Human Computer Interaction
Group Project

Chatbot for a Library
Group #15

Name: Harman Singh Chawla
Net ID: hsc367

1. Introduction

We are rapidly paddling into an era which tackles the problem of man-hours being a limited resource by introducing agents to automate support systems. However, the opportunities provided by chatbots go beyond mere customer support, opening avenues like collecting information about users, helping book study rooms, check availability of books and navigate the site. For the purposes of this project we will be talking in context to our own library at NYU Tandon School of Engineering.

Bern Dibner Library's website is a hub for resources available to all Tandon students. This intrinsically makes the sitemap complex and navigating the website a challenge for people not well-versed with the interface of the website. Of course, it is not simple to create an interactive agent that the user will really trust and gaining sufficient user engagement to make the effort lucrative and worth the time.

Through the means of this project, we would like to take away the element of humans from the chat feature on the Dibner Library website. By introducing a chatbot we wish to bring a wide array of features. The bot is essentially an assistant that communicates with us through text messages, a virtual companion that integrates into websites, applications or instant messengers and helps get closer to the users, in our case, professors and students.
A strong case can be made in favour of augmenting the existing chat feature with an automated system. It allows you get rid of the routine tasks and simultaneously processing of multiple requests from users. Besides, a tremendous speed of processing users' request, resolving queries or pointing them in the right direction with a simple understandable chatbot helps gaining students' loyalty, increasing the chances they return and use the resources available online.

We borrow concepts from artificial intelligence, specifically natural language processing, and human-computer interaction to create a smart chatbot which is intuitive and easy to use. The objective is to enhance users' interaction with the library's website and eliminate the annoying wait or limitations (such as simple FAQs) of the current system.

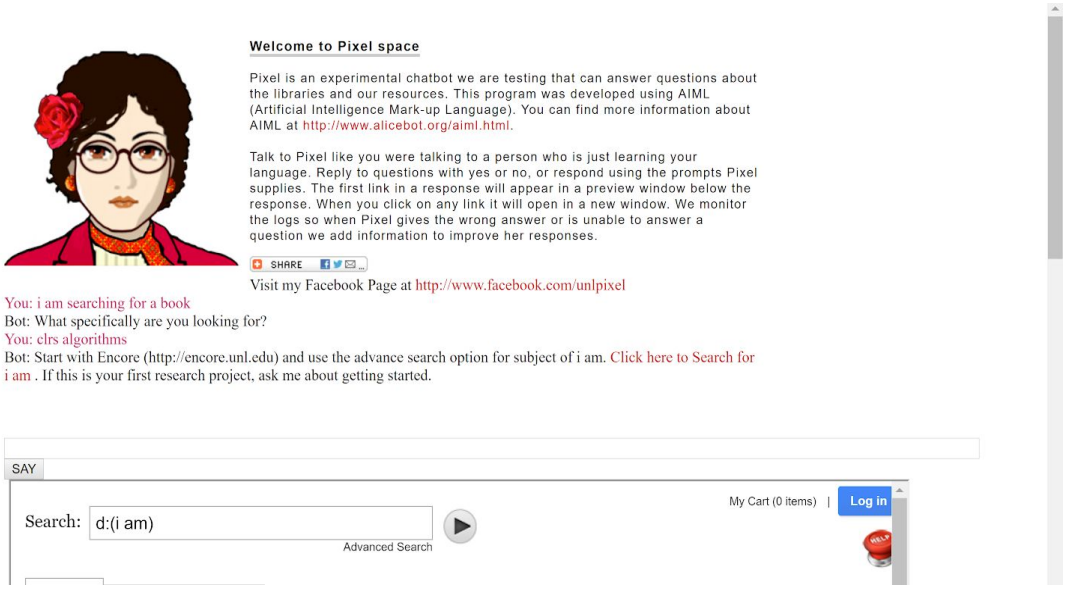
2. Background and Literature Review

There has been a shift in the way we communicate in the recent years. But the advancements in the field of communication has not directly reflected in the ease of access to answers which users expect out of a system. There is a gap which is to be bridged between a user with a question and the answers he or she is looking for.

