

Issues in Informing Science + Information Technology

An Official Publication of the Informing Science Institute InformingScience.org

IISIT.org

Volume 16, 2019

How to Program a Chatbot – An Introductory Project and Student Perceptions

Ju Long*	Texas State University, San Marcos, United States	julong@txstate.edu
Michael Juntao Yuan	Ringful Health, Austin, United States	michael@ringful.com
Hsun-Ming Lee	Texas State University, San Marcos, United States	<u>samlee@txstate.edu</u>

* Corresponding author

ABSTRACT

Aim/Purpose	One of the most fascinating developments in computer user interfaces in recent years is the rise of "chatbots". Yet extent information system (IS) curriculum lacks teaching resources on chatbots programming.
Background	To better prepare students for this new technological development and to en- hance the IS curriculum, we introduce a project that teaches students how to program simple chatbots, including a transactional chatbot and a conversational chatbot.
Methodology	We demonstrated a project that teaches students how to program two types of simple chatbots: a transactional chatbot and a conversational chatbot. We also conducted a survey to examine students' perceptions on their learning experi- ence.
Findings	Our survey on students' perception of the project finds that learning chatbots is deemed very useful because chatbot programming projects have enabled the students to understand the subject better. We also found that social influence has positively motivated the students to learn chatbot programming. Though most of the students have no prior experiences programming chatbots, their self-efficacy towards chatbot programming remained high after working through the programming project. Despite the difficult tasks, over 71% of respondents agree to various degrees that chatbot programming is fun. Though most students agree that chatbot programming is not easy to learn, more than 70% of respondents indicated that they will use or learn chatbots in the near future. The overwhelmingly positive responses are impressive given that this is the first time for the students to program and learn chatbots.

Accepting Editor: Eli Cohen | Received: April 16, 2018 | Revised: July 15, October 2, December 31, 2018, March 12, March 26. 2019 | Accepted: March 29, 2019.

Cite as: Long, J., Yuan, M. J., & Lee, H-M. (2019). How to program a chatbot – An introductory project and student perceptions. *Issues in Informing Science and Information Technology, 16,* 1-31. <u>https://doi.org/10.28945/4282</u>

(CC BY-NC 4.0) This article is licensed to you under a <u>Creative Commons Attribution-NonCommercial 4.0 International</u> <u>License</u>. When you copy and redistribute this paper in full or in part, you need to provide proper attribution to it to ensure that others can later locate this work (and to ensure that others do not accuse you of plagiarism). You may (and we encourage you to) adapt, remix, transform, and build upon the material for any non-commercial purposes. This license does not permit you to use this material for commercial purposes.

Recommendations for Practitioners	In this article, we introduced a step by step project on teaching chatbot pro- gramming in an information systems class. Following the project instructions, students can get their first intelligent chatbots up and running in a few hours using Slack. This article describes the project in detail as well as students' per- ceptions.
Recommendations for Researchers	We used UTAUT model to measure students' perception of the projects. This study could be of value to researchers studying students' technology learning and adoption behaviors.
Impact on Society	To our best knowledge, pedagogical resources that teach IS students how to program chatbots, especially the introductory level materials, are limited. We hope this teaching case could be of value for IS educators when introducing IS students to the wonderful field of chatbot programming.
Future Research	For future work, we plan to expand the teaching resources to cover more ad- vanced chatbot programming projects, such as on how to make chatbot more human-like.
Keywords	information system education, chatbot, artificial intelligence, student perceptions

INTRODUCTION

One of the most fascinating developments in computer user interfaces in recent years is the rise of "chatbots". Some people hail it as the "new command line" (Roach, 2016; Zyane, 2017), but it is much more than that. Chatbots are Artificial Intelligence "robots" used in human users' preferred environments, mostly in messaging applications. Chatbots converse with users in a natural language, understand what users want, and do the humans' bidding through a large network of connected services with natural and pervasive user interfaces. With the rising popularity of messaging apps, users are increasingly interested in this mode of communication—the next step in the continuous user interface evolution from text command line to Graphical User Interface to web apps to mobile apps and then to "bots".

One of the first chatbots, ELIZA (Weizenbaum, 1966), was developed 50 years ago at the Massachusetts Institute of Technology. It simulated a Rogerian psychotherapist, someone who just repeats the human user's words back to the human. In recent years, with the development of smartphone technology, smartphone-based chatbots have gained wide interest in the industry with high profile products such as the Apple Siri and Amazon Alexa as examples. Popular messaging platforms, such as Facebook Messenger and Skype, announced developer programs to support chatbot applications. At the same time, smaller messaging platforms, such as Slack and Telegram, launched "bot stores" and investment funds to attract developers.

There are already many innovative uses of bots. For example, Github's intelligent bot HuBot performs tasks from running software builds on demand, deploying software to production websites, scheduling and running meetings, and even translating between human languages. All these tasks were once done by highly paid human employees (<u>https://hubot.github.com/</u>, accessed December 2017).

Clearly, as chatbots become more prevalent in business applications, chatbot programming will become more a desirable skill for IS students. However, since this field is still quite new and developing fast, there are relatively little pedagogical resource on teaching IS students how to program chatbots. Most of the existing resources are reference type of books or magazine articles. An exception is some pioneering research done in education setting to facilitate student learning. Griol, Molina, and Sanchis De Miguel (2014) studied how to use a conversational agent to help children learn to appreciate their environment. Oliveira, Silva, and Silva (2016) examined the mechanisms of human language formation to simulate a more human-like conversational chatbot. However, despite this early research on chatbot, there is no step-by-step instructional material to introduce chatbot programming to beginning learners, especially students in the information systems fields. Given the rising popularity of chatbots and their increasingly extensive applications in the industry, the skills set to be able to program chatbot becomes more marketable and desirable. To better prepare students for this new technological development, to help students build more marketable skillsets, and to enhance the information system (IS) curriculum, we introduce a project that teaches undergraduate students major in computer information systems how to program a simple chatbot. Following the project instructions, students can get their first intelligent chatbot up and running in a few hours using Slack. This article describes the project in detail as well as students' perceptions. To our best knowledge, pedagogical resources that teach IS students how to program chatbots, especially the introductory level materials, are limited. We hope this teaching case could be of value for IS educators when introducing IS students to the wonderful field of chatbot programming.

