LAP Visual IDE

Limitless Automation Possibilities

Team Info

Andrew

-

Will

- Just another broke college senior
- Javascript flavors aficionado

Jacob

- Big gamer
- College is hard

Alex

-

Wesley

-

Client Info

Blue Ridge Automation

- Develops automation software designed to make monotonous tasks easier
- Notable clients include:
  - Nestle
  - Kroger
  - Ingredion
  - Starbucks
- LAP is their main software that is sold
  - This is what our IDE will be for
  - Helps programmers develop automation code for factories visually and intuitively.

Mentor Feedback

Mid-iteration

- Project looks good
- Recommendations
  - Create a scrum master
  - Add more clarification on user features in the app
  - Setup a clarification meeting with the client
- These mid-iteration suggestions were implemented successfully.

Pre-release

- Project is ready for release
- Recommendations
  - More use of feature cards with team member assignments to each one
  - API needs fleshed out once iteration 2 starts
- Fleshing out of API will begin later this week

Client Feedback

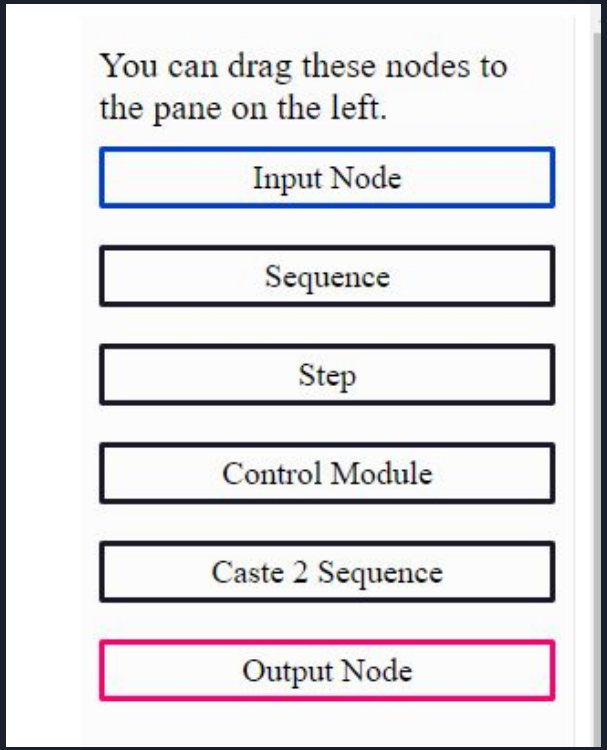
Really enjoyed the first iteration features.

- Drag and drop was clean and easy to use
- Recommendations
  - Save and restore is inefficient to use and not user friendly
  - Implement a menu based system that doesn't require the user to remember the configuration name
  - Client encouraged us to explore the SPROCs more

