Computer Science Department

CS3733-B17

Professor Wilson Wong

Team F: The Fuschia Faeries

Final Iteration User Manual

Project Manager: William Godsey

Assistant Project Manager: Talal Jaber

Lead Software Engineer: Luke Ludington

Assistant Software Engineers: Benjamin Mattiuzzi and Nikolaos Kalampalikis

Product Owner: Floris van Rossum

Test Engineer: Stephanie Racca

Documentation Analyst: Griffin Roth

Software Engineers: Yuan Wang, Andrew Schueler, and Parmenion Patias

Team Coach: Binam Kayastha

Table of Contents

Table of Contents 1

Intro\Safety Precautions 2

System Requirements: 2

Minimum System Requirements: 2

Running the Application: 4

Start Screen and Basic Operation: 4

Login In: 4

Step-by-step operation of all features: 6

Pathfinding: 6

Algorithms and Characteristics: 7

Map Builder: 8

Add/Edit Node 9

Delete Node 9

Add/Edit Edge 10

Delete Edge 10

Edit Users: 10

Add User: 11

Remove User: 11

Edit User: 11

Service Request Components: 12

Food Request 13

Assistance Request/Cleaning Request/Security Request/Kiosk Issue 13

Transport Request 14

Overwatching and Editing requests: 14

Refreshing Requests: 14

Timeout and how to Change Timeout: 15

Application Program Interface (API): 15

Search Engine: 16

Feature 1 Autocomplete: 16

Feature 2 Error Fix: 16

Feature 3 Synonym Check: 17

Visitor/User Features 17

The Emergency Button: 17

The Clear Button: 17

The Reverse Button: 17

Maintenance **18**

Troubleshooting Tips **18**

Limited Warranty (1 week) **18**

Intro\Safety Precautions

Please review this document before using the application as it can prove crucial in facilitating ease of use and ensuring minimal issues.

System Requirements:

Before running our application please make sure your system covers the minimum requirements.

!Disclaimer: *The application may not run, and/or may stop running at any point if your system doesn't cover the minimum requirements.*

Minimum System Requirements:

Ram: 4GB
Graphics: Integrated Graphics
Hard Drive: 4GB
OS: Windows: Vista and older
Linux: Ubuntu 14.04 or older or similar
Macintosh: OS X 10.10 Yosemite
Software Required: Java 8 or older

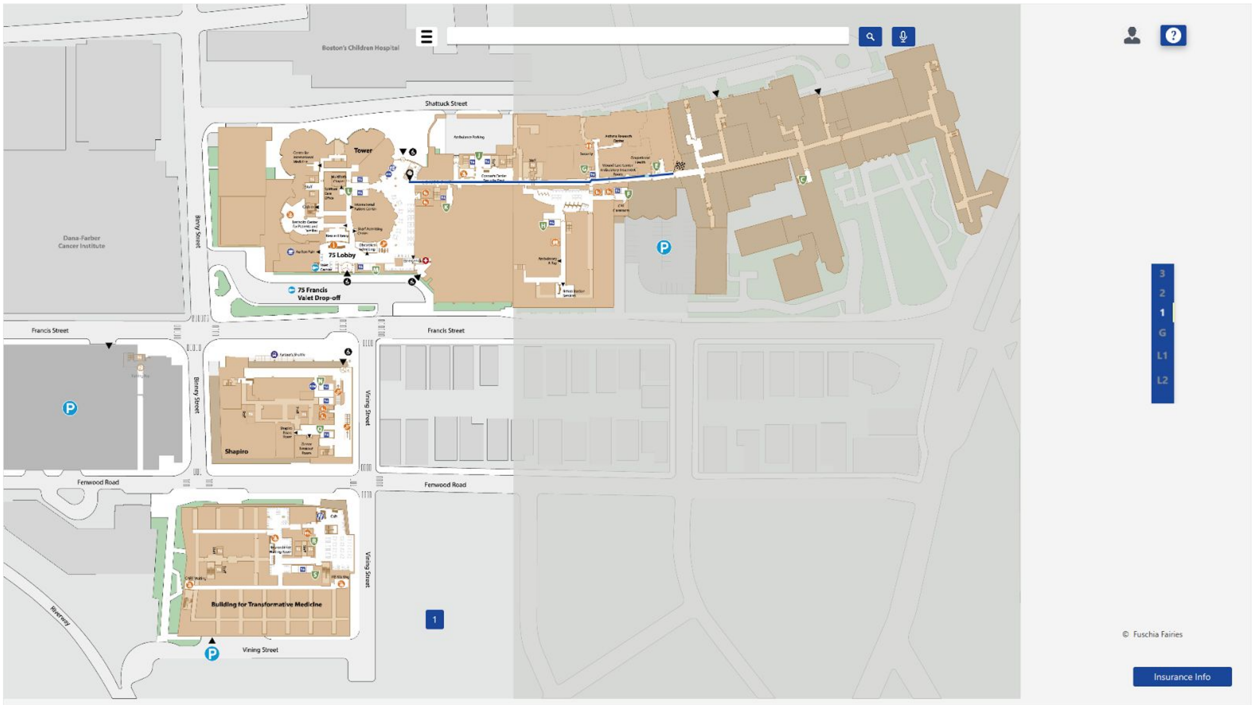
! For optimal performance the developers suggest:
Ram: 8GB or more
Graphics: GeForce 2GB 960
Hard Drive: 8GB or more
OS: Windows: 7 or older
Linux: Ubuntu 16.06 or older or similar
Macintosh: MacOS High Sierra or older
Software Required: Java 8 or older

Running the Application:

The application has been delivered to you as a .Jar file. To run the application you need to have Java 8 (or older) installed in your system. If you have Java installed but your system is unable to run the application try downloading and installing Java's latest update found here: https://www.java.com/en/download/

Start Screen and Basic Operation:

Once you have made sure your system covers the minimum requirements, you can open the .Jar file and you will be instantly transferred to the following screen.

