Jan 14, 2021
Me: “Hello Xx, I was informed that you are the operations manager at our school and therefore are involved with the contact tracing process at our school.”

Client: “If there could be some sort of a website for bussing that would be great, and will save us a lot of time in contact tracing.”

Jan 15, 2021

Client: “The website/app should, in a situation where one is positive, flag the student and generate a list of the other students that have shared the bus or are siblings with the positive student. At the moment this is done manually.”

Jan 20, 2021

Me: “What are the current processes you and your team conduct for contact tracing?”

Client: “We will get an email or call from the parent asking for a lab report. The bus transport managers get the names and bus info and then check the bus attendance sheets and who shared the same seat as the infected case. Then we plot the students’ route. When did they get on the bus? First or the last person on? Where they the first drop-off? How long did they stay on the bus? We are trying to measure exposure. This is to measure the amount of exposure. We also check social distancing on the bus. We check if the windows were open and contact tracing will be conducted on the driver. We only do 1 level of contact tracing so only things like who sat next to you in order to contain it to the busses. We also check if all students wore masks on the bus. Also, of course, we ask for their temperature.”

Client: “We use tools such as the bus list that have pick-up times. Another document we use is a temperature list to check their temperature. Both of these are manual.”

Me: “Do you have experience with technology?”

Client: “We use PowerSchool and cameras which we familiarize ourselves with. We can interact and reveal specific times and locations the bus is. We use a tracking system and a live video system on each bus. In case of emergencies, we can locate the bus. We also have communication devices such as radios.”

Me: “What should this system achieve? What algorithms would enhance this website?”

Client: “If the system could display accurate data. This will make the process easier and quicker. It needs to be purely easy, it should display the number of students per bus, embarkment information per student. We want it to show us the number of students per busses.

Me: “I think you should upload this data for it to be displayed and it should assume possible red flags with a list of bus members.”

Client: “And the time they went in and out.”

Client: “If maybe the data could be linked with Powerschool but Ill share the form which includes temperatures, bus numbers, and more.”

Me: “What features to make it easier to use and more efficient?”

Client: “If maybe there could be a dashboard called ‘student information’ which shows the traveling, the distance, the capacity of the bus and time spent on the bus.”

Client: “If the data could be in an electronic format that would help with notification emails. Or even a PDF report per bus on excel.”

Client: “It would be good if we could download the information from the website to excel for printing purposes.”

May 6, 2021

Me: “I was thinking when one person has a bad temperature, they are not allowed to go on the bus, right? In this case, the only useful input would be the embankment times because the temperature should be good if you are on the campus, right? Also, I thought that the user, you, would log the temperature of each bus rider from the day of the infected case using the manual sheet, therefore I only accommodated one cell for temperature (Even this is useless as how can a 'healthy' temperature be used to check who could possibly have COVID-19). Even in terms of my IA for the IB, I will not get good marks for storing all the temperatures if I won't use them. What I plan on doing and what will be rewarded by the IB is comparing social distancing values and embankment times, which will help with generating a list of potential infected students, getting the bus driver's input for the risk rating. This will all be hopefully displayed on the website.”

Client: “Yes, you are correct, no one with a bad temperature will be allowed on campus.”

August 18, 2021

Me: “[The meeting] can be a quick check-in of what you expect of the end-product and me showing my progress, you are ultimately giving feedback on it. As such, when are you free to meet so that we can have this short check-in discussion?”

Client: “Thanks, I'll be there.”

August 19, 2021

Client: “This product does great with collecting data points to process. The buttons are simple to use. However, one thing to fix can be for the input box for who to send the email to, it should be able to take more than one at a time. Also, it would be appreciated if the bus XY Plane layout could be displayed on the website for clarity. Overall great work.”

September 2, 2021

Me: “Hello Xx, I have finished off the final product. If you want to see the distances calculated, press the calculate distances button after inputting the bus number and the infected coordinates and then go to the bus sheet to see the calculations. For the functions, you can press the button and they run. For the Risk-Rating, Sorted Times, & the Name Validation please go to their specific sheets to see the outputs as the displaying does not work at the moment.”

