Final Report

CS Information Session Reservation

Group: Barbara O’Connor, Nicole Moore, Michael Hoff, Quentin Simoneaux
Clients: Ruth Labbe Hale and Debbie Zier
Professor: Dr. Edward A. Fox
# Table of Contents

Table of Contents .................................................................................................................. 2

Table of Figures ..................................................................................................................... 3

Executive Summary / Abstract ............................................................................................. 5

Introduction .......................................................................................................................... 6

Requirements ....................................................................................................................... 7
  Consolidating information ................................................................................................. 7
  Ease of use ......................................................................................................................... 7
  Automation and notifications ............................................................................................ 7
  Integration .......................................................................................................................... 7

Design .................................................................................................................................. 8

Implementation ..................................................................................................................... 12

Testing / Evaluation / Assessment ....................................................................................... 17

Users’ Manual ....................................................................................................................... 18

Developer’s Manual ............................................................................................................. 21

Lessons Learned .................................................................................................................. 36
  Timeline/Schedule ........................................................................................................... 36
  Problems ......................................................................................................................... 36
  Solutions ......................................................................................................................... 37
  Future Work ..................................................................................................................... 38

Acknowledgements ............................................................................................................. 39

References ............................................................................................................................ 39
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Diagram</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Sign Up Form</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Confirmation Email - Family</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Confirmation Email - CS Staff/Client</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Google Sheet with Form Responses</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>Google Calendar Event</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Reminder Email</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>No Ambassador Email</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>Standard Google Form</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>Main HTML <code>&lt;form&gt;</code> Element</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>“Name” Attributes</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>Full Custom Google Form</td>
<td>14</td>
</tr>
<tr>
<td>13</td>
<td>Google Apps Script</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>Google Sheet Database</td>
<td>19</td>
</tr>
<tr>
<td>15</td>
<td>Control-F Searching</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>Column Sorting</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>Column Sorting 2</td>
<td>21</td>
</tr>
<tr>
<td>18</td>
<td>Column Sorting 3</td>
<td>21</td>
</tr>
<tr>
<td>19</td>
<td>System Users and Goals</td>
<td>22</td>
</tr>
<tr>
<td>20</td>
<td>Session Attendee Sign Up Goal</td>
<td>22</td>
</tr>
<tr>
<td>21</td>
<td>Session Attendee Check Email Goal</td>
<td>22</td>
</tr>
<tr>
<td>22</td>
<td>CS Admin Review Calendar Goal</td>
<td>23</td>
</tr>
<tr>
<td>23</td>
<td>CS Admin Check Email Goal</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>Column Sorting 3</td>
<td>23</td>
</tr>
</tbody>
</table>
Executive Summary / Abstract

The CS Information Session Reservation project is an effort to assist recruiting processes for the Computer Science department at Virginia Tech. The department has an ambassador program that is responsible for hosting information sessions for prospective students and their families when they visit campus. The main goals of this project are to provide individuals with a means of signing up for information sessions online, and to automate behind the scenes processing for the ambassador program.

Our clients for this project are the leads of the ambassador program for the CS department, Ruth Labbe Hale and Debbie Zier. After meeting with them multiple times, we came up with a list of requirements and proposed deliverables for the project design. Our clients emphasized the desire to have the manual steps they complete on a day to day basis automated. Their current process heavily utilizes common Google Suite applications such as Gmail, Google Sheets and Google Calendar. As they don’t have a technical background, we incorporated these Google Suite applications into our design to facilitate the transition to our new, proposed system.

The proposed design that our clients would like put in place at the end of the semester includes the following pieces: an online form for prospective students and their families to use to sign up for information sessions, a shared online calendar for student ambassadors and our clients to see upcoming events, a spreadsheet containing relevant information for our clients, and various reminder emails for those signed up for events. Our design utilizes Google Suite applications as well as HTML, CSS, and JavaScript to achieve the aforementioned goals. Upon user submission of the signup form, customized confirmation emails are sent to both the user and our clients reminding them of the upcoming information session. The collected form response is used to automatically populate a spreadsheet with information about the attendee, so our clients can utilize the data if needed. The spreadsheet automatically updates the shared Admin Google Calendar for the selected information session, which is then used by student ambassadors to sign up for these events.

Our project is currently being hosted on a team member’s GitHub, but will be integrated into the CS department’s website once the development and testing stages are completed. After we have accomplished the base requirements for the project, we have plans to work on additional features that we think may be of use to our clients.
Introduction

Now, more than ever, there has been a push for students to pursue both bachelor’s and master’s degrees in the STEM industry. Computer Science is one of the most lucrative fields to go into, and Virginia Tech’s College of Engineering is working to get more students interested in its Department of Computer Science. According to VT’s College of Engineering website the Department of Computer Science is “ranked No 13 for its undergraduate program, moving the college upward in this national ranking against peers and aspirational peers… The department enrolls over 900 undergraduate majors and over 250 graduate students. During academic year 2018-19, we awarded 300 BS degrees, 52 MS degrees and 21 PhDs.” [1]

While this may seem like an impressive feat, it’s only the beginning for the legacy that the Computer Science department is trying to establish. Virginia Tech recently announced plans for a new Innovation Campus in the Northern Virginia area, which will be the hub for graduate students interested in technical degrees. According to an article on the Virginia Business website, “The state is investing nearly $1 billion in technology education, and Virginia Tech will be the largest recipient — along with bearing responsibility for producing more than 16,000 computer science degree-holders over the next two decades.” [2]

The future growth of the Computer Science department directly depends on the events led by its ambassador program. The improvements proposed by our team’s project could greatly impact the program, making it easier to host these information sessions, and in turn resulting in a greater number of students attending the university to study Computer Science.