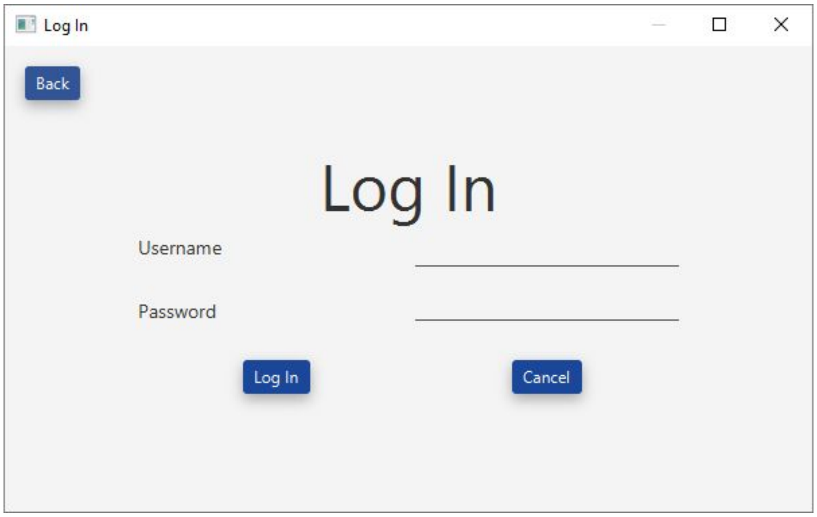


This is the start screen of the program from here you can move and utilize all our features.

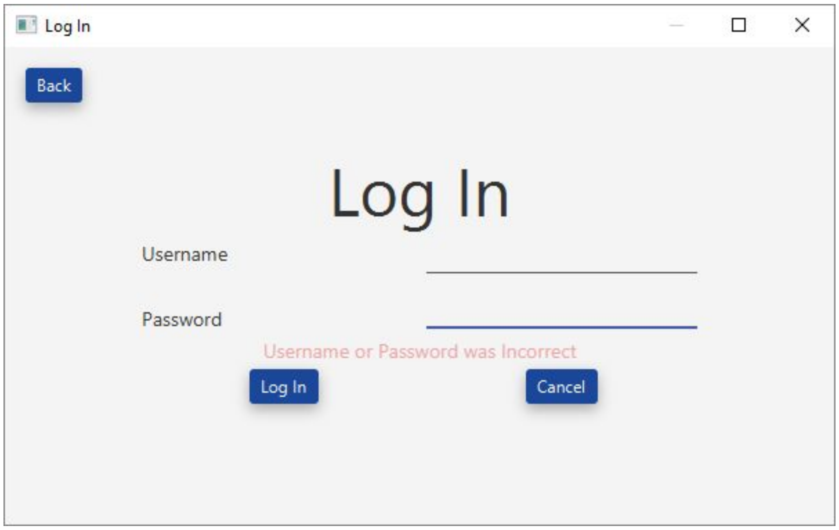
Login In:

To login the user has to click on this icon 👤 located at the top right part of most screens.

This will get you to the login screen, which will ask you for a Username and Password.



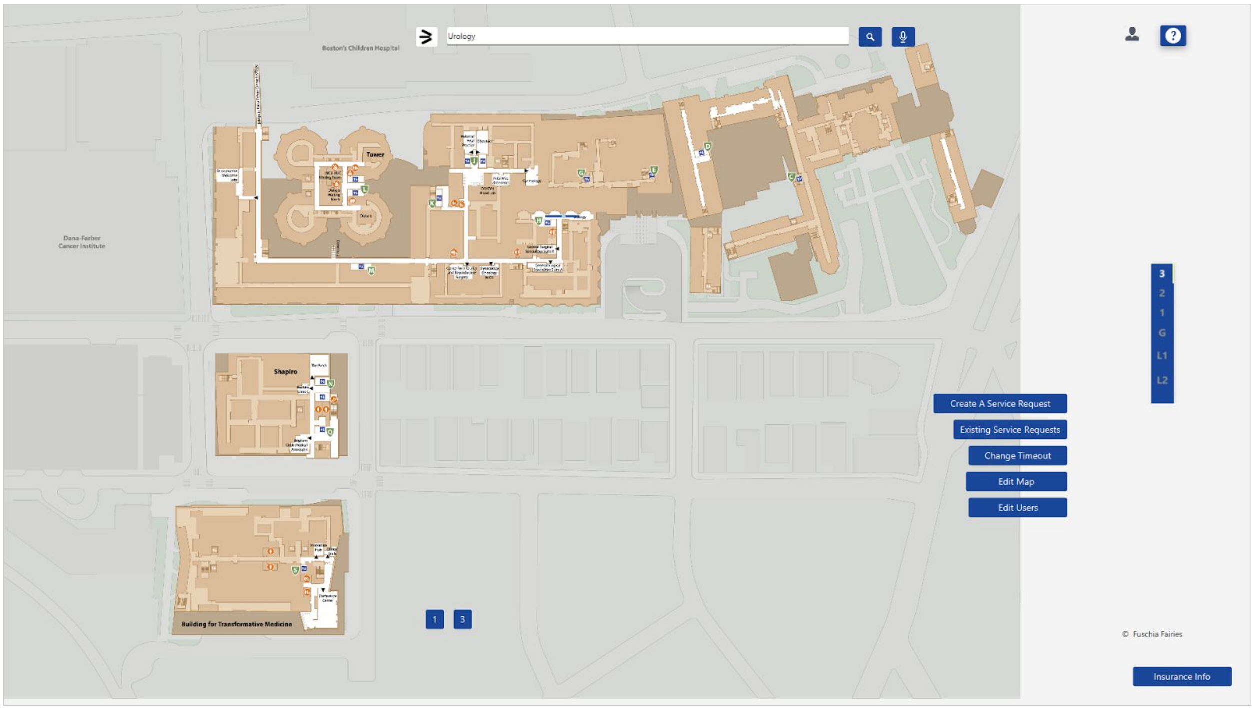
Putting in a wrong username or password will result in this following message on the login screen:



Username or Password was Incorrect

!Note: Please remember that both the username and password are case sensitive so you need to remember which characters are capitalized and which are not.