Chatbots have been used in the recent times for increasing the ease of access to answers to frequently asked questions. Artificial intelligence was initially was used in IBM's Watson. Various advances in

artificial intelligence when combined with availability of online resources makes it time to seriously consider artificial intelligence as a tool for the library. Chatbots are computer applications which imitate human personality in an interactive manner. Chatbots respond to queries in a conversational manner and also keep a track of conversations in a way which is meaningful to humans. An interactive library experience would be more human like and lively compared to a search engine and would also fit well with students who have queries at the library.

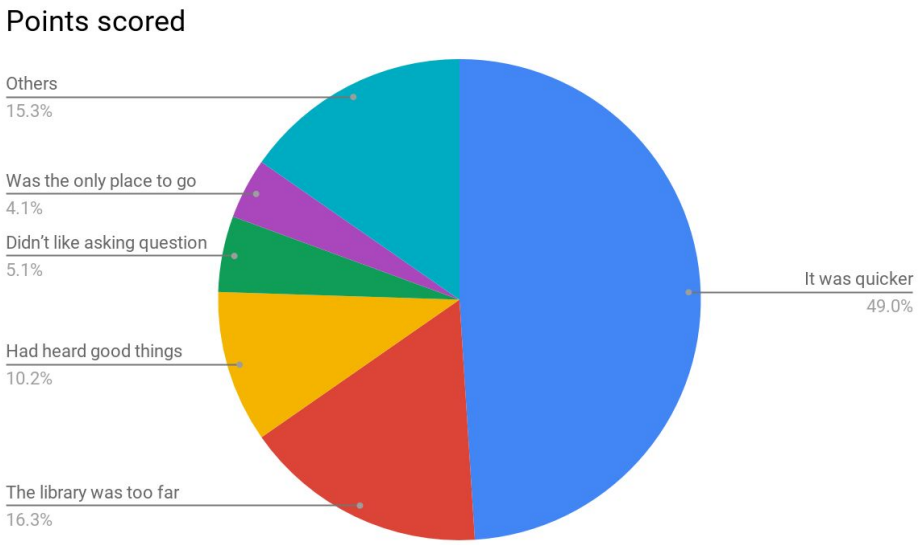
In a similar system implemented at the University of Nebraska- Lincoln, a chat bot was created so as to give answers to frequently asked questions. The project was called the “PIXEL” project which began development in the fall of 2010. PIXEL was locally hosted in an SQl database and was developed using the open source PHP interpreter designed for AIML called program O.



One of the selling points for these bots is their ability to handle common directional and predictable questions. They excel at routine, repetitive tasks that can free librarians from the most common questions. Bots flatten a website, when someone chats with a bot they don’t need to know the layout of the website, or the resources available to them. The chatbot is programmed with that information and pulls together the necessary information, formatting and presenting it in a manner that meets the needs of the information seeker. Chatbots can handle known repetitive searches by programing the searches into the bot responses. Chatbots can also handle basic multi-step searches through a more complex series of questions and answer algorithms. Exploratory searches are more challenging but a chatbot can provide general guidance and make referrals to librarians for more additional assistance. Text messaging and chat services are becoming popular reference services in libraries, and both are similar to a chatbot experience in that neither are face to face and involve a technology interface. For this reason, we may be able to apply lessons from human chat session to the design of a chatbot.

A 2010 analysis of a chat log at the University of Nebraska-Lincoln revealed some interesting patterns. This study of over 500 chat sessions (see chart 1) indicate that 35% of the searches were inquiries about database subjects or look-ups for specific titles, which approximates the monitoring category in the Nardi/O'day study. The next highest percentage, 16% involved questions about services (how do I renew a book, etc.), the next category was 15% for research questions that would equate with the exploratory and planned searches in the Nardi and O'Day article. Reference (ready reference), which Nardi and O'Day decided not to explore, was 11% of the chat questions. Ten percent of the queries were about system problems (login authentication issues, etc.). The last three categories were simple answer questions (6%), personal (4%) questions that involved non-library issues, and directional inquiries about where to find something (4%). A deeper analysis of the logs shows evidence of the personalization and refocusing behavior noted by Nardi and O'Day that was important in the reference interview. Overall, the chat questions in the UNL log appeared to be quite varied and sometimes resulted in referrals to librarians when the question appeared to be too difficult or complex. It would be
difficult with current technology to satisfy the personalization of the search process, or the collaborative refining of searches that occurs in one-on-one reference sessions with software, however through the use of a conversation agent, such as an artificial intelligent chatbot, it is possible to come close.

