Towards Best Practices for Chatbots

by

Maria Antonieta Ferman Guerra
B.Sc., Universidad Villa Rica, 2005
M.A., Universidad Villa Rica, 2007

A Project Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Towards Best Practices for Chatbots

by

Maria Antonieta Ferman Guerra
B.Sc., Universidad Villa Rica, 2005
M.A., Universidad Villa Rica, 2007

Supervisory Committee

Dr. Margaret-Anne Storey, Supervisor
(Department of Computer Science)

Dr. Hausi A. Müller, Department Member
(Department of Computer Science)

Supervisory Committee

Dr. Margaret-Anne Storey, Supervisor
(Department of Computer Science)

Dr. Hausi A. Müller, Department Member
(Department of Computer Science)

ABSTRACT

A chatbot is a computer program designed to simulate conversation with human users, especially over the Internet [3].

Designing Chatbots is a new emerging area for conversational interfaces. Chatbots are a great tool to allow users get answers to their inquiries. Users just need to ask their question and a chatbot using natural language will ideally reply with an appropriate answer. Despite being a new hot technological area, chatbot creation is not a straightforward task. There is a large amount of blogs and best practices to create chatbots, making the chatbot process less troublesome. However, there are still many challenges to overcome.

The process that designers need to follow is more effective when they use specific practices, this will allow them to have a smooth process by avoiding commonly made mistakes. This project gathers a comprehensive list of best practices to give designers a specific structure on how to design a script-based chatbot effectively.

Additionally, this project describes the two implemented evaluations of the best practices: The first was to design and implement a customer support chatbot for a software application, and the second was to carry out two interviews with chatbot designers of a software company.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My parents and sisters who believed in me when I didn't believe in myself. Thank you for providing me with unfailing support in tough times and share with me my success and failure. Especially thank you for being with me despite the distance.

My supervisor, Dr. Margaret-Anne Storey for giving me the great opportunity of having a brighter future and all her patience, support and kindness.

All the members of the CHISEL lab for all your knowledge and great moments that we spend together. Especially thanks to Cassandra Petrachenko for the act of kindness that you showed me, even though you did not know me.

My Canadian family and friends for all your support, guidance and love. Especial thanks to Maryi Arciniegas for everything you have taught me, all your patience, knowledge and care.

All the wonderful people in Redbrick for all your help, support, patience and the lessons learned. I would especially thank to the Shift team and Aaron Sundberg.

Life can only be understood backwards; but it must be lived forwards.
Søren Kierkegaard

To my family, you live in my heart and in everything I do.

Chapter 1

Introduction

1.1 Motivation

Inexperienced computer users require software that is accessible and customized to their needs. For this type of user, interaction with software can be challenging and a potential barrier to accessing software services; fortunately, chatbots can help. Chatbots allow users to interact and communicate with software through their own words or through specific commands, making it easier to understand and use.

In addition, Chatbots can provide cognitive support to users when they are performing a certain task. According to Falconer et al. [14] "The goal of cognitive support within a software system is to offload some of the user's cognitive processes involved in performing a task to the software. This can reduce the number of items that a user must internally track and process, allowing them to concentrate their expertise on other parts of the task."

Therefore, Cognitive Support is the aid that chatbots might give to users. According to Walenstein [37], "The cognitive support provided by a tool is the computational advantages that the tool provides. Cognitive support can therefore be understood entirely in computational terms: support is the pro-vision of computational advantage."

Chatbots should not be confused with conversational agents. According to Radziwill et al. [24], "Chatbots are one category of conversational agents, which are software systems that mimic interactions with real people. They are typically not embodied in the forms of animals, avatars, humans, or humanoid robots (those programs are considered to be embodied conversational agents)."

Chatbots efficiency can be augmented by allowing software to consider user preferences and adapt to their needs. According to Shevat, “Bots are a new user interface which let users interact with services and brands using their favorite messaging apps. Bots are a new way to expose software through a conversational interface.” [29]

However, creating a chatbot is not a straightforward task, and designers must keep several elements in mind when creating them. They also need to fully understand the tasks they want to accomplish; in other words, they need to think about the problem they want to solve and consider the best way to solve it.

Tool design is more effective when designers use specific practices to avoid starting from scratch and making common mistakes [17]. To address this, I developed a comprehensive list of best practices that provide designers a structure of how to design a script-based chatbot effectively.

The best practices I developed make chatbot designers aware of the most important aspects that should be considered when designing a script-based chatbot.

Many chatbot design elements are complex and require a considerable amount of planning. One example is smoothly mapping chatbot conversations into the user’s flow: many designers agree that mapping a full conversation is one of the hardest things to do [35, 39]. Designers need to fully understand the conversational script and flow that the chatbot needs to follow. Therefore, it is important for designers, who want to create chatbots, to follow a set of best practices.

There are several best practices and guidelines for designing script-based chatbots. However, to the best of my knowledge, there is not a complete set of best practices, because there is no consistency across authors. There is a large amount of sources highlighting some individual best practices, however, there is no complete set of practices to design a script-based chatbot.

The best practices outlined in this work describe the necessary elements of a chatbot, making the design process more effective and efficient.

1.2 Project Methodology

The project methodology is presented in Figure 1.1. The top section of the figure shows the four phases of the project, and a thin arrow is used to represent the flow between phases. The bottom section of the figure represents the output of the phases, and a thick arrow represents the output produced. The red box shows the main result of my project: best practices for effectively designing a script-based chatbots.

This project included four main phases as represented in Figure 1.1. **Phase I** comprises the literature review on the current guidelines for creating chatbots. In this phase, I carried out an extensive literature review on well-known online best practices to understand the most common practices to design a chatbot effectively (cf. Fig. 2.3). In addition, I also selected and used the most popular chatbots in order to understand their most important features. As a result, I noticed there is no standard for designing a script base chatbots.

Phase II describes the creation of the best practices to design a script-based chatbot effectively. The best practices are classified in four categories: Chatbot's purpose, User-Chatbot Communication, Chatbot Features, and Human Factor Concerns. The complete list of best practices is described in Chapter 2.

In **Phase III**, I created a customer support chatbot, using the best practices identified in Phase II as a first evaluation of the best practices. The chatbot process creation is described in Chapter 3.

The original idea was to implement a visualization chatbot, however, after having a meeting with the Shift's Director of Technology, the chatbot purpose was changed. Because the best practices can be used to design any type of chatbot, the change in the chatbot's purpose does not affect the evaluation of the best practices.

Finally, **Phase IV** contains an evaluation of the best practices through interviews with Chatbot creators. The process followed and results produced from this phase are described in Chapter 4.

After carrying out the four phases of this Master's project, I gained a great deal of knowledge and understanding of how a script-based chatbot should be designed.