September 3, 2021

Me: “[All the buttons] should work now. You can just access the website and the sheet. Please only edit names, grades, and times in the sheets so the functions can properly run. Also, please note that to allow updated information to be considered in the function, please refresh the website page and on the google sheet, click off of any cells.”

September 7, 2021

Me: “Would it be okay if you could send me the evaluation through this google doc I've shared.”

Client: “

- The add button works consistently well and I liked how it was able to find gaps to place the student into.
- The displaying sorted times had a simple button and displayed the times quickly; I liked how the times were put into a sheet for backup.
- The risk rating is very easy to use and well-formulated, taking into account ventilation, use of masks, and social distancing questions.

Me: “Okay thanks for the feedback so far. Anything else?”

Client: “I am very busy at the moment so sorry about the short bullet points. Upon first notice:

- The boxes made me enter information so that it matched the information required from it.
- After each click of the buttons, the notification gave quick feedback of what data has been changed and where to find it.
- Calculating distances was done efficiently and very quickly which helps me save time with this contact tracing process.

- Risk scores were calculated efficiently and quickly and set into the matching cells. The sorting was quick and sorted the corresponding names as well, setting up the email to be clear.
- The email worked all the time. Like I said in the face-to-face discussions, I wish it was enabled to send multiple emails. But for now, it worked well.
- The name validator worked very well. I had a chance to try it but putting a number instead of the last name, and highlighted the errors consistently, which allowed me to correct them.
- For the buttons, I found that I could only run one function at a time. In some instances, I might want to run two or even more scenarios at the same time, instead of one by one."

Me: "Great, thanks for the feedback on the product, and thanks for your help along the way."

Client: "Firstly thank you so much for looping me into this project. You have taught me so much. I'm sure you've realized, I'm not that tech-savvy. Thank you for your patience and understanding, I've had a glimpse of what is possible, and how coding can solve our everyday problems. The product's scope will continue to grow, as the Covid-19 continues to be around, with the need of schools to be able to effectively and efficiently minimize and conduct contact tracing. The overall product is offering an excellent baseline from which the Tech department can work on to improve on and tweak as the requirements change."