A 2005 study by Ward (Ward 2005) found chat services were used by graduates and undergraduates for a variety of reasons: it was quicker (48%), the library was too far away (16%), had heard good things about the service (10%), didn't like asking question in person (5%), was the only place to go (4%), and other (15%). These reasons were independent of the type of question so Ward concluded libraries planning to offer chat should expect to "field all types of questions." Whether or not we approve, clearly any chat service must be able to handle just about any type of question. Fortunately, the availability of online resources makes it possible to program a chatbot with this functionality.



3. Identifying users and defining the problem

Having a clear image of the targeted user-base is essential for a successful product. It is given that library websites can often be intimidating. The pages can have large depths, the user might lose track of his present state, the multiple search bars might be ambiguous or perhaps the user is not sure which tab in the navigation bar will take him closer to the desired goal. In such scenarios, a helping hand by someone who knows the sitemap or isn't confused by the mundane user interface might save the day.

Many libraries have realized the issue and have already deployed chats as a feature to reach out and seek help. This solution has its limitations when we consider the cost of manning the conversation on the other end.

The desire to navigate the library website in search of a specific resource is something that would be common amongst all users of the chatbot. We can't assume that they will be well-versed with navigating complex digital interfaces or have a background in computer science. For this reason, we will be keeping the interface as simple and clean as possible, using minimum number of elements and repeating them as much as possible. Furthermore, we will need to ensure that if the user's queries are not answered by the bot, he can go back and have a conversation with an actual person or leave a message for the librarian.

The chatbot will need to be gender-neutral and may assume the level of intellect expected by a person at an undergraduate level. If we take up the case of Bern Dibner Library specifically, given that it serves an engineering college, we may frame our replies expecting a certain level of analytical skills such as knowledge about hyperlinks and holding an online conversation.

4. User interface design and functionality

In this section, we intend to explain the design choices, UI elements and the experience design of our project. In the spirit of the project, we have **not** used any frameworks or libraries to build the user interface and back-end functionality of our chatbot. The entire project has been developed from scratch and is a responsive design. The red arrows presented in Figures 4(a) - 4(e) are meant to attract attention and are not a part of the user interface. The yellow box in the image below marks the extent of the webpage.

