



Lev Tours

Dr. Michael Leverington - Sponsor
David Failing - Mentor

Members

Erik Clark
Kyle Savery
Alexis Smith
David Robb
Ariana Clark-Futrell

User Manual

4/27/2020

Overview

This document provides an in depth description as to how a front or backend user can operate the software. These instructions principally rely on screenshots of our Graphical User Interface.

Table of Contents

1. Introduction.....	2
2. Installation.....	2
3.1 Configuration and Daily Operation (Frontend).....	3
3.2 Configuration and Daily Operation (Backend).....	6
4. Maintenance.....	12
5. Trouble-shooting.....	13
6. Conclusion.....	13

1. Introduction

The purpose of this user manual is to give guidance on how our touring product works from the perspective of both the back and frontend. Frontend usage pertains to the touring capability of the Graphical User Interface (GUI), while backend use will cover how to set up and edit building data. Some importance features of this product include:

- Inputting a destination to receive orientation dependent directions to that destination
- Selecting a tour to receive directions to each location in the tour
- Setting up a building with floor maps and locations of nodes on each floor
- Creating tours based off of these nodes
- Storing Wi-Fi information at each node to estimate position during the operation of this software

2. Installation

For what is expected of our product, installation purely in terms of our software is straightforward as the product is already on the provided laptop with all of the necessary modules installed (the program will be located in the Lev Tours directory under the Desktop). In terms of installing the product into the robot this is left up to future capstone teams, but the essential idea is the robot will act as a frontend user to access a building's locations and directions to any nodes.

Nonetheless, here is a set of steps that can be taken to use our software on a new device.

- Make sure that Python is installed and updated to a 3.X version. The position estimates using Wi-Fi relies on the functionality of a Linux machine, but all other functionality can be achieved on any machine. If planning to use our Wi-Fi capabilities it is important to disable the need for the sudo password within the terminal. The call to scan nearby Wi-Fi signals must be run with sudo and disabling the sudo password was the only solution our team found. To disable the sudo password, enter 'sudo visudo' in the command line. A properties file will appear, and you need to append this line "USERNAME ALL=(ALL) NOPASSWD:ALL" at the bottom of the file. Replace USERNAME with the username of the machine. If the line is entered anywhere other than at the bottom this may cause an issue. Once this is complete our software can freely scan Wi-Fi information.
- All of the libraries listed below need to be installed.
 - datetime
 - logger
 - math
 - os
 - PIL
 - rssi
 - shutil
 - threading
 - time
 - tkinter
 - webbrowser

The majority of these libraries are part of the Python standard library, however PIL may need to be installed separately. The rssi library is included with our software.

- After all of these dependencies are successfully installed, load the source code into a directory on the desired machine, then open a terminal and change into that directory.
- Once there execute the command 'python3 Gui.py'. This will launch the software. If the software fails to launch, the most likely error is that a module has not been installed correctly and possibly resides in a directory that is not part of the running version of Python's path.

3.1 Configuration and Daily Operation (Frontend)

The section will cover the frontend usage of our software, that is, the components that all users are able to access. These are the information and tours page, with the information page supplying a simple frequently asked questions list and the tours page providing the user (and eventually the robot) with a way to navigate around the building.

3.1.1 Information Page

From the homepage (the first page that loads upon startup) a user can access additional information such as an FAQ and a link to our team website where they can view all of our documentation. Once directed to our website a landing page will appear informing the user that when they want to return to the application, they just need to exit the browser.



Figure 3.1: Info Page Button

3.1.2 Tours Page

When a user clicks on the Tours Page button from the homepage (Figure 3.2), they will be directed to a page where they can select a building (Figure 3.3). From here, users can double

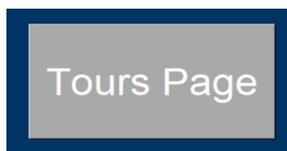


Figure 3.2: Tours Page Button

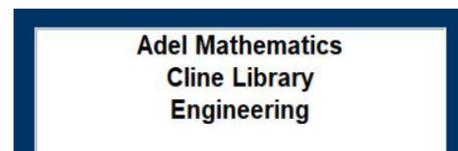


Figure 3.3: Building Selection

click on a building which will direct them to the navigation page. A user can also input a specific building in a search bar. If a backend user does not see a building that has been created, this is because only *connected* buildings are displayed on the front end. To be connected, a building must have at least one floor and on each floor any point must be accessible from any other point.