Appendix Code

```
function doGet(e){
 return HtmlService.createTemplateFromFile('Index').evaluate();
}

function getCoordinates(xposition, yposition){
 //var url =
'https://docs.google.com/spreadsheets/d/1Gysu2BjFxd6QaVciAap_LSNlJh9XMU4yxJbw8
nnz1As/edit#gid=1795252282';
 var coordinates = SpreadsheetApp.getActive();
 var main_interface = coordinates.getSheetByName('Inputting Risked Student');
 transfer_data = main_interface.appendRow([xposition, yposition, new Date()]);
 Logger.log(xposition);
 Logger.log(yposition);
}
function getNumber(bus_number){
 var coordinates = SpreadsheetApp.getActive();
 var main_interface = coordinates.getSheetByName('Inputting Bus Number');
 transfer_data = main_interface.appendRow([bus_number, new Date()]);
 Logger.log(bus_number);
}

function getEmail(email_address){
 //var url =
'https://docs.google.com/spreadsheets/d/1Gysu2BjFxd6QaVciAap_LSNlJh9XMU4yxJbw8
nnz1As/edit#gid=1795252282';
 var email = SpreadsheetApp.getActive();
 var Main_interface = email.getSheetByName('Email Address');
 Main_interface.appendRow([email_address, new Date()]);
}
```

```
<!DOCTYPE html>
<html>
 <head>
   <base target="_top">
   <style>
.topnav {
 background-color: #4169E1;
 overflow: hidden;
}
.topnav a {
 float: left;
 color: #000000;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
 font-size: 17px;
}
.topnav a:hover {
 background-color: #ddd;
 color: black;
}
.topnav a.active {
 background-color: #e60000;
 color: white;
}
  </style>
   <script>
     function collectInput(){
       var xposition = document.getElementById("xpos").value;
       var yposition = document.getElementById("ypos").value;
       google.script.run.getCoordinates(xposition, yposition);
```

```
        document.getElementById("xpos").value = '';
        document.getElementById("ypos").value = '';
        window.alert("The infected student coordinates are (" + xposition + " ,
"+ yposition+")")
      }

      function collectInput2(){
        var bus_number = document.getElementById("bus_num").value;
        google.script.run.getNumber(bus_number);
        document.getElementById("bus_num").value = '';
        window.alert("All functions will now be performed on Bus Number " +
bus_number)
      }

      function collectEmail(){
        var email_address = document.getElementById("email").value;
        google.script.run.getEmail(email_address);
        document.getElementById("email").value = '';
        window.alert("The red-flag list in order of highest risk to lowest will
be sent to "+ email_address)
      }

      function getValidation(){
        google.script.run.withSuccessHandler(runValidation).NameValidator();
      }
      function runValidation(option1, option2){
        if (option1 == undefined){
          window.alert(option2)
        } else if (option2 == undefined){
          window.alert(option1)
        } else if (option1 == undefined && option2 == undefined){
          window.alert(option3)
        }
```

```
}

function runDistance(){
  google.script.run.CalculateDistance();
}

function getRisk(){
  google.script.run.withSuccessHandler(runRisk).RiskRatingSystem();
}

function runRisk(printed_score){
  window.alert(printed_score);
  }

function getSortedTimes(){
  google.script.run.withSuccessHandler(runSortTime).SortByTime();
}

function runSortTime(embarkment_times){
  window.alert(embarkment_times);
}

function getRedFlag(){
  google.script.run.withSuccessHandler(runRedFlag).RedFlag();
}

function runRedFlag(report){
  window.alert(report)
}

function getAdd(){
  google.script.run.withSuccessHandler(runAdd).Add();
}
```

```
    function runAdd(report){
      window.alert(report)
    }
  </script>
 </head>
 <body>