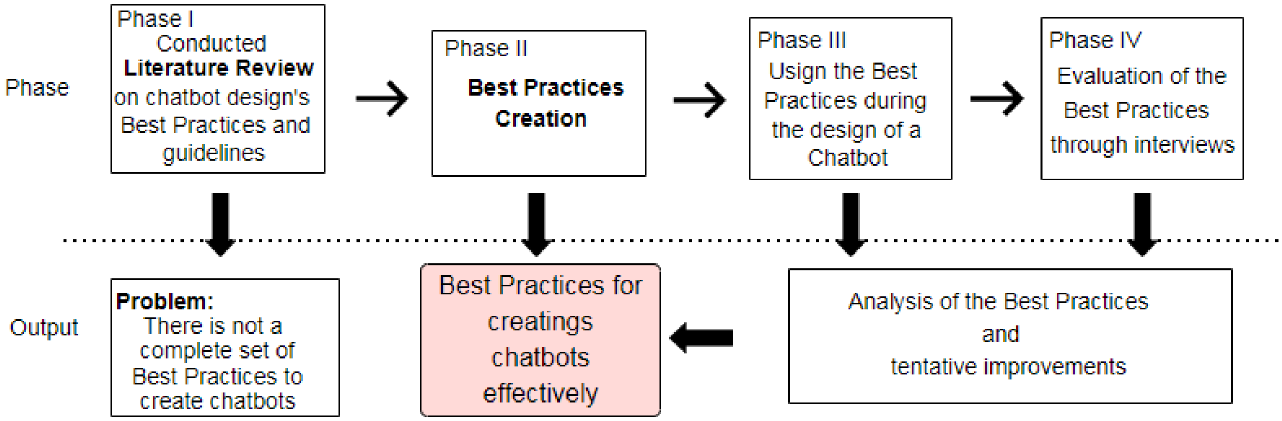


Figure 1.1: Project Methodology

Chapter 2

Identifying Best Practices to Create Chatbots Effectively

2.1 Methodology for Identifying Best Practices

After conducting an extensive literature review of existing best practices and guidelines in Phase I, and identifying the lack of a comprehensive list of best practices, I decided to create such a list in Phase II. Using the literature explored and my own knowledge, I created a list of 18 best practices. The methodology followed in this process is presented in Figure 2.1.

A white box represents one step of Phase II, and a pink box represents the best practices (output). A thin arrow represents the flow between steps and a thick arrow represents the output. In this diagram, the outputs are the 18 best practices and the improvements gathered from steps four and five.

The literature review started with chatbot's papers about the different types of chatbots. In this step, I decided to create best practices for script-based chatbots. After this, the literature review was conducted on all possible sources of data regarding chatbot design. This process was followed until no new best practices were found. After this, I reviewed the different technologies to create chatbots. Finally, I reviewed papers about the role of chatbots in customer service support.

The majority of the sources that support the best practices came from real world experiences in industry and were written by software engineers, chatbot designers, and technologists. The exception of this criteria is "the best practice #12 Social intelligence," which came from Katharine Schwab, an associate editor at Co.Design

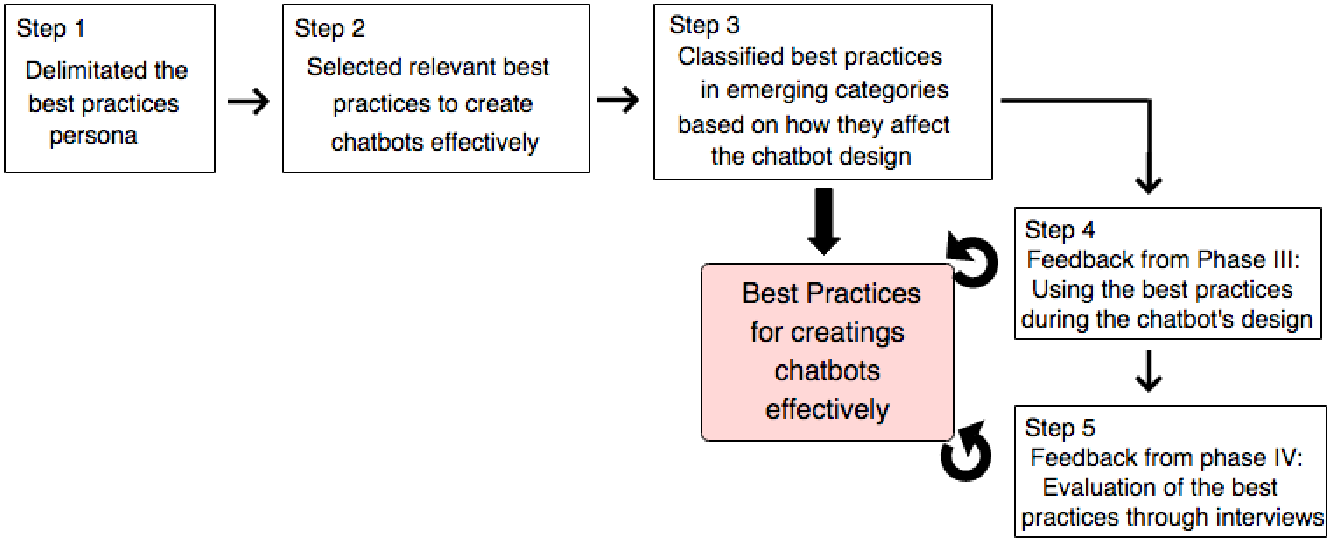

Step 1
Delimitated the best practices persona
Step 2
Selected relevant best practices to create chatbots effectively
Step 3
Classified best practices in emerging categories based on how they affect the chatbot design
Best Practices for creatings chatbots effectively
Step 4
Feedback from Phase III: Using the best practices during the chatbot's design
Step 5
Feedback from phase IV: Evaluation of the best practices through interviews

Figure 2.1: Methodology of Phase II (Steps 4 and 5 represent the feedback that came from Phases III and IV to refine the best practices)

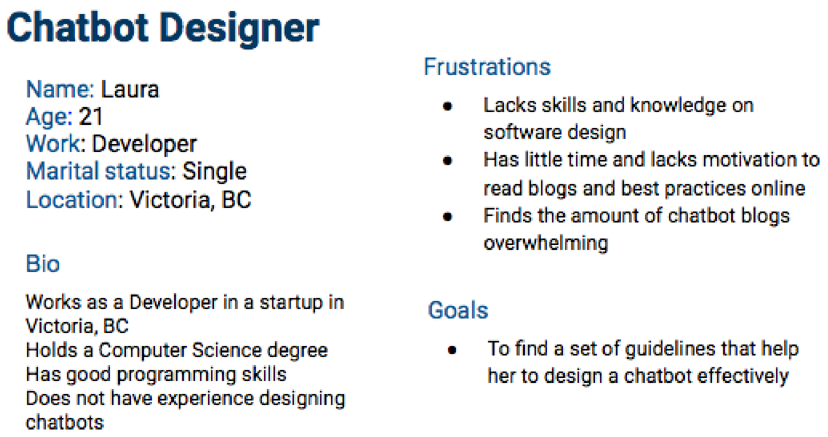
who covers technology, design, and culture. According to Schwab [26], "Artificial intelligence isn't just about smarts it's about social intelligence." I strongly believe that this element will become crucial for the effective development of a chatbot. More and more designers will create chatbots with sophisticated Artificial Intelligence algorithms, so while chatbots become more intelligent, more social and emotional knowledge will be needed.

To the best of my knowledge there is a limited amount of research papers about chatbot creation. The research papers used in this literature review support the ideas founded in the online blogs and best practices with definitions or conversational examples. Some ideas were also gathered from those papers such as "the best practice #15 User's control."