Iteration 1 Features

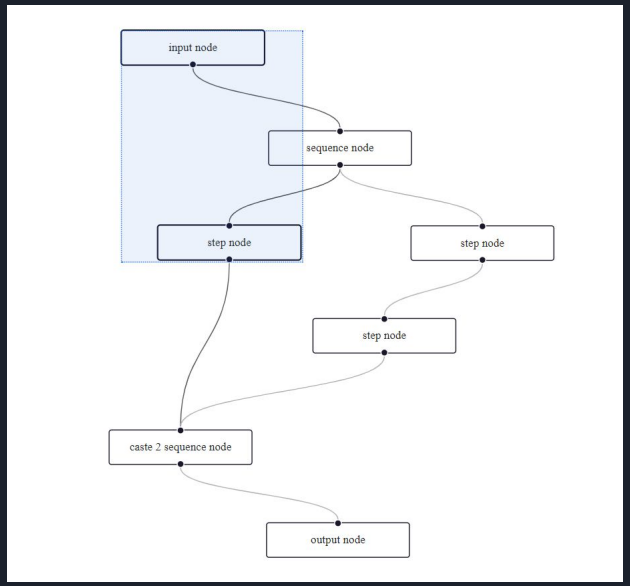
FR1: Sidebar of Node Types

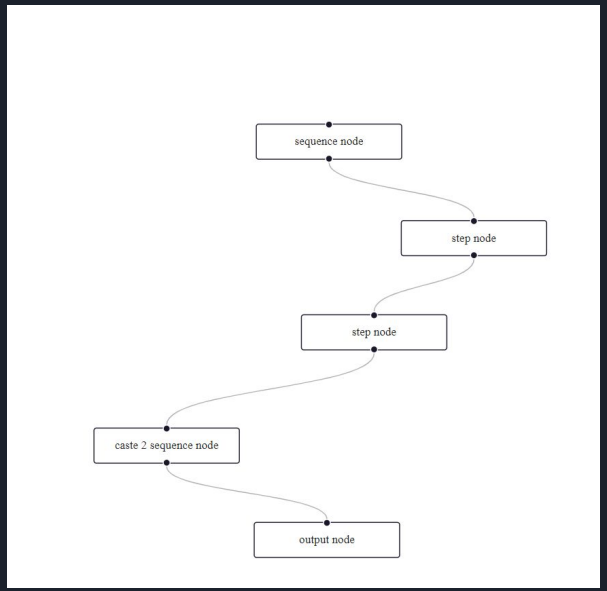
[FR1] The system shall give the user the option to browse from several types of items. Sequences, steps, or control modules.[BR2] [HIGH]



FR5: Remove Nodes

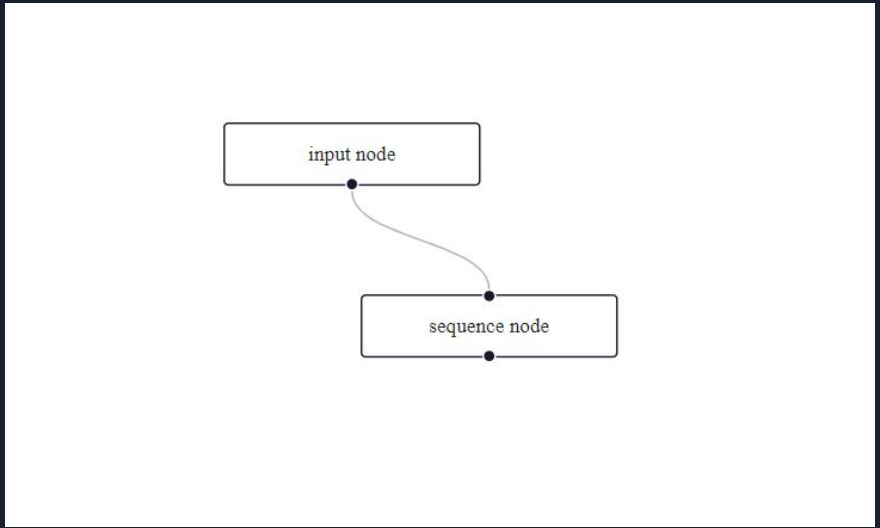
[FR5] The system shall allow the user to remove a node from the workspace. [BR1] [HIGH]





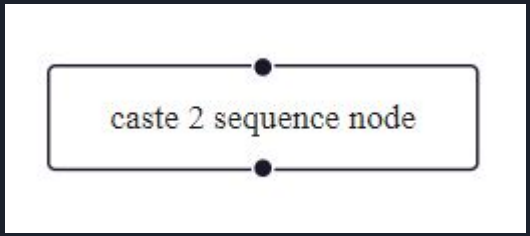
FR8: Connect Nodes

[FR8] The system shall allow the user to connect two nodes together with a line. [BR1] [MEDIUM]