LITERATURE REVIEW

A chatbot can be defined as "an artificial construct that is designed to converse with human beings using natural language as input and output" (Brennan, 2006). Conversational agents, or chatbots, provide a natural language interface to their users (Kerly, 2007). Over the years, their design has become increasingly sophisticated and their use adopted in education (e.g., Jia, 2004) and commerce (e.g., Statista, 2017). More recently, smartphone-based chatbots have gained wide interest in the industry with the introduction of high profile products such as the Apple Siri and Amazon Alexa, as well as messaging platforms such as Facebook Messenger and Skype. For instance, More than 2 billion business-related messages are sent through Facebook Messenger chats according to Facebook IQ (2018). 67% of that study participants expressed expectations of having the possibility to get in touch with the sellers and service providers by means of chats.

According to 2017 worldwide Statista survey (Statista, 2017), 34% of respondents declared they would prefer to answer the questions from AI by means of a chatbot or a virtual assistant in regard to e-commerce. Healthcare (27%) and telecommunications (25%) were detected to be the second and the third correspondingly.

ELIZA (Weizenbaum, 1966) was developed as one of the first chatbots. ELIZA analyzed input sentences and created its response based on reassembly rules associated with a decomposition of the input. Dr. Richard Wallace (2009) developed A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) as a chatbot built using Artificial Intelligence Markup Language (AIML). The chatbot is based on categories containing a stimulus, or pattern, and a template for the response. Category patterns are matched to find the most appropriate response to a user input. Furthermore, AIML tags provide for consideration of context, conditional branching, and supervised learning to produce new responses. Overall, A.L.I.C.E. is a viable and experienced system.

There are scarce resources on teaching IS students how to program chatbots. The main reason for the lack of extent literature on teaching chatbot programming is probably because modern chatbot development is still a relatively nascent field. However, we do find that chatbots have been used in education to facilitate student learning. For instance, based on natural language processing, intelligent tutoring, text-to-speech generation, voice recognition, and virtual reality, a prototype of a chatbot was developed to serve as a language instructor assistant. The chatbot could simulate an intelligent dialogue capable of conducting a conversation, correct some grammar and spelling mistakes, as well as provide definitions and translations of words and phraseological units (Pietroszek, 2007). Another simulation system focuses on supplying a virtual chatting partner (chatbot), which can chat in English with the English learners anytime, anywhere. It generates communicative responses according to the user input, the dialogue context, the user's and its own personality knowledge, common sense knowledge, and inference knowledge (Jia, 2009).

Besides assisting language learning, chatbots have been used as a tool for network management training. Dorothy, a network management-aware chatbot, was created to interact not only with the user, but also with the managed network via management protocols (Leonhardt, Tarouco, Vicari, Santos, & Silva, 2007). A multimodal conversational agent has also been developed to help children appreciate and protect their environment (Griol et al, 2014). Mechanisms of human language formation (semantic, syntactic, pragmatic, and morphologic analysis) were used to simulate a normal conversation in an almost human way, which could eventually replace a human being (Oliveira, et al., 2016). In another research, a community of students, teachers, and interested developers programmed the database of the educational chatbot with educational content, including questions and answers on school subject information and other digital content, to add data to an educational chatbot's knowledge base (Bii & Too, 2016).

To extend the existing research on chatbot, it is valuable to examine ways to include teaching chatbot programing in the IS curriculum. Thus our project introduces a step-by-step project on how to program for chatbots. IS educators could use this project as an introduction to students interested in chatbot programming. Given the increasingly extensive development and utilization of chatbot applications in the IT industry, teaching students how to program chatbots could enhance students' marketable skill sets. In the following section, we will describe the project we designed to teach undergraduate students in the IS field how to program chatbots.

RESEARCH METHODOLOGY

In this project, students are instructed to develop two types of chatbots – transactional bots and conversational bots – and our project is designed as a phased project. We divide the project into two phases because we focus on different types of chatbots in each phase: students first create a transactional bot (the Movie Info bot) and then a conversational bot (the Movie Line bot). The project uses Slack as the platform to build the chatbot on. We also suggested to students the free web hosting services (e.g., Heroku, IBM Bluemix, RedHat OpenShift) to host the PHP code.

Three class sessions (each session is 80 minutes) were devoted to explaining and programming the projects. These sessions were held in the computer lab. During each session, students spent about 40 minutes working in their groups and coding for the chatbot based on the codes and instructions given by the instructor. During the second 40 minutes, the instructor explained the codes and helped students debug. There is no student tutor available. Students learned from interactions with their peers and the instructor.

To gauge the students' interests in programming for chatbots, and subsequently examine how to improve the teaching methodology and learning experience, we conducted a survey on students' perceptions and we share the survey results in this section too.

Two Types of Intelligent Bots

First, it is crucial to understand that there are two types of intelligent bots. The first is transactional bots. These bots respond to user commands to complete a single task at a time. Of course, the transactional task could be quite complicated, such as "join this meeting and take notes." The key here is that the user simply tells the bot what needs to be done. There is no "conversation" per se. The above-mentioned GitHub HuBot is a good example of transactional bots, as is Apple Siri. Example tasks for a transactional bot include report weather, look up movie times or ratings, build software, translate a term, schedule a reminder, schedule a meeting, etc.

The second kind of intelligent bots are conversational bots. These type of bots can carry a conversation with human users and figure out what the human users want before executing the task. Conversational bots are much harder to program than transactional bots because they need to keep track of the conversation context (i.e., when the user says "this is correct," what does "this" refer to?) and detect the user's sentiment. The conversational bots also need to derive a certain conclusion at the end of the conversation and act on it.

SLACK CHATBOT BASICS

For this teaching project, we build our bots on Slack. Slack is a group messaging platform for workrelated teams. As of September 2017, it had 6 million daily active users (Konrad, 2017), and it is growing fast. Slack is one of the pioneers in the current wave of chatbot renaissance. Slack provides one of the first "bot stores" in the industry so that teams can discover and install bots easily. As expected, most Slack bots are work and productivity related. We choose Slack to build our bots because of the following reasons. First and foremost, Slack provides a ready-to-use infrastructure to plug-inn-play students' bots. It already has a very user-friendly interface for a sophisticated messaging app. Second, Slack allows users to create an unlimited number of teams and messaging channels, making it easy to test students' bots in private. Third, Slack is very popular, and many people are already familiar with it. Fourth, Slack is free for small teams such as student teams. Fifth, and finally, Slack provides compelling tools to help students make their bots available to other teams and users in Slack's vast community.

THE MOVIE INFO BOT: A TRANSACTIONAL BOT

In Slack, there are many ways to build bots. The first part of the project is to create a transactional bot. For transactional bots, one approach is to take advantage of the slash commands. Slash commands consist of a slash followed by a command keyword. The objective is for students to build a custom command called */movie* that can give the user movie information. Any user can invoke this command followed by a movie title, and it will give him or her brief information about this movie. For example, type in */movie Star Wars episode IV* (Figure 1):



Send

Figure 1. Slash command example

The response should be as shown in Figure 2.