  <h1><center> Contact Tracing Website</center></h1>
  <div class="topnav">
    <a
href="https://docs.google.com/spreadsheets/d/1Gysu2BjFxd6QaVciAap_LSNlJh9XMU4y
xJbw8nnz1As/edit#gid=0" class="active" target="_blank">Bus Log</a>
    <a href="https://forms.gle/EVGn4GDsMPJ9qWiAA" target="_blank">Risk Rating
Calculator Form</a>
    <a href="https://forms.gle/y8wzBQqg6kk8HEv37" target="_blank">Add
Form</a>
  </div>
   <h2>Inputting Infected Student Case Coordinates or Bus Number for
functions</h2><br>
    <label> X-Coordinate: </label><input type="number" id="xpos"/><br><br>
    <label> Y-Coordinate: </label><input type="number" id="ypos"/><br><br>
    <button id="button" onclick="collectInput()">Submit</button><br><br>

    <label> Bus Number to perform function: </label><input type="number"
id="bus_num"/>
    <button id="button2" onclick="collectInput2()">Submit</button>

    <br><br><label> Email Address to send Red Flag Report to: </label><input
type="text" id="email"/>
    <button id="sub" onclick="collectEmail()">Submit</button><br><br>

    <h2>Buttons to Run Each Function</h2><br>
```

```
     <button id="distance" onclick="runDistance()"> Calculate Distances
</button><br><br>
     <button id="red-flag" onclick="getRedFlag()"> Run RedFlag Function
(Calculating Risk Scores, Sorting Risk Scores and Emailing ) </button><br><br>
     <button id="run_risk" onclick="getRisk()">Display Risk Rating
</button><br><br>
      <button id="add" onclick="getAdd()"> Run Add Function </button><br><br>
     <button id="cell" onclick="getValidation()"> Name Validation
</button><br><br>
     <button id="time" onclick="getSortedTimes()">Display Sorted Times
</button><br><br><br><br><br>

  </body>
</html>
```

```
function NameValidator(){
 var Bus_Log = SpreadsheetApp.getActive();
 var BusNumber_log = Bus_Log.getSheetByName('Inputting Bus Number');
 var lastRow = BusNumber_log.getLastRow();
 var bus_row = 1;
 var bus_number = BusNumber_log.getRange(lastRow, bus_row).getValue();
 var bus_number2 = "Bus " + bus_number;
 Logger.log(bus_number2)
 var sheet = Bus_Log.getSheetByName(bus_number2);
 var FirstName_row = 3;
 var FirstName_column = 2;
 var FirstName_Array = [];
 var LastName_Array = [];
```

```
 var LastName_Row = 3;
 var LastName_Column = 3;

 var count1 = 0;
 var count2 = 0;
  var sheet2 = Bus_Log.getSheetByName('Name Validation');
 var actuallastRow = sheet2.getLastRow();
 var reallastRow = actuallastRow + 1;
  var errorArr1 = [];
 for (FirstName_row = 3; FirstName_row < 24; FirstName_row++){
   var FirstName = sheet.getRange(FirstName_row, FirstName_column).getValue();
   //Logger.log(typeof FirstName)
   FirstName_Array[FirstName_row-3] = FirstName;
   Logger.log(FirstName);
   if(typeof FirstName != 'string'){
     var count1 = count1 + 1;
     Logger.log(count1)
     var errorRow1 = FirstName_row;
     errorArr1[count1 - 1] = errorRow1;
     Logger.log(errorArr1)
     option1 = "There are Error(s) in:  \n\n*row(s): " + errorArr1.join(", ")
+"\n*column: " + FirstName_column +" \n\nin the Bus Log of " + bus_number2;
     sheet2.getRange(reallastRow, 1).setValue(option1);
     sheet2.getRange(reallastRow, 10).setValue(new Date());
   }
 }
 var errorArr2 = [];
 for (LastName_Row = 3; LastName_Row < 24; LastName_Row++){
   var LastName = sheet.getRange(LastName_Row, LastName_Column).getValue();
   LastName_Array[LastName_Row-3] = LastName;
   Logger.log(LastName);
   if(typeof LastName != 'string'){
     var count2 = count2 + 1;
```

```
      Logger.log(count2)
      var errorRow2 = LastName_Row;
      errorArr2[count2 - 1] = errorRow2;
      Logger.log(errorArr2)
      option2 = "There are Error(s) in:  \n\n* row(s): " + errorArr2.join(", ")