Phase II consisted of five steps (cf. Fig. 2.1). To illustrate the specific needs and goals that the best practices should satisfy I created a user persona in the first step. A persona is used by designers as an archetypal user of a system [11]. In this project the user persona is named "Laura" and she is defined as the type of person that will use the chatbot's best practices. Her definition is presented in Figure 2.2.

The second step was to select relevant best practices and guidelines to create script-based chatbots effectively.

The identification criteria for the selection was to choose best practices from sev-



Chatbot Designer

Name: Laura
Age: 21
Work: Developer
Marital status: Single
Location: Victoria, BC

Bio

Works as a Developer in a startup in Victoria, BC
Holds a Computer Science degree
Has good programming skills
Does not have experience designing chatbots

Frustrations

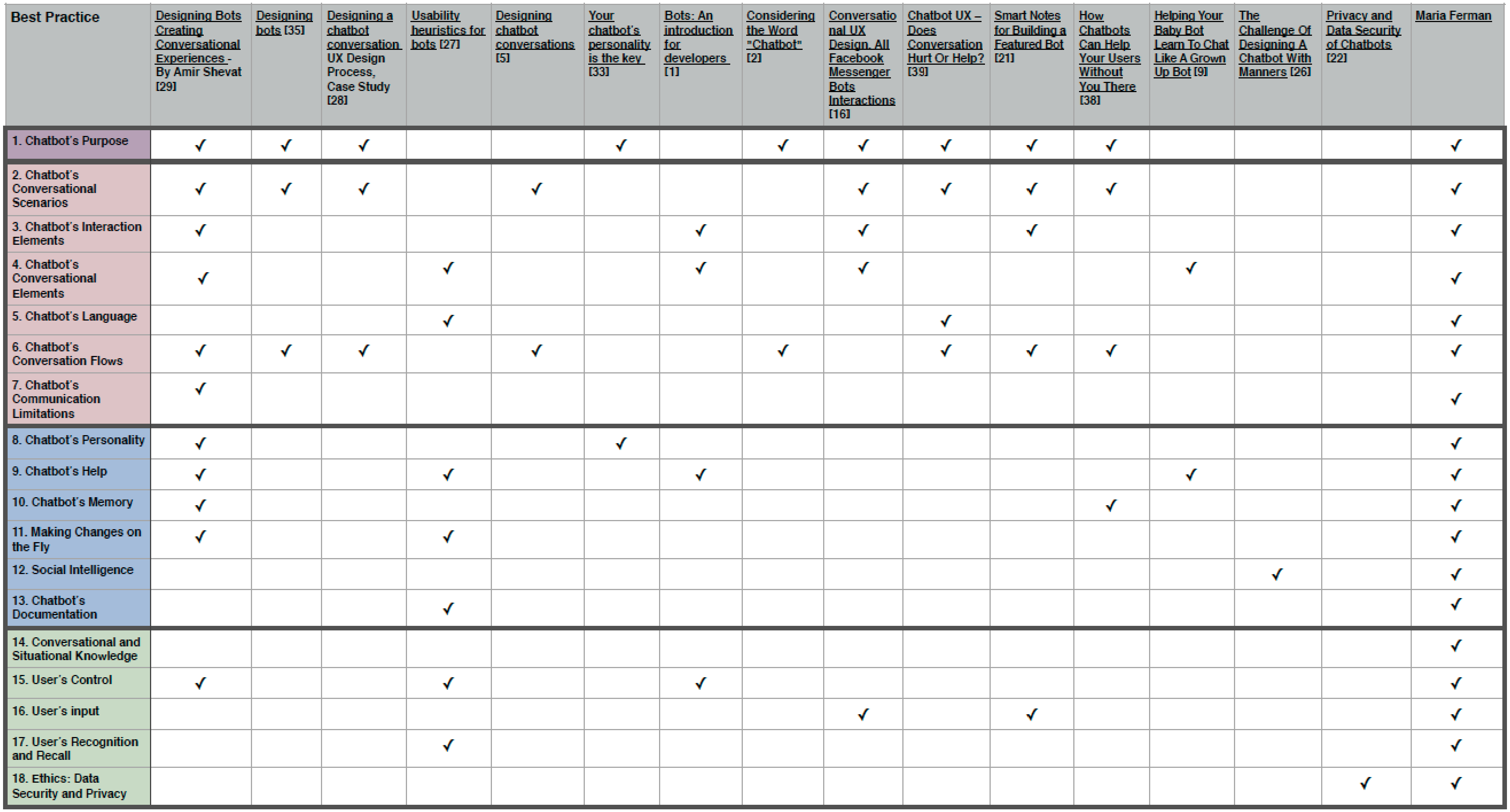
- Lacks skills and knowledge on software design
- Has little time and lacks motivation to read blogs and best practices online
- Finds the amount of chatbot blogs overwhelming

Goals

- To find a set of guidelines that help her to design a chatbot effectively

Figure 2.2: Persona used for selecting the best practices

eral online and printed sources (cf. Fig. 2.3.)



Best Practice	Designing Bots Creating Conversational Experiences - By Amir Shevat [29]	Designing bots [35]	Designing a chatbot conversation UX Design Process, Case Study [28]	Usability heuristics for bots [27]	Designing chatbot conversations [5]	Your chatbot's personality is the key [33]	Bots: An introduction for developers [1]	Considering the Word "Chatbot" [2]	Conversational UX Design. All Facebook Messenger Bots Interactions [16]	Chatbot UX – Does Conversation Hurt Or Help? [39]	Smart Notes for Building a Featured Bot [21]	How Chatbots Can Help Your Users Without You There [38]	Helping Your Baby Bot Learn To Chat Like A Grown Up Bot [9]	The Challenge Of Designing A Chatbot With Manners [26]	Privacy and Data Security of Chatbots [22]	Maria Ferman
1. Chatbot's Purpose	✓	✓	✓			✓		✓	✓	✓	✓	✓				✓
2. Chatbot's Conversational Scenarios	✓	✓	✓		✓				✓	✓	✓	✓				✓
3. Chatbot's Interaction Elements	✓						✓		✓		✓					✓
4. Chatbot's Conversational Elements	✓			✓			✓		✓				✓			✓
5. Chatbot's Language				✓						✓						✓
6. Chatbot's Conversation Flows	✓	✓	✓		✓			✓		✓	✓	✓				✓
7. Chatbot's Communication Limitations	✓															✓
8. Chatbot's Personality	✓					✓										✓
9. Chatbot's Help	✓			✓			✓						✓			✓
10. Chatbot's Memory	✓											✓				✓
11. Making Changes on the Fly	✓			✓												✓
12. Social Intelligence														✓		✓
13. Chatbot's Documentation				✓												✓
14. Conversational and Situational Knowledge																✓
15. User's Control	✓			✓			✓									✓
16. User's input									✓		✓					✓
17. User's Recognition and Recall				✓												✓
18. Ethics: Data Security and Privacy															✓	✓

Figure 2.3: Best practices' references

The third step was to come up with a classification of the best practices (cf. Fig. 2.4.) After the literature review, I realized that the purpose of the chatbot defines how the others best practices must be applied. For example, based on the purpose of the chatbot, the personality will be defined. A chatbot designed to allow teenagers to share images and memes will have a more juvenile personality than a chatbot that is oriented to give news to people. For this reason, the Best Practice #1 is the chatbot purpose, and only after defining this can the rest of the best practices be applied. Figure 2.4 shows the list of best practices created.

