# Goal:

The goal of the program is to allow a classroom of badge-users to simulate the spread of a disease. The simulation should rely mostly on Netlogo and the badges should do very little of the heavy-lifting computation-wise.

# Specifications:

## During Runtime:

* The chance of spread of the disease should be variable and controlled by the NetLogo program
* There should be a way for the NetLogo program to “disinfect” a badge user that carries the disease, and the NetLogo view and the user’s badge should reflect this change
* When an infected badge comes into contact with a non-infected one, the infection should spread to the uninfected badge according to the user-defined chance.
* There should be an option in the NetLogo program to allow the user to define the % chance that any user is born immune to the disease, or the chance that a badge cannot be infected at all
* The users should not know who is infected and who isn’t
* The user should be able to change the number of initially infected users in NetLogo
* If a badge disconnects, simply turning it off and back on again should reconnect it to the same server, and the badge should continue interacting as if nothing happened.

## After Runtime:

* User can play back the simulation like a movie in the view
* The options to change chance of spreading, chance of immunity, and to disinfect an infected user should all be applicable to the playback. This means that one round of interactions in the classroom could be used to show a large number of possible routes the simulation could’ve gone, had the input variables been different.
* If no variables are changed, the playback feature should play back the SAME THING. The movie should only change if the input variables change

## Variables that can be changed in NetLogo:

* % chance the infection spreads
* % chance a person is “born” immune
* # of initially infected users

## Other NetLogo functionality:

* Select either a specific badge or a random badge to disinfect

# Notes:

* Every time badge interacts with another badge, it should send its module id, the module-id of the badge it infected, and the timestamp to Netlogo
  + NOTE: For the timestamp, there are essentially two options, and each has its flaws: I could use the badges for time keepers, which is fine unless one badge disconnects, or I could use NetLogo, which would be more reliable, except that Netlogo only checks for new interactions every so often, so interactions that happen back-to-back in real life would appear to happen at the exact same time in NetLogo, which isn’t necessarily a problem, except that its less accurate – USE TIMESTAMP FROM WEB SERVER
  + NetLogo will save the data from the interaction in a master array that will be ordered according to time. So, the first interaction to occur will be the first entry in the array and the last interaction will be the last entry. In addition, (THOUGH IM NOT SURE THIS IS 100% NECESSARY YET) each badge-object will keep a list of all the interactions it’s been involved in.
* When badges interact and Netlogo interprets the interaction, the infection state of the badge could be represented visually by a different color or secretly by only an object variable that no one can see except the badge-object
* Since Netlogo should be performing nearly all of the calculations, the badges main task is simply to interact with other badges and report the important interaction data to Netlogo. In Netlogo, then, there will be one main list that contains every interaction recorded by the badges, and each interaction datapoint will contain the module id’s of the two badges involved and the timestamp Netlogo adds when it receives those interactions.
  + Then, once it comes time to draw the links, whether that be live during the simulation or after the fact during analysis, a new list will be created that contains an extra piece of information about each interaction: whether or not it resulted in the spread of the disease.
* Interactions are not generally directional: that is, neither badge initiates the interaction: it is mutual. So, if in an interaction either of the two badges is infected, the non-infected one should be attempted to be converted to infected
* When watching in the view window, it’s strange when an infected badge infects a badge that already has a number of other non-infectious interactions because it looks like the disease should’ve spread down the line but didn’t. Maybe implementing a way to show how old an interaction is would be useful, or perhaps it’s good how it is now
* CURRENTLY, the playback feature doesn’t account for time, and just shows the progression of interactions one-by-one. I think adding a feature to allow you to play back the simulation this way OR over time would be beneficial
* Something else