Other departments at Virginia Tech already have online signup processes in place for their information sessions, but they’re not as helpful as they could be. For example, the ECE department’s online form for signing up for information sessions doesn’t utilize a calendar component. [3] This was one of the main asks from our clients, who would like us to provide users with a sharp looking, easy to use experience. We also included questions on the form that will provide the ambassador program with more personal information about the attendees, in hopes that they can provide them with a more personalized experience during their information sessions.

Our focus is on creating a seamless experience for all the intended users of our system. Signing up to attend information sessions should be easy for prospective students and their families, and equally as easy for student ambassadors interested in signing up to host them. Utilizing Google Suite applications allowed us to automate the entire process from start to finish, so that our clients will no longer need to manually do it. How we accomplished these tasks, how developers and users can utilize our system, and problems we faced along the way are some of the many things that will be discussed throughout the remainder of this report.
Requirements

As previously mentioned, the base requirements for this project to be deemed successful were agreed upon after meeting with our client multiple times. We all agreed that the main requirements included consolidating attendee/ambassador information, increasing ease of use of the system, automating the process with necessary notifications, and integrating the final product online. In these meetings we also discussed potential stretch goals that our team thought of, but these will be addressed in a later section of the report.

Consolidating information

Our clients expressed their interest in having all of the information recorded from our Google Form submissions on a continually updated spreadsheet. To achieve this we connected our form to a Google Sheets spreadsheet, which includes all of the information about the prospective students and their families attending an information session.

Ease of use

All of the intended users of our proposed system currently have a difficult time signing up for and hosting information sessions. The clients requested that our design be simple and straightforward, so that it will be easy as to how to complete tasks such as filling out forms and signing up for calendar events. Another factor affecting ease of use is the automation goal for the project, which will be discussed in more detail next.

Automation and notifications

While the design must be clean and simple, the biggest goal for this project is to automate the entire end to end process for our clients. Having to manually control sign ups is cumbersome and wastes time, so they requested that our system automatically do repetitive tasks for them. All of the tools used in our design are connected so that submissions to our form sets off a chain reaction of updates. Another automation request was to have email notifications sent to the prospective students and their families on completion of the form, as well as to our clients to notify them that a new family is attending an event.

Integration

The last requirement is to have our fleshed-out design fully implemented and tested, and integrated into the CS Department’s website upon completion of the other requirements. Our clients were very adamant about using this system right away and would like it prominently displayed on the home page of their website.
Design

The client had general expectations of how this project would be designed. So as to use familiar applications for the client and the potential users, we chose to utilize Google Suite tools such as Google Forms, Google Calendar, and Google Sheets as the main applications with which the user and the client would interact. The client expected a simple and clean design that would be easy for users with no technical experience to navigate and use.

*Figure 1. System Diagram*

*Figure 1* shows how the client and expected users will interact with the system. Our design will automate the process of prospective families signing up for information sessions, in turn making it easier for the CS Ambassador lead as well as CS Administrators to coordinate scheduling processes. The general flow is described below.
Prospective families use a custom form to sign up for a CS Information Session; see Figure 2. Google Forms layout is used to provide familiarity for the user. The form is designed to follow with Virginia Tech colors and styles to smooth future integration into the CS @ VT website. The form includes a calendar view for the user to choose a date in which their information session will take place. The dates that are unavailable for an information session are disabled. This allows for easier and more convenient usage, and minimizes errors and accidental signups for the client. The form includes added questions such as where the student is from, what their grade level is, and if there are any additional questions for the CS Ambassador. This helps make the CS Ambassadors better prepared for the information session. The form process automatically sends out a confirmation email (see Figure 3) to the prospective family that has just signed up for an information session. A simple design of the email is used so that it is to the point, and informative for the user. A separate confirmation email is also sent out to the CS Administrators who run the CS Ambassador program so they are aware that a new family has signed up for an information session.
The information from the form is populated into a shared Google Sheet (see Figure 5) that is accessible to the client as well as the CS Ambassadors. The client requested a way to have all information about the potential families organized in a single spot. This sheet is a way for the clients to consolidate all information that has been entered into the form. The sheet provides a column for each question from the form. Each submission of the form is recorded. This provides further automation of the previous process of the client sorting through emails to collect information about Information Sessions.
Figure 5. Google Sheet with Form Responses

Each time a user submits the custom form, a Google Calendar event (see Figure 6) is created and updated based on the information provided by the user. Google Calendar is used to keep with the same applications that the client is familiar with. This calendar is accessible to the client as well as the CS Ambassadors. The families are added as a guest to the event. Previously the family name and email were written manually in the notes section of the event. This design provides automation so that the event is automatically updated with the correct time, date, and guests. CS Ambassadors can see these events and "sign up" as the Ambassador for the information session. The event is created with the color set to red. Once an ambassador signs up for an information session the color should be changed to green to mark that no more staff is needed for that event. This provides an accurate representation of the state of the application to make things clear for the client and the CS Ambassadors.

