**Product/Users:**

Our users are CPS parents who are logging in to access their child’s information. Our chatbot Parent Helper would interact with parents to alert them to key items - poor attendance, grades, patterns in data (missing specific class, poor grades in humanities or STEM, etc.).

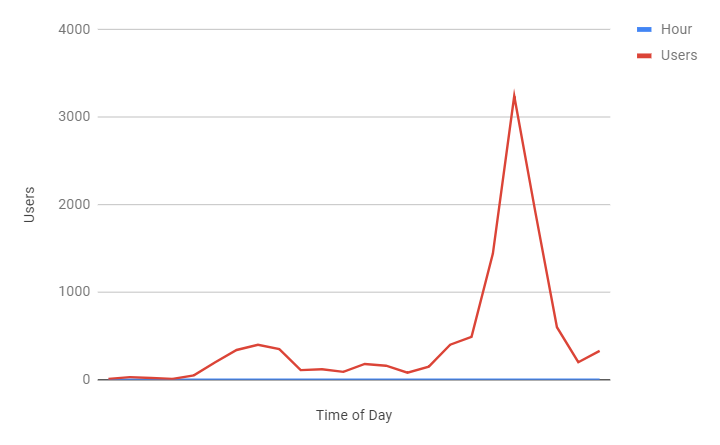
Our chatbot Parent Helper will be free. This is a service of Chicago Public Schools for parents and students. Parent Helper by design will use current CPS servers and be free to all parent users.

* **What would interface look like? Friendliness, simplicity, usability?**

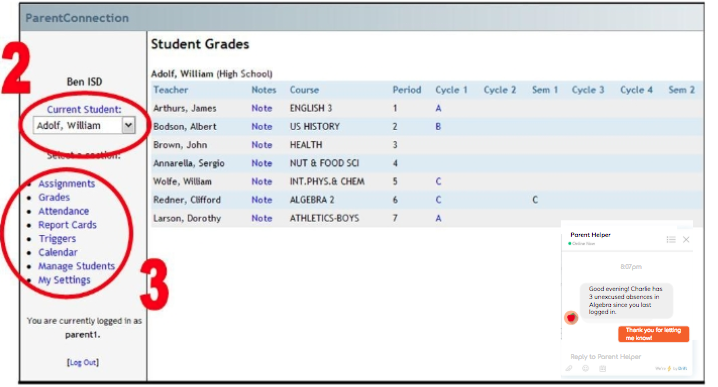
The user interface will be very simple. We will use the same login that parents already have for the Parent Portal. This will mean that parents do not have to learn any new usernames, passwords or login information. They will be able to access the Parent Helper from Parent Portal for the chatbot.

**User/Server Expectations:**

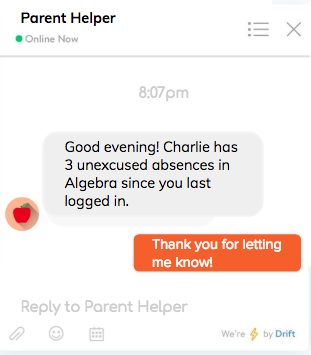
* **User breakdown**
  + There 326,978 students in CPS.
  + There are 15 networks divisions in CPS as of May 14 2018, and approximately 35 schools per network.
  + Because the network are (based on recent adjustments) of similar size and population, we can assume roughly 22,000 students per network.
  + 44,000 users (two parents per child) is the expected maximum load of any server
  + We expect no more than 10% of the maximum number of users to log in at the same time, as 50% can be eliminated as 2 parents of each child would be unlikely to use the service at the same time, only 40% of parents have ever logged in and activated their username and updated their email with the current portal service, which reduces our numbers to 8800.   
    There has never been a log of more than 4000 users at any given time.
* **Server Usage**
  + Different for grade-school and high-schools as grades are distributed on different dates. (the week before report card pickup, high school).
  + Peak hours anticipated to be after 9-5 work hours: 6:00-9:00

****

**General Mock-up**



**Detailed Mock-up**



* **How is this chatbot user friendly?**

*Well-Designed*

Our chatbot Parent Helper is similar in design to mainstream chat boxes. The chatbot is consistent with the general design of the Parent Portal website with a white background. Since the navigation menu is on the left-hand side, the chatbot will be on the right hand side, where is there is ample space and it does not overwhelm the user.