3.1.3 Navigation Page

After selecting a building you will be directed to a page that resembles the example in Figure 3.4.

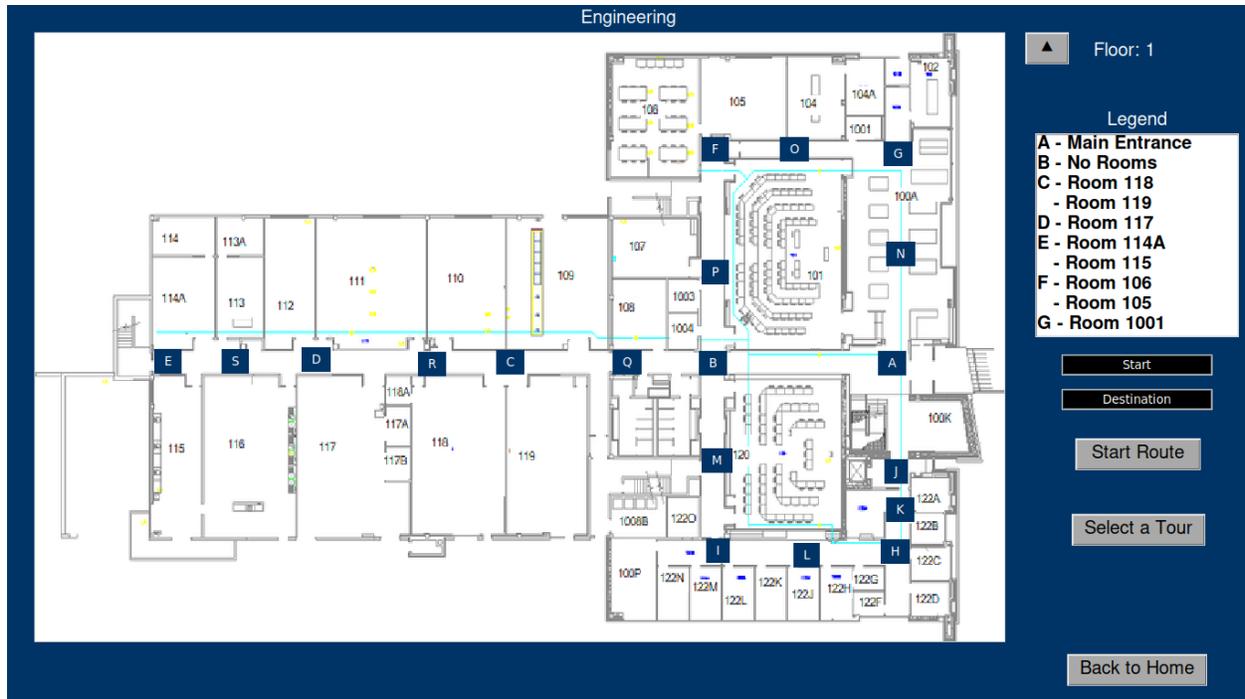


Figure 3.4: Navigation Page

This page will serve as the main area for all navigational activity. From here you can change the floor, start a route, or a tour. Shown is a map of the current floor, this will default to floor 1 (or the lowest floor on the building). On the map, all nodes will show up as a navy box with a label in them. In the top right all labels will be listed in a legend along with all of the rooms or areas they are meant to represent. For instance, node C represents rooms 118 and 119. There are two entry boxes below this listing as well: one for entering a starting location and one for entering a destination. To begin a route, enter appropriate values into these boxes and then click “Start Route” (details in 3.1.4). There is also a “Select a Tour” button, this will replace the list of rooms with a list of tours that have been given meaningful names (such as “Mechanical Engineering Tour”). Choosing one of these tours will begin a series of routes taking the user to several significant locations within the building (details in 3.1.5). To view a different floor, click on the arrows next to the floor name at the top right. If the user is already on the lowest floor then no down button will appear, likewise if they are on the top floor.



Figure 3.5: Close Up of Swapping Floors

3.1.4 Running a Route

You can input either a node's label or one of the associated names in order to visit it. Once you have entered starting and ending locations into the boxes shown above, clicking the "Start Route" button will mark your current location with a small robot image. The path to the destination will light up yellow in the order that the nodes will be traversed. Then the destination will be marked red.