Figure 6. Google Calendar event

Each guest signed up for each information session event is sent a reminder email about the upcoming session 24 hours before the event. This email, shown in Figure 7, provides relevant information about the upcoming session. This allows for a smoother process for the client to ensure that those signed up for the event will be fully aware of all of the details for the information session.
The client requested that they be notified if there is not a CS Ambassador signed up for the information session on any given day. So, an automated email, seen in Figure 8, will be sent to the admin email each day, notifying the client if there is no ambassador signed up on the calendar event. This email will only be auto-sent if there is no ambassador for the information session on that day. This email has the subject “IMPORTANT: Need ambassador for info session today” so as to draw more attention to the urgency of the email. There is minimal text in the message to maintain simplicity.

**Implementation**

**Information Form**

The first part of our solution starts with the Information Form. This is the HTML form that can be served on a website and then sent to guests to sign up for an information session. To create this form we first created a normal Google Form with all of the essential questions (Figure 9).
You’ll notice some extra questions: Event Name, Start Time and End Time. These will be explained in the Responses Sheet section.

Next, we wanted to give the form a different look and feel to customize it for our needs. The customization options offered by Google are minimal, so we decided we should create our own HTML page. Google Forms use a standard HTML <form> element to submit data, so all we had to do was copy the endpoints for where to send the data, and we could create our own page with custom HTML and CSS styling (Figure 10).

The only other requirement was to copy the “name” attributes from the questions on the Google Form. Upon submission, the backend expects the data to come from specific name attributes. Thus, the first question has a name attribute of entry.1238163387
With the `<form>` element and the “name” attributes, we have all the required information from the Google Form, and can create our own custom form and submit to the same endpoint. The Google Form already has some good formatting, so we copied the general layout and look of the questions. We then made some minor cosmetic edits to make the design better fit with the VT CS department. This included hiding the Session Name, Start Time, and End Time questions, as they are not meant for users to change. See Figure 12.

Lastly, we had to remove some `<script>` links that were linking to Google stylesheets (this causes an error when we publish the HTML form on our own site). We then replaced the functionality lost with our own JavaScript (Date Picker response converted to specified format, as well as Time questions). After all those edits, we were able to upload the HTML file to a GitHub repo and start serving the page. The live webpage can be found at http://csinformationsession.github.io/.
Responses Sheet

The next part of our solution is the responses sheet. This is where all the information submitted via the form is collected. To implement this, we created a Google Sheet to accompany the Google Form, and used the provided tools to have submissions automatically populate this sheet. The sheet in question can be seen in Figure 5. This allows the admins to see all of the submission data in one place.

Updating Calendar Events

Next, we wanted to implement a way to use this information to update an event on an internal Google Calendar, so the event can be shared and seen by the admins and CS Ambassadors.

We did some research and found that the best way was to write our own custom script using Google Apps Script (GAS) that is connected to the Google Sheet. The admin gmail account has pre-made calendar events created for each day that an information session is being held. The script is run each time a user submits a response to the form. This trigger is set up by the built-in triggers, made by GAS. It uses the data from the submission and the relevant data fields (Email Address, Event Date), and finds the Google Calendar event for the date that the user selected. The corresponding CalendarEvent object then updates that event by adding the user's email address as a guest. This is implemented by accessing the shared Google Calendar by the calendar id, and using the Google Calendar object to search for the events on the date that was submitted to the form. This is where the Event Name, Start Time, and End Time questions from the above Information Form became useful. It was helpful to have these fields accompany every submission to the form, so that we could later utilize them in the script. To do so, we added these questions to the original Google Form, but prefilled the responses and hid the questions from the user. From the user’s perspective the experience is as expected, but on the backend, these hidden questions in partner with the script allowed us to add some extra useful functionality. An example Google Calendar event that is updated can be seen in Figure 6.

Confirmation Emails

After updating the Google Calendar event, we wanted to figure out how to send confirmation emails both to the user who fills out the form, as well as the admin. We did some research and
found that a custom script, also using GAS, gave us the best resources and customization to complete this task. The script is run each time a user submits a response to the form. This trigger is set up by the built-in triggers, made by GAS. The script uses the email that was submitted by the prospective family, as well as other relevant data from the form, and customizes an email for them. This email is immediately sent to the prospective family from the admin email, confirming their sign up information. The script uses the MailApp object and sends the confirmation email from the admin email with a customized subject and body. The body of the email is created by updating an HTML file that is also linked to the GAS in the Google Sheet. This email can be seen in Figure 3. This script uses the same user information and sends an additional email to the admin Gmail notifying them that a family has signed up for an information session. While it utilizes the same user information, this email uses a separate HTML file to update the body of the email. This email can be seen in Figure 4.

Reminder Emails
The reminder emails for the guests for each calendar event were created using the same GAS tools. We wrote a custom script that is linked to a time-based trigger that will run the script 24 hours before the information session is held. This script accesses the shared Google Calendar by the calendar ID, and uses the Calendar object to access the information session for the following day. The CalendarEvent object is used to access each email in the guest list for that event. A custom reminder email, seen in Figure 7, is created using a separate HTML file that accesses the date, start, and end time for the current CalendarEvent object. The MailApp object is then used to send out this reminder email with a custom subject and body to each email that is on the guest list.