+

Movie Info BOT 12:50 PM

'Star Wars: Episode IV - A New Hope' 1977. Rating: 8.7 IMDB: http://www.imdb.com/title/tt0076759/ Plot: Luke Skywalker joins forces with a Jedi Knight, a cocky pilot, a wookiee and two droids to save the galaxy from the Empire's world-destroying battlestation, while also attempting to rescue Princess Leia from the evil Darth Vader.

Message

To create a bot for slash command *movie*, students need to first create this command in Slack's customization tool. It is under the "Custom integrations \rightarrow Slash Commands" menu. Here, students need to specify an external web server to respond to the command (Figure 3).

Command	/movie
	Examples: /deploy, /ack, /weather
	Reserved: /help, /me, /remind, /task, /bug, /feed, /shrug, /halp, /topic, /status, /away, /dm, /msg
	/open, /join, /close, /leave, /mute, /who, /archive, /unarchive, /feedback, /hangout, /color,
	/colour, /invite, /commands, /play, /active, /part
UKL	http://michaelyuan.com/download/bots/omdb_title.php
	The URL to request when the slash command is run.
Method	POST

Figure 3. Slack customization tool - Slash Commands

Then, when users type:

/movie Star Wars episode IV

Slack sends an HTTP POST to the specified server. The POST parameter "text" will have the value "Star Wars episode IV":

text=Star Wars episode IV

Slack then waits for the server to respond and displays whatever text the server returns. In our project, the server queries the OMDB API for the movie title. OMDB API returns movie information in a JSON object like this:

```
ł
    "Title":"Star Wars: Episode IV - A New Hope",
    "Year":"1977",
    "Rated": "PG",
    "Released":"25 May 1977",
    "Runtime":"121 min",
    "Genre": "Action, Adventure, Fantasy",
    "Director": "George Lucas",
    "Writer": "George Lucas",
    "Actors":"Mark Hamill, Harrison Ford, Carrie Fisher, Peter Cush-
ing",
    "Plot":"Luke Skywalker joins forces with a Jedi Knight, a cocky pi-
lot, a wookiee and two droids to save the galaxy from the Empire's
world-destroying battle-station, while also attempting to rescue Prin-
cess Leia from the evil Darth Vader.",
    "Language": "English",
    "Country": "USA",
    "Awards": "Won 6 Oscars. Another 38 wins & 27 nominations.",
    "Metascore":"92",
```

```
"imdbRating":"8.7",
"imdbVotes":"852,420",
"imdbID":"tt0076759",
"Type":"movie",
"Response":"True"
```

}

The server then extracts relevant information from the JSON and assembles and formats the HTTP response text like this:

```
'Star Wars: Episode IV — A New Hope' 1977. Rating:8.7 IMDB:
http://www.imdb.com/title/tt0076759/ Plot: Luke Skywalker joins forces
with a Jedi Knight, a cocky pilot, a wookiee and two droids to save the
galaxy from the Empire's world-destroying battle-station, while also
attempting to rescue Princess Leia from the evil Darth Vader.
```

The full server code for the omdb_title.php file in PHP is below. We suggested students use many free web hosting services that can run PHP (e.g., Heroku, IBM Bluemix, RedHat OpenShift, etc.) to host the PHP code.

```
<?php
```

```
$json_str = file_get_contents("http://www.omdbapi.com/?t=" . urlen-
code($ REQUEST["text"]) . "&y=&plot=short&r=json");
```

```
$json = json_decode($json_str, true);
```

```
echo "'" . $json['Title'] . "' " . $json['Year'] . ". Rating:" .
$json['imdbRating'] . " IMDB: http://www.imdb.com/title/" .
$json['imdbID'] . "/ Plot: " . $json['Plot'];
?>
```

But, in the case of getting movie information, having to enter a full title of a movie and getting back a short description left much to be desired. Therefore, the next phase of the project is to create a conversational bot.

THE MOVIE LINE BOT: A CONVERSATIONAL BOT

In this part of the project, students are instructed to build a conversational bot that can interact with the user to discover exactly what movie information the user wants.

In Slack, a good place to build a conversational bot is a channel. A channel is a type of "group chat" in Slack. Any user can join a channel and start chatting. All the messages in the channel are shown to every user in the channel. However, besides human users, the channel itself can also respond to messages, making it a "bot" in some circumstances. We set up a channel called #movie_line. The screen shots that follow (Figure 4 to Figure 8) show the sequence of commands to drill from an ambiguous title search to show times. Please note that in Figure 4, we could not show all the options in the screenshot due to space limitations, and Figures 6 and 7 show the continuous flow of the commands.

●●000 T -	Mobile Wi-Fi 🗢	12:59 PM	@ ◀ ∦	58% 🔳 🖓
#	#movie_line	•	Q	•••
	Politician(2011 9 Kylie - Aphro 10 Football Ma	bdite: Les Fol anager 2011	ies Tour 2011(: (2010)	2011)
	juntao 12:58 P 6	Μ		
ß	BOT 12:58 PM You selected R Cleveland (201 Please respond showtime, cast,	A L USH Time M L1) I with one of trivia, quote	1achine 2011: the following: s or <i>done</i> for m	Live in <i>detail</i> , ore
	juntao 12:59 P Done	Μ		
&	BOT 12:59 PN Bye juntao. See	⁄l e you next ti	me!	
+	Star Wars			Send
	Figure 4. Sear	ch by movie	e title example	2
●●000 T -	Mobile Wi-Fi 🤶	12:59 PM	@ 1 *	57% 🔳 🖓
#	#movie_line	•	Q	•••
(-)	I have found th your search. Pl learn more abo 1 Star Wars: F	ne following ease respond out a movie. Inisode IV - 4	movies matching d with a number	ng er to 977)

#	#movie_line 👻	Q	•••
6-9	I have found the following movies may your search. Please respond with a nu learn more about a movie.	tching mber	; to
	1 Star Wars: Episode IV - A New Hop 2 Star Wars: Episode V - The Empire S Back(1980)	e(197 trikes	/)
	3 Star Wars: Episode VI - Return of the Jedi(1983)	е	
	4 Star Wars: Episode I - The Phantom Menace(1999)		
	5 Star Wars: Episode III - Revenge of t Sith(2005)	he	
	6 Star Wars: Episode II - Attack of the Clones(2002)	•1	
	 7 Star Wars: The Force Awakens(2015) 8 Star Wars: The Clone Wars(2008) 9 Star Wars: The Clone Wars(2008–20) 10 Star Wars: Clone Wars(2003–2005) 	9) 015) 5)	