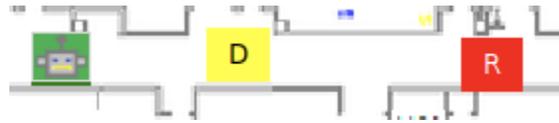


Figure 3.6: Route Example

Once all the necessary nodes have been highlighted the route will begin and instructions will be displayed at the bottom helping to guide you on your route.

Directions: Turn right and continue straight to L

Directions: Proceed straight to V

Figure 3.7: Example Directions

If you need to pause the current route, a "Pause" button will appear in the same location as the "Start Route" button. The software will then wait at the next node in the route and display a message box asking you when you want to resume.



Figure 3.8: Pause Button

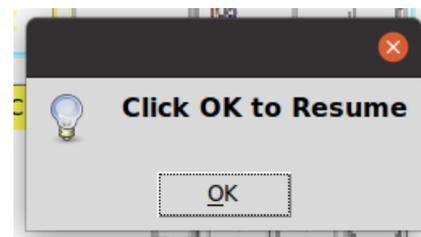


Figure 3.9: Resume Box

Once you have finished your route a box similar to the pause box will appear, letting you know you have arrived at your destination. Once you have clicked the 'OK' button on this box you can run another route or tour.

3.1.5 Running a Tour

To run a tour, click the "Select a Tour" button shown in Figure 3.4. The user can then select any of the available tours that appear in the tour list (Figure 3.10). Once they select a tour they can click the "Start Tour" button. The tour will then start routing you from place to place within the

tour much like how running a route worked. Once the tour is done a box will appear informing the user that the tour is complete.



Figure 3.10: Tour Selection Box

3.2 Configuration and Daily Operation (Backend)

This section will cover how backend users are able to edit building data as well as how to start up the program.

3.2.1 Starting the Software

To start the software, open the terminal on the machine. Navigate to the folder where the code is stored, as of writing this document the code is stored on the Desktop in a folder called "Lev_Tours". To run our software type in the following command: `python3 Gui.py`. To exit the software, hit the 'esc' key. This will take the GUI out of fullscreen mode and allow the user to terminate the software.

3.2.2 Logging Into the Admin Section

To log into the system from the Home Page, click the Admin Page button. This will take the user to a page where they are able to enter their username and password. The administration section of our GUI will initially be set up to have a login of **Username:** SmilingTourGuide and **Password:** capstone. After entering your username and password the GUI will display the Admin Landing Page as shown below.



Figure 3.11: Admin Landing Page

3.2.3 Adding a User

To add a new user to the software just click the “Add User” button on the admin landing page which will bring you to a section where you can type in a new username and password for a backend user. It will also have instructions on what type of characters can be used.

3.2.4 Creating a New Building

To create a new building click on the New Building button which will bring you to a page that allows you to add buildings into the software. To create the building, type the building’s name into the entry box (Figure 3.12) then click enter. This will then bring you to the new floor page. When adding floors to a building from this page, add them in order from the lowest to highest floor (e.g. 1 to 3). Type the name of the next floor to be added in the “Floor Number” entry box (Figure 3.13). Next you will need to add a map of this floor, to do this click the “Upload Map” button and a file browser will open up allowing you to select a map image from anywhere on the machine (Figure 3.14). The only files that are supported are ‘.jpg’ and ‘.png’. Once you have uploaded the map click the “Add” button. You can repeat this process as many times as you need to get all the floors in a building (Figure 3.15). Return to the admin landing page when you are done by clicking the “Back” button.

 The image shows a small form titled 'Enter the building's name'. It features a dark blue background. At the top, the text 'Enter the building's name' is displayed in white. Below this is a black input field containing the text 'Building Name'. At the bottom of the form is a grey button with the text 'Enter' in white.