*Easy Communication*

Our chatbot Parent Helper initiates conversations with parents by providing new and key information for them. As the parent accesses the Parent Portal website to find out important information about academic achievement and attendance, the Parent Helper quickly analyzes information about a student and relays it to the parent.

* **How many users can it support?**

As many port numbers are available. We anticipate a max of 4,000 simultaneous users.

**Ideas:**

* **How do you know the chatbot works?**

We have tested connectivity by using Geni a networking testbed at the same UIC network servers to test on Geni.

* **How will you test the chatbot?**

We tested our code by setting up server/client programming to see

* **When would you test the server to accommodate for user need?**

The server will tested at key times when parents access Parent Portal. As most CPS

parents work throughout the day, traffic to Parent Portal is considerably higher during the evenings, particularly after dinner, and the weekend. For parents who are frequent visitors to the CPS Parent Portal, there is some traffic to the website during lunch.

**Process:**

* **What will be the process of creating a chatbot?**

Our first step in creating a chatbot Parent Helper is to create the code for our chatbot.Parent Helper would utilize an algorithm while scanning through student data for attendance and academics. The algorithm would prioritize information in both academics and attendance. For example, if student data for attendance and/or academics dips below a certain level (there are automatic settings from CPS, but these settings can be modified by parent/guardian). The algorithm also includes a pattern recognition component, including tardies or absences during a particular class or during a specific time period (morning and/or afternoon classes), low grades in a particular class, and academic deficiencies in general areas (literacy, STEM, art, etc.).

The second step to create Parent Helper is to employ a graphic designer and user experience designer to create our interface. To ensure that our chatbot visual design is successful, we will host a small focus group of CPS parents to ensure that any kinks in the design will be ironed out with user feedback.

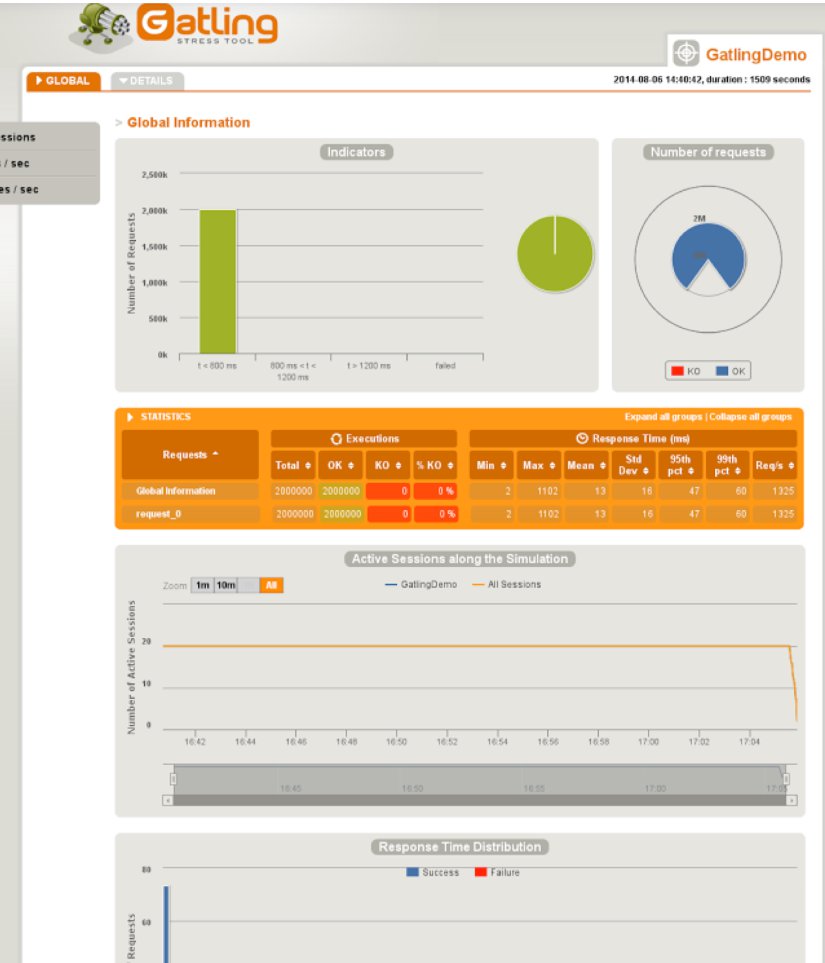
The third step is to test Parent Helper in Geni to determine whether all the parts of this chatbot work. If there are any major flaws, our team may have to resort back to the first and/or second step(s). Once these flaws have been resolved, we will move on to the pilot program to test it within a real context. The chatbot would be piloted in 5 schools, which have different demographics. They may differ by being an elementary, a high school, a school that has high poverty levels, a selective enrollment/magnet school. After parents begin using the interface with our chatbot “parent helper” we would seek information for how this has changed behaviors. We would look for patterns to see if certain populations benefitted more than others. We would then use feedback and survey data to modify the chatbot for broad implementation.