(+) 7 Send

Figure 5. Multiple results returned

●●000 T -	Mobile Wi-Fi 훅	12:59 PM	@ ◀ ∦	57% 🔳 🗋
#	#movie_line	•	Q	•••
	Menace(1999) 5 Star Wars: Eg Sith(2005) 6 Star Wars: Eg Clones(2002) 7 Star Wars: Th 8 Star Wars: Th 9 Star Wars: Th 10 Star Wars: C	bisode III - Rev bisode II - Atta ne Force Awak ne Clone Wars ne Clone Wars Clone Wars(20	venge of the ck of the vens(2015) v(2008) v(2008–2015) v(03–2005)	
	juntao 12:59 P 7	Μ		
ß	BOT 12:59 PN You selected S (2015) Please respond showtime, cast,	م tar Wars: The with one of t trivia, quotes o	Force Awake he following: or <i>done</i> for mo	ns detail, ore
(+) S	howtime			Send
●●000 T-	Figure 6. D Mobile Wi-Fi 중	Prill down to o	one result. @ 7 *	57% 💶 🗅
#	#movie_line	-	Q	•••
	9 Star Wars: Th 10 Star Wars: (juntao 12:59 P 7	ne Clone Wars Clone Wars(20 M	(2008–2015) 003–2005)	
ß	BOT 12:59 PM You selected S (2015) Please respond showtime, cast,	م t ar Wars: The d with one of t trivia, quotes o	Force Awake he following: or <i>done</i> for mo	ns detail, ore
	juntao 12:59 P Showtime	Μ		
ß	BOT 12:59 PM Follow this link www.imdb.cor	/ c for showtime n/showtimes/	s: http:// title/tt24884	96
(+)	Aessage			

Figure 7. Search by showtime

+	Star V Awake ₽G13 ★8.4	Vars: Episode ens (2015) 2 h 15 min 81 Metascore	VII - The Fo	orce
THEATE	RS NEAR YO	DU		
\heartsuit	Alamo Dra Slaughter 5701 West S	fthouse Ciner Lane Slaughter Lane,	na - Austin	0.8 mi 🔉
Star Wa	rs The Force	Awakens		
12:45p	m 4:45 8	3:05 11:25		Tickets
	Regal Wes	tgate Stadiun	า 11	
	Figure 8	8. Return show	time results	

In Slack, you could add "Custom integration \rightarrow Outgoing WebHooks" to any channel. Here is how to specify another PHP script on the server to receive messages from this channel (Figure 9 and Figure 10).

Cust	om Inte	grations		
Insta	alled Apps	Custom Integrations	Connected Accounts	
	Bots Connect a bo	t to the Slack Real Time Messag	ng API.	1 configured >
ß	Outgoing We Get data out o	bHooks of Slack in real-time.		2 configured ≯
2.	Slash Comma Customized S	nds ilack commands for your team.		2 configured >

Figure 9. Add Outgoing WebHooks

▼ a new channel
a new channel
a channel is

Figure 10. PHP Script Example

When anyone posts a message in the channel, Slack sends an HTTP POST to the server. The POST contains the message in the "text" parameter, as well as other parameters describing the sender of the message:

```
token=OpDFurkJD0XMKVXtPinfeHVl
team_id=T0000
channel_id=C0000000
channel_name=test
timestamp=1355517523.000005
user_id=U00000000
user_name=juntao
text=Star Wars
```

When the server responds, it needs to send back a JSON text to describe what the bot "says." Compared with the plain text response from the previous Slash command-based transactional bot, this JSON response allows for structure and rich formatting of the response. For instance, the simple JSON response below indicates that the text should be formatted using a modified markdown convention.

```
{
    "text": "*bold* `code` _italic_",
    "mrkdwn": true
}
```

Now, the students are instructed to write the server PHP script. This script is more complicated than the previous transactional bot since it needs to keep track of conversation state (i.e., is the user asking to search a title or responding to one of the choices presented by the bot?) In this project, for the sake of simplicity, we create a cache file for each user session. The cache file is named after the token submitted by Slack in the HTTP POST to uniquely identify the conversation channel.

The overall flow of the PHP script is as follows:

First, the script checks if the cache file already exists. If not, the user is asked to search for a new movie. The search result is saved in the cache file. The cache file now exists and a new session starts.

If the cache file exists, the script examines its content. If the content is in the cached search result, the script will check if the user entered a number to select a movie from the results list, and save that selected movie into the cache file.

If the cache file content is a single selected movie, the script asks the user to drill down into details of the movie. If the user says "done," the script deletes the cache file and waits for a new search query.

The code below shows the PHP script structure:

```
<?php
// The bot should NOT respond to its own messages!
if ($ REQUEST["user name"] == "slackbot") {
  exit;
// This is the cache file name
$cache file = $ REQUEST["token"] . `.txt';
if (!file exists($cache file)) {
 // 1. No cache file. This is a new conversation. Start with search
  // The search results JSON from OMDB is in $json str, we will store
it in the cache file now
  file put contents($cache file, $json str);
 // RESPONSE: Create movie list and ask the user to select one.
} else {
  $cache content = file get contents($cache file);
  if (strlen($cache content) < 10) {</pre>
    // 2. The cached file has a selected IMDB movie ID. We are expect-
ing a "detail" command here
    $command = strtolower($ REQUEST["text"]);
    if ($command == "done") {
        unlink ($cache file); // Delete the cache file
        exit;
    }
    // RESPONSE: Give the user the details link
  } else {
    // 3. The cached file has the search results. We are expected a
number to select a movie.
    // Save the selected movie's IMDB ID to the cache file
    // RESPONSE: ask the user to enter a "detail" command for the se-
lected movie
  }
```

}

The PHP script queries the OMDB server for a movie search, transforms the OMDB results into Slack JSON, returns the Slack JSON as the response, and then saves the OMDB results in a cache file. The OMDB return result is as follows:

```
{"Search":[
    {
        Title":"Star Wars: Episode IV - A New Hope",
        "Year":"1977",
        "imdbID":"tt0076759",
```

```
"Type": "movie",
    "Poster":"http://ia.media-
imdb.com/images/M/MV5BMTU4NTczODkwM15BM15BanBnXkFtZTcwMzEyMTIyMw@@. V1
SX300.jpg"
  },{
    "Title":"Star Wars: Episode V - The Empire Strikes Back",
    "Year":"1980",
    "imdbID":"tt0080684",
    "Type": "movie",
    "Poster": "http://ia.media-
imdb.com/images/M/MV5BMjE2MzQwMTgxN15BM15BanBnXkFtZTcwMDQzNjk20Q@@. V1
SX300.jpg"
  },{
    . . . . . . .
  },
 . . . . . . .
],
"totalResults":"294",
"Response":"True"}
```