The categories in my classification are: Chatbot's purpose, User-Chatbot Communication, Chatbot Features, and Human Factor's Concerns. The second category, displayed in Figure 2.4 in pink is the User-Chatbot communication. This category contains the elements that allow a user and the chatbot to have a comprehensive and smooth communication.

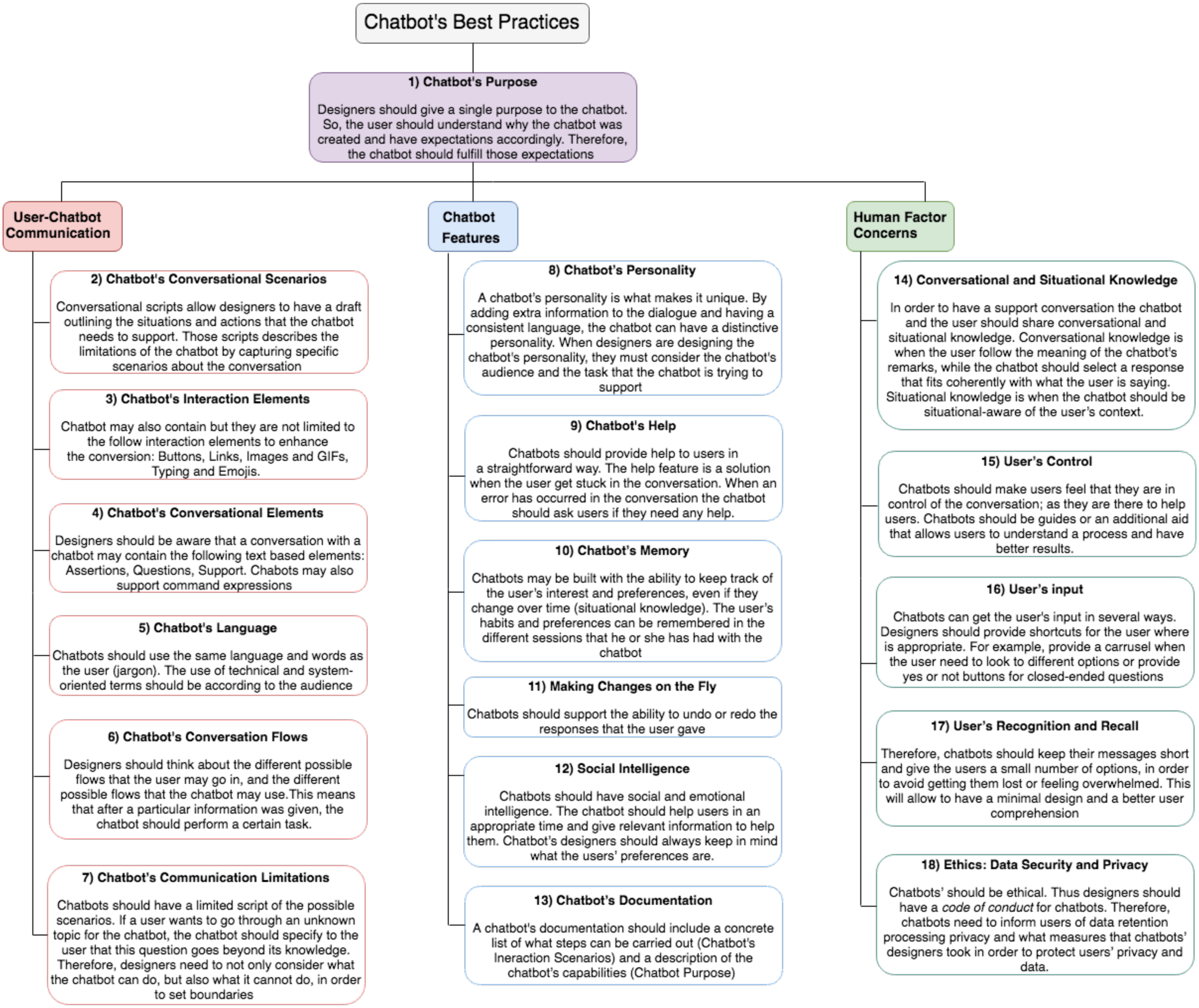
The third category is Chabot features, which is displayed in blue in Figure 2.4. This category lists the elements that allow the chatbot to be more efficient and to increase user engagement.

Finally, the last category is Human Factor's Concerns, which is displayed in green in Figure 2.4. This category deals with certain concerns that users may have, such as data security and how logs should be managed by chatbot designers. This category also shows the human part of the interaction, such as defining the kind of input the user may use.

Step four represents Phase III of this Master's project, where a chatbot was created based on the best practices. And step five represents Phase IV of the project, in which an evaluation of the best practices was carried out by interviewing two chatbot designers. Steps four and five allowed me to improve the best practices in an iterative process by collecting feedback and using the feedback to refine the best practices.

2.2 The Best Practices

These are oriented to people that are interested in designing chatbots, regardless of their programming skills. According to Shevat [29], "Bot builders have several tools at their disposal. Some require more technical skills than others while providing higher levels of flexibility." People that want to design a chatbot can choose between different tools. For those that do not have any programming skills, they can use tools like "Chatfuel' or "Manychat." For those that have deeper knowledge of programming,



Chatbot's Best Practices

1) Chatbot's Purpose

Designers should give a single purpose to the chatbot. So, the user should understand why the chatbot was created and have expectations accordingly. Therefore, the chatbot should fulfill those expectations

User-Chatbot Communication

2) Chatbot's Conversational Scenarios

Conversational scripts allow designers to have a draft outlining the situations and actions that the chatbot needs to support. Those scripts describes the limitations of the chatbot by capturing specific scenarios about the conversation

3) Chatbot's Interaction Elements

Chatbot may also contain but they are not limited to the follow interaction elements to enhance the conversion: Buttons, Links, Images and GIFs, Typing and Emojis.

4) Chatbot's Conversational Elements

Designers should be aware that a conversation with a chatbot may contain the following text based elements: Assertions, Questions, Support. Chabots may also support command expressions

5) Chatbot's Language

Chatbots should use the same language and words as the user (jargon). The use of technical and system-oriented terms should be according to the audience

6) Chatbot's Conversation Flows

Designers should think about the different possible flows that the user may go in, and the different possible flows that the chatbot may use.This means that after a particular information was given, the chatbot should perform a certain task.

7) Chatbot's Communication Limitations

Chatbots should have a limited script of the possible scenarios. If a user wants to go through an unknown topic for the chatbot, the chatbot should specify to the user that this question goes beyond its knowledge. Therefore, designers need to not only consider what the chatbot can do, but also what it cannot do, in order to set boundaries

Chatbot Features

8) Chatbot's Personality

A chatbot's personality is what makes it unique. By adding extra information to the dialogue and having a consistent language, the chatbot can have a distinctive personality. When designers are designing the chatbot's personality, they must consider the chatbot's audience and the task that the chatbot is trying to support

9) Chatbot's Help

Chatbots should provide help to users in a straightforward way. The help feature is a solution when the user get stuck in the conversation. When an error has occurred in the conversation the chatbot should ask users if they need any help.