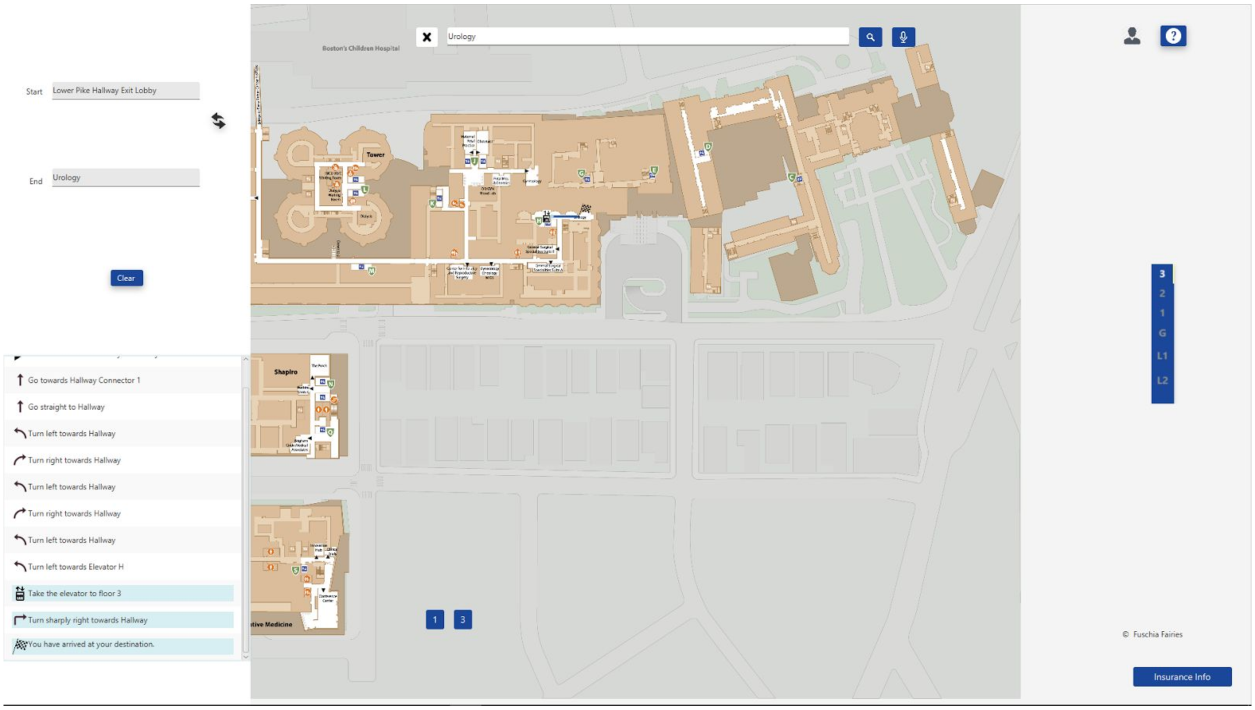
Once logged in, a screen like the following will appear.
!Note: The following screen is the login screen of an Administrator, other screens will be similar but with less functionalities.



Step-by-step operation of all features:

Pathfinding:

Pathfinding is one of the included features, that has been implemented and tested before been integrated in the application. In pathfinding you can choose the start and end node you want to traverse between and the algorithm you want to use to calculate your path. Pathfinding also allows you to find an *accessibility path* for people that cannot use stairs or escalators. Each algorithm has different characteristics, and one should make sure she/he is using the algorithm that best suits her/his needs. As of now the pathfinding algorithm for non-logged in users is A*.



Start Lower Pike Hallway Exit Lobby

End Urology

↑ Go towards Hallway Connector 1

↑ Go straight to Hallway

↰ Turn left towards Hallway

↱ Turn right towards Hallway

↰ Turn left towards Hallway

↱ Turn right towards Hallway

↰ Turn left towards Hallway

↰ Turn left towards Elevator H

Take the elevator to floor 3

↱ Turn sharply right towards Hallway

You have arrived at your destination.

In the photo above you can see the UI once you have searched for a destination. The line shows the path one needs to follow. In the lower left part of the program one can see the Hierarchical directions. Things highlighted in blue are in your current floor. The flag points the final destination. In the lower central part of the code the user can see all the floors he needs to go through to get from the initial to final destination. Clicking on the floor changes the map, to that of the selected floor and highlights the directions that refer to that floor.

Algorithms and Characteristics:

A* (A star): This is the best algorithm to find the fastest route between 2 points. It takes the least amount of time to calculate. More on the algorithm can be found here: http://ieeexplore.ieee.org/document/4082128/

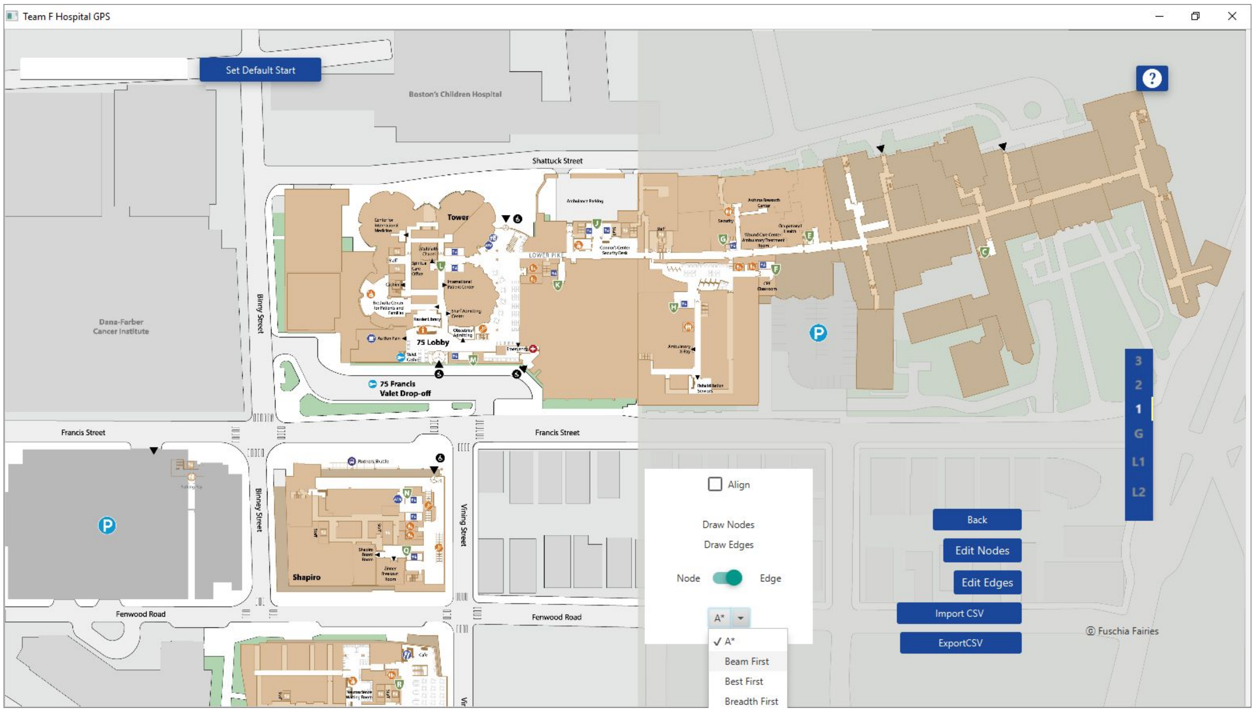
Dijkstra: This algorithm will find the same route as A* BUT will take longer to calculate.
More on the algorithm can be found here:
http://www-m3.ma.tum.de/foswiki/pub/MN0506/WebHome/dijkstra.pdf