After the above PHP script processes it, the JSON data sent back to Slack becomes:

```
{ "text": "I have found the following movies matching your search.
Please respond with a number to learn more about a movie.\n
1 Star Wars: Episode IV - A New Hope(1977)
2 Star Wars: Episode V - The Empire Strikes Back(1980)
3 Star Wars: Episode VI - Return of the Jedi(1983)
4 Star Wars: Episode I - The Phantom Menace(1999)
5 Star Wars: Episode III - Revenge of the Sith(2005)
6 Star Wars: Episode II - Attack of the Clones(2002)
7 Star Wars: The Force Awakens(2015)
8 Star Wars: The Clone Wars(2008-2015) 10 Star Wars: Clone Wars(2003-
2005) ",
"mrkdwn": true }
```

Below is PHP script in more detail to show how the search is performed.

```
<?php
if (!file exists($cache file)) {
  // This is a new conversation. Start with search
  $json str = file get contents("http://www.omdbapi.com/?s=" . urlen-
code($ REQUEST["text"]) . "&r=json");
 file put contents($cache file, $json str);
  $movies list str = "";
  $json = json decode($json str, true);
 for($i=0; $i<count($json['Search']); $i++) {</pre>
    $movies_list_str = $movies_list_str . ($i+1) . " " .
$json[`Search'][$i][`Title'] . "(" . $json[`Search'][$i][`Year'] .
")\n";
 }
?>
"text": "I have found the following movies matching your search. Please
respond with a number to learn more about a movie.\n <?= $mov-
ies list str ?>",
```

```
"mrkdwn": true
}
<?php
}
?>
```

Next, the user sends in a number to select a movie from the list. At this point, the cache file should already exist, and the content of the cache file is the OMDB result. Once the PHP script detects a valid number, it selects the movie and writes its IMDB ID into the cache file. The script responds with a message that highlights the selected movie and asks the user to further specify what he or she wants to know about that movie. The PHP script section is as follows.

```
<?php
 // This is number selector on JSON array
 $json = json decode($cache content, true);
 $num = intval($ REQUEST["text"]);
 if ($num < 1 || $num > count($json['Search'])) {
?>
ł
"text": "Sorry, I did not understand. Please respond with a number in
the list",
"mrkdwn": true
}
<?php
   exit;
 } else {
    $movie = $json['Search'][$num - 1];
?>
{
"text": "You selected *<?= $movie[`Title'] ?>* (<?= $movie[`Year'] ?>)
\nPlease respond with one of the following: detail , showtime ,
cast , trivia , quotes or done for more",
"mrkdwn": true
<?php
   file put contents($cache file, $movie['imdbID']);
   exit;
 }
?>
```

Finally, the user sends a message to ask for details for that movie. The PHP script creates a custom IMDB link to send back to the Slack channel. If the user says "done," the PHP script will delete the cache file and be ready for the next movie query.

```
<?php
// this is IMDB ID
$command = strtolower($_REQUEST["text"]);
if ($command == "detail") {
    $imdb_url = "Follow this link for detail:
    http://www.imdb.com/title/" . $cache_content;
} elseif ($command == "showtime" || $command == "showtimes") {
    $imdb_url = "Follow this link for showtimes:
    http://www.imdb.com/showtimes/title/" . $cache_content;
} elseif ($command == "cast") {
    $imdb_url = "Follow this link for full cast and crew:
}
</pre>
```

```
http://www.imdb.com/title/" . $cache_content . "/fullcredits";
} elseif ($command == "trivia") {
   $imdb_url = "Follow this link for trivia:
http://www.imdb.com/title/" . $cache_content . "/trivia";
} elseif ($command == "quotes") {
   $imdb_url = "Follow this link for quotes:
http://www.imdb.com/title/" . $cache_content . "/quotes";
}
?>
{
   "text": "<?= $imdb_url ?>",
   "mrkdwn": true
}
```

The full script of omdb_search.php can be found in Appendix A. It is more elaborate than the code listings above, as it handles basic user input errors.

STUDENT PERCEPTION SURVEY

A survey was conducted at the end of the semester to examine the students' perception of the chatbot programming project during the semester of Spring 2017. The goal of the survey is to help us understand the drivers of students' acceptances of the chatbot programming project and chatbot technology. It could also provide insights on designing interventions targeted at the students who may be less inclined to use or program chatbots.

The respondents consisted of undergraduate students majoring in Computer Information Systems. None of them had background experiences in chatbot programming. A total of 35 valid responses were collected and analyzed. The survey was anonymous and institutional review board (IRB) review was exempted. The survey was voluntary and was given at an IS project management class. The level of students were seniors.

The survey instrument was created based on the Unified Theory of Acceptance and Use of Technology (UTAUT) model (Venkatesh, Morris, Davis, & Davis, 2003). The UTAUT model is a widelyused instrument to assess the likelihood of success for new technology introductions. To determine new technology usage intention and behavior, UTAUT theory holds that there are several key constructs: 1) performance expectancy, 2) effort expectancy, 3) attitude, 4) social influence, 5) facilitating conditions, 6) self efficacy, and 7) anxiety. These are direct determinants of usage intention and behavior. Based on the UTAUT model, we have examined the above determinants of intention and usage. We also collected responses on Behavioral Intention, specifically the intention to use and learn chatbots in the near future. The answers to the questions were created and coded in a 7-point Likert Scale (Strongly Disagree is coded as 1, Strongly Agree is coded as 7). The survey instrument is included in Appendix B of this article.

The means and standard deviations of the responses to each survey question are listed in Table 1. Figure 11 shows the response mean of each factor. For each factor, we calculated the average of the responses to the survey questions, and use the average in Table 2 to Table 8 and Figure 12 to Figure 18. For instance, if the respondent chooses 6 in Performance Expectancy question 1, 5 in question 2, 6 in question 3, the computed score for this response is 5.67, the average of 6, 5, and 6.