Ambassador Checks
The emails sent to the client, admin Gmail, to notify the staff that no ambassador is signed up for the event for the current day, uses the same tools that come with GAS. We wrote another custom function in the script that is connected to a time-based trigger that runs each morning. This function accesses the shared Google Calendar by calendar ID, and verifies if an ambassador (who would be using the @vt.edu email address) is signed up for the CalendarEvent for the current day. If no guest with an @vt.edu email address is found, the script uses Google MailApp to send a custom email to the admin email address (seen in Figure 8). This customized email body is created using an HTML file, similar to the files used for the reminder and confirmation emails.
Testing / Evaluation / Assessment

Testing
Throughout our development process we used a variety of testing methods, although most of them were informal due to the nature of our project. Most of our testing was done via usability tests and test cases. We explored the use of the entire system from the perspective of a CS program visitor and the CS administration. First we conducted basic tests to confirm that all information that was requested on the form was captured by its submission. To do this we created test accounts to walk through form submissions as an information session attendee. This helped us discover and work out bugs in the form itself. Most of these issues occurred in form submission and in form formatting and were quickly resolved by editing our forms source code. We also used this trial to check the creation and format of the confirmation emails.

We also observed the flow of information from the administrative side of the program. Most of this testing involved configuring a variety of addons/plugins that automated the flow of information from the form. Ensuring that critical information was captured and organized on the form was essential and involved running a variety of test cases on form submissions. Once the form submissions and spreadsheet were verified the Google Calendar was automatically populated with relative ease. Not much testing needed to be done outside of testing multiple events and creating new custom events.

Evaluation/Assessment
Our evaluation and assessment portions were relatively similar and were logically grouped due to their similarity. The evaluation of our overall project came from comparing its capabilities to the specifications requested by our client. Our minimum requirements were to create a semi-autonomous sign-up system that removed the need for phone or email event reservations. Our client hoped to achieve this through the use of an online sign up form that captured basic attendance information and generated confirmation emails. After testing our system for those basic functionalities, we have concluded that our solution does support these functions. Beyond this we discussed some additional functionalities that could be implemented such as custom confirmation emails, custom automated Google Calendar events, and a customizable Google Form. All this has been implemented thus far and has been tested as described above. With the minimum requirements being met and our secondary goals initially implemented we used the performance of these characteristics as our evaluation and assessment criteria. Our client has expressed their satisfaction with the project thus far and our team has worked out all the bugs we have encountered in our system. We have used client satisfaction and system performance in trial scenarios as evaluation for this project.
Users’ Manual

Usability Overview

This Google Suite solution can be used to accomplish a variety of tasks that are critical to the Computer Science (CS) Ambassador program. The overall system is capable of completing the following tasks:

- Collect sign-up information from prospective CS information session attendees.
- Provide confirmation of sign ups to attendees and CS administrators.
- Automatically update sign-up information (which is further detailed below) in a portable spreadsheet format.
- Automatically generate custom Google Calendar events to indicate event attendance and provide attendance information.

This system is implemented with Google Suite applications and can be completely accessed through the use of a custom Google Form, Google Sheets, Google Calendar, and virtually any email application.

System Use

To demonstrate the full functionality of this system a walkthrough of its functionality will be provided below. The use of this system will be different for CS information session visitors and CS Ambassador program members. The only modules to which they both have access are the custom Google Form and associated confirmation emails.

The form shown in Figure 10 can be used to capture required and supplementary information about those who wish to attend an information session.

For attendees:
This form can be filled out to provide information about a requested visit. It allows visitors to ask preliminary questions that they want answered during their session. Some of these questions are required and are denoted by a red asterisk; those without this indication are optional. The questions found on this form are subject to change. Once the form is complete clicking the submit button at the bottom will send the provided information to the CS Ambassador team.

For CS Ambassador members:
This custom Google Form can be edited to request a variety of information from prospective visitors. This form will not be directly used by administrators, but the information that is collected will be used to populate information in the confirmation emails, information spreadsheet, and event calendar as described below.
After the Google Form has been filled out and submitted, this information is used by both the confirmation email automation and the associated Google Sheet. We will take a closer look at the confirmation emails first.

The confirmation emails, shown in Figure 3 and Figure 4, are automatically sent to both the form submitter and members of the CS Ambassador program. This email's primary use is to confirm the entered information and to serve as an official reminder/confirmation of event attendance for the visitor and for the event coordinators. The generation of this email is automatic. It is sent to the designated administration email address as well as the address entered by the visitor on the Google Form. The email contains a short greeting as well as specific information that the administrators wish to provide to the attendees. The email formatting is subject to change based on administrator decisions.

While these confirmation emails are being generated and sent, the Google Sheet which collects all form submissions is automatically updated.

This sheet (Figure 14) collects all the information that was submitted with each form. Each visitor's response is recorded in its own line, so all submissions are associated horizontally in the spreadsheet. The purpose of this document is to have a record of all CS information sessions in one place to serve as a reference for the CS ambassador program. For example, if CS ambassador members wanted to verify the demographics of a specific attendee to provide a better tailored information session they could find their entry by searching the event date in the spreadsheet. Finding specific information in the sheet can generally be done in one of two ways.

If some defining feature of a submission is known, the spreadsheet user can press Ctrl + F to allow for exact match searching. This will bring up the search bar shown in Figure 13. This would be useful for finding a specific entry once the sheet has been used to capture a large number of records by removing the need to manually search for an entry. This exact match search will highlight any information found in the document that matches the user's search.
The other useful searching mechanism is the ability to sort the columns of the spreadsheet to group entries by a specific element. For example, this can be done with the date parameter to review information of those who will attend an information session on a specific date. To do this right click on the top of the column that contains the parameter to be sorted. Choose “Sort sheet A to Z” as highlighted in Figure 15. This will alphabetically and numerically sort the entries so that specific dates or date ranges can be found easily.