Figure 3.12: Naming a Building

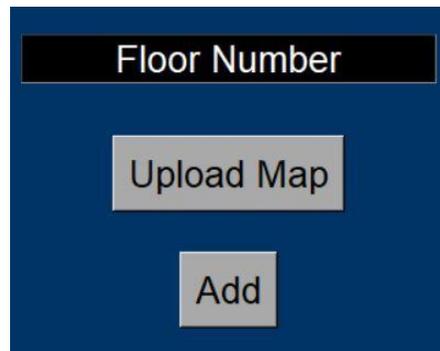


Figure 3.13: Adding a Floor

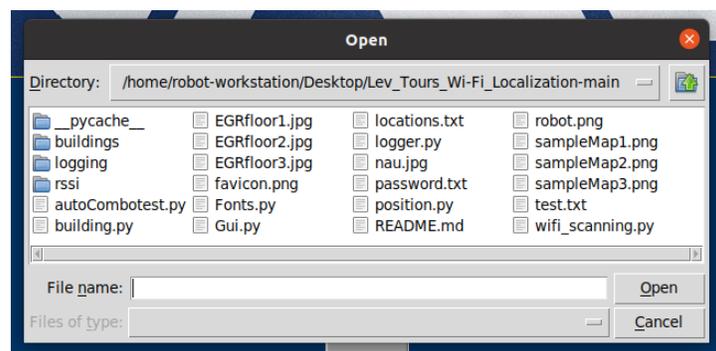


Figure 3.14: Uploading a Map

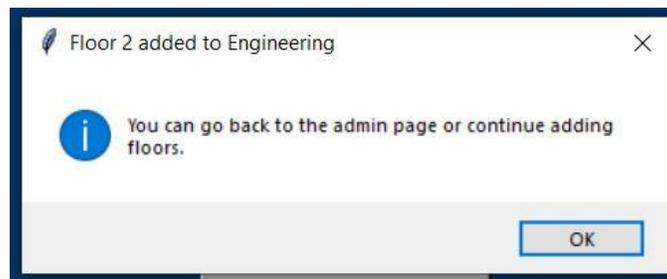


Figure 3.15: Floor Added Confirmation

3.2.5 Editing a Building

To start editing a building, hit the “Load Building” button on the Admin Landing Page. This will bring you to a section that allows you to choose which building you would like to edit. This works the same as the building selection in section 3.12, except every created building will be displayed regardless of connectedness. Once you have selected a building a screen that looks like the one below will appear.

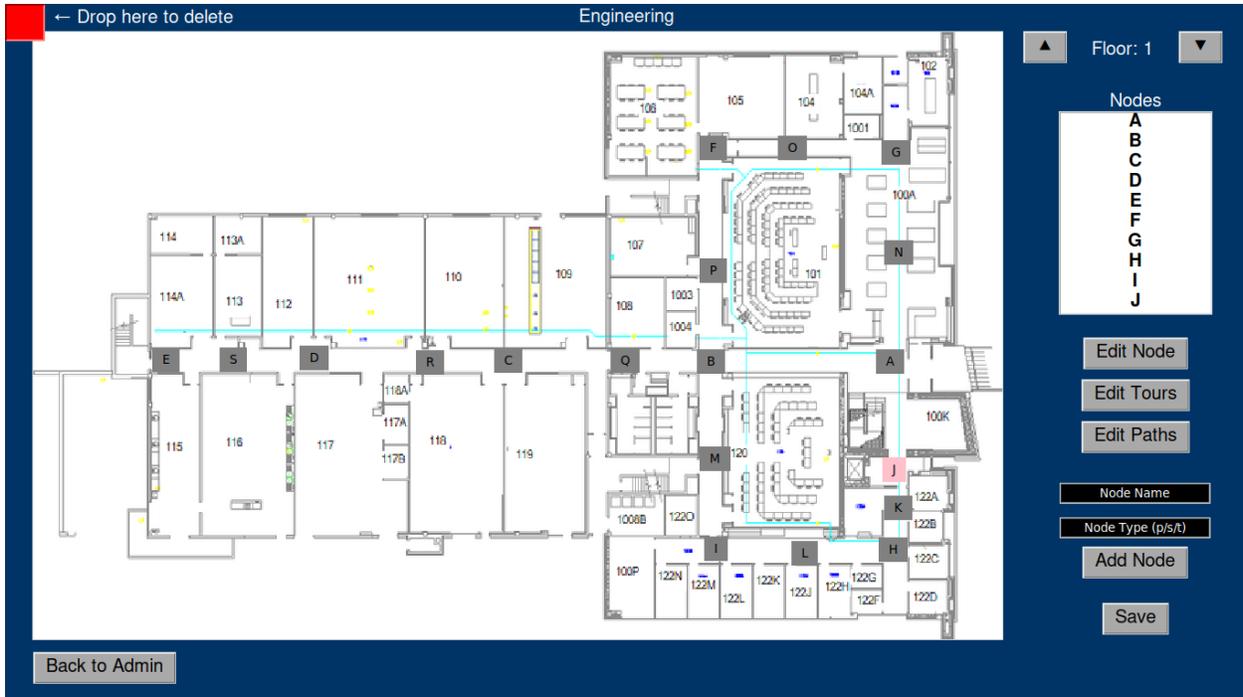


Figure 3.16: Building Editing Page

3.2.6 Nodes

A node in our software is a location within a building that has been given a label and is connected to other nodes. A node can have any number of associated names as shown in Figure 3.4. These are locations that the node is meant to represent.