+"\n* column: " + LastName_Column +" \n\nin the Bus Log of " + bus_number2;
      sheet2.getRange(reallastRow, 5).setValue(option2);
      sheet2.getRange(reallastRow, 10).setValue(new Date());
    }
  }

  option3 = "No Errors for the First and Last Name Cell Ranges for the bus log
of " + bus_number2+"."

  if(count1 != 0 && count2 == 0){
    return "Error, please enter a firstname with no numbers.\n" + "Please note
the following: " + option1;

  } else if (count1 == 0 && count2 != 0){
    return "Error, please enter a lastname with no numbers.\n" + "Please note
the following: " + option2;

  } else if(count1 > 0 && count2 > 0){
    return "Error, please enter a first and lastname with no numbers.\n" +
"Please note the following: " + option1+ " and,\n"+ option2;
  } else{
    return option3
  }
}

function CalculateDistance(){
  var Distance_Log = SpreadsheetApp.getActive();
  var BusNumber_log = Distance_Log.getSheetByName('Inputting Bus Number');
```

```
 var lastRow = BusNumber_log.getLastRow();
 var bus_row = 1;
 var bus_number = BusNumber_log.getRange(lastRow, bus_row).getValue();
 var bus_number2 = "Bus " + bus_number;
 Logger.log(bus_number2)
 var sheet = Distance_Log.getSheetByName(bus_number2);

 var X_Position_Row = 3;
 var Y_Position_Row = 3;
 var X_Position_Column = 6;
 var Y_Position_Column = 7;
 var X_Pos_Array = [];
 var Y_Pos_Array = [];

 for (;  X_Position_Row < 24; X_Position_Row++){
   var X_Position = sheet.getRange(X_Position_Row,
X_Position_Column).getValue();
   X_Pos_Array[X_Position_Row - 3] = X_Position;
 }
 for (;  Y_Position_Row < 24; Y_Position_Row++){
   var Y_Position = sheet.getRange(Y_Position_Row,
Y_Position_Column).getValue();
   Y_Pos_Array[Y_Position_Row - 3] = Y_Position;
 }
 Logger.log(X_Pos_Array);
 Logger.log(Y_Pos_Array);

 var X_index = X_Pos_Array.length;
 Logger.log(X_index)
 var Y_index = Y_Pos_Array.length;

 var infected_coordinates = Distance_Log.getSheetByName('Inputting Risked
Student');
```

```
 var lastRow = infected_coordinates.getLastRow();
 var x_column = 1;
 var y_column = 2;
 var Infected_Student_X_Pos = infected_coordinates.getRange(lastRow,
x_column).getValue();
 var Infected_Student_Y_Pos =  infected_coordinates.getRange(lastRow,
y_column).getValue();

 Logger.log(Infected_Student_X_Pos)
 Logger.log(Infected_Student_Y_Pos)

 var Distance_Array = [];
 for (i = 0; i < X_index; i++){
   var Raw_Distance = Math.sqrt( ((X_Pos_Array[i] -
Infected_Student_X_Pos)**2) + ((Y_Pos_Array[i] - Infected_Student_Y_Pos)**2));
   var Distance = Raw_Distance.toFixed(0);
   Distance_Array[i] = Distance;
   }
 Logger.log(Distance_Array)
 var Distance_Row = 3;
 var Distance_Column = 8;
 for(; Distance_Row < 24; Distance_Row++){
   var Calculated_Distance = sheet.getRange(Distance_Row,
Distance_Column).setValue(Distance_Array[Distance_Row-3]);
 }
}
function SortByTime(){
 var Time_Log = SpreadsheetApp.getActive();
 var BusNumber_log = Time_Log.getSheetByName('Inputting Bus Number');
 var lastRow = BusNumber_log.getLastRow();
 var bus_row = 1;
 var bus_number = BusNumber_log.getRange(lastRow, bus_row).getValue();
 var bus_number2 = "Bus " + bus_number;
```

```
 Logger.log(bus_number2)
 var sheet = Time_Log.getSheetByName(bus_number2);
 var Time_Column = 5;
 var Time_Row = 3;
 var Embarkment_Times_Array = [];
 for (; Time_Row <24; Time_Row++){
   var Time = sheet.getRange(Time_Row, Time_Column).getValue();
   Embarkment_Times_Array[Time_Row-3] = Time;
   Logger.log(Time);
 }
 var element_index = Embarkment_Times_Array.length;
 var temp = 0;
  for(var i = 0; i < element_index; i++){
   for(var j = 1; j < (element_index - i); j++){
     if(Embarkment_Times_Array[j-1] < Embarkment_Times_Array[j]){