The final piece of this system is the Google Calendar, which is used to track requested events. This calendar is automatically generated by the information found on the associated spreadsheet (Figure 16).
Each entry found on the spreadsheet corresponds to one submitted response from the sign-up form. Each date found on the spreadsheet generates a unique event that contains the visitors' names/emails. If there are multiple visitors for a specific date, only one event is created on the calendar with all visitor information.

Figure 18 shows an example of a unique event that is generated on the calendar. To view an event, simply click the entry on the calendar to view the visitor information. This can then be used to locate the visitor’s full form submission via the searching methods on the Google Sheet.

Developer’s Manual

Users and User Goals

Below is a table to describe the goals of our three user groups.
CS 4624 - Multimedia, Hypertext and Information Access

<table>
<thead>
<tr>
<th>Session Attendees</th>
<th>CS Ambassador Admins</th>
<th>CS Ambassadors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign up for information session</td>
<td>Review ambassador calendar sign up</td>
<td>Sign up to host information sessions</td>
</tr>
<tr>
<td>Be reminded of information session</td>
<td>Be reminded about attendees</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Be informed of prospective family information</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 19. System Users and Goals*

**Breakdown of Goals**

**Session Attendees**

Goal: Sign up for information session
- Navigate to CS @ VT website
- Click on sign up for info session link
- Fill out and submit provided Google form
- END: successfully sign up for information session

*Figure 20. Session Attendee Sign Up Goal*

Goal: Check Confirmation emails
- Log in to personal email
- Check inbox and review confirmation email
- END: Be reminded of information session sign up date/time/location

*Figure 21. Session Attendee Check Email Goal*

**CS Ambassador Admins**

Goal: Review ambassador calendar sign up
- Sign into admin account
- Open calendar
- Click events to see guest list
- END: See if an event has ambassador, and if so, who it is

![Figure 22. CS Admin Review Calendar Goal](image)

Goal: Check confirmation emails about attendees
- Sign into admin account
- Check inbox and review confirmation email
- END: Be reminded of information session sign up date/time/location

![Figure 23. CS Admin Check Email Goal](image)

Goal: Review attendee information
- Sign into admin account
- Open responses sheet
- END: Be informed about prospective family information

![Figure 24. CS Admin Review Attendees Goal](image)

CS Ambassadors

Goal: Sign up to host information sessions
- Navigate to shared Admin Calendar
- Review open CS info sessions
- Edit open event
  - Add name to guest list on event
  - Update event color
- END: Successfully signup for info session
Implementation-based Service

Session Attendees

Goal: Sign up for information session

<table>
<thead>
<tr>
<th>(Sub-)Task</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigate to CS Website</td>
<td>cs.vt.edu website</td>
</tr>
<tr>
<td>Navigate to signup for info session</td>
<td>cs.vt.edu website</td>
</tr>
<tr>
<td>Fill out &amp; submit provided Google Form</td>
<td>Custom Google Form</td>
</tr>
</tbody>
</table>

![Figure 26. Session Attendee Sign Up Service](image)

Goal: Be reminded of information session sign up date/time/location

<table>
<thead>
<tr>
<th>(Sub-)Task</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log into personal email</td>
<td>Preferred email sign-in service</td>
</tr>
<tr>
<td>Check inbox &amp; review confirmation email</td>
<td>Preferred email service</td>
</tr>
</tbody>
</table>

![Figure 27. Session Attendee Reminder Email Service](image)

CS Ambassador Admins

Goal: See if an event has an ambassador, and if so, who it is
Figure 28. CS Admin Calendar Service

Goal: Be reminded of information session sign up date/time/location

Figure 29. CS Admin Reminder Email Service

Goal: Be informed about prospective family information

Figure 30. CS Admin Review Attendees Service

CS Ambassadors
Goal: Sign up to host information session
Workflows

Session Attendees

Figure 31. CS Ambassador Sign Up Service

Figure 32. Session Attendee Workflow Diagram
Information Form

To add new features to the information form, you just need an understanding of HTML, CSS, and JavaScript. A code editor will be helpful as well; we recommend Visual Studio Code. The VS Code addon, “Beautify,” is also very helpful to make the HTML more readable as you work with it. Additionally, the Google Chrome DevTools are excellent as you edit the document. You can hover over the items and the appropriate section in the HTML code will be highlighted. We have only one file, index.html, so this is the only file one needs to look at. We recommend working incrementally, changing one thing at a time and loading the HTML file in a browser to
see the effects. Finally, after being satisfied with changes, the HTML file needs to be uploaded to the server of the webpage. Currently it is a GitHub page, but the file can be served anywhere.

Responses Sheet and Script

To make changes to the responses sheet, sign into the admin account and access the Responses sheet. Minor edits can be made, such as changing the order of the fields. However, the sheet is already linked to the form, and will automatically populate new information. The main item to edit here is the Google Apps Script (GAS) for this Google Sheets. When logged in and viewing the sheet, the script can be found by clicking on "Tools" in the menu bar and then selecting "<> Script Editor" from the corresponding dropdown menu. The script we wrote to perform various aspects of the automation process is found in the “Code.gs” file. This script file is automatically connected to the responses sheet that collects the form responses, so connecting these pieces need not be done manually. The shared admin Google Calendar is connected to the responses sheet by using its specific calendar ID in the various script functions. Should a different Google Calendar need to be connected to the responses sheet, the old calendar ID would need to be replaced by the new one. To find the calendar ID for the new Google Calendar, open the calendar and navigate to its settings page. In the settings page scroll down to the section titled “Integrate calendar” copy and paste the string listed under “Calendar ID” into the script code.

