Mistakes must be forgivable. The user will first select a configuration, then a type of node, then add a node to the workspace, then remove it.
- **Login before accessing the IDE.** Users: Automation Engineers. This is a use case so others do not edit a user's work without their knowledge. The user will first attempt to navigate to the LAP visual IDE, but will first be redirected to login if they have not already done so.

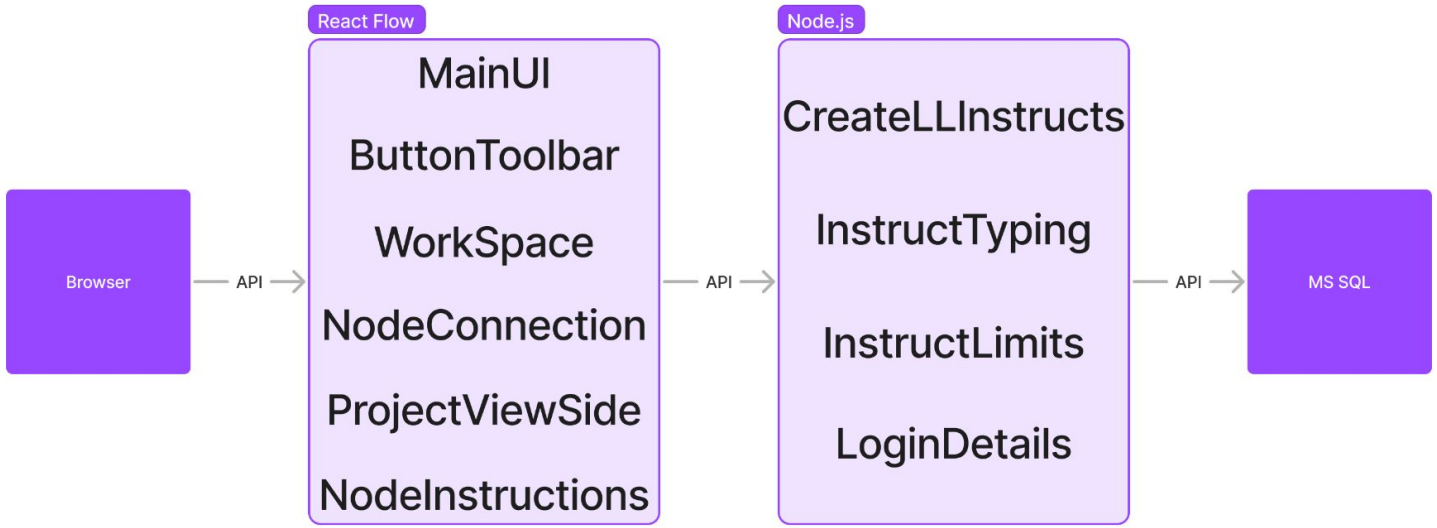
Functional Requirements

- The system shall give the user the option to browse from several types of items. Sequences, steps, or control modules.
- The system shall pull the different types of items from the database and display them in a pane.
- The user shall be able to select a configuration to act as a container for the project.
- The user shall be able to drag an item into the workspace, causing it to create a new node.
- The system shall allow the user to remove a node from the workspace.
- A node should have an input and output connection able to connect to and from other nodes.
- A node should store a single item.
- The system shall allow the user to connect two nodes together with a line.
- The nodes should be able to be resized.
- The user shall be able to download the project to an existing PLC.
- The user shall be required to login before accessing the visual IDE.
- A configuration will not be able to connect to or from anything.
- The system shall only allow sequences to connect to other sequences or steps.
- The system shall only allow steps to connect to sequences of other steps.
- The system shall only allow control modules to connect to caste 2 sequences or the bottom level.

