User Manual

Berri - Beta System

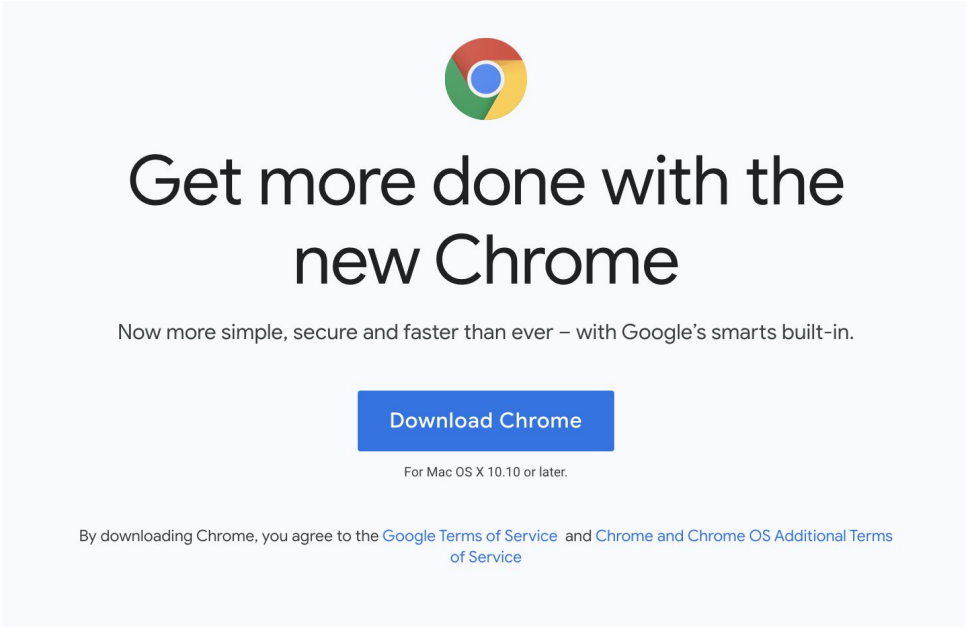
BEN WEISS
GLEN XU
ANTHONY ANASTASOPOULOS

Table of Contents

Prerequisite

In order to use our application Berry, you will need to use the Google Chrome browser since the app was developed as a Google Chrome extension.

If you do not already have Google Chrome installed, you can do so by visiting their website at https://www.google.ca/chrome/



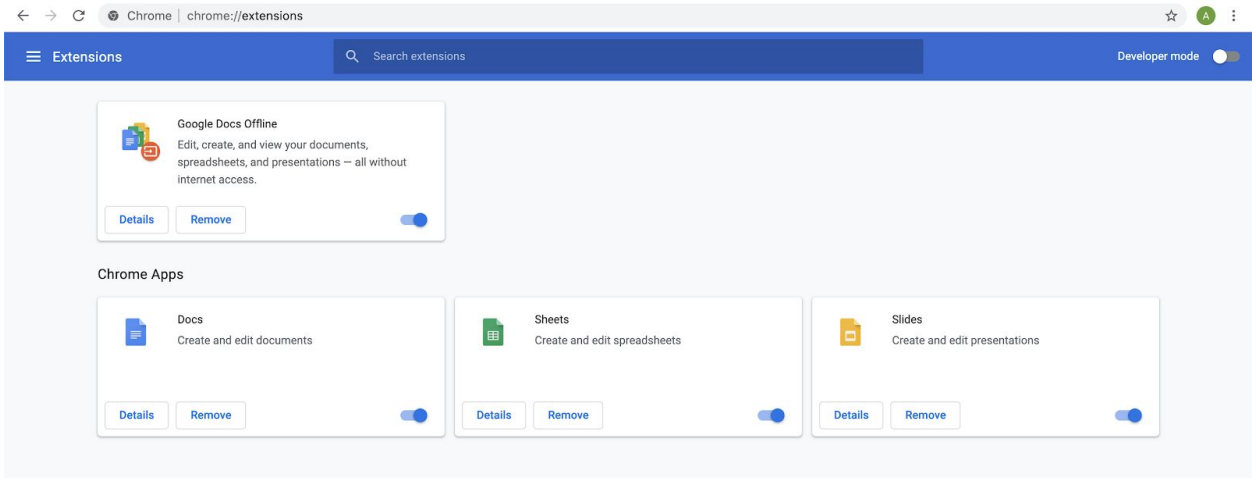
You will be met with the following prompt on the website. Clicking the "Download Chrome" button will download a .dmg or .exe file which contains the Chrome browser. Follow the necessary steps once the file is downloaded, at which point you should have the browser installed.