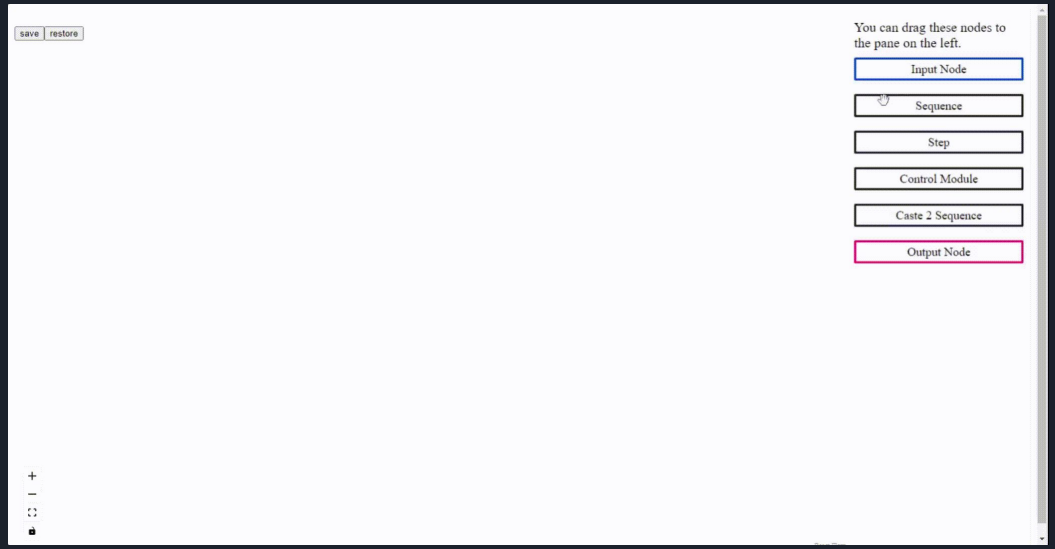
FR6: Node Connectors

[FR6] A node should have an input and output connection able to connect to and from other nodes. [BR1] [HIGH]



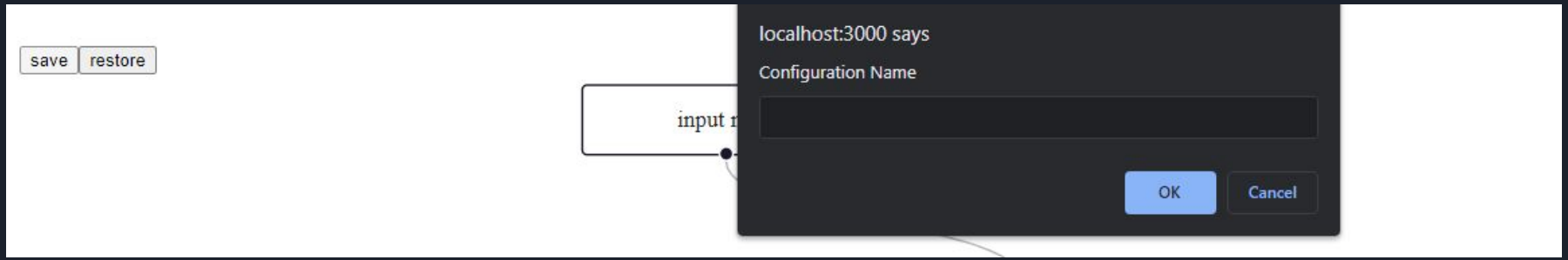
FR4: Drag and Drop

[FR4] The user shall be able to drag an item into the workspace, causing it to create a new node. [BR2] [HIGH]



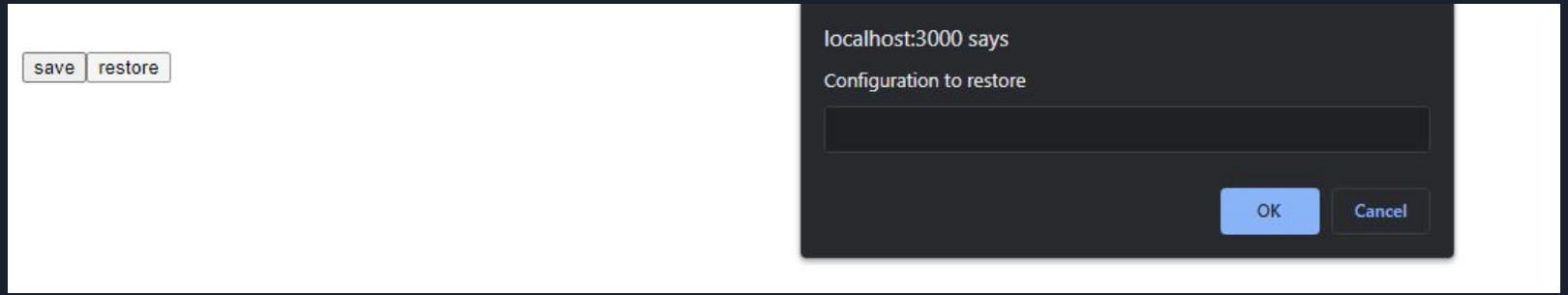
FR7: Saving Configurations

[FR7] The user shall be able to save a configuration to be worked on later [BR1] [HIGH]



FR3: Restoring Saved Configurations

[FR3] The user shall be able to restore a saved configuration and edit it. [BR1] [HIGH]