Depth First Search: This algorithm will usually find one of the longest paths to get from start to end node. This is the best for people that want to get to know the hospital and most (if not all) of its floors and departments. More on the algorithm can be found here: https://arxiv.org/pdf/math/0610935.pdf

Breadth First Search: This algorithm will find a good enough (not optimal) path to get you from your start to your end node. More on the algorithm can be found here: http://ieeexplore.ieee.org/document/5219222/?arnumber=5219222

Best First Search: This algorithm will also find a good enough but not optimal path that consists of the the nodes that are the closets between each other. More on the algorithm can be found here: http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume28/coles07a-html/node11.html#modifiedbestfs

Beam Search: This algorithm will also find a good but not optimal path. The path created by this algorithm is usually better than that of Best First Search, but rarely anything close to the path given by A* or Dijkstra. More on the algorithm can be found here: http://www.springer.com/us/book/9780387988320



In the photo above you can see the algorithm choosing dropdown bar, which can be accessed after login in as an administrator.

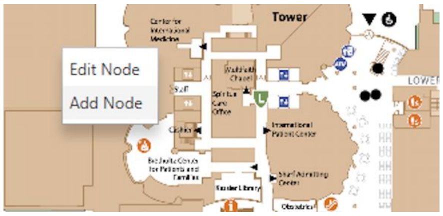
Map Builder:

The MapBuider is the part of the application that deals with parsing through the .csv files that contain all information regarding the Nodes and Edges of the map. Once these are parsed *the data is then saved in the database of the code.* In your .jar file we have added all nodes available to the production team (Fuschia Fairies) as of 12/12/2017. We know that some nodes may change so we have implemented functions that allow you to add and delete nodes and edges in the system. To avoid system breaches and difficulties these functionalities can be only accessed by the system administrators.

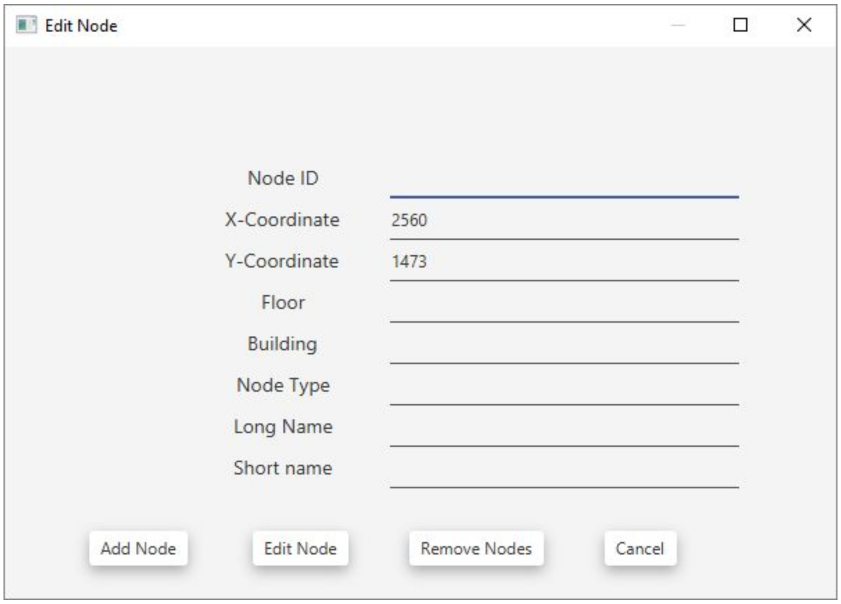
Note: *We call the points of interest* ***Nodes*** *and the connection between two points of interest,or Nodes,* ***Edges****.*

Add/Edit Node

1 Click the *logIn* button
2 Log in as an Administrator
3 Click on the *Edit Map* button under settings
4 *Right Click* on the map where you want to position the new node
5 Click on the *Add Node* pop up button



6 On the new pop up window fill out the rest of the fields
7 Once you have filled all fields click *Add Node*



Edit Node

Node ID
X-Coordinate 2560
Y-Coordinate 1473
Floor
Building
Node Type
Long Name
Short name

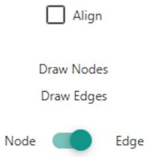
Delete Node

1 Click the *logIn* button

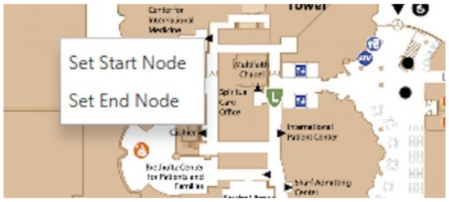
2 Log in as an Administrator
3 Click on the *Edit Map* button under settings
4 On the map left click near the node you want to delete
5 On the popup window click *Remove Nodes*

Add/Edit Edge

1 Click the *logIn* button
2 Log in as an Administrator
3 Click on the *Edit Map* button under settings
4 Move the bar from N… to E…