 Attached below is the code that makes the calendar add a guest upon form submit:

```javascript
/**
 * Updates the calendar event in the CS Information Session Calendar to add the primary contact email of the
 * family as a guest. Also sends a confirmation email to the family and the administrators.
 */
function newSignUp(e) {
    let cal, desc, i, calId, loc, row, thisEvent, title, tstart,
    tstop;
    let sheet = SpreadsheetApp.openById("1xVQG-SJ-oFjTjIZAH4u6UgFtUR4e9x726nXE9wP-g"); // id of the test sheet
    let date = e.values[6];
    if (!isNaN(new Date(date))) { // if it a date in the correct format, then update the event.
        let cal = CalendarApp.getCalendarById("869ulujrhd01burgms96gme5o0group.calendar.google.com");
        let contactEmail = e.values[1];
        let events = cal.getEventsForDay(new Date(date));
        let calEvent;
        if (events.length > 0) {
            calEvent = events[0];
            calEvent.addGuest(contactEmail); // add the family as a guest to the event
        }
        sendConfirmEmail(e, calEvent);
    }
}
```

Figure 35. New Sign Up Code

Custom Notification Emails

The confirmation emails sent after a prospective family submits an entry to the above Information Form, go to both the family email on file and the admin Gmail account. These emails are generated by using two separate HTML files in the GAS and by calling the MailApp.sendMail() function. The MailApp.sendMail() function has four parameters to fill in
before sending an email: to, subject, htmlBody, and name. Since the admin account is the one that this script file is automatically connected to, the emails are sent from it, and it isn't one of the parameters of the function. Should the subject or name of the email sender need to be changed for either email, these can be found in the sendConfirmEmail() function in the script file. Should the body of either confirmation email need to be changed, the corresponding HTML file would need to be updated, either confirmation-email-admin.html or confirmation-email-student.html. These files can be found in the left hand side of the script editor window with the Code.gs script file.

There are two different types of reminder emails in our design, the reminder to families and ambassadors 24 hours before an information session, and the reminder to the admin account if there isn’t an ambassador signed up the day of an event. The reminder email sent to the families and ambassador can be edited in the MailApp.sendMail() function call found in the remindGuests() function of the script file. This sendMail() function call sends a reminder to all emails listed as a guest on the corresponding Google Calendar event for the following day. Should a different Google Calendar need to be used to find these information session events, the calendar ID would need to be updated in this function as well. The body of this reminder email can be edited by updating the reminder-email.html file also found on the left hand side of the script editor window.

The reminder email sent to the admin account if there isn’t an ambassador signed up the day of an event can be edited in the MailApp.sendMail() function call found in the checkAmbassador() function of the script file. This function checks to make sure that there is a guest with an email ending in “@vt.edu” in the guest list for the calendar event that day. If there isn't a guest with a VT email, then there isn't an ambassador signed up for the information session, and a reminder email is sent to the admin account to notify them of this. The body of this reminder email can be edited by updating the no-ambassador-email.html file also found on the left hand side of the script editor window.

The associated code has been pasted below for reference:
function remindGuests(e) {
    let cal = CalendarApp.getCalendarById('860uluw5d9blq2lburgms96gme9o@group.calendar.google.com'); // test calendar id
    let tomorrow = new Date();
    tomorrow.setDate(new Date().getDate() + 1);

    let eventsTomorrow = cal.getEventsForDay(tomorrow);
    if (eventsTomorrow.length > 0) {
        let event = eventsTomorrow[0];
        let guests = event.getGuestList();
        let template = HtmlService
            .createTemplateFromFile('reminder-email');

        // Convert date data into the correct format
        let formattedStartTime = Utilities.formatDate(event.getStartTime(), 'America/New_York', 'h:mm a');
        let formattedEndTime = Utilities.formatDate(event.getEndTime(), 'America/New_York', 'h:mm a');
        let formattedStartTime = formattedStartTime.substring(0, 4);
        let formattedEndTime = formattedEndTime.substring(0, 4);
        let day = event.getStartTime().getDay();
        let month = event.getStartTime().getMonth();
        let formattedDate = days[day] + ' ' + months[month] + ' ' + event.getStartTime().getDate();

        template.templatetomorrow = formattedDate;
        template.startTime = formattedStartTime;
        template.endTime = formattedEndTime;
        let message = template.evaluate().getContent();
        for (let i = 0; i < guests.length; i++) {
            MailApp.sendEmail({
                to: guests[i].getEmail(),
                subject: 'Reminder: CS50 Information Session',
                htmlBody: message
            });
        }
    }
}

<!DOCTYPE html>
<html>
<head>
    <base target="_top">
</head>
<body>
<p>Hello. </p>
<p>This is a friendly reminder that you are signed up for an Information Session with CS50</p>
<p><strong>tomorrow</strong> <strong>tomorrow</strong>, <strong>tomorrow</strong> from <strong>start</strong> to <strong>end</strong> from <strong>8</strong> to <strong>9</strong> pm. </p>
<p>If you have any questions, concerns, or will no longer be able to attend, please contact us.</p>
<p>You can reach us at <strong>vcsinfo@session@gmail.com</strong> or <strong>540-231-6931</strong>.</p>
<p>We look forward to seeing you soon. </p>
<p>CS @ VT</p>
</body>
</html>
function sendConfirmEmail(e, calEvent) {
  // Convert date data into the correct format
  let formattedStartime = Utilities.formatDate(calEvent.getStartTime(), "America/New_York", "h:mm a");
  let formattedEndtime = Utilities.formatDate(calEvent.getEndTime(), "America/New_York", "h:mm a");
  formattedStartime = formattedStartime.substring(6, 4);
  formattedEndtime = formattedEndtime.substring(8, 4);
  let day = calEvent.getStartTime().getDay();
  let month = calEvent.getStartTime().getMonth();
  let formattedDate = days[day] + ', ' + months[month] + ' ' + calEvent.getStartTime().getDay();