10) Chatbot's Memory

Chatbots may be built with the ability to keep track of the user's interest and preferences, even if they change over time (situational knowledge). The user's habits and preferences can be remembered in the different sessions that he or she has had with the chatbot

11) Making Changes on the Fly

Chatbots should support the ability to undo or redo the responses that the user gave

12) Social Intelligence

Chatbots should have social and emotional intelligence. The chatbot should help users in an appropriate time and give relevant information to help them. Chatbot's designers should always keep in mind what the users' preferences are.

13) Chatbot's Documentation

A chatbot's documentation should include a concrete list of what steps can be carried out (Chatbot's Ineraction Scenarios) and a description of the chatbot's capabilities (Chatbot Purpose)

Human Factor Concerns

14) Conversational and Situational Knowledge

In order to have a support conversation the chatbot and the user should share conversational and situational knowledge. Conversational knowledge is when the user follow the meaning of the chatbot's remarks, while the chatbot should select a response that fits coherently with what the user is saying. Situational knowledge is when the chatbot should be situational-aware of the user's context.

15) User's Control

Chatbots should make users feel that they are in control of the conversation; as they are there to help users. Chatbots should be guides or an additional aid that allows users to understand a process and have better results.

16) User's input

Chatbots can get the user's input in several ways. Designers should provide shortcuts for the user where is appropriate. For example, provide a carrusel when the user need to look to different options or provide yes or not buttons for closed-ended questions

17) User's Recognition and Recall

Therefore, chatbots should keep their messages short and give the users a small number of options, in order to avoid getting them lost or feeling overwhelmed. This will allow to have a minimal design and a better user comprehension

18) Ethics: Data Security and Privacy

Chatbots' should be ethical. Thus designers should have a *code of conduct* for chatbots. Therefore, chatbots need to inform users of data retention processing privacy and what measures that chatbots' designers took in order to protect users' privacy and data.

Figure 2.4: 18 best practices for designing a chatbot effectively

they can use Wit.ai, IBM Watson or build the chatbot from scratch. The following best practices allow people to have a good design for creating a chatbot, regardless of whether or not they have a strong background as a programmer.

The following section describes 18 best practices for designing a chatbot effectively. The best practices and definitions are represented in Figure 2.4. Best Practice #1, Chatbot's Purpose, is one of the most important. Based on the purpose of the chatbot, designers can apply the other categories of the chatbot elements: User-Chatbot Communication, Chatbot Features, and Human Factor Concerns.

The **User-Chatbot communication** category groups all the interaction elements that users may have with a chatbot when they are exchanging information or ideas. The second category, **Chatbot Features**, groups the characteristics that a chatbot may have. The final categoty, **Human factor concerns**, groups the factors that are considered important by users when interacting with a chatbot.

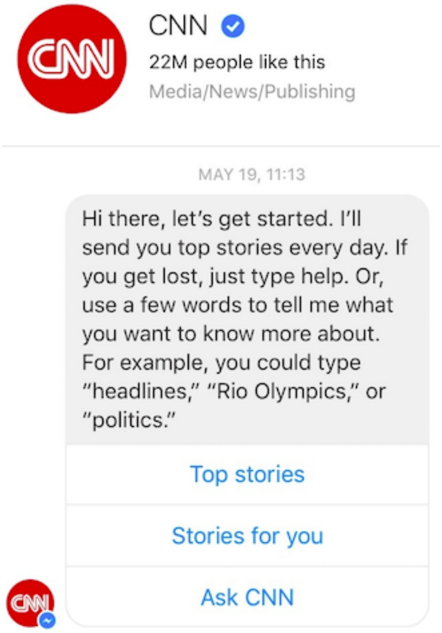
Category I: Chatbot's Purpose

2.2.1 Best Practice #1 Chatbot's Purpose

The purpose of a chatbot should be clear to its users. A chatbot that makes a user feel confused about its purpose is futile. **Designers should give a single purpose to the chatbot [2], however, virtual assistants such as Siri and Alexa may have more than one purpose. Users need to have a clear understanding of why the chatbot was designed and what its capabilities and limitations are.**[1]

Currently, chatbot designers can deal with this issue in two different ways. The first is to have a clear explanation about the chatbot's capabilities at the beginning of interactions with users. A clear example of this is the CNN chatbot (cf. Fig. 2.5). When the user opens the chatbot for the first time, the chatbot describes the top stories and explains how to get more news according to the user's preferences. The second option is to allow users to discover the purpose and capabilities of the chatbot by themselves during the interaction. However, an issue with this approach is that the user may feel confused and this may cause them to leave the conversation with the chatbot sooner.

[1] The text that is presented in bold as it is the description of the best practices that appears in the Chatbot's best practices diagram.



CNN

22M people like this

Media/News/Publishing

MAY 19, 11:13

Hi there, let's get started. I'll send you top stories every day. If you get lost, just type help. Or, use a few words to tell me what you want to know more about. For example, you could type "headlines," "Rio Olympics," or "politics."

Figure 2.5: Example of the Best Practice #1 Chatbot's Purpose: CNN chatbot

Chatbot designers need to identify a balance between the constraints of the interface and the user's goals. What's more, the kind of chatbot needed should be decided based on the final user requirements, and if it is feasible to address the user requirements or not. Only after making this analysis can one start designing the chatbot's capabilities and features (i.e., what the chatbot can do in order to help users in their tasks).

Understanding the chatbot's purpose helps the designer choose the right solution for the user, whether it is a chatbot or a direct manipulation interface (e.g., graphic user interface). According to Maes [31], "Understanding means that the agent-user collaboration can only be successful if the user can understand and trust the agent." Chatbots can be very useful as long as they are well-developed and oriented to the purpose.

Category II: User-Chatbot Communication

This category groups the communication elements in a session between a user and a chatbot when they are exchanging information or ideas.

2.2.2 Best Practice #2 Chatbot Conversational Scenarios

Chatbots should have a well-designed series of conversations with their users. Every single intended conversation should have a specific script of the different scenarios that the chatbot and the users may have.

Conversational scripts **allow users to have a "draft outlining the situation and actions the chatbot needs to support" [28]. Those scripts describe the limitation of the chatbot by capturing specific scenarios about the conversation.**

A conversational script is a structured development of a conversation [21] to engage users in the conversation, guide them through the interactions [27], and avoid references that are not clear to the participants. The conversation between the user and the chatbot should not lead to any ambiguity; it should be clear and concise.

One of the most important and complex parts of designing a chatbot is the creation of the conversational script because the user can take many paths (i.e., conversation scenarios) in order to achieve one task [5]. One effective way to create those different scenarios is using mind maps. According to Eppler [13], "Mind maps are defined as a multicoloured and image-centred radial diagram that represents semantic or other connections between portions of learned material hierarchically." Figure 2.6 shows a mind map regarding the different conversational scenarios of a chatbot that allow novice users to create visualizations. After creating the mind map, designers may use tools like InVision[2] to create the conversational chatbot mockups.