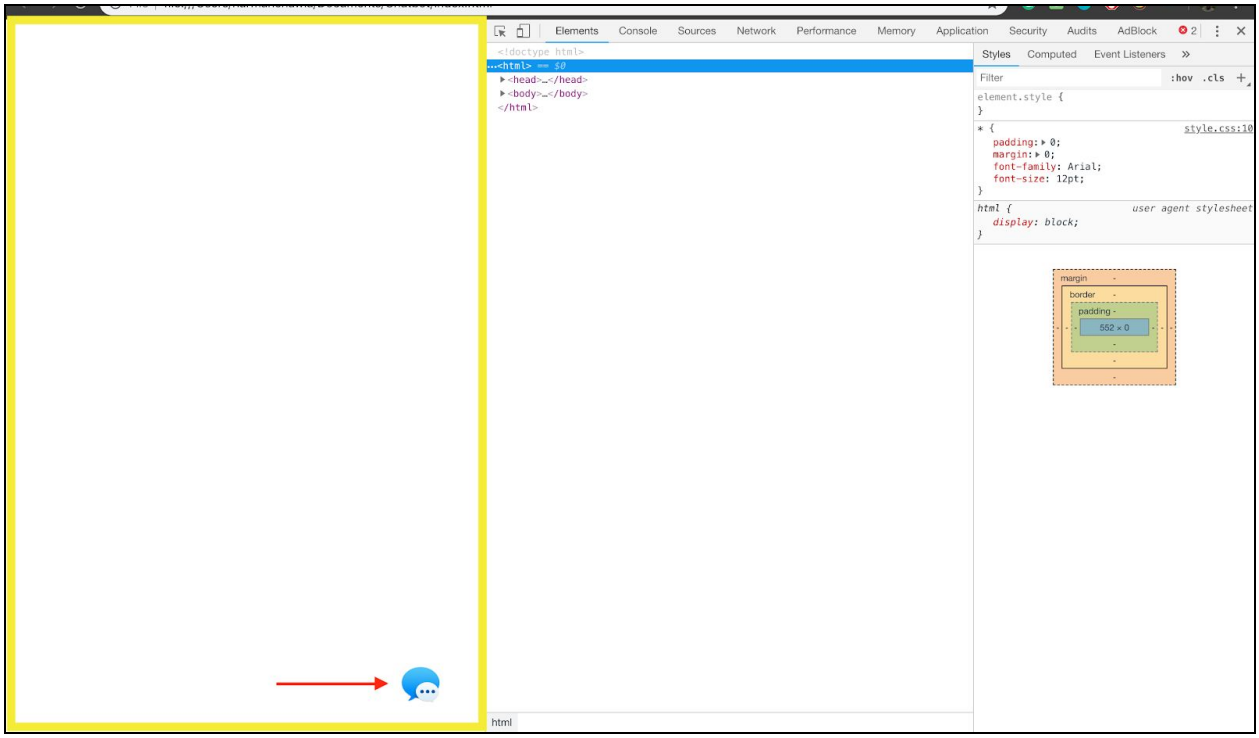


Figure 4(a). Icon marking the chat functionality.

The gateway to the chat functionality is a simple non-intrusive icon which floats at the bottom-left corner of the screen, as visible in Figure 4(a). We chose a familiar looking icon to make its purpose more evident. Since this element floats above everything on the webpage, discoverability shouldn't be a hassle.

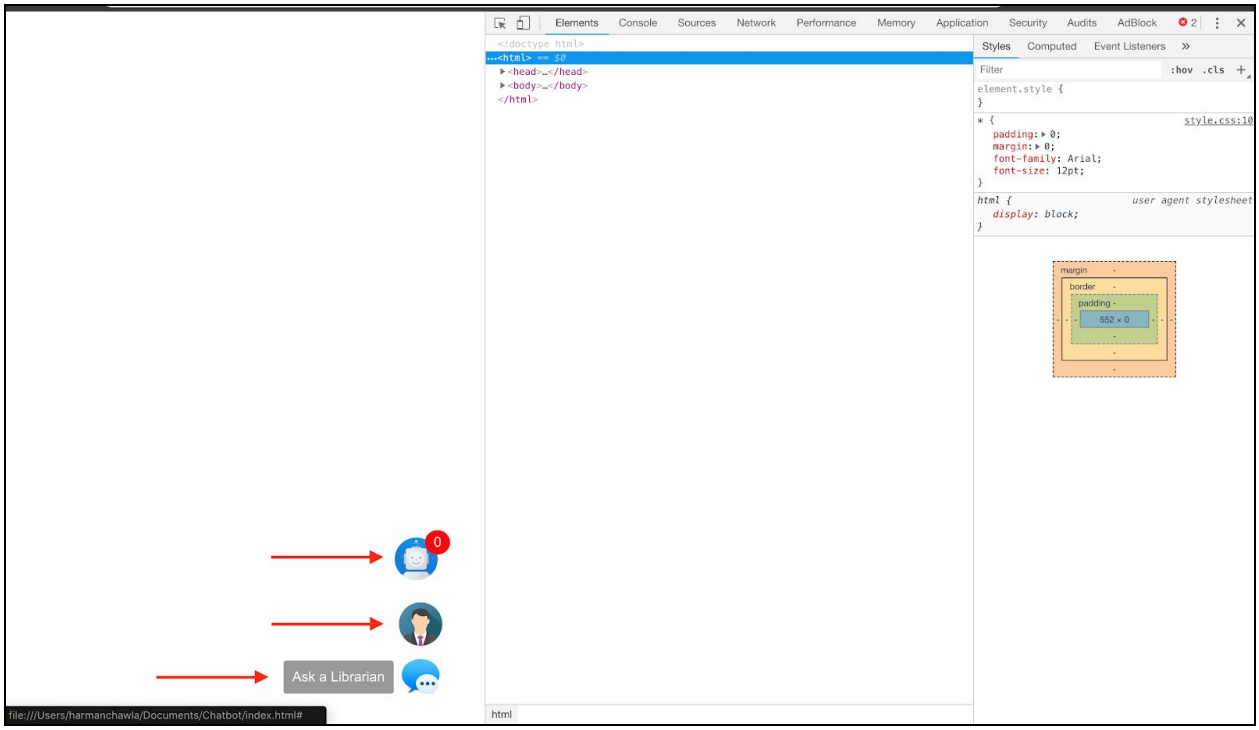


Figure 4(b). Hovering over the chat icon.