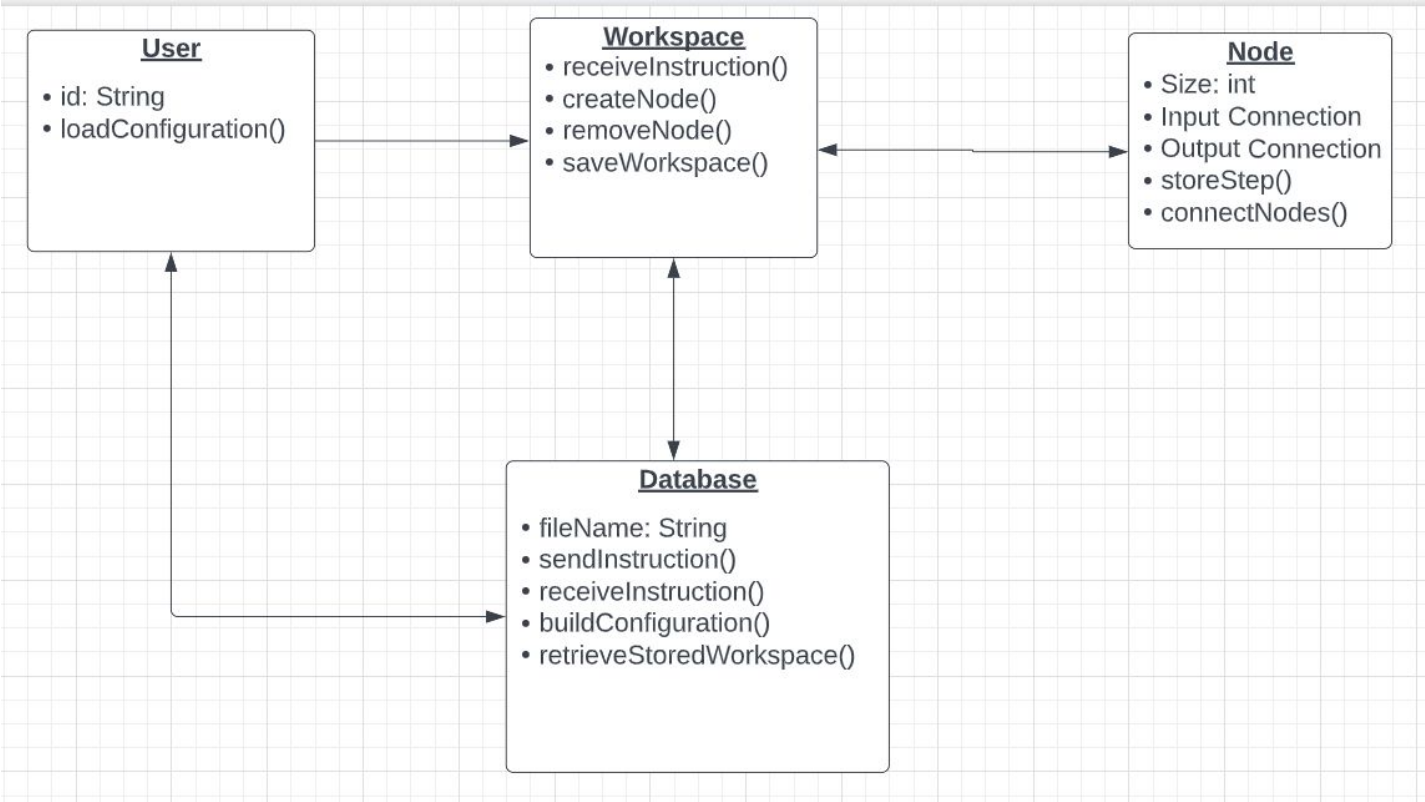
Nonfunctional Requirements

- The system shall load the options for nodes dynamically and quickly from the database.
- The workspace should be responsive and intuitive to use.
- The properties (length, shape) of the line should be dynamic according to where the two nodes are located.

Architecture

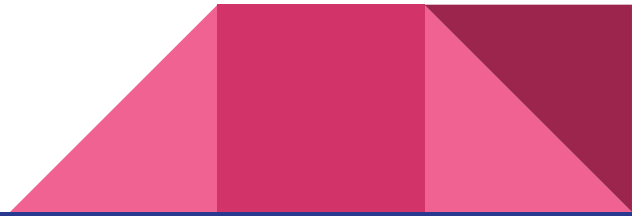


Domain Model



Tech Stack: PERN

- Node.js/Express.js
- PostgreSQL
- React Flow



Prototype

https://www.figma.com/proto/uKmmOhZwV9SzGHE5cxtLBT/LAP-VISUAL-IDE?node-id=7%3A8&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=7%3A8

- A very early idea of what the LAP visual IDE could look like.
- Several changes will be made in lieu of our mentor's and client's recommendations.