5 Right Click on the Map close to the start or end node of the edge you want to add



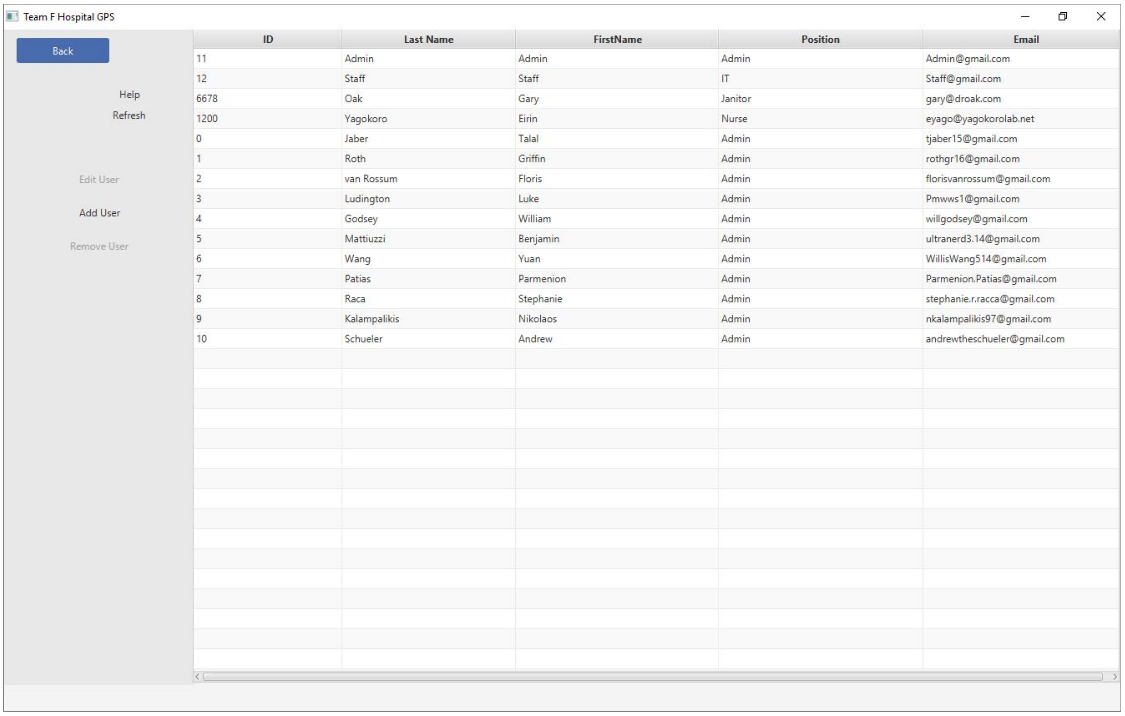
6 Right Click on the Map close to the end or start node of the edge you want to add
7 On the popup button click *Add Edge*

Delete Edge

1 Click the *logIn* button
2 Log in as an Administrator
3 Click on the *Edit Map* button under settings
4 Move the bar from N… to E…
5 Right Click on the Map close to the start or end node of the edge you want to delete
6 Right Click on the Map close to the end or start node of the edge you want to delete
7 On the popup button click *Remove Edge*

Edit Users:

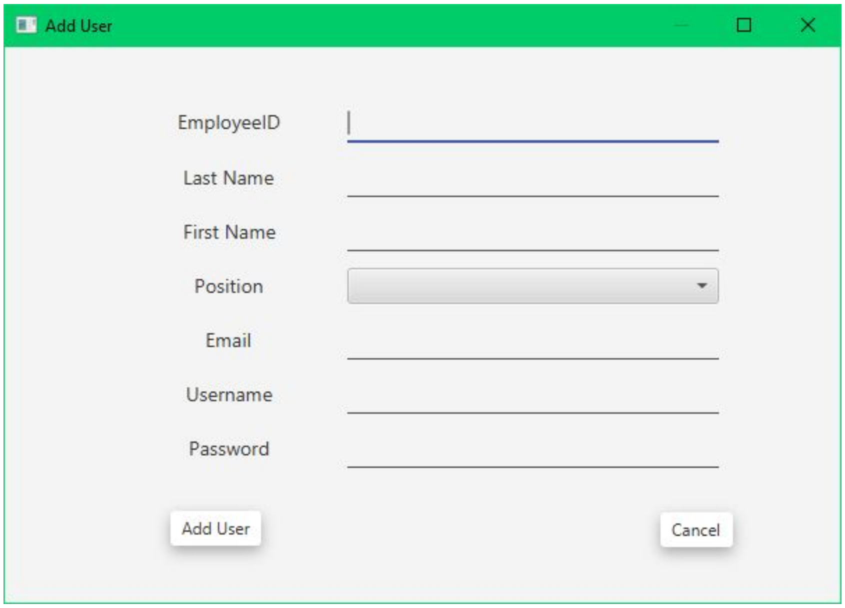
Edit users allows administrators to edit, add and/or remove users. By users we mean people that have Login access to the application.



ID	Last Name	FirstName	Position	Email
11	Admin	Admin	Admin	Admin@gmail.com
12	Staff	Staff	IT	Staff@gmail.com
6678	Oak	Gary	Janitor	gary@droak.com
1200	Yagokoro	Eirin	Nurse	eyago@yagokorolab.net
0	Jaber	Talal	Admin	tjaber15@gmail.com
1	Roth	Griffin	Admin	rothgr16@gmail.com
2	van Rossum	Floris	Admin	florisvanrossum@gmail.com
3	Ludington	Luke	Admin	Pmwws1@gmail.com
4	Godsey	William	Admin	willgodsey@gmail.com
5	Mattiuzzi	Benjamin	Admin	ultranerd3.14@gmail.com
6	Wang	Yuan	Admin	WillisWang514@gmail.com
7	Patias	Parmenion	Admin	Parmenion.Patias@gmail.com
8	Raca	Stephanie	Admin	stephanie.r.racca@gmail.com
9	Kalampalikis	Nikolaos	Admin	nkalampalikis97@gmail.com
10	Schueler	Andrew	Admin	andrewtheschueler@gmail.com

Add User:

To add a user, click on the *Add user* button, and fill in the relevant fields. Under the position drop down are the options Admin, Helper, Cleaning, Medical, Security, and IT.