Factors	Questions	Mean	S.D.
Performance	1. I find learning chatbots is useful.	5.60	1.42
Expectancy	2. Developing chatbots enables me to understand the CIS subject better	5.43	1.44
	3. Learning chatbots improves my knowledge in CIS.	5.71	1.47
Effort Expec-	4. It is easy for me to become skillful in developing the chatbots.	3.74	1.56
tancy	5. I find developing chatbots is easy to learn.	2.80	1.43
	6. Learning to program chatbots is easy for me.	2.77	1.61
Attitude	7. Programming chatbots is fun.	4.49	1.72
	8. I like programming chatbots.	4.26	1.70
	9. Programming chatbots makes the class more interesting.	4.91	1.79
Social Influ-	10. People on my team motivate me to learn chatbots.	4.31	2.18
ence	11. The instructor of this course encourages me to learn chatbots.	5.54	1.58
	12. My team has supported each other in learning chatbots.	5.00	1.91
Facilitating	13. I have the resources necessary to develop the chatbots.		2.02
Conditions	14. I have the knowledge necessary to program the chatbots.	3.49	1.80
	15. A specific person (or group) is available for assistance with development difficulties.	4.14	1.94
Self Efficacy	16. I could develop the chatbots if there is no one around to tell me what to do.	3.11	1.86
	17. I could develop the chatbots if I have a lot of time to complete the tasks.	5.46	1.17
	18. I could develop the chatbot if I could find help information when I got stuck.	6.03	1.27
Anxiety	19. I feel anxious about programming the chatbot.		1.77
	20. Programming chatbots is intimidating to me.	4.29	1.98
	21. I hesitate to develop the chatbot for fear of making mistakes.	3.03	1.85
Behavioral	22. I plan to use chatbots in the near future.	4.40	1.56
Intention	23. I intend to learn more about chatbots in the near future.	5.06	1.63
	24. I predict I will use/learn chatbots in the near future.	4.89	1.41

Table 1. Means and Standard Deviations of the Responses



Figure 11. Mean Responses of Each Factor

Performance expectancy (Table 2, Figure 12) is the highest rated factor among all the determinants. A bit over Seventy-four percent of responses agree or strongly agree that learning chatbots is very useful, chatbot programming projects have enabled them to understand the CIS subject better, and learning chatbots improves their knowledge in CIS.

Table	2. Free	uency a	and (Cumulat	ive P	ercentag	e of	Resp	onses -	Perf	formance	Exp	oectancy
								P				r	

Responses	Frequency	Percent %	
Strongly Disagree	0	0.00%	
Disagree	0	0.00%	
Somewhat Disagree	2	5.71%	
Neutral	2	5.71%	
Somewhat Agree	5	14.29%	
Agree	17	48.57%	
Strongly Agree	9	25.71%	



Figure 12. Frequency and Cumulative Percentage of Responses - Performance Expectancy

Students also tend to agree that Social Influence has positively motivated them to learn chatbot programming (Table 3, Figure 13). More than 65% students agree to some extent that people on their project team motivated them to learn chatbots, the instructor encouraged them to learn chatbots, and their teams have supported each other in learning chatbots.

Responses	Frequency	Percent %	
Strongly Disagree	1	2.86%	
Disagree	1	2.86%	
Somewhat Disagree	4	11.43%	
Neutral	6	17.14%	
Somewhat Agree	4	11.43%	
Agree	10	28.57%	
Strongly Agree	9	25.71%	

Table 3. Frequency and Cumulative Percentage of Responses – Social Influence



Figure 13. Frequency and Cumulative Percentage of Responses - Social Influence

Though most of the students have no prior experiences programming chatbots, their self-efficacy towards chatbot programming remains quite high after working through the programming project (Table 4, Figure 14). Over seventy-four percent of respondents agree to various extents that they could develop chatbots even if there is no one around to tell them what to do, if they have a lot of time to complete the tasks, and if they could find help information when they became stuck.

Responses	Frequency	Percent %	
Strongly Disagree	0	0.00%	
Disagree	0	0.00%	
Somewhat Disagree	2	5.71%	
Neutral	7	20.00%	
Somewhat Agree	11	31.43%	
Agree	12	34.29%	
Strongly Agree	3	8.57%	

Table 4. Frequency and Cumulative Percentage of Responses - Self Efficacy



Figure 14. Frequency and Cumulative Percentage of Responses - Self Efficacy

Despite the difficult tasks, over 71% of respondents agree to various degrees that chatbot programming is fun, they like programming chatbots, and programming chatbots makes the class more interesting. Apparently, they truly enjoyed the programming projects (Table 5, Figure 15)

Responses	Frequency	Percent %	
Strongly Disagree	2	5.71%	
Disagree	2	5.71%	
Somewhat Disagree	4	11.43%	
Neutral	2	5.71%	
Somewhat Agree	12	34.29%	
Agree	9	25.71%	
Strongly Agree	4	11.43%	

Table 5. Frequency and Cumulative Percentage of Responses - Attitude



Figure 15. Frequency and Cumulative Percentage of Responses - Attitude

With regards to effort expectancy questions (Table 6, Figure 16), a little over 22% of responses somewhat agree that it is easy to become skillful in developing chatbots, and it is easy to learn chatbot programming. About 54% of students find it difficult (strongly disagree, disagree, or somewhat disagree). The rest of the students responded neither agree nor disagree. This result is understandable since chatbot programming is completely new technology for students. In addition, when programming for the chatbot, students also need to use a new programming language (PHP), set up web hooks, and interact with APIs. To most of the students, this is new knowledge to learn aside from chatbot programming. Therefore, it is not surprising most students agree that chatbot programming is not easy to learn and becoming skillful in developing the chatbots takes great effort.

Responses	Frequency	Percent %	
Strongly Disagree	3	8.57%	
Disagree	6	17.14%	
Somewhat Disagree	10	28.57%	
Neutral	8	22.86%	
Somewhat Agree	6	17.14%	
Agree	2	5.71%	
Strongly Agree	0	0.00%	

Table 6. Frequency and Cumulative Percentage of Responses - Effort Expectancy





Besides effort expectancy, we also measured the students' anxiety factor (Table 7, Figure 17). Again, because programming chatbots are new to all the students, it is not surprising to see the anxiety level is high. Around 48% of the respondents agree to various extents that they feel anxious about programming the chatbot, programming chatbots is intimidating, and they hesitate to develop the chatbot for fear of making mistakes.

Responses	Frequency	Percent %	
Strongly Disagree	0	0.00%	
Disagree	8	22.86%	
Somewhat Disagree	5	14.29%	
Neutral	5	14.29%	
Somewhat Agree	5	14.29%	
Agree	11	31.43%	
Strongly Agree	1	2.86%	

Table 7. Fre	equency and	Cumulative	Percentage	of Respo	nses – Anxietv
10010 11 11	equeiney and	Gainanative	rereentage	or neopo	noco immety



Figure 17. Frequency and Cumulative Percentage of Responses - Anxiety

The most encouraging result is the positive responses on Behavioral Intention (Table 8, Figure 18), which measures students' intention to learn and use chatbots in the near future. More than 70% of respondents indicated that they will use or learn chatbots in the near future; among them, 40% of responses agree or strongly agree with those statements. The overwhelmingly positive responses are impressive given that this is the first time for the students to program and learn chatbots. Despite the high effort expectancy and high level of anxiety in learning chatbot programming, students are not discouraged by the challenging projects. Rather, their interests in learning chatbots remain high and the performance expectancy also is ranked high. In future research, we plan to examine more the motivational factors for students to learn chatbots, and also design intervention to reduce the high level of anxiety.