       temp = Embarkment_Times_Array[j-1];
       Embarkment_Times_Array[j-1] = Embarkment_Times_Array[j];
       Embarkment_Times_Array[j] = temp;
       Logger.log(Embarkment_Times_Array)
     }
   }
 }
 var sort_column = 1
 var bus_column = 4
 var sheet = Time_Log.getSheetByName('Sorted Times')
 var lastRow2 = sheet.getLastRow();
 Logger.log(lastRow2);
 embarkment_times = Embarkment_Times_Array.toString();
 var sorted = sheet.getRange(lastRow2+1,
sort_column).setValue(embarkment_times);
 var bus = sheet.getRange(lastRow2+1, bus_column).setValue(bus_number2);
 Logger.log(embarkment_times);
 return embarkment_times;
```

```
}

function RedFlag(){
 var MainSheet = SpreadsheetApp.getActive();
 var BusNumber_log = MainSheet.getSheetByName('Inputting Bus Number');
 var lastRow = BusNumber_log.getLastRow();
 var bus_row = 1;
 var bus_number = BusNumber_log.getRange(lastRow, bus_row).getValue();
 var bus_number2 = "Bus " + bus_number;
 Logger.log(bus_number2)
 var sheet = MainSheet.getSheetByName(bus_number2);
 var Time_Row = 3;
 var Time_Column = 5;
 var Distance_Row = 3;
 var Distance_Column = 8;
 var Risk_Score_Column = 9;
 var First_Name_Row = 3;
 var First_Name_Column = 2;
 var Last_Name_Row = 3;
 var Last_Name_Column = 3;

/*Get Name from same row as time and divide time by distance and sort that
in*/
 var FullName_Array = [];
 var RiskScore_Array = [];
 for (First_Name_Row = 3; First_Name_Row < 24; First_Name_Row++){
   var First_Name = sheet.getRange(First_Name_Row,
First_Name_Column).getValue();
   var Last_Name = sheet.getRange(First_Name_Row,
Last_Name_Column).getValue();
   var Time = sheet.getRange(First_Name_Row, Time_Column).getValue();
   var Distance = sheet.getRange(First_Name_Row, Distance_Column).getValue();
   var Raw_Risk_Score = (Time/Distance);
```

```
   if (Time == 0 && Distance == 0){
     var Raw_Risk_Score = 0;
   }
   if (Raw_Risk_Score == 'Infinity' || Raw_Risk_Score == 'NaN' ){
     var Raw_Risk_Score = 0;
   }
   var Risk_Score = Raw_Risk_Score.toFixed(0);
   var Risk_Score_List = sheet.getRange(First_Name_Row,
Risk_Score_Column).setValue(Risk_Score);
   FullName_Array[First_Name_Row-3] = First_Name + " " + Last_Name;
   var risk_score = sheet.getRange(First_Name_Row,
Risk_Score_Column).getValue();
   RiskScore_Array[First_Name_Row-3] = risk_score;
 }


 Logger.log(FullName_Array);
 Logger.log(RiskScore_Array);
 var Student_Info_index = RiskScore_Array.length;
 Logger.log(Student_Info_index);
 var temp = 0;
 var temp1 = 0;
  for(var i=0; i < Student_Info_index; i++){
   for(var j = 1; j < (Student_Info_index -i); j++){
     if(RiskScore_Array[j-1] < RiskScore_Array[j]){
       temp = RiskScore_Array[j-1];
       RiskScore_Array[j-1] = RiskScore_Array[j];
       RiskScore_Array[j] = temp;
       temp1 = FullName_Array[j-1];
       FullName_Array[j-1] = FullName_Array[j];
       FullName_Array[j] = temp1;
   }
   Logger.log(RiskScore_Array)
```

```
   }
 }
 var Worded_FullNames = FullName_Array.join(", ")
 console.log(Worded_FullNames);
 var Worded_RiskScores = RiskScore_Array.join(", ")
 console.log(Worded_RiskScores);

 var Subject = 'Potential Infected Students';
 var processed_ouput = Worded_RiskScores + "\n\n which correspond to: \n\n" +
Worded_FullNames;
 var Message ="We would like to inform you that the contact tracing website
for bussing has detected the following red flags. Please notify them to
quarantine for 7-10 days and return a negative test before returning on
campus: " + "\n\n" + processed_ouput;
  var EmailAddress_log = MainSheet.getSheetByName('Email Address');
 var lastRow = EmailAddress_log.getLastRow();
 var email_row = 1;
 var EmailAddress = EmailAddress_log.getRange(lastRow, email_row).getValue();