3.2.7 Creating Nodes

When creating nodes, it is recommended to use an uppercase single or double letter name (e.g. A or AB). This allows all the nodes displayed on the map to be a consistent size. These labels are not the actual names of the nodes, just a label that is used to represent them on the map. Actual location names from the building can be added later (section 3.2.8). Once a label for a node is made, the user can enter it in the “Node Name” entry box. After this, they need to decide a type for the node which can be primary, secondary, and transitional. A primary node is something that will always be shown on a map. These nodes should be things such as any turns or cross sections in hallways, as well as end points of hallways. Secondary nodes are less important parts of a building. These can be rooms in the middle of a hallway where they are in between two primary nodes (or locations such as water fountains). A transitional node is used to distinguish an elevator on the floor that the robot will eventually use to traverse a building. Note that transitional nodes can only be elevators and not stairs. During editing, these transitional nodes will appear pink to stand out from the others. Once the type of node is decided, enter either “p”, “s”, or “t” in the “Node Type (p/s/t)” entry box. Now this node can be created by selecting the “Add Node” button. Node labels must be unique within a building and the software will alert the user if they have entered an existing label. After adding the node, a square icon will appear directly to the left of the “Add Node” button (Figure 3.17).

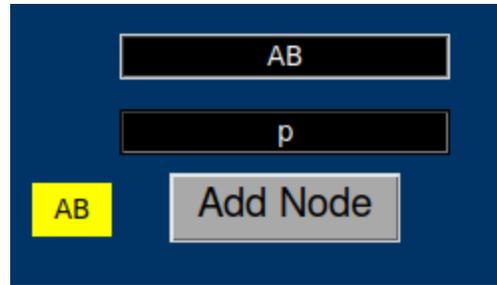


Figure 3.17: Adding a Node

After this step, the node can be dragged and placed anywhere on the map. To delete a node, drag it into the upper left region of the editing page and drop it into the red box.

3.2.8 Edit Node

Once a node has been created, click the “Save” button to update the list of available nodes. These entries can now be selected using the “Edit Node” button. As shown below, a new menu will open on the right side of the screen. This list, which will be empty when editing a new node, displays all the associated rooms to the node. To add an associated room enter its name in the “Enter Nearby” box and click “Add Nearby”. This location will then be displayed in the list. To remove a location, select one from the list and then click “Delete Nearby”. The last way to edit a node involves performing a series of scans that measure the Relative Strength Indicator Values (RSSI) for each detected access point. If a node already has RSSI information stored, then there will be a label that reads “Wi-Fi Stored: YES”. Otherwise the label will read “Wi-Fi Stored: NO”. If you wish to store an RSSI data set with this node, travel to the physical location in the building and then click “Scan Wi-Fi”. The software will then display a message reminding you not to move until all of the scans are complete. Selecting “Delete Wi-Fi Data” will remove all RSSI values stored with a node.



Figure 3.18: Edit Node Menu

3.2.9 Edit Paths

A path defines how nodes are connected in the building. Transitional nodes are logically linked to every other transitional node in the building automatically. In order to edit a path or see which paths exist on the current floor, click the “Edit Paths” button as shown in Figure 3.16. This will bring up the edit paths menu displayed in Figure 3.19 on the right side of the screen. To add a path, enter the labels of the two nodes to be connected in the “Node 1” and “Node 2” boxes (order does not matter). Then put the distance in feet into the “Node Distance” box and hit “Set Path” and a path will be added to the list. To remove a path, enter the labels of the two nodes you want to disconnect and enter either “inf” or “infinity” into the distance box.

Figure 3.19: Edit Paths Menu

3.2.10 Edit Tours

Tours are multiple routes linked together. Any tour can span multiple floors or just a single floor. They can also visit a node more than once if they are designed this way (such as a tour that starts and ends in the same spot). To create a tour, click the “Edit Tours” button (Figure 3.16), this will pop-up a menu on the right hand side of the screen just like for editing paths and nodes (Figure 3.20). Here you can enter a descriptive name for the new tour in the “Enter Tour Name” box. Then select “Add Tour” and make this tour available in the Created Tours list. If you want to edit a tour, select one from this list and then click “Edit Tour”. Another menu (Figure 3.21) will appear allowing you to choose what nodes you want to add to the tour. In this menu you can select a node from the top list and add it to the bottom of the lower list. You can also select a node on the bottom list and move its position with the up and down arrows located on the left and right of this list. You can select a node from the lower list and hit the “Delete from Tour” button to remove that node from the tour. Once the tour is complete you can select the “Back” button to return to the main editing page. Tours are saved automatically.