Also, the app was designed to work with Google Meet. This means that if you wish to use the app, it will only work with Google Meet meetings, not with Zoom. This is because it wasn't possible for our Chrome extension to obtain the chat data from Zoom.

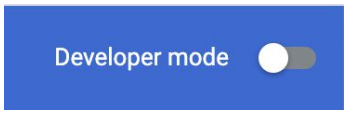
Installation

In order to install our application Berry and get it running, you must first download the application folder from our repository, the link is provided in our website. Once you have downloaded the necessary files, save them somewhere that you can access shortly.

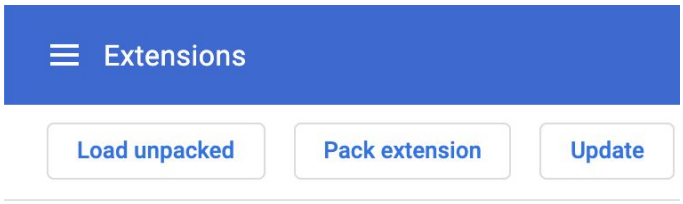
Now, open a Chrome tab, and type chrome://extensions/ into the search bar. You should arrive at a screen that looks similar to this:



From here, you need to toggle the “Developer mode” switch at the top right.



Doing so will add some new options near the top left of the screen, as shown below.



Now, click on the “Load unpacked” button, and search for the application file you downloaded from our website. Select this file, and confirm your selection. This will add our application to your list of extensions. To use the application, click the extensions icon which is right next to the Chrome search bar, and select the Berry app. The icon for the app is shown below:

User Guide

Changes from Alpha System

Several changes were made to our app based on the computer prototype version. The details of these changes as well as the reasoning behind them are detailed in the Design Evolution section of the deliverable, but a summary of these changes is as follows:

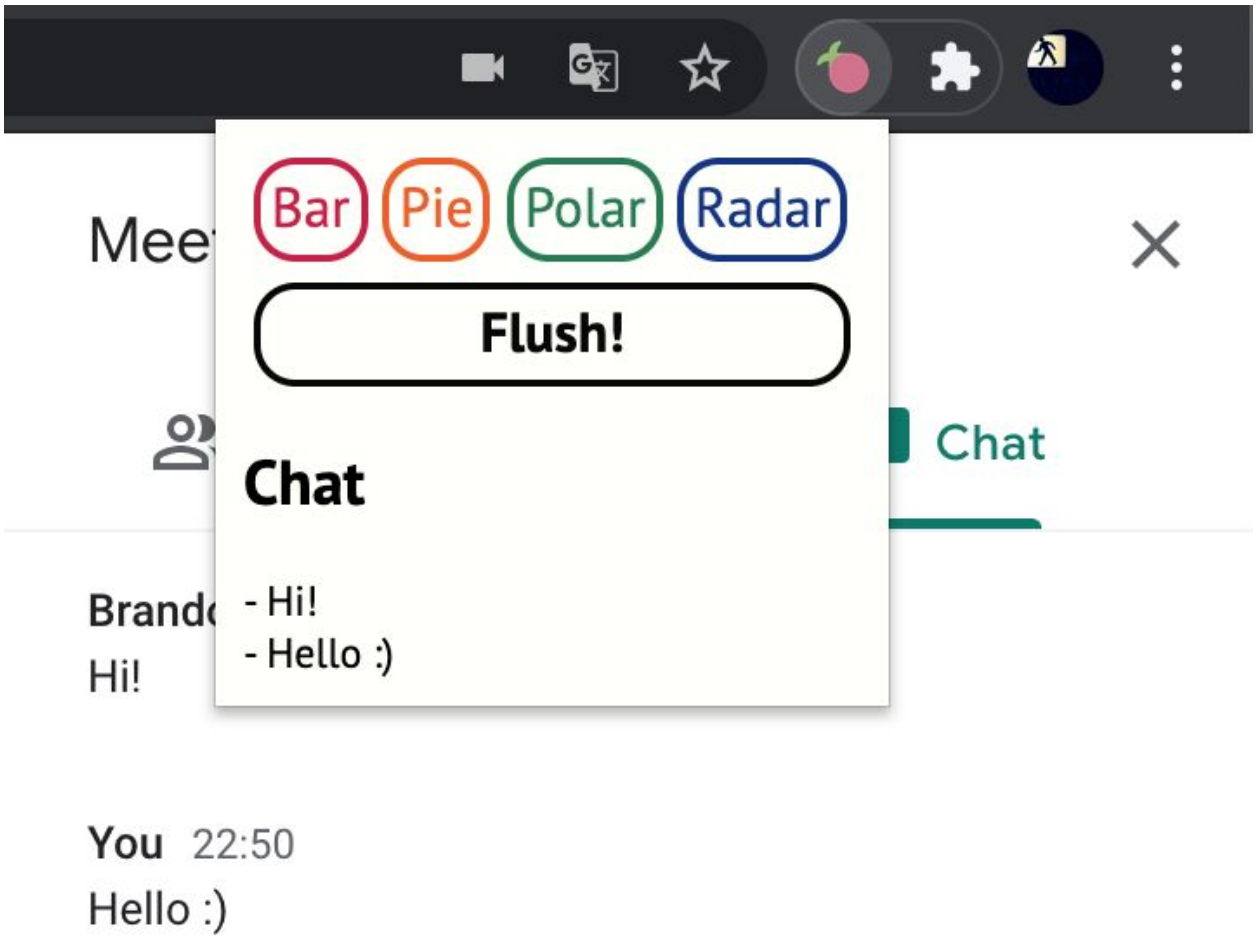
- More customization features for the graphs, as well as a new graph type were added in order to satisfy the customization and optionality usability goals respectively
- Added a new flush feature, which removes the messages from the chat history, allowing for new results to be submitted
- Fixed a bug that led to some confusion for the users, to make the system more familiar
- Added some colours and CSS designs to the user interface