Figure 4(b) shows the next stage of the experience. Upon hovering over the icon, a text banner underscores the purpose of the icon. We keep the same name “Ask a Librarian” on purpose. Maintaining the legacy here will help returning users navigate the new UI elements and make the learning process more swift. The hover also reveals two more icons - a bot and a person. The images for these icons have been chosen to avoid any ambiguity among the two options. The purpose of the project is not to replace the actual human interaction just yet, which makes keeping the option of live chat with an actual person a necessity. The user can click on either of the two icons and the respective chat window will pop-up.

We have added an additional UI element to indicate the number of unread messages the user has from the bot. The attempt it to drive more people to use the bot instead of live chat with a person. In future, when there is chatbot has been deployed and tested, we could interchange the position of the icons to drive even more users to select the bot instead of the librarian.

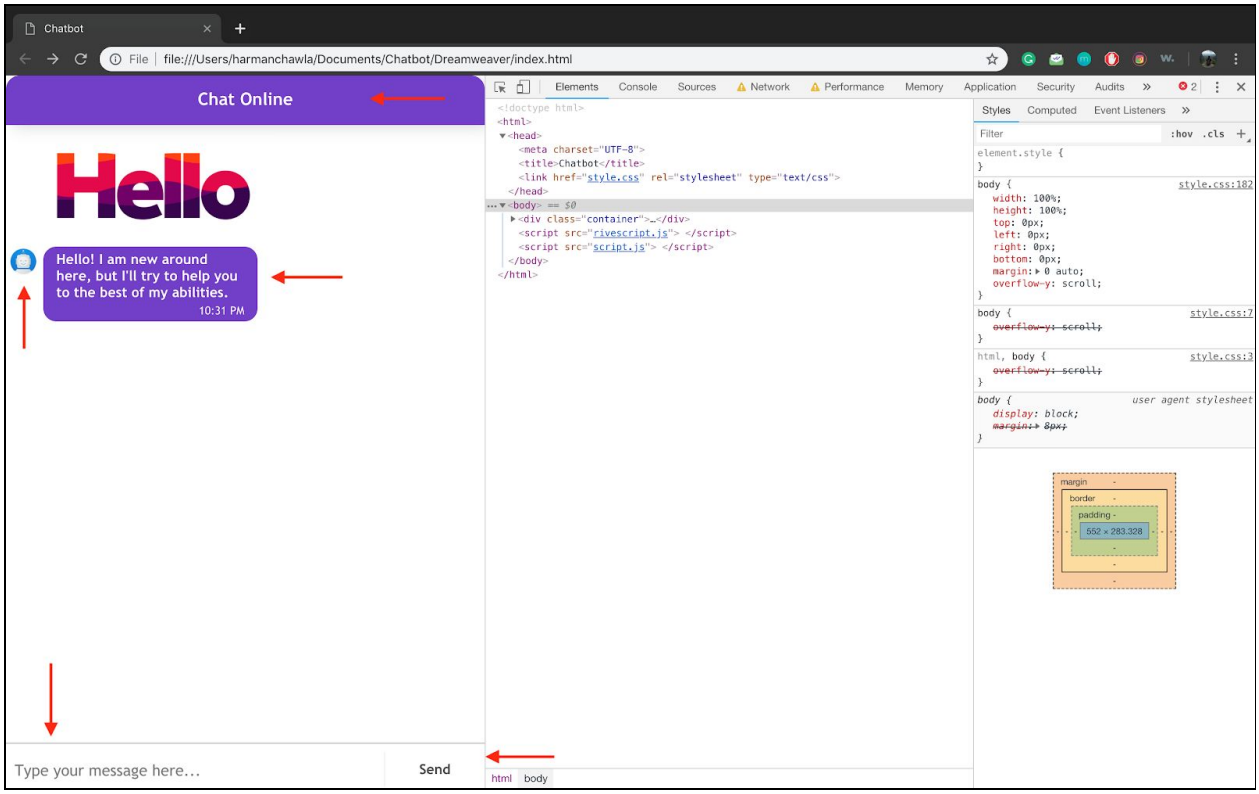


Figure 4(c). Chat window

If the user wishes to go ahead and chat with the bot, a chat window pops up. The chat window greets the users with a similar looking chat window. The bar at the top indicates the status of the chat giving the user a clear idea of his present state. At the bottom of the page, we find an input bar consisting of a text element and a "Send" button. The use of bold fonts and high contrast colors makes the text easily readable. The text bar is clearly labelled which helps the user identify its purpose. The labeling helps the user map the functionality of the element and save it as a mental model, thus accelerating the learning process.

The conversation is initiated by the bot which sends a short greeting message and a vibrant GIF image ("Hello.") The message is accompanied with an image of the sender, in this case, the bot. We use the same image for the bot which the user clicked on to open the chat image. This helps the user create mental affiliation with the image and the *bot* as it is consistent throughout the interface. The bot's message appears in white-colored font over a blue background. This contrast makes the text easily readable. Furthermore, shades of blue inspire safety and serenity, making the user calm and relaxed. A timestamp accompanying each message aids in keeping track of the conversation.