The fourth step is the broad implementation of Parent Helper.

The fifth step is to troubleshoot any bugs in our code by using Github.

* **How will it be tested?**

Our chatbot Parent Helper will be tested in Geni. We will also access the Geni service using the free load testing tool Gatling and [Loader](https://loader.io/), chosen for their clear graphical representations of the results.

[](https://www.blazemeter.com/blog/open-source-load-testing-tools-which-one-should-you-use)

* **How much bandwidth will be needed?**

Assuming an average destination distance of 10 km to our network at UIC, a transmission speed of 23 ms (from our Ping search to UIC), a packet size of 64 bytes, and a network card that transmits at 10 Gbps

Ping is the real latency delay, 23 ms.

Propagation delay, extremely small, .0001 ms.

Theoretical isn’t providing much information, so it is more important to use the real delay, which accounts for traffic on the network.

A 10 Gbps network card costs $212 for Intel x710 quad port. With higher speeds, there is more throughput for parents.

* **How many network cards will the server need to provide the necessary bandwidth for the users? Will people need larger network cards?**

Given 4,000 users, who will be limited to 100 characters per text message. There will be no images or videos transmitted.

1 byte per character of data, would have 100 bytes of data per user, this would result in 400,000 bytes for the total users. A typical 10 Gbps network card will be sufficient.

People will not need special network cards, because they will not be exchanging video and images.

* **Who will be consulted? What feedback will be sought?**

The chatbot would be piloted in 5 schools, which have different demographics. They may differ by being an elementary, a high school, a school that has high poverty levels, a selective enrollment/magnet school. This would provide feedback from a variety of stakeholders - those at different grade levels, different levels of parental involvement, families who may have a variety of challenges.

Surveys and focus groups would be used to seek feedback. An initial survey of users would gather demographic information, and behavioral information such as how often they check grades/attendance, what steps they usually take (contact teacher, conversation with child, etc.), and what they usually do when their child needs help.

After parents begin using the interface with our chatbot “parent helper” we would seek information for how this has changed behaviors. We would look for patterns to see if certain populations benefitted more than others. For example, did parents that usually felt helpless when looking at bad grades, now use the chatbot suggestions to take action?

* **If this project were expanded to the entire US to scale for more users, would you be able with the current design? How will it be redesigned?**

This would be designed specifically within a certain program, IMPACT, so it would need to be modified to work within other dashboards for other districts.

The current design has general features that would be useful for all districts, such as alerts with low grades, missed attendance days, etc. Perhaps in other regions parents may want alerts to low standardized test scores and ways for students to receive test prep support.

**Demo**

* **What does successful chatbot look like? How will you get to implementation?**

A successful chatbot would greet a parent when they log in. The chatbot would use the data for that particular student and run through the algorithms that are coded. It would identify important information to tell the parent, and allow the parent to chat back. It would then provide concrete suggestions, such as “tutoring for this class is Mondays after school,” and parents would be able to act on this information.

We would use the process described to get to implementation for a pilot, then use feedback and survey data to modify the chatbot for broad implementation.