Frequency	Percent %	
0	0.00%	
2	5.71%	
2	5.71%	
6	17.14%	
11	31.43%	
9	25.71%	
5	14.29%	
	Frequency 0 2 2 6 11 9 5	

Table 8. Frequency and Cumulative Percentage of Responses – Behavioral Intention



Figure 18. Frequency and Cumulative Percentage of Responses – Behavioral Intention

DISCUSSION

In this article, we introduced a teaching case on teaching chatbot programming in an information systems class. We explained the development environment using Slack with step-by-step project instructions. Additionally, we conducted a survey on students' perceptions of the projects and discussed the results. This teaching case could be useful for educators in the IT education field when introducing chatbot programming to students.

In this introductory project, we focused our discussions on how to construct an intelligent bot to query movie information. It has the three key components for any intelligent bot:

A messaging user interface. This project uses Slack. However, it could be other messaging platforms or even a custom app the students build themselves.

A server to receive messages and create responses. In our project, students create simple PHP scripts for this purpose.

Backend services are needed to provide the knowledge required for the conversation or perform actions the user demands. In this project, we use OMDB web services for movie information.

There are many enhancements students can do to improve their bots. The first and most obvious is to make the bot a "person" in the chat application. In our example, we used slash commands or messaging channels as bots. They served our purposes, but a "person" could do even more. For instance,

Any user can have one-on-one conversation with the bot user.

The bot user can be brought into a message channel to participate in a conversation, and then leave once it is no longer needed.

The bot user can also proactively message human users to start a conversation rather than simply respond to a human user's requests.

In addition, the "intelligence" of the bot is determined by the backend service it has access to. In our example, all the knowledge about movies comes from OMDB web services. In the "API economy," more and more services are available as web services, and the bots can do more and more. A good example is the previously mentioned HuBot in GitHub—it can already do many human tasks around the office. Thus, students could utilize many interesting APIs to enhance their bots.

CONCLUSION

We believe this teaching case is among the first to introduce IS students to chatbot programming. For our students, learning this new technology and learning how to actually program a chatbot has tremendous benefits. The skills they learned can be directly applied in their future career given how many companies are developing the chatbots in serving customers better. Although currently there is a lack of teaching cases and resources on chatbot programming, as chatbots become more prevalent in business applications and as chatbot programming becomes more a desirable skill for IS students, hopefully, more teaching resources will be available.

There are some limitations of this teaching case, which hopefully we will be able to address in the future. Given the small size of the class, we only had 35 valid survey responses. 35 data records do not give sufficient statistical power for us to conduct structural equation modeling this time around. However, we plan to continue collecting data using the same UTAUT survey instrument. Once the number of data records is sufficient for structural equation modeling, we will analyze how determinants, including performance expectancy, anxiety, social influence, attitude, self efficacy, etc., may have impacts on behavior intention. Hopefully, more insights will be generated. Moreover, in future research, it will be interesting to track the actual adoption behavior instead of behavioral intention. Last but not least, given the high level of anxiety and effort expectancy, in the future, more support from the instructors and more help resources for the students are essential to ease the anxiety level in learning chatbots, especially for first-time learners.

REFERENCES

- Bii, P., & Too, J. (2016). What will be in those lap tops: Empowering students and teachers to add content to an educational chatbot's knowledge base. Universal Journal of Educational Research, 4(5), 941-948. <u>https://doi.org/10.13189/ujer.2016.040502</u>
- Brennan, K. (2006). The managed teacher, emotional labour, education, and technology. *Educational Insights,* 10(2), 55–65.
- Facebook IQ. (2018). How messaging moves business. Retrieved from https://www.facebook.com/business/insights/series/messaging-moves-business?ref=ens_rdr
- Griol, D., Molina, J. & Sanchis De Miguel, A. (2014). Developing multimodal conversational agents for an enhanced e-learning experience. Advances in Distributed Computing and Artificial Intelligence Journal, 3(1). <u>https://doi.org/10.14201/ADCAIJ2014381326</u>
- Jia, J. (2004). CSIEC (Computer Simulator in Educational Communication): An intelligent web-based teaching system for foreign language learning. *ED-MEDLA (World Conference on Educational Multi-media, Hypermedia & Telecommunications)*, Lugano, Switzerland.
- Jia, J. (2009). CSIEC: A computer assisted English learning chatbot based on textual knowledge and reasoning. *Knowledge-Based Systems*, 22, 249–255. <u>https://doi.org/10.1016/j.knosys.2008.09.001</u>
- Kerly, A. (2007). Bringing chatbots into education: Towards natural language negotiation of open learner models. *Knowledge-Based Systems*, 20,177–185. <u>https://doi.org/10.1016/j.knosys.2006.11.014</u>
- Konrad, A. (2017, Sept. 12). Slack passes 6 million daily users and opens up channels to multi-company use. *Forbes.* Retrieved December, 2017 from <u>https://www.forbes.com/sites/alexkonrad/2017/09/12/slack-passes-6-million-daily-users-and-opens-up-channels-to-multi-company-use</u>

- Leonhardt, M., Tarouco, L., Vicari, R., Santos, E., & Silva, M. (2007). Using chatbots for network management training through problem-based oriented education. *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT)*. <u>https://doi.org/10.1109/ICALT.2007.275</u>
- Oliveira, V., Silva, E., & Silva, P. (2016). The development of a chatterbot for environmental education. Proceedings of International Conference for Information Communication Technologies in Education.
- Pietroszek, K. (2007). Providing language instructor with artificial intelligence assistant. *International Journal of Emerging Technologies in Learning, 2*(4).
- Roach, A. (2016) Chat interfaces are the new command line. Retrieved from <u>http://ajroach42.com/chat-interfaces-are-the-new-command-line/</u>
- Statista. (2017). Acceptance of artificial intelligence chatbots by customers worldwide, as of 2017, by service. Retrieved from https://www.statista.com/statistics/717098/worldwide-customer-chatbot-acceptance-by-industry/
- Venkatesh, V., Morris, M., Davis, G., & Davis, F. (2003). User acceptance of information technology: Toward a unified view. MIS Quarterly, 27(3), 425-478. <u>https://doi.org/10.2307/30036540</u>
- Wallace, R. S. (2009). The anatomy of A.L.I.C.E. In R. Epstein., G Roberts, & G. Beber (Eds), Parsing Turing Test: Philosophical and methodological issues in the quest for the thinking computer (pp. 181-210). Dordrecht: Springer.
- Weizenbaum, J. (1966). ELIZA A computer program for the study of natural language communications between man and machine. *Communications of the ACM*, 9(1), 36–45. <u>https://doi.org/10.2307/30036540</u>
- Zyane, R. (2017, March 18). How ChatOps can help you DevOps better. *Chatbots Magazine*. Retrieved: https://chatbotsmagazine.com/how-chatops-can-help-you-devops-better-5-minutes-read-507438c156bf