  // send admin email
  let templateAdmin = HtmlService
    .createTemplateFromFile('confirmation-email-admin');
  templateAdmin.date = formattedDate;
  templateAdmin.startTime = formattedStartime;
  templateAdmin.endTime = formattedEndtime;
  templateAdmin.name = e.values[2];
  templateAdmin.email = e.values[1];
  templateAdmin.phoneNumber = e.values[3];
  let adminMessage = templateAdmin.evaluate().getContent();
  //TODO: update email to vtcsinfoSession@gmail.com
  MailApp.sendEmail({
    to: 'vtcsinfoSession@gmail.com',
    subject: 'New Information Session Sign-up',
    htmlBody: adminMessage,
    name: 'CS0VT Ambassador Team'
  });

  // send family email
  let templateStudent = HtmlService
    .createTemplateFromFile('confirmation-email-student');
  templateStudent.contactName = e.values[2];
  templateStudent.studentName = e.values[4];
  templateStudent.date = formattedDate;
  templateStudent.startTime = formattedStartime;
  templateStudent.endTime = formattedEndtime;
  let studentMessage = templateStudent.evaluate().getContent();
  MailApp.sendEmail({
    to: e.values[1],
    subject: 'Confirmation of CS0VT Information Session',
    htmlBody: studentMessage,
    name: 'CS0VT Ambassador Team'
  });
}

<html>
  <head>
    <base target="_top"/>
  </head>
  <body>
    <p>Mails us the contactName ?</p>
    <p>Thank you for expressing your interest in CS0VT!</p>
    <p>We look forward to seeing you on the date ? from ? to ? for your information session in McKrude Hall Rm 114.</p>
    <p>If you have any questions or concerns, feel free to reach out to us at any time.</p>
    <p>You can reach us at vtcsinfoSession@gmail.com or 646-231-6031</p>
    <p>We look forward to seeing you and the studentName ? soon.</p>
    <p>CS & VT</p>
  </body>
</html>

Figure 38. Confirmation Email Code

Figure 39. Confirmation Email (Student) HTML
Google Apps Script Triggers

The aforementioned emails (confirmation and reminder) are set up to be run based on triggers. For example, the confirmation emails are sent on a form submit trigger. This is when a user submits the Google Form. The reminder emails are set up on a time based trigger (once a day to check events that day). To edit the triggers for the project, navigate to the “My Triggers” tab within Google Apps Script. From the admin Google Form, you can get here by going to Tools->Script Editor. Then Edit->Current project's triggers.
Edit Trigger for CS Information Session

Choose which function to run

- newSignUp

Failure notification settings

- Notify me immediately

Which runs at deployment

- Head

Select event source

- From spreadsheet

Select event type

- On form submit

Figure 41. New Sign Up Trigger
Figure 42. Check Ambassador Trigger

Choose which function to run

- checkAmbassador

Failure notification settings

- Notify me immediately

Which runs at deployment

- Head

Select event source

- Time-driven

Select type of time based trigger

- Day timer

Select time of day

- 8am to 9am
Figure 43. Remind Guests Trigger

Logging & Errors

Google Apps Scripts also includes logging and error reports. These can be useful for debugging and maintenance. To access these, navigate to the App Scripts dashboard. Then click the “Executions” tab. Here, all the details of the triggers and functions executions can be seen. Errors will be explained in detail, and any logging will be accessible here.

Custom Calendar

The calendar on the form has several dates disabled based on real life availability. These dates can be easily edited. The source code has a <script> tag containing a ‘flatpickr’, which allows for the customization in enabled/disabled dates. To find this <script> easily, ctrl+f search for
'flatpickr'. This will jump to the code. The ‘disable’ section of this code allows entry of specific dates to disable, as well as create a function to create more sophisticated rules. Additionally, here is a link to the flatpickr documentation: https://flatpickr.js.org/.

Lessons Learned

Timeline/Schedule

Our team kept to a general timeline and schedule throughout the project. We met weekly or bi-weekly depending on the workload and tasks for the given week. The team made sure to keep in contact with each other during key points of the project. The team also was transparent and clear in the work that needed to be done. Below is a general timeline of the work completed as well as the future work to be done.

<table>
<thead>
<tr>
<th>Date Range</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/2-2/17</td>
<td>• Meet with client</td>
</tr>
<tr>
<td></td>
<td>• Initial Research</td>
</tr>
<tr>
<td></td>
<td>• Presentation 1</td>
</tr>
<tr>
<td>3/6-3/31</td>
<td>• Functional prototype created</td>
</tr>
<tr>
<td></td>
<td>• Majority of requirements met</td>
</tr>
<tr>
<td></td>
<td>• Testing</td>
</tr>
<tr>
<td></td>
<td>• Presentation 2</td>
</tr>
<tr>
<td>4/1-4/20</td>
<td>• Interim Report</td>
</tr>
<tr>
<td></td>
<td>• Update implementation</td>
</tr>
<tr>
<td></td>
<td>• Further Testing</td>
</tr>
<tr>
<td>4/20-Beginning of May</td>
<td>• Possibly work on stretch goals</td>
</tr>
<tr>
<td></td>
<td>• Integration onto CS@VT website</td>
</tr>
</tbody>
</table>

Problems

We ran into several problems while working on our project. Notably, we initially attempted to use third-party applications and add-ons to create our solution. We tried using Zapier, a third-party application used to automate workflows, to connect our shared Google Calendar and Google Sheet for the project. We attempted to use this tool to create a Google Calendar event from information available on the Google Sheet, but were unable to do so because of issues linking Zapier to our admin Gmail account.