Based on these changes, we arrived at the following design for our application.

Home Page

The home page of our app consists of:

- A chat history region, updating and showing the chat messages from Google Meet simultaneously.
- Four graphing buttons, each of which allows the user to present the data they obtained in a different graphical format. For this version, the options include:
    - Bar Graph
    - Pie Graph
    - Polar Graph
    - Radar Graph
- A flush button, which allows the user to clear the chat history.



CHAT HISTORY

The chat history region is used to show all the messages received by the host. In Google Meet, the host can be sharing not only their camera or the entire screen, but also a window, or another tab in Chrome. The chat history region in the extension enables the host to see the chat history even when they are not present in the Google Meet tab. For example, they can be sharing a youtube video in another tab in Chrome, while still being able to see the incoming messages.

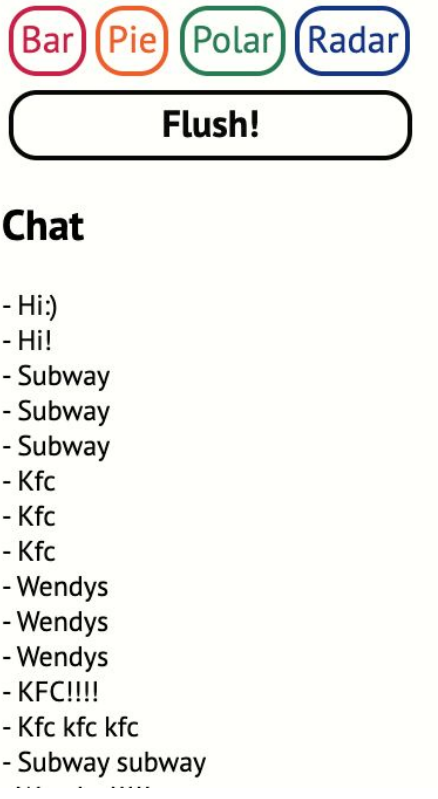
VISUALIZING ANSWERS

The host can press one of the **Bar Graph**, **Pie Graph**, **Polar Graph**, or **Radar Graph** buttons in order to visualize the answers obtained through the chat. In this current version, the answers that are visualized include any messages that were repeated at least three times in the chat. This is so that any unwanted or unrelated messages that were sent while waiting for valid answers to the question are not included in the visualisation. Note that the method for selecting which data to visualize may change in a future version of the application.