 Logger.log(EmailAddress);
 var Email = GmailApp.sendEmail(EmailAddress, Subject, Message);
 Logger.log(Email);

 var report = "The Risk Scores have been calculated in the bus log for " +
bus_number2 + " and an email has been sent with the sorted risk scores and
corresponding names to " + EmailAddress +  "."
 return report;
}


function RiskRatingSystem() {
 var Risk_form = SpreadsheetApp.getActive();
 var sheet = Risk_form.getSheetByName('Risk Rating Form Response');
 var index = 2;
```

```
 var Social_Distancing_Column = 3;
 var Ventilation_Column = 4;
 var Embarkment_Time_Column = 5;
 var Mask_Wearing_Column = 6;
 var RiskScore_Column = 7;
 var lastRow = sheet.getLastRow();
 Logger.log(lastRow)
 for (; index <=lastRow; index++){
   var Social_Distancing = sheet.getRange(index,
Social_Distancing_Column).getValue();
   Logger.log(Social_Distancing);
   var Ventilation = sheet.getRange(index, Ventilation_Column).getValue();
   Logger.log(Ventilation);
   var Embarkment_Time = sheet.getRange(index,
Embarkment_Time_Column).getValue();
   Logger.log(Embarkment_Time);
   var Mask_Wearing = sheet.getRange(index, Mask_Wearing_Column).getValue();
   Logger.log(Mask_Wearing);
}
 var Rate_of_Risk = (Social_Distancing + Ventilation + Embarkment_Time +
Mask_Wearing)/4;
 Rate_of_Risk = Math.round(Rate_of_Risk);
 var printed_score = "The risk is rated a " + Rate_of_Risk + " out of 5";
 var setting_to_sheet = sheet.getRange(lastRow,
RiskScore_Column).setValue(printed_score);
 var getting_from_sheet = sheet.getRange(lastRow,
RiskScore_Column).getValue();
 return printed_score;
}


function Add() {
 var Add_Change_Delete_Log = SpreadsheetApp.getActive();
```

```
 var sheet = Add_Change_Delete_Log.getSheetByName('Add Form Responses');
 var Action_column = 3;
 var Firstname_column = 4;
 var Lastname_column = 5;
 var Grade_column = 6;
 var Add_Bus_column = 7;
 var lastRow = sheet.getLastRow();
 Logger.log(lastRow);
 var action = sheet.getRange(lastRow, Action_column).getValue();
 Logger.log(action)
 var firstname = sheet.getRange(lastRow, Firstname_column).getValue();
 Logger.log(firstname)
 var lastname = sheet.getRange(lastRow, Lastname_column).getValue();
 Logger.log(lastname)
 var grade = sheet.getRange(lastRow, Grade_column).getValue();
 Logger.log(grade);
 var add_bus = sheet.getRange(lastRow, Add_Bus_column).getValue();
 Logger.log(add_bus);

 var sheet_name = "Bus "+ add_bus;
 var bus_sheet = SpreadsheetApp.getActive().getSheetByName(sheet_name);
 var firstname_row = 3;
 var firstname_column = 2;
 var lastname_column = 3;
 var grade_column = 4;

 var firstname_row = 3
 var firstname_column2 = 2
 var name_check = bus_sheet.getRange(firstname_row,
firstname_column2).getValue();
 for(firstname_row=3; firstname_row < 24; firstname_row++){
   var name_check = bus_sheet.getRange(firstname_row,
firstname_column2).getValue();
```

```
   if(name_check == ''){
     var added_firstname = bus_sheet.getRange(firstname_row,
firstname_column).setValue(firstname);
     var added_lastname = bus_sheet.getRange(firstname_row,
lastname_column).setValue(lastname);
     var added_grade = bus_sheet.getRange(firstname_row,
grade_column).setValue(grade);
     Logger.log(firstname_row)
     break
   }

 }
  /*var firstname_values = bus_sheet.getRange("B1:B").getValues();
 var last_firstname = firstname_values.filter(String).length;
 var add_cell = last_firstname + 2;
 Logger.log(last_firstname);
 */

 var report = "Please check the bus log of "+sheet_name +" for the addition of
"+ firstname +" "+lastname+ " in grade "+grade+".\nPlease DO NOT Forget to
include their embarkment times if "+ firstname + " "+lastname+" was on the bus
with the infected student."
 return report;


 }
```