As in human conversation, chatbots and users can have similar parts of a conversation, such as greetings, conversation and farewell. The welcome or greeting section is the first interaction in which chatbots should clearly communicate their purpose in order to avoid functionality misunderstandings (cf. Fig. 2.5).

The welcome section of the conversation should be short and concise, to avoid users feeling overwhelmed due to the amount of information that may be provided by the chatbot. The first section on the top right of Figure 2.6 shows the *greeting conversational script* for a Visualization chatbot as an example. This section shows the possibles greetings that a user may utilize for starting the conversation with the chatbot.

The conversation section is when the central motive of the dialogue will be developed. According to Mishra [20], "Effective communication is getting the message

[2] www.invisionapp.com

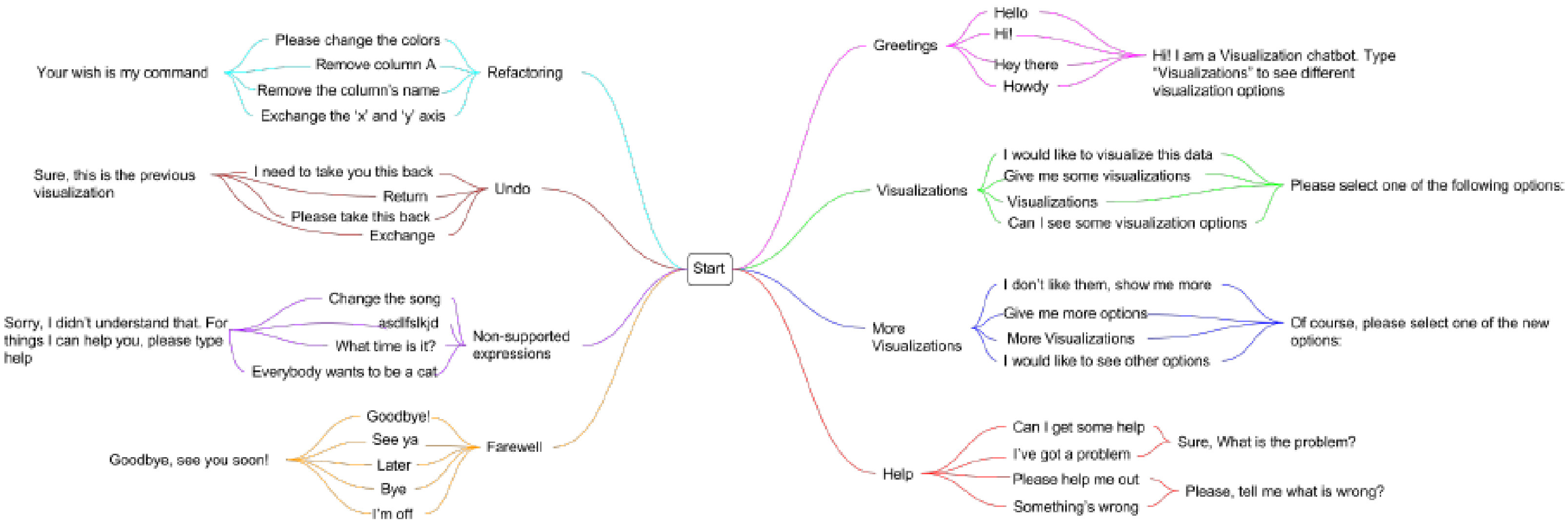

Start
Refactoring
Please change the colors
Remove column A
Remove the column's name
Exchange the 'x' and 'y' axis
Your wish is my command
Undo
I need to take you this back
Return
Please take this back
Exchange
Sure, this is the previous visualization
Non-supported expressions
Change the song
asdlfslkjd
What time is it?
Everybody wants to be a cat
Sorry, I didn't understand that. For things I can help you, please type help
Farewell
Goodbye!
See ya
Later
Bye
I'm off
Goodbye, see you soon!
Greetings
Hello
Hi!
Hey there
Howdy
Hi! I am a Visualization chatbot. Type "Visualizations" to see different visualization options
Visualizations
I would like to visualize this data
Give me some visualizations
Visualizations
Can I see some visualization options
Please select one of the following options:
More Visualizations
I don't like them, show me more
Give me more options
More Visualizations
I would like to see other options
Of course, please select one of the new options:
Help
Can I get some help
I've got a problem
Please help me out
Something's wrong
Sure, What is the problem?
Please, tell me what is wrong?

Figure 2.6: Mind map that I designed of the different conversational scenarios of a chatbot that helps to create visualizations

across as intended and getting desired feedback by influencing and attracting attention". The chatbot's conversation "should provide information the user wants to know" [38]. This section can have assertions, questions, comments, and feedback. The participants of the conversation should take turns in order to communicate a clear message. For example, a conversation with a visualization chatbot may have the following sections: *Visualizations, More visualizations, Help, Refactoring, Undo, and Non-supported expressions* (cf. Fig. 2.6).

The farewell section is the last part of a conversation. By this phase, the user should have had the help they needed to finish their task. In this section, the chatbot should ask the user if the provided information or help was useful, and if the user would like to do anything else. In addition, designers can add a chatbot survey in order to get feedback about the performance of the chatbot. The last section of Figure 2.6 shows the *farewell conversational script* for a visualization chatbot.

2.2.3 Best Practice #3 Chatbot Interaction Elements

There are several interaction elements that a chatbot may use in a conversation. The use of them should be tied closely to the purpose and personality of the chatbot to improve the flow of a conversation and improve user engagement. On the other hand, the user may have different elements to provide information to the chatbot, those elements are being described in best practice #16 User's Input.

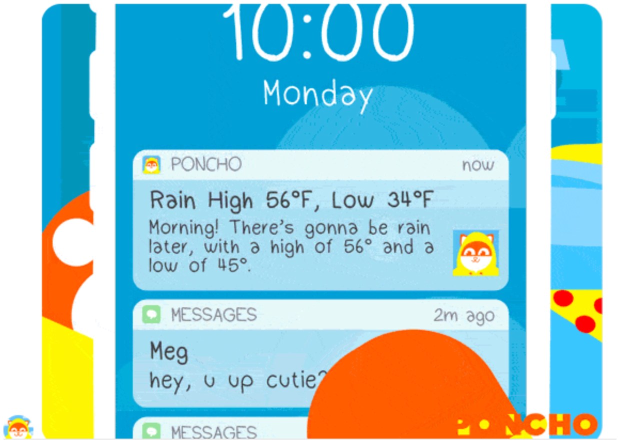
In the following list I present and describe the chatbot interaction elements and how they should be used. However, chatbots interaction elements are varied and not limited to the following list:

- Text integration: Script-based chatbots use text to express the information they want to provide to the user. Designers need to be careful of the amount of text they provide to the user in a single text box (best practice #17 User's Recognition and Recall). According to Amunwa [7], "text messaging offers users a simple, frictionless interface thats easily understandable in any culture, and is available to everyone worldwide."