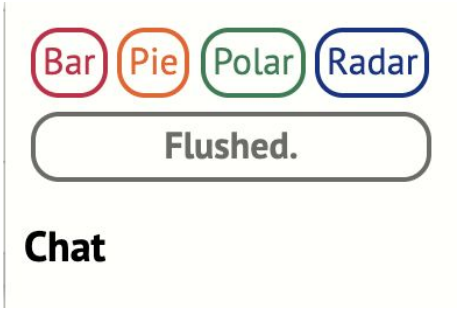
FLUSHING CHAT HISTORY

The host can press the “Flush!” button in order to clear the current chat history. This is so that they can gather a new set of answers to visualize. One limitation with the flush feature is that, in order for it to work properly, the host must keep the extension open between the time they flush and the time they visualize the new results. Whenever you close and reopen the extension, it grabs all the data from the Google Meet chat again. Therefore, if you flush the chat history and then close the extension, all the messages in the Google Meet chat will once again appear in the chat history once you reopen the extension.

For instance, given the following chat history:



The host can then click the “Flush!” button, which yields the following:



Notice how the chat history is now clear, and the “Flush!” button is replaced by a greyed out “Flushed.” button. These are both indications that the flush action was completed. The button will be enabled only when new messages come in.

Now, if the host wants to record new answers, they must keep the above window open, and as new messages are entered in the chat, they will appear in the chat history. As mentioned before, if the host closes and reopens the app, they will see all the messages that they flushed again.

The functions of the “Flush!” button are demonstrated in more details in the following video:

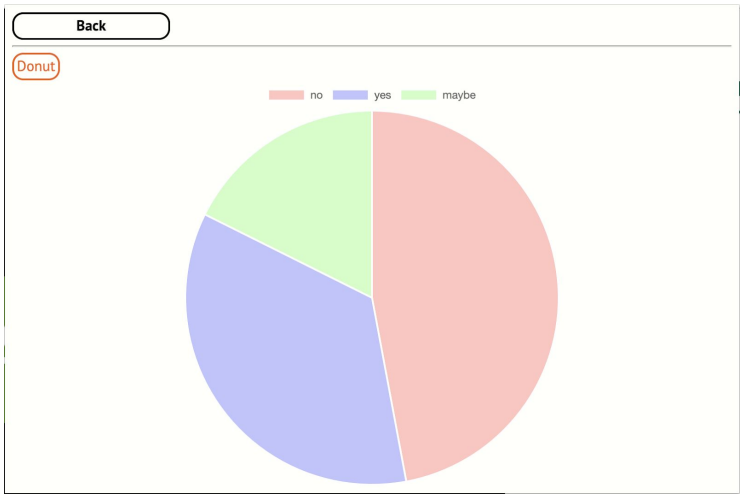
How “Flush!” Works. **<https://youtu.be/PbJdD---sb8>**

Visualization Pages

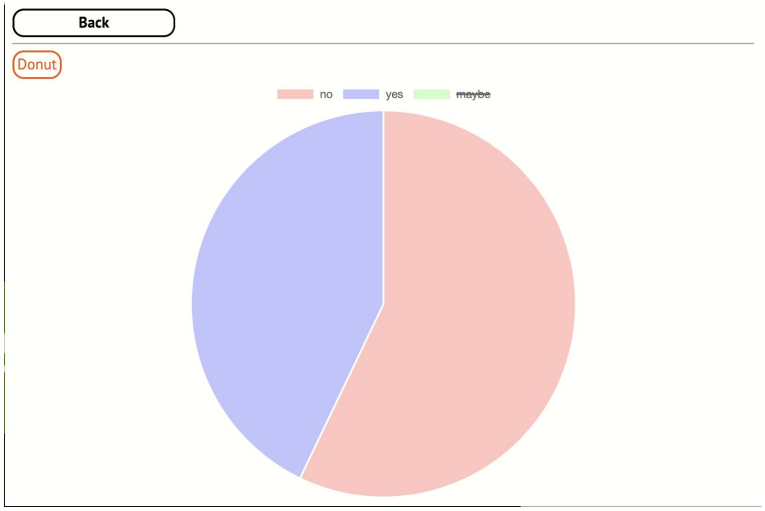
Each visualization page (i.e. the resulting page from each graphing button) consists of a **Back** button in the top left which allows the user to return to the home page, and a visual representation of the answers that were recorded.