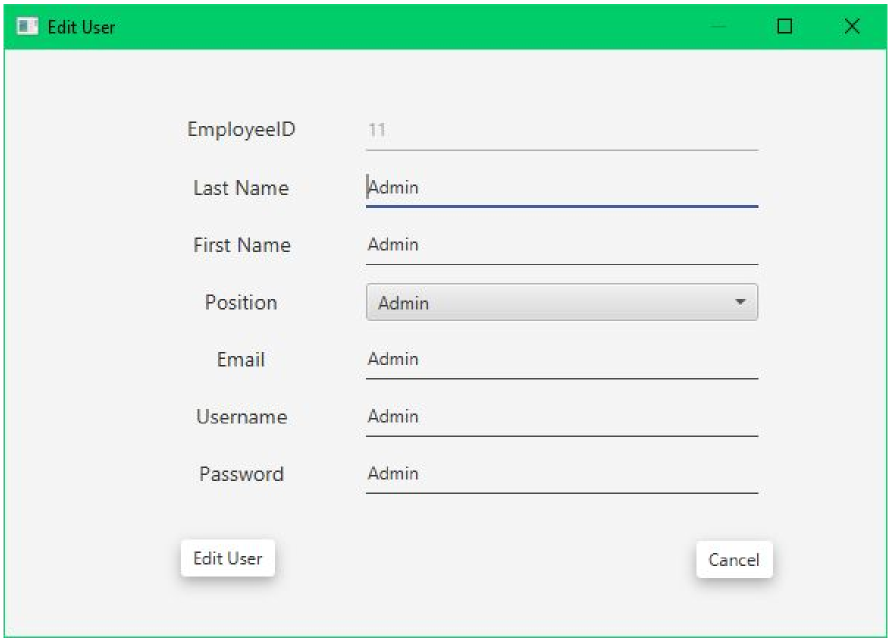


Remove User:

To remove a user an user, logged in as an administrator has to click on the user he wishes to remove and then click the *Remove User* button on the lower left part of the screen.

Edit User:

To edit a user the administrator has to click on the user he wants to edit and then click the edit user button. This will bring up the following screen. Once the user has been edited, clicking on the *Edit User* button will save the changes and clicking on *Cancel* will discard all changes.



EmployeeID	11
Last Name	Admin
First Name	Admin
Position	Admin
Email	Admin
Username	Admin
Password	Admin

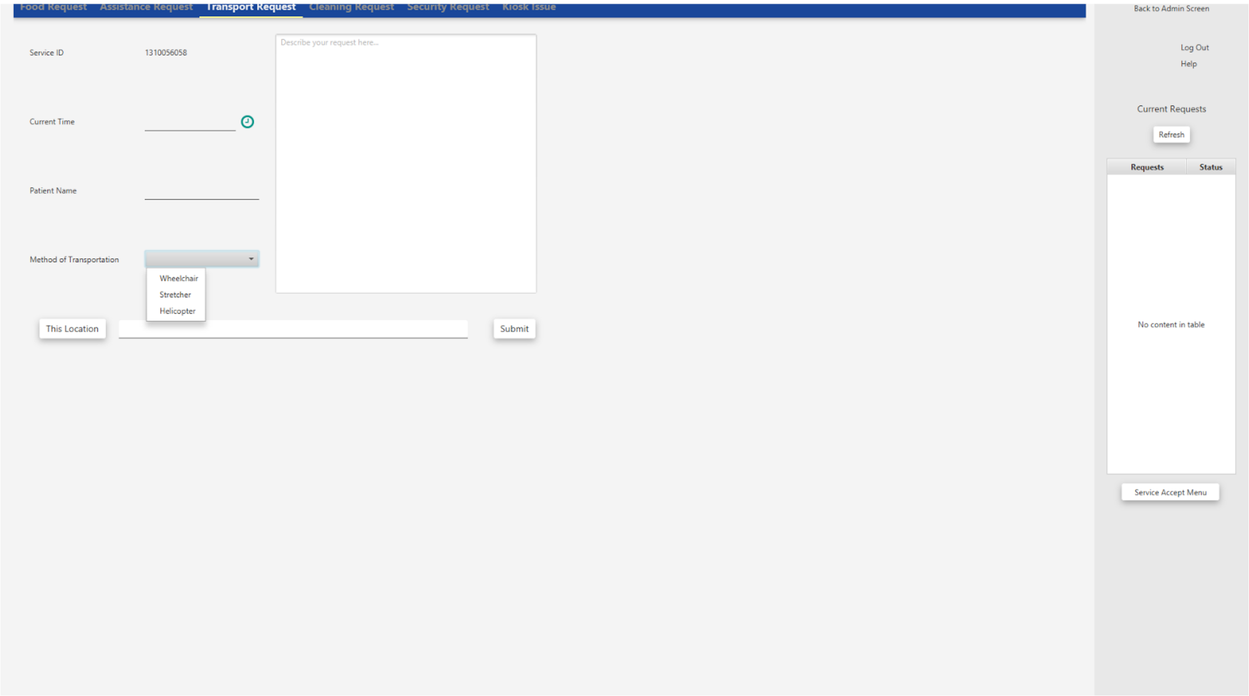
Service Request Components:

Making requests is one of the most important parts of the application and efficient operation of a hospital, that varying from a clean request (requesting cleaning personnel) to an emergency helicopter transportation of a patient (requesting helipad clearance, a helicopter and the people needed to make the transfer). Our development team has included 6 different types of requests one can make once she/he Logs In the application.

Steps to access the *Create A Service Request* menu:

1 Log in the application
2 Click on the Create A Service Request button
3 Click the request you would like to make

Note: *Once you click on the Create A Service Request button the application automatically assigns a Service ID to your request.*

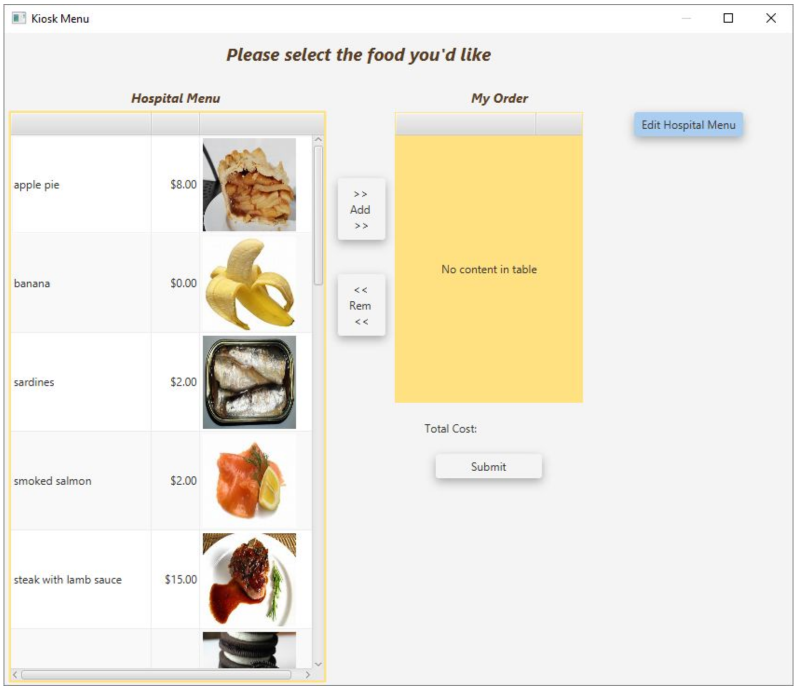


Service ID 1310056058

The photo above shows the *Create a Service Request* menu (specifically the create a Transport Request). Using the selection menu at the top the user can choose the request she/he wishes to make.
!Note: Directions on how to make each request follow below