Figure 3.20: Create Tour Menu

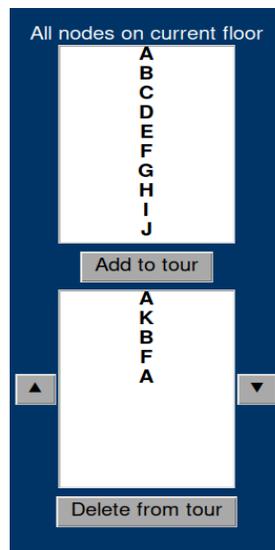


Figure 3.21: Edit Tour Menu

3.2.11 Adding Extra Floors

If you would like to add a floor to the building you are working on, go all the way to the top floor and hit the up arrow button again. This will ask you if you would like to add another floor to this building. Selecting “Yes” on this prompt will direct the user to the Add Floor Page as depicted in Figure 3.13. Follow the same steps taken to create a new building in order to add another floor.

4. Maintenance

This section will provide information as to how our software product should be maintained over the course of its life. Our system is relatively low maintenance as it is completely self contained. Currently, the provided laptop is running Ubuntu 20.04. While we do not anticipate that this

should cause issues with our program, after the laptop is updated ensure that all of the features described in sections 3.1 and 3.2 are still operational. If any errors do occur, they may be minor and easy to fix, otherwise we would recommend reverting to a previous Ubuntu version if at all possible.

The only output from our program is a log file that is generated on every run of the system. These log files are stored inside of the 'logging' directory. Inside this directory there are some number of directories named for each day the program was executed. These directories contain indexed log files denoting a unique run of the program. This is to ensure previous data is always saved in an accessible way and any file can be removed if desired.

Other than updating the laptop and manipulating the log files, there are no other operations our team would deem as "maintenance". However, in terms of future teams expanding the software it is important to note where Wi-Fi scanning operations take place. Within the file 'wifi_scanning.py', there exists an object named *RecurringScans* that subclasses the built-in "Thread" object. This object accepts a list of any number of node objects, and contains a 'run' function that will repeatedly scan nearby Wi-Fi and estimates the device's current position against the provided nodes. These position estimates are sent to the log file under the prefix "POSITION ESTIMATE", which is mainly for demonstration purposes and these outputs can eventually be used by the robot. Currently, the 'wifi_scanning.py' module is imported to 'Gui.py' as 'ws'. Within the GUI, scanning is started by calling `ws.start_scanning(node_list)` and stopped by calling `ws.stop_scanning()`. With these position estimates future teams (along with a more advanced Wi-Fi analysis technique, such as Artificial Intelligence) may make the robot's localization functionality to be accurate to within the sub 2 meter range.

5. Troubleshooting

- If the program becomes unresponsive after attempting to scan Wi-Fi during node setup or after starting a route, it is likely that the machine has not had the sudo password disabled. Unfortunately, the best solution to this problem is to restart the laptop and check the configuration of sudo using the instructions in section 2. A restart is required because the terminal will wait on the sudo password input from the user, but also prevents the GUI from being exited via the 'esc' key.
- If you have set up a new building, created nodes, and paths for those nodes, but the building does not show up in the listbox prior to the Tour Page then at least one node has been left disconnected from the rest. To resolve this issue, travel back to the building editing page and select "Edit Paths". Here you should ensure that every node you meant to connect has actually been connected.

6. Conclusion

With this user manual complete we would like to thank you, Dr. Leverington, for being a very insightful client and it was a pleasure working with you over the course of this project. Our team wishes you the best of luck going forward with other capstone teams in getting the 30-Gallon Robot to be a truly autonomous tour guide. Below is a list of our members' contact information,

feel free to reach out at any time in the future and we would be happy to help answer any questions about our software.

- Erik Clark - eaclark597@gmail.com
- Kyle Savery - kyledsavery@gmail.com
- David Robb - Dmrobb7@gmail.com
- Ariana Clark-Futrell - Arianafutrell@gmail.com