EDITING THE DISPLAYED RESULTS

The host can edit the graph by choosing which results they would like for it to show. For instance, in the Pie Graph, we first see a visualization as such when we generate it.



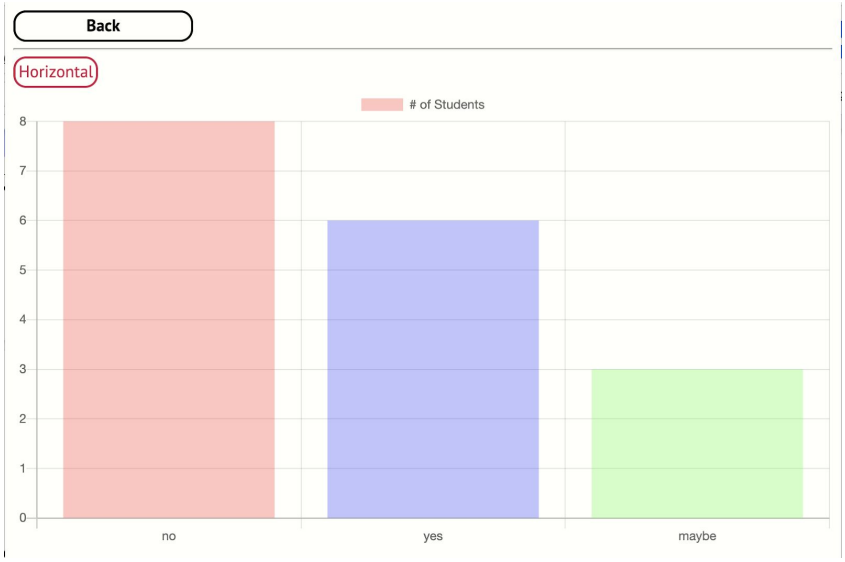
However, the host can decide that they do not want to show the “Maybe” answers. To do so, they simply click on the word “Maybe” above the Pie Graph, and this will remove it from the graph. This results in the following Pie Graph.



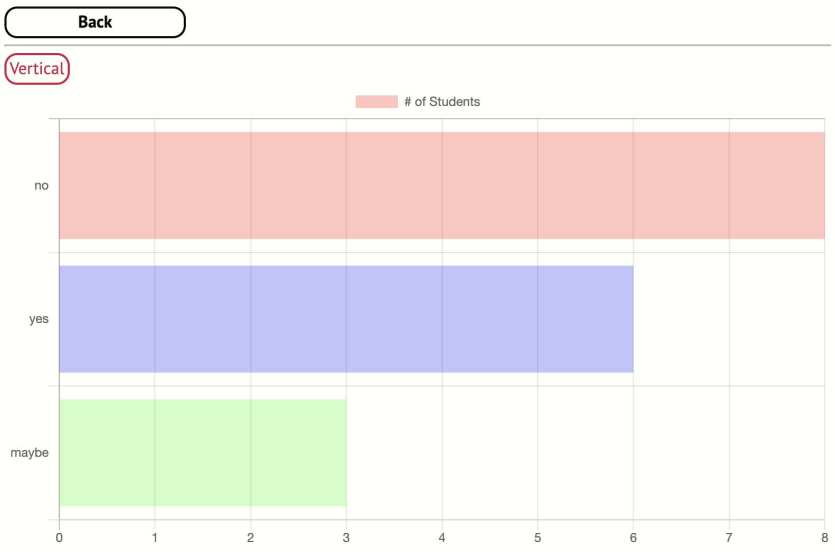
Note that this feature works in both the Pie and Polar Graphs.