Food Request

Given that you have completed the steps above, you now have fill out the *Current Time* and the *Time to Be Served* Button to set the time that you are making the request and the time you want the food to be served respectively. Then write the name of the patient you want the food served to. Then clicking on the food button takes you to a new menu where you can choose the food you want delivered from the foods available. The application also shows the user the price they will need to pay. Then choose the location where you want the food submitted and if you have any comments you can write them on the assigned space over the submit button. Once you have filled all fields you can click the Submit button and the request will be saved.



apple pie	$8.00
banana	$0.00
sardines	$2.00
smoked salmon	$2.00
steak with lamb sauce	$15.00

!Note: This is the Choose Food menu found in the Food Request. You can add and delete items, Total Cost shows the final price of the meal and by clicking submit you can submit the food request.

Assistance Request/Cleaning Request/Security Request/Kiosk Issue

Given that you have completed the steps above, you now have fill out the *Current Time* field with the time you are making the request. You then need to fill the urgency of the request, the location where you are requesting help and if you have any comments you can write them in the assigned space above the Submit Button. Once everything is filled you can click the submit button and the request will be saved.

Transport Request

Given that you have completed the steps above, you now have fill out the *Current Time* field with the time you are making the request. You then need to fill the method of the transportation, the location where you are requesting help and if you have any comments you can write them in the assigned space above the Submit Button. Once everything is filled you can click the submit button and the request will be saved.

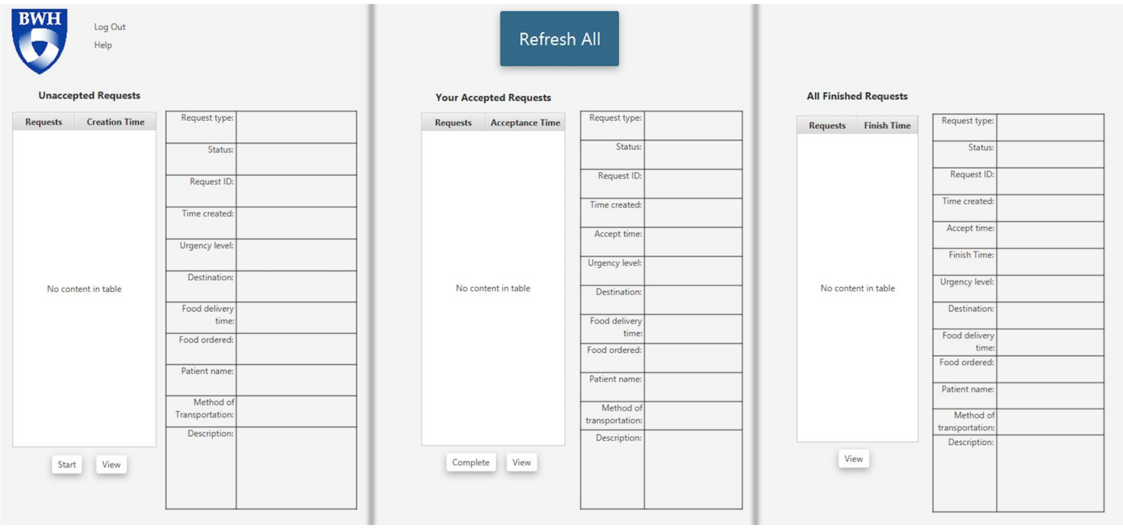
Overwatching and Editing requests:

An administrator can overview and manage all submitted requests in the current requests, as well as update and edit them. To get to that screen you have one has to follow the succeeding steps:

1 Log in as an administrator
2 Click on the Existing Service Requests button
3 Click the on the request you would like to edit on the Current Requests list
4 You can either start, view or set the request as completed

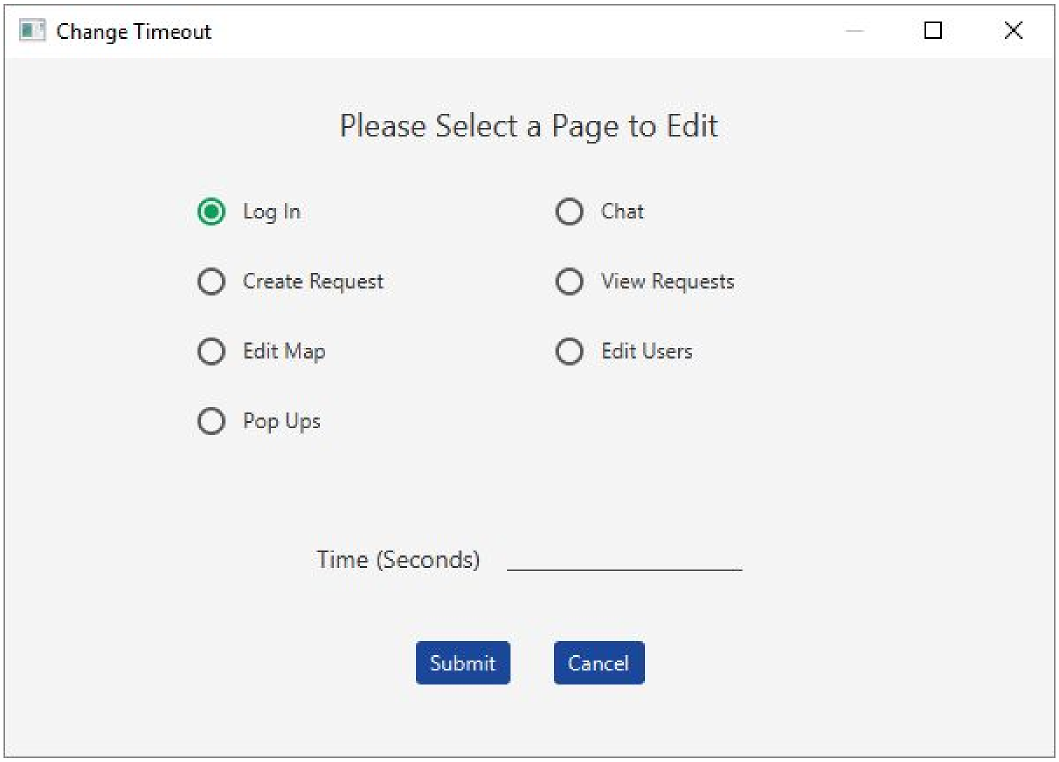
Refreshing Requests:

While on the "Existing Service Requests" menu (step-by-step details on how to get to that menu can be found above) you can click on the *Refresh* button to make sure all 3 request screens are up to date.