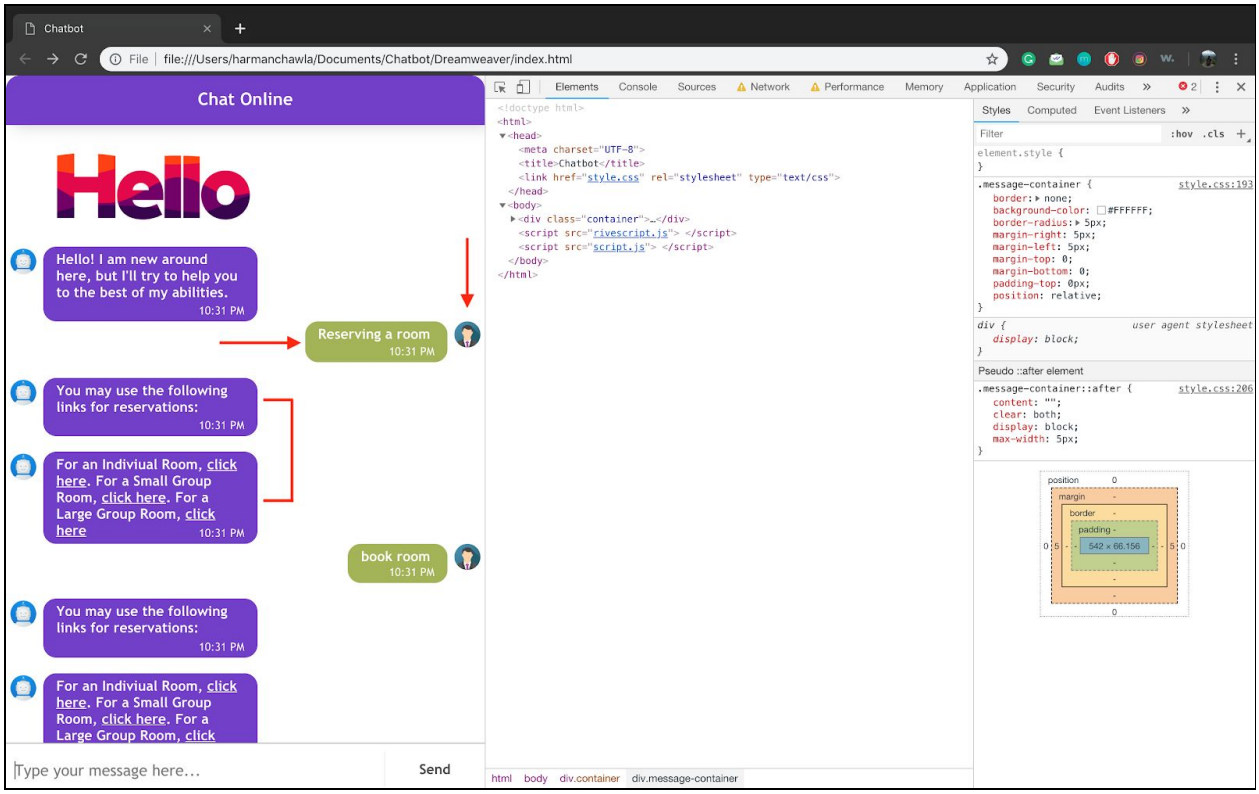


Figure 4(d) Exploring the chat experience

Once the user types in an input and hits send, the sent message gets displayed back to the user. These messages too are accompanied with an image. By default, this image the image of user saw on the last page referring to a person. If we keep adding novel elements each time user interactions with the interface, we risk burdening the short-term memory of the user as the user would have to map each element with its purpose.. Repetition, in this case, adds an element of familiarity and reduces mental load.

A good interface will fail to be useful if the system lacks the desire functionality. In figure 4(d), we see an excellent example which underscores the efficiency and low error rate of the chatbot. The bot was able to comprehend the statement “book room” and judge that the user won’t be requesting information about a “room for/of book”. So, it returns the answer to the question “how to reserve a room?” This appended functionality makes it easier for the user to interact with the user without typing complete sentences and offer just certain keywords. Many users would be familiar with this model of querying through their interactions with search engines.

The functionality extends to the user interface as well. The bot will break the reply into two or more messages as a single lengthy text may lead to a daunting or tiring experience. Note how the bot returns a hyperlink to the desired page and not the actual URL. Presenting the URL would have made the reply unnecessarily long and so it is avoided.