OTHER CUSTOMIZATION FEATURES

There are also some other ways in which the host can edit the graphs. For the Bar Graph, the host can toggle the orientation of the bars between vertical and horizontal, by clicking on the toggle button which is right under the back button in the top left. For instance, consider the following Bar Graph:

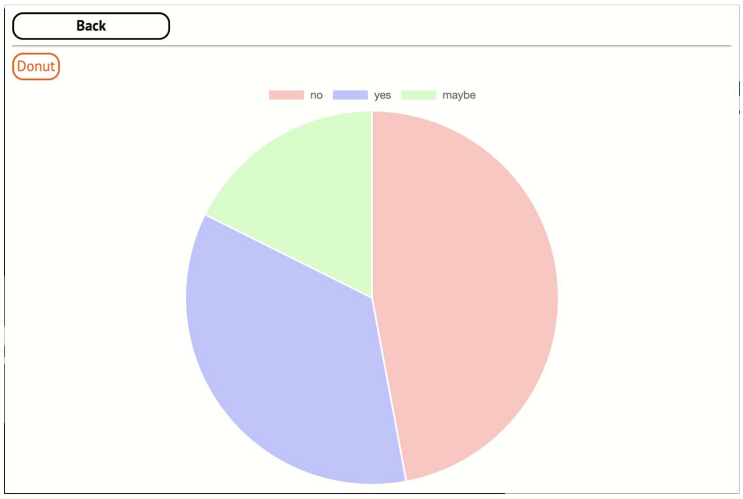


Right under the back button, we see a button that currently says “Horizontal”. Clicking this button transforms the above graph as such:

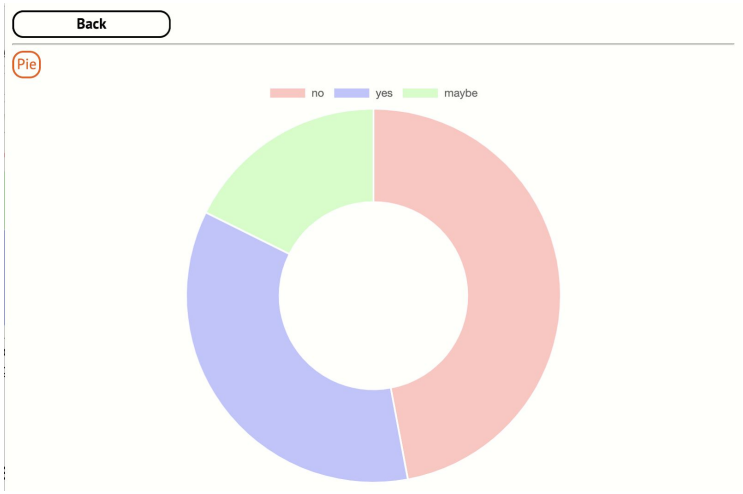


Notice how the “Horizontal” button now says “Vertical”. The host can switch between the two layouts as they please.

Similarly, for the Pie Graph, the host can toggle between a Pie and Donut Graph, by clicking on the toggle button which is right under the back button in the top left. For instance, consider the following Pie Graph:



Right under the back button, we see a button that currently says “Donut”. Clicking this button transforms the above graph as such:



Notice how the “Donut” button now says “Pie”. The host can switch between the two layouts as they please.

For both of these customization features, the change is purely from a visual standpoint, since both versions of the graph represent the exact same thing in each case.

Known Limitations and Potential Bugs

Here are some limitations that the development team is aware of during testing. If you encounter them during use, please refer to this section to resolve the issue.

Limitations of Some Charts

In our current bar chart and radar chart, the labels are not displayed as in pie chart and polar chart. This is due to the design of Chart.js. We will continue to perfect these charts, so that they

behave consistently with the pie chart, or we will replace them with more suitable chart types in the next version.

Residue of Previous Chat History

Sometimes when you start the Berri extension, you will notice that the chat history of the previous meeting is still left in the interface. This is due to the design of the chrome storage. We have eliminated all possible situations that might cause a bug to our product. If you see the residue of previous chat history when you first open the extension, just send some random messages into the chat box, or click the “Flush!” button for a few times. It will not affect your normal usage of the application.

Thank you so much for using the computer prototype of this online video conference visualization tool. We hope that this user manual addresses all possible curiosities you may have about the system usage.

In case of any subtle discrepancy or inconsistency between the final release of the beta version and this user manual, the former shall prevail.

If there is anything not working on your device or environment, please contact us, we are more than happy to look up the issue and offer a solution.

Ben Weiss <benjamin.weiss3@mail.mcgill.ca>
Glen Xu <jiayuan.xu@mail.mcgill.ca>
Anthony Anastasopoulos <anthony.anastasopoulos@mail.mcgill.ca>