Timeout and how to *Change Timeout:*

Timeout is the function that allows the application to go back to the start screen once x seconds of inactivity have passed. When you first run the application the timeout is set at 30 seconds for each screen, but that can be changed by an administrator. To do that, an administrator has to go to the *Change Timeout* screen and then click on the screen whose timeout he wants to change, and the duration after which he wants the screen to timeout.



Change Timeout

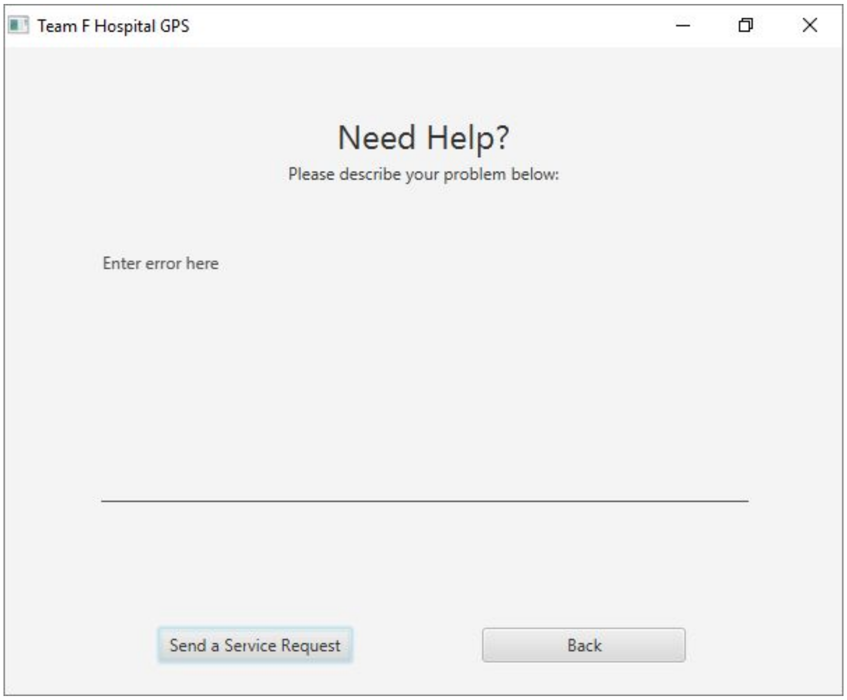
Please Select a Page to Edit

● Log In
○ Chat
○ Create Request
○ View Requests
○ Edit Map
○ Edit Users
○ Pop Ups

Time (Seconds) ________________

Application Program Interface (API):

Our in-house developed API allows direct communication between two parties (User-Admin communication). To access the API one can click on the help buttons or question mark icon which can be found in most scenes. This automatically opens a popup window (seen below) where the user can ask for help and automatically be connected to a supervisor or someone training to help him.



Team F Hospital GPS

Need Help?

Please describe your problem below:

Enter error here

Search Engine:

The search bar is most likely the single most used part of the application as it allows users to find points and areas of interest. To increase the efficiency of the system and facilitate the use of the application we have boosted our search by implementing a *clever* search engine that has two features that help make the user's experience better.

Feature 1 Autocomplete:

The autocomplete feature of the search engine actively helps the user find what he is looking for by showing him all the nodes that contain everything that has been types by the user to that point. *For example when the user types "bath", the program shows "bathroom" under search bar and by clicking on it the program fills and searches for bathroom, saving the user time.*

Feature 2 Error Fix:

We understand that using a new system for the first time can be more challenging than expected and people using a new keyboard for the first time may may encounter sizing issues which may lead to typos. To solve this issue and make sure we enhance user experience our code (given that the node searched for doesn't exist) actively searches for the node(s) that looks the most with the point of interest that the user searched and find the path between the start and the point found. *For example searching for bothroom or bethroom, will instead find the closest bathroom.*

Feature 3 Synonym Check:

In the development team we know that people may be looking for something that has been saved in the system in a different way than the one being searched for. This is why once the user hits the search button, our code will check for the closest item that has the same characteristics as the word that has been searched. *For example searching for Restrooms, will actively also search for WC and bathroom and will tell you the closest node between your start and end position.*

Visitor/User Features

As Visitor/User we define everyone who doesn't have a Login and password for the application. This could be either hospital visitors or hospital personnel. This type of user can use only parts of the application mainly focusing on the pathfinding feature and search engine feature. Additionally with the Visitor/User in mind a couple more features were added to increase the efficiency of this type of user even more. These features are:

The Emergency Button:

This button can be used in times of crisis and by clicking this button the program finds the closest to your position exit and gives you the fastest path for it.

The Clear Button:

This button clears the start and end node inputs allowing the user to start and make a new search.

The Reverse Button:

This button allows the user to find the path that gets him back where we came from.

!Disclaimer: All button described above can be found at the main pathfinding screen.

Maintenance

No active maintenance is required for our system. As long as the jar file is not tampered with and the system requirements are met the application should work without any issues. Still, if you have any issues please check the Troubleshooting section (below) for solutions.

Troubleshooting Tips

In case of code malfunction you should close the program and rerun it. This should delete all changes that have implemented by the user and return the program in its original state. If problems remain after this step, delete your .Jar file and the download and run the original .jar file. If you are still experiencing issues please contact the Fuschia Fairies Support Team at *soft_eng_f_17@wpi.edu*

Limited Warranty (1 week)

Any feature: If any feature in the application provided by the Fuschia Fairies fails due to a bug in the code within 1 week after submission, the Fuschia Faeries will fix the bug and resubmit the code free-of-charge.
!Disclaimer: The Fuschia Faeries are not liable to any problems caused by a system not covering the minimum systems requirements.