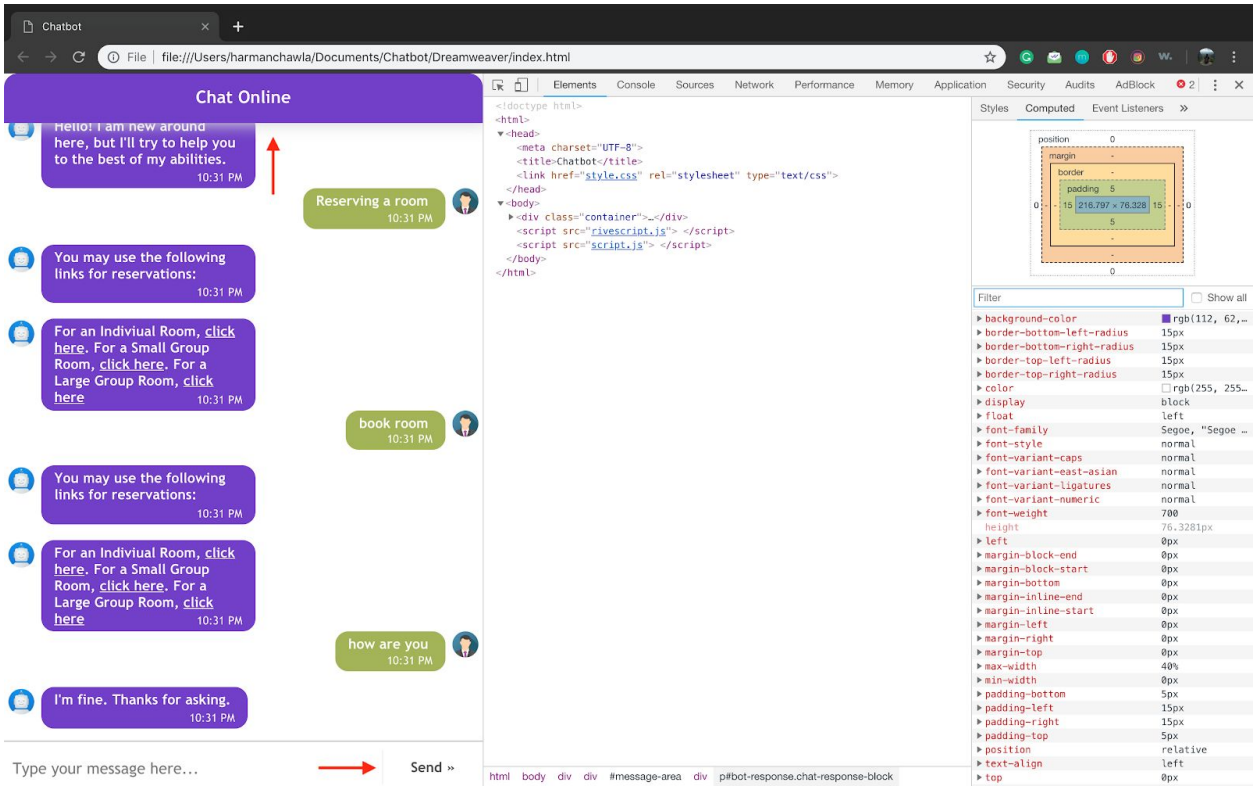


Figure 4(e) Hidden treats of the user interface

One of the advantages of having a bot at the other end of the conversation is the ability to perpetuate the conversation. The user may interact with the bot for as long as he wants. We added simple UI elements to indicate the past state of the conversation. A hint of white-colored blur on the top, suggesting the presence of messages beyond the current screen, helps avoid the mixing of blue colored top-bar with the blue texts. Coupled with the presence of a scroll bar on the the right margin of the chat helps the user navigate up and down the conversation. Another feature of the user interface is the small animation in the send button. This animation is triggered when the user hovers over the button, further indicating the purpose of the button. This can be beneficial when the user is trying to figure out the use of each element (discovery phase)

5. Brain of the chatbot

The basis of our chatbot is an algorithm built on the concept of pattern matching. RiveScript is a text-based scripting language meant to aid in the development of interactive chatbots. It derives concepts from AIML (Artificial Intelligence Markup Language) but has a simpler syntax. This can be helpful for a limited domain deployment bot like ours i.e. there are so many questions the users can answer.

RiveScript is a system that uses pattern matching to comprehend the input and generate the output. For a very high level understanding of a chatbot, if the input is “Hello” (that the user says). This goes into the chatbot’s brain. The brain does some processing and gives a reply, which we might program any way we

want. So we can make it reply “Hi” or assign multiple legit replies {“Hi”, “Hello”, “Howdy!”} and then RiveScript will choose one reply at random. Furthermore, we can assign a weight to each of these replies, a percentage, for which the program accounts.

The bot we made is made using javascript and python for accepting input and displaying output to the user. The *brain* is run and designed using RiveScript and this brain uses the technique of pattern matching we mentioned earlier. A naive methodology would be to give the brain a set of sample inputs and configure relevant outputs for each input. This basically means asking “Did the user say any of these things?” If the answer is yes (returns true), we output the configured to the user.

Here is a glimpse of how the code works:

```
! version = 2.0
    + Hello bot!  //possible input