- Typing: This is a very important element because it allows users to understand that more information is coming, and that they should not add a new input until the chatbot finishes its message. The typing time should be proportional to the number of words that a user needs to read. In this element, timing is crucial: if the period of time is too short, users may skip reading the first message and jump to the following message, causing misunderstandings. If the period of time is too long, users may feel bored from waiting for the message to come.

- Links: These allow designers to provide extra information that has not been included in the chatbot's conversational scenarios. Links can be used to direct the user to learn more about a product, offer them a webpage to provide feedback, or allow the user to convey insights that the chatbot does not provide.

- Images and GIFs: An image is a visual representation of an idea, and it is a common practice to use them to convey complex information. According to Shevat [29], "Images are useful as some information is much easier to relay in an image than in words." Graphical Interchange Format files or GIFs are images animated in a single file. They are a great way to engage people into a conversation because they are usually used to make jokes or sarcastic remarks about what is being said. Designers can create a GIF to explain how to interact with the chatbot and use it to educate people about the chatbot features. An example can be seen in the Poncho chatbot's greeting section (cf. Fig. 2.7): Poncho has a GIF at the beginning of the conversation to explain how to use the chatbot.



Rain High 56°F, Low 34°F
Morning! There's gonna be rain later, with a high of 56° and a low of 45°.

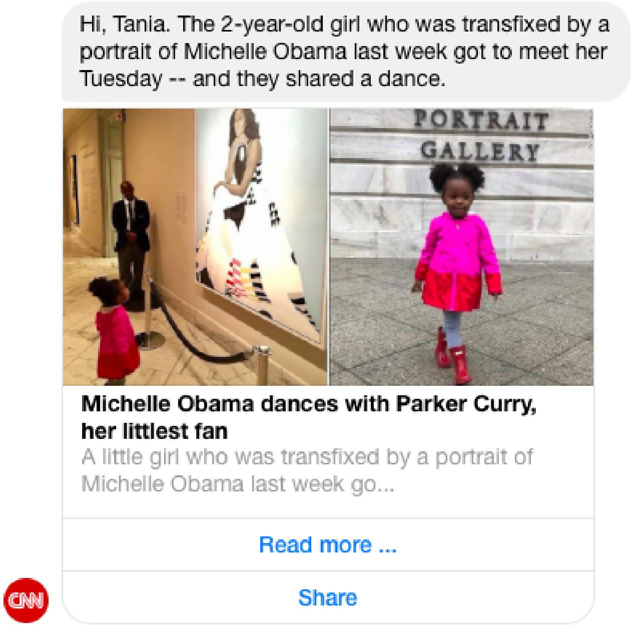
Meg
hey, u up cutie?

Figure 2.7: Example of the Best Practice #3 Chatbot's Interaction Elements: Poncho chatbot's greeting section with GIF's

- Emojis: This element allows the chatbots to show its "emotions" about the conversation. If the chatbot does not understand the user input, a sad emoji will allow designers to convey the feeling of "I'm sorry" to the user. Emojis also allows the user to feel that the chatbot "speaks" in the same way as them. This is particularly useful for younger audiences.

- Carousel: When the designer needs to give the user different options to select, the use of a carousel is the most efficient way. According to Pernice [23], "Carousels enable more than one piece of content to occupy the same piece of prime real estate on the homepage, which can help diffuse any infighting about whose content is most deserving" and it should not be included more than five frames in each carousel because it is hard for users to remember more than five topics at the same time.

- Buttons: According to Lurchenko [16], "Buttons give people choices and direct them toward certain actions." Thus, buttons allow designers to create a specific set of options from which the user can choose. Buttons are especially useful when the conversation requires certain input from the user. A good example of this is when the CNN chatbot ask the user to select if they prefer to "Read more..." or "Share" (cf. Fig. 2.8). The user is given two options: to read more about the selected news or to share them with others.



Hi, Tania. The 2-year-old girl who was transfixed by a portrait of Michelle Obama last week got to meet her Tuesday -- and they shared a dance.

Michelle Obama dances with Parker Curry, her littlest fan

A little girl who was transfixed by a portrait of Michelle Obama last week go...

Figure 2.8: Example of the Best Practice #3 Chatbot's Interaction Elements: CNN chatbot using buttons

2.2.4 Best Practice #4 Chatbot Conversational Elements

According to Allen et al. [6], Conversation can be defined as the "process occurring in real time, and constitutes a reciprocal and rhythmic interchange of verbal emissions. It is a sharing process which develops a common social experience." The authors also claim that **the speaker can emit several elements, such as assertions, and questions in a conversation.** These conversational elements can be emitted by humans and chatbots when they are interacting with each other. The following interaction text-based elements are described from the chatbot's perspective:

- Assertions are affirmations or messages that the chatbot gives to the user; assertions determine the content of the conversation and allow the user to increase their knowledge about a certain topic.

- Questions can be used by the chatbot to make sure the provided information is required by the user. According to Stiver [34], questions should be designed according to the type of answer the speaker wants to get. For example, if the speaker wants a yes or no response or to choose between options. An example of a question is when Poncho,[3] a weather forecasts chatbot, asks users about which city weather they are interested in viewing. If the user would like to know the weather in Manhattan, Poncho will ask the following verification question: "New York, NY? Right now it's mostly cloudy there. Is that the right city?" After the user clicks "Yea", Poncho will give the complete forecast (cf. Fig. 2.9). According to Shevat [29], a chatbot may ask to confirm its correctness before carrying out a process. There are two types of confirmation that bots may use: explicit and implicit. Explicit is when the chatbot asks about the correctness of an input provided by the user. They are also used to allow the chatbot to ask the user permission to carry out a task (process). Implicit confirmation is when the chatbot confirms to the user that it received the user's input. However, designers should avoid open-ended questions, because users may respond in a non-supported way. A solution for this may be the use of "yes" or "no" buttons or quick replies. Quick replies are questions that provide different options in buttons inside the conversation, and this type of question allows users to select just one option. Each provided option will invoke different actions. If the options provided by the chatbot contain several images, a good way to display them is the use of carousels [23] (see Best Practice Chatbot Interaction Elements).

Figure 2.9 shows an example of a closed-ended question in which the chatbot asks the user to select one option.