First Iteration Features

- The system shall give the user the option to browse from several types of items. Sequences, steps, or control modules.
- The system shall pull the different types of items from the database and display them in a pane.
- The system shall only allow control modules to connect to caste 2 sequences or the bottom level.
- The system shall only allow sequences to connect to other sequences or steps.
- The system shall only allow steps to connect to sequences of other steps.
- The user shall be able to download the project to an existing PLC.
- A node should store a single item.
- The system shall allow the user to connect two nodes together with a line.
- A node should have an input and output connection able to connect to and from other nodes.
- The user shall be able to select a configuration to act as a container for the project.
- The user shall be able to drag an item into the workspace, causing it to create a new node.
- The system shall allow the user to remove a node from the workspace.

Mentor Feedback

Recommended some project ideas

- Add grouping to nodes in order to label their functions
- Use cases should be more like user stories
- Business Requirements should be less broad

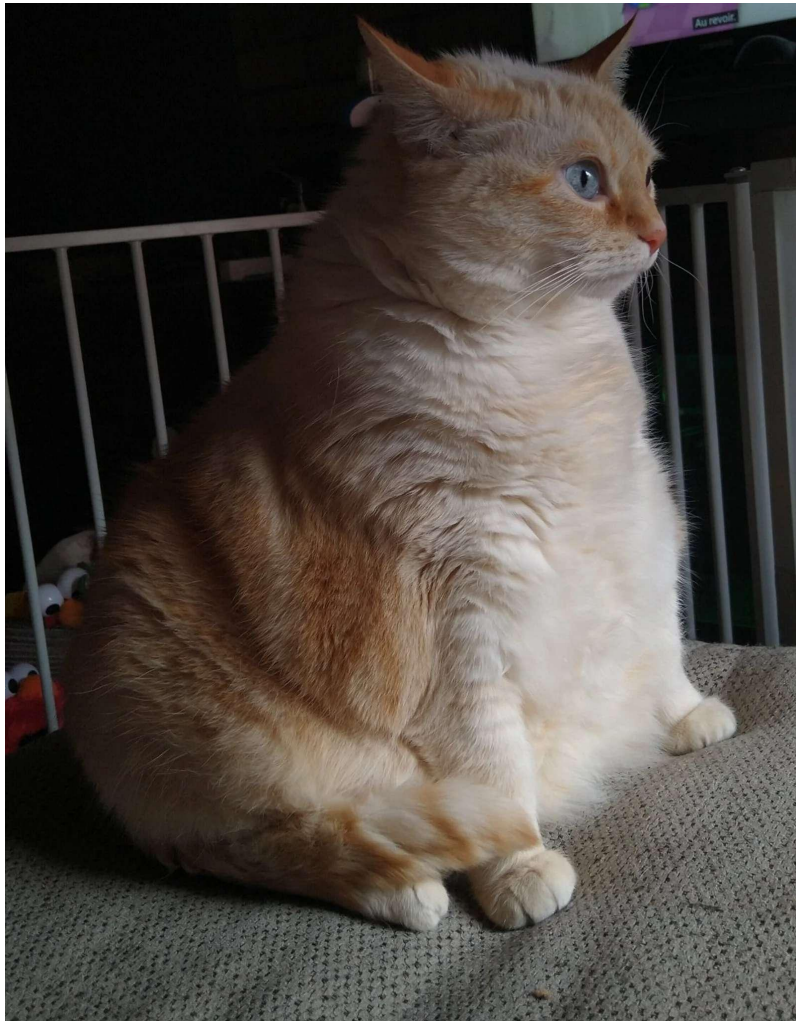
We implemented these changes before meeting with our client

Client Feedback

Informed us of things that will change in the project as a whole

- Clarified what configurations were and how they will be used
- Recommended an incremental approach, where requirements will not be known until previous requirements have been met.
- We will not have to download to the PLC
- A user will need to be able to fluently switch from an old GUI to our new one
- First iteration will focus mainly on the UI functionalities

These changes will most likely be reflected in our github by the end of this week



Just Keep Swimming
-Albert Einstein