        - Hello human! I am <bot name>  //possible output
    + * //generic input for which we don’t have a reply
        - Sorry, I don’t have a reply for that. I’ll forward your concern to an assistant
```

RiveScript allows you to make several such specific and general cases. The development of the bot is primarily dependant on the number of cases we can hard-code into the program. The bot records and logs every query for which it couldn’t return a response for and allows the administrator to add a reply to the code of the bot. This is a form of human-aided learning. The idea is to allow the growth of the chatbot as time progresses and program statements can be added to the bot to keep it relevant to the cause.

The user interface is based on a wide array of technologies such as basic HTML, CSS and Javascript, but these aren’t related to the functioning of the chatbot.

6. Conclusion

The zeal of automating a service we use on a routinely basis drove us to integrate an interactive chatbot. The goal was to enhance users’ experience with the library’s website and bridge the gap between the students’ query and the solutions. This system helps saving valuable man-hours and spark future engagement. The baseline infrastructure provided by the chatbot can be embedded into the communication channel of choice such as websites, mobile apps or as an independent tool for enhancing the user experience.

The prototype developed for this project is a customised version built to be embedded into the a library website, say Bern Dibner Library at our institute. It allows users to query FAQs and returns search results. The bot maintains a knowledge of the chat history with the user (structured data) and an access to a knowledge base (unstructured data). It also aids in guiding amatuer users and helps them navigating the complicated websites. Furthermore, the chatbot takes-off pressure from the manning the chat system and allows interaction using a simple user interface. The system helps in adding to the plethora services provided by the library without much human resource requirement.

The field of human-computer interaction borrows traits from computer science, psychology and social science. Our system has a similar alignment and borrows concepts from several fields to deliver a functional product without compromising on the aesthetics. We have tried to sketch a user interface which lives up to the five pillars of usability and achieves the said functionality within the defined constraints.