Next, to solve the same automation task we attempted to use Qualtrics. This solution looked promising at first because it would allow us to use VT-standard styling similar to that of the form created by the ECE department. However, we were also unable to use this as we could not get the proper admin rights for the automation.
After these two attempts proved unsuccessful, we found a potential solution in a third-party Google Sheets add-on called Event-o-Matic. Utilizing this add-on enabled us to automatically create Google Calendar events based on new rows of information in our Google Sheet. While this seemed to solve the automation problem at hand, it wound up creating additional problems later on. Every time someone signed up for an information session for a given day, it would create new Google Calendar events, rather than updating the guest list on the primary event. Our clients asked us to prioritize updating events rather than creating new ones, which we came to realize was impossible while using Event-o-Matic.

We were also asked to add a confirmation and reminder notification system in our design, which we again tried to implement using the third-party application Zapier. While this approach seemed useful at first, we again couldn’t utilize our admin Gmail account to apply the necessary customizations.

Not having previous experience with any of the aforementioned third-party applications and add-ons led to a lot of trial and error situations for our team. Should we have had some experience with them, or more time to do initial research, we could have saved a lot of time spent solving these problems and instead spent it on stretch goals.

Solutions
To combat the above problems, we spent a lot of time researching to find other methods that work. The original solution using Event-o-Matic for Google Calendar automation was scrapped, and replaced by one that utilized Google Apps Script (GAS) instead. According to Google’s website, “Apps Script is a rapid application development platform that makes it fast and easy to create business applications that integrate with G Suite.”[4] Using this new development platform, we were able to write our own script, connecting the form responses we collected in our Google Sheets with the rest of the Google Suite applications in our design. Using various event and time triggers at our disposal in GAS, we were able to attain the level of automation that our clients desired.

Existing Google Calendar events can now be updated rather than created, which was a much cleaner solution than the one we originally implemented. When a prospective family signs up for an event, separate confirmation emails are sent to both them and the admin Gmail account. To achieve the reminder notifications desired, we used time-based triggers to go off the day before an information session, which shoots off reminder emails to the prospective families and CS ambassadors signed up. The last automation problem we solved was checking to make sure that there is an ambassador signed up the day of an information session. Our clients asked if there was a way we could notify them if an ambassador was still needed, so we added an additional time-based trigger to our script which emails them that day if no one is signed up.
Future Work

After full integration into the CS@VT website is completed, there are a few potential additions to the system that our team thinks would be useful to the ambassador program. The primary goals for this project were to add an online signup option for information sessions and automate the end to end process for our clients. Ideally, should others continue to work on our project, the overall goal would be to continue to increase automation. Our current design allows for our clients to keep track of prospective family data through the use of an automatically updating Google Sheets. After speaking with our clients we all agreed that it would also be helpful to keep track of ambassador hours and events signed up for. Creating another resource, such as an additional Google Sheets, would allow both our clients and ambassadors to keep track of how many hours ambassadors have completed. This would also make it easier for our clients to determine which ambassadors deserve CS Ambassador awards at their end of the year banquet. More research would need to be conducted regarding this, since we’re not sure if we would be able to automate the process of ambassadors signing up for calendar events and populating a spreadsheet. Following this, reminder emails could be sent out to the student CS Ambassadors who have not yet met the required number of hours for the program. There is already infrastructure in place that could be utilized to send reminder emails, once the tracking of hours portion is figured out.

Another potential future adaptation to this project would be to update the date picker on our custom Google Form. We had issues with a previous option, so we chose to add a new one which was cleaner looking and better functioning. This newly implemented date picker allowed us to grey out dates that were unavailable, so we were able to grey out weekends and days of the summer when information sessions are not offered. While this solution worked, the unavailable days array had to be manually hardcoded. In the future, we think it would be beneficial to do more research on how to dynamically disable date picker days directly based on days from the shared Google Calendar. We started to look into this towards the end of the semester but were unable to find enough research on the topic in order to implement this.

As a team, we thought that we should also address how COVID-19 could impact this project in the future. Since the campus is shut down, information sessions are being held online via Zoom meetings on Mondays, Thursdays and Fridays from 1:30-2:30pm. While the link to this Zoom meeting room is listed on the CS@VT website, prospective families are still urged to sign up for the meeting via email or phone call. We think that small adjustments to our design could be made so that our clients could start to use this automated system right away. Since the department is already doing a lot to adjust to the switch to online schooling, the integration of our system as soon as possible could relieve some of the COVID-19 related stress.
Acknowledgements

We would like to thank our client, the CS Department, and more specifically our primary contacts:

   Ruth Labbe Hale, Student Support & Enrollment Specialist, ruth2018@vt.edu
   Debbie Zier, Receptionist & Program Support Tech, debzi5@vt.edu

We would also like to acknowledge Dr. Fox for his continued support of this project.

References