* **What will be the location of our network? What is the rationale of choosing this network location?**

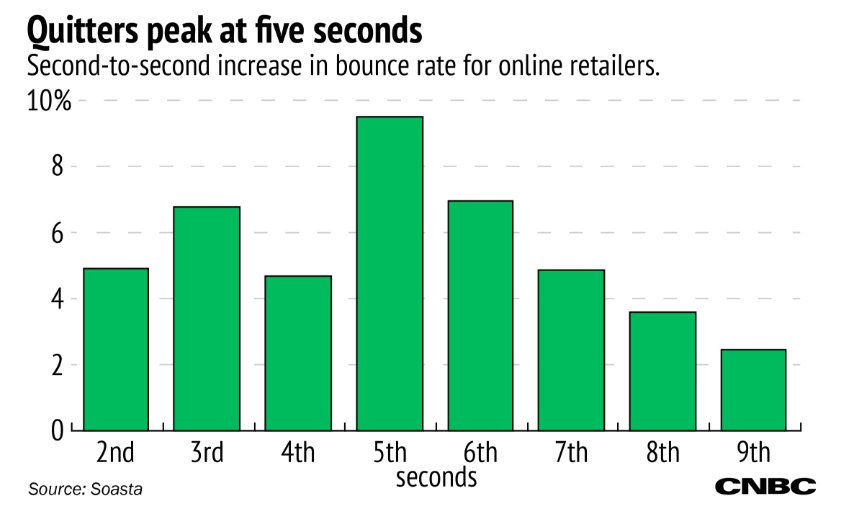
We determined that we want to keep our servers local. Our users are all City of Chicago parents. This geographic region would allow us to use University of Illinois at Chicago as a regional server. **We determined the average ping speed or the real delay/latency is 23 ms. In other words, our round trip time was 23 milliseconds.**

We have not chosen University of Chicago and Northwestern University as server locations because in our experiences, these servers did not respond on GENI.

**Reducing Delay**

* **How long would you want your user to wait?**

This is part of our product testing. We will not need “top” speed for this site. We do want to do user testing to determine how long parents will stick with the Parent Helper chat when they have some delay. We want to determine through user testing the maximum and optimal time for users. Our research shows 5 seconds, but we will verify this in our testing.

[](https://www.cnbc.com/2016/09/21/internet-retailers-and-site-performance-online-shoppers-wont-wait-5-seconds-for-pages-to-load.html)

* **Did you consider network delay when developing your project?**

We initially did not consider the network delay when developing our project. It was after we used Geni and worked in the testbed that we needed to think about the network delay. We determined that we want to keep our servers local. Our users are all City of Chicago parents. This geographic region would allow us to use University of Illinois at Chicago as a regional server. **We determined the average ping speed is 23 ms.** In addition, when trying to use University of Chicago with Geni, it would not connect.

* **How would latency delay affect your chatbot?**

This would impact our chatbot if the delay was longer than parents are willing to wait. As latency delay is the lag between inputting the information and it leaving the machine, this delay could be frustrating to parents. They may give up if they feel their messages aren’t being sent out, and if it is coupled with network delay, it could multiply to significant delays before the chatbot responds.

Our average ping speed is 23ms. A latency delay of 20 ms has a maximum throughput of 26.21 MBPS. A latency delay of 30 ms has a maximum throughput of 17.47 MBPS.

* **How will you distribute code if bug comes along?**

If we have a bug in the code, we will use Github and download webpage.

* **What does Github do?**

**GitHub** is free and open source. There is a paid option for users that want unlimited personal repositories.

**GitHub** is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, **GitHub** provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.

The **purpose of Git** is to manage a project, or a set of files, as they change over time. **Git** stores this information in a data structure called a repository. Basically, **GitHub** is like a Wiki or a place/resource where fixing bugs is open sources to developers and other users.

**Client/Server**

The server doesn’t need to know the client’s IP address, but the client needs to know the server’s IP address. The server needs to be up first, because it is waiting to receive the message from the client. All chatbots make you both a client and a server

For our network we are using TCP/IP to send packets between client and server.