FR9: Zoom

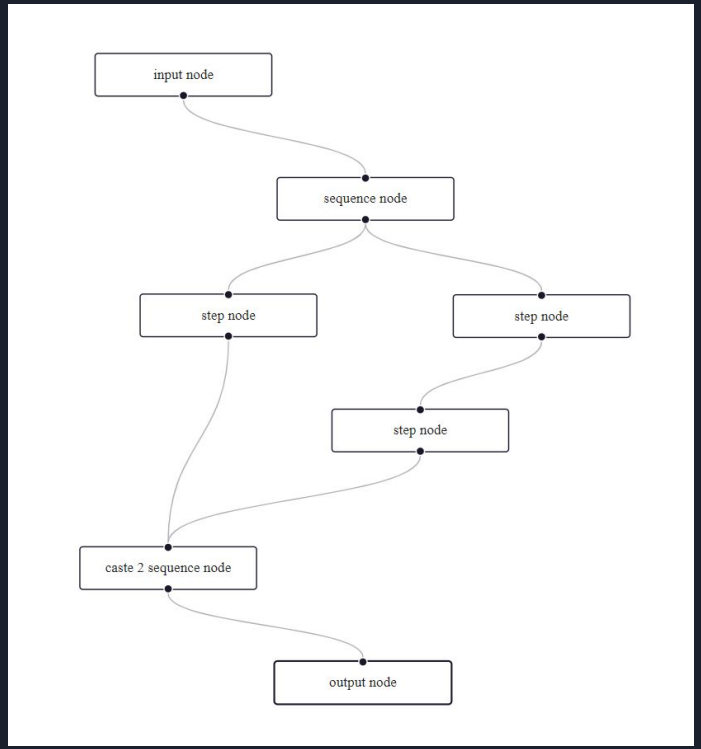
[FR9] The workspace can be zoomed in and out. [BR2] [LOW]

This was low priority, but it turns out this is built into reactflow

NR3: Dynamic Line Calculation

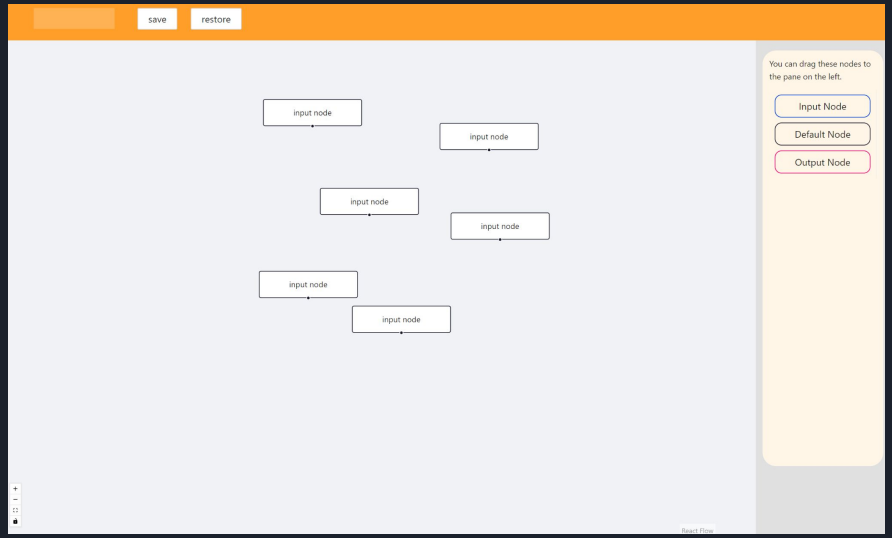
[NR3] The properties (length, shape) of the line should be dynamic according to where the two nodes are located. [BR2] [LOW]

*This was low priority, but was also built into the reactflow library



FR16: Menu Layout

[FR16] The application should have a layout that includes a space for; a file/database browser, a workspace and a menu [BR1] [MEDIUM]



Planned Iteration 2 Features

1. The configuration to edit will be pulled from the database.
2. Each node should call a stored procedure in the database upon saving.
3. The user shall be required to login before accessing the visual IDE.
4. The system shall only allow sequences to connect to other sequences or steps.
5. The system shall only allow steps to connect to sequences of other steps.
6. A REST API will run alongside the application to call existing stored procedures.
7. The workspace should be responsive and intuitive to use.

Random Slide

When there’s an agreed upon time to meet and nobody shows up