APPENDIX A. OMDB_SEARCH.PHP CODE

```
omdb search.php
<?php
if ($ REQUEST["user name"] == "slackbot") {
    exit;
}
$cache file = $ REQUEST["token"] . '.txt';
if (!file exists($cache file)) {
    // This is a new conversation. Start with search
    $json str = file get contents("http://www.omdbapi.com/?s=" . urlen-
code($ REQUEST["text"]) . "&r=json");
    file put contents($cache file, $json str);
    $movies list str = "";
    $json = json decode($json str, true);
    for($i=0; $i<count($json['Search']); $i++) {</pre>
        $movies list str = $movies list str . ($i+1) . " " .
$json['Search'][$i]['Title'] . "(" . $json['Search'][$i]['Year'] .
")\n";
    }
?>
{
"text": "I have found the following movies matching your search. Please
respond with a number to learn more about a movie. \n <?= $mov-
ies list str ?>",
"mrkdwn": true
}
<?php
} else {
    $cache content = file get contents($cache file);
    if (strlen($cache content) < 10) {</pre>
        // this is IMDB ID
        $command = strtolower($ REQUEST["text"]);
        if ($command == "detail") {
            $imdb url = "Follow this link for detail:
http://www.imdb.com/title/" . $cache content;
        } elseif ($command == "showtime" || $command == "showtimes") {
            $imdb url = "Follow this link for showtimes:
http://www.imdb.com/showtimes/title/" . $cache content;
        } elseif ($command == "cast") {
```

```
$imdb url = "Follow this link for full cast and crew:
http://www.imdb.com/title/" . $cache content . "/fullcredits";
        } elseif ($command == "trivia") {
            $imdb url = "Follow this link for trivia:
http://www.imdb.com/title/" . $cache content . "/trivia";
        } elseif ($command == "quotes") {
            $imdb url = "Follow this link for quotes:
http://www.imdb.com/title/" . $cache content . "/quotes";
        } elseif ($command == "done") {
?>
{
"text": "Bye <?= $ REQUEST["user name"] ?>. See you next time!",
"mrkdwn": true
}
<?php
            unlink($cache file);
            exit;
        } else {
?>
{
"text": "Sorry, I did not understand. Please respond with one of the
following: _detail_, _showtime_, _cast_, _trivia_, _quotes_ or _done_",
"mrkdwn": true
}
<?php
            exit;
        }
?>
{
"text": "<?= $imdb url ?>",
"mrkdwn": true
}
<?php
    } else {
        // This is number selector on JSON array
        $json = json decode($cache content, true);
        $num = intval($ REQUEST["text"]);
        if ($num < 1 || $num > count($json['Search'])) {
?>
```

```
{
"text": "Sorry, I did not understand. Please respond with a number in
the list",
"mrkdwn": true
}
<?php
            exit;
        } else {
            $movie = $json['Search'][$num - 1];
?>
{
"text": "You selected *<?= $movie['Title'] ?>* (<?= $movie['Year'] ?>)
\nPlease respond with one of the following: _detail_, _showtime_,
_cast_, _trivia_, _quotes_ or _done_ for more",
"mrkdwn": true
}
<?php
            file put contents($cache file, $movie['imdbID']);
            exit;
        }
    }
}
?>
```

APPENDIX B. SURVEY INSTRUMENT

CHATBOT PERCEPTION SURVEY

All questions have the same choices based on 7 Point Likert Scale:

Strongly disagree (1) Disagree (2) Somewhat disagree (3) Neither agree nor disagree (4) Somewhat agree (5) Agree (6) Strongly agree (7)

Performance Expectancy

Q1. I find learning chat bots is useful.

Q2. Developing chat bots enables me to understand the CIS subject better.

Q3. Learning chat bots improves my knowledge in CIS.

Effort Expectancy

Q4. It is easy for me to become skillful in developing the chat bots.

Q5. I find developing chat bots is easy to learn.

Q6. Learning to program chat bots is easy for me.

Attitude Toward Developing Chat Bots

Q7. Programming chat bots is fun.

Q8. I like programming chat bots.

Q9. Programming chat bots makes the class more interesting.

Social Influence

Q10. People on my team motivate me to learn chat bots.

Q11. The instructor of this course encourages me to learn chat bots.

Q12. My team has supported each other in learning chat bots.

Facilitating Conditions

Q13. I have the resources necessary to develop the chat bots.

Q14. I have the knowledge necessary to program the chat bots.

Q15. A specific person (or group) is available for assistance with development difficulties.

Self Efficacy

Q16. I could develop the chat bots if there is no one around to tell me what to do.

Q17. I could develop the chat bots if I have a lot of time to complete the tasks.

Q18. I could develop the chat bot if I could find help information when I got stuck.

Anxiety

Q19. I feel anxious about programming the chat bot.

Q20. Programming chat bots is intimidating to me.

Q21. I hesitate to develop the chat bot for fear of making mistakes.

Behavioral Intention to Use/Learn Chatbot

Q22. I plan to use chat bots in the near future.

Q23. I intend to learn more about chat bots in the near future.

Q24. I predict I will use/learn chat bots in the near future.

BIOGRAPHIES



Dr. Ju Long is an associate professor of Computer Information Systems at the McCoy College of Business at Texas State University. She has a Ph.D. degree from the University of Texas at Austin, and a Master degree from the University of Michigan in Ann Arbor. She studies technology's impact on business and how new technologies are changing our world. She has published in many academic journals in the areas of information technology security, open source software development, mobile technologies, and healthcare IT.



Dr. Michael Juntao Yuan received his PhD from University of Texas at Austin. He is currently the chief scientist at the CyberMiles Foundation. His primary research interests include technology innovation, healthcare IT, and entrepreneurial studies.



Dr. Hsun-Ming Lee received his PhD in Industrial Engineering with a focus on information engineering from Arizona State University, USA. He is currently an Associate Professor of Computer Information Systems in the McCoy College of Business Administration at Texas State University, USA. His research interests include web engineering, technology-enhanced learning, and decision support systems.