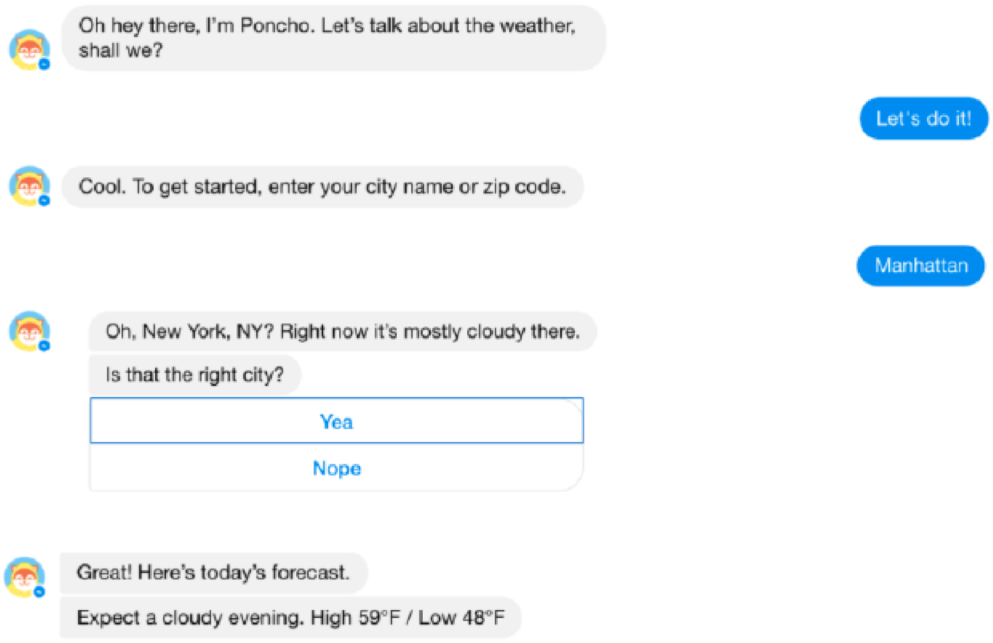
2.2.5 Best Practice #5 Chatbot's Language

According to Allen et al. [6], in order to have a meaningful conversation, it is necessary to share a language and have vocabulary in common. **Chatbots should use the same language and words as the user (jargon). The use of technical and system-oriented terms should be tailored to the audience.**

If the design of the conversational agent is oriented towards certain populations, chatbots can mimic how these users normally speak. Therefore, before the creation of the chatbot, it is necessary to have a "solid understanding of the audience we seek to appeal to, and to have a vocabulary familiar to them" [27]. Figure 2.10 shows a conversation between a user and Iris[4], the data science chatbot. Iris is oriented to a

[3] https://poncho.is/

[4] https://github.com/Ejhfast/iris-agent/



Oh hey there, I'm Poncho. Let's talk about the weather, shall we?

Let's do it!

Cool. To get started, enter your city name or zip code.

Manhattan

Oh, New York, NY? Right now it's mostly cloudy there.

Is that the right city?

Great! Here's today's forecast.

Expect a cloudy evening. High 59°F / Low 48°F

Figure 2.9: Example of the Best Practice #4 Chatbot's Conversational Elements: Poncho chatbot using a question

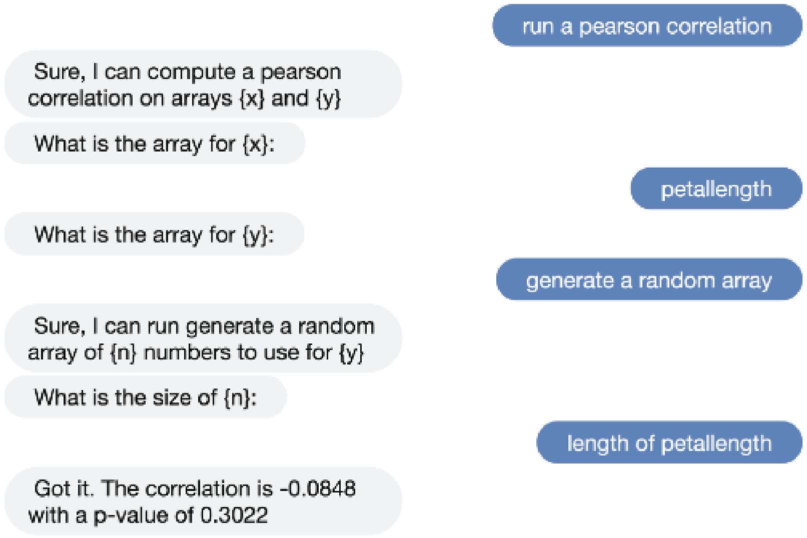
scientific audience, therefore the language that Iris uses is similar to their users.

Designers need to keep in mind the kind of language the chatbot will understand, this may be Natural language or command expressions. According to the chatbot's user, the designer needs to select whether or not use commands expressions besides the Natural language interaction. If the chatbot's users are developers or scientists, among others, designers may add command expressions such as "*/show.*" However, if users are not familiar with them, designers should avoid include those command expressions.

2.2.6 Best Practice #6 Chatbot's Conversation Flows

According to Reichman [25], in a usual conversation many details are shared. To avoid an incoherent conversational flow, it is necessary that individuals understand when and why the conversational topic has been changed. In accordance with Angara et al. [8], "Users may want to change something in the input or switch topics in the middle of a conversation."

The conversational chatbot flow should follow a certain order. This means that after a particular piece of information is given, the chatbot



run a pearson correlation

Sure, I can compute a pearson correlation on arrays {x} and {y}

What is the array for {x}:

petallength

What is the array for {y}:

generate a random array

Sure, I can run generate a random array of {n} numbers to use for {y}

What is the size of {n}:

length of petallength

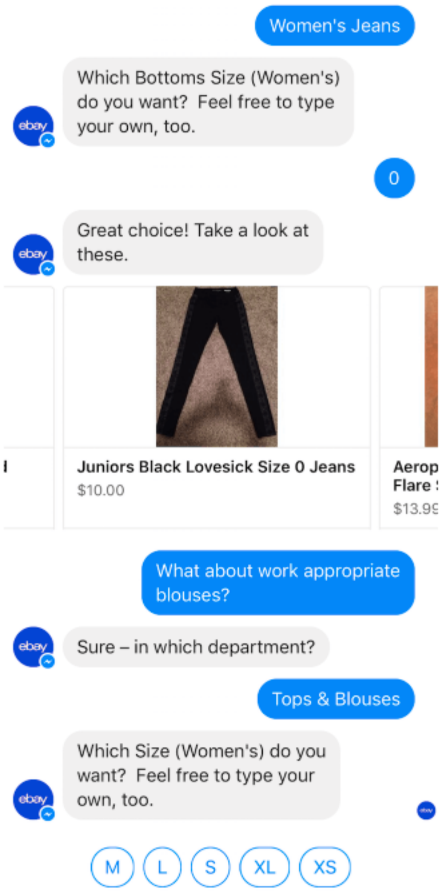
Got it. The correlation is -0.0848 with a p-value of 0.3022

Figure 2.10: Example of the Best Practice #5 Chatbot's Language: Iris Data Science Chatbot using specific language for its scientific audience

should perform a certain task. Designers should think about the different possible conversational paths the user and chatbot may take. In addition, designers should provide a way to escape from an erroneous conversational flow. Users and chatbots can change the topic of the conversation at any point. Any shift in the topic should be made carefully to avoid misunderstandings. For example, eBay's ShopBot[5] helps users selecting a pair of jeans, and after that, the user can also ask for a blouse from a different department (cf. Fig. 2.11). It is important to understand that the purpose of the conversation did not change: the purpose remains the same, that is, buying clothes. However, the flow of the conversation switches from buying jeans in one department to buying blouses from a different department. Chatbots may also change the flow of a conversation, e.g., Rescue.io (cf. Fig. 2.12). This chatbot was created to help in emergency situations. According to Clegg [10], "during an emergency when the user cannot phone someone for help, Rescue can quickly alert the user's emergency contacts of what is happening and where the user is so they can help". Rescue leads the conversational flow by asking a series of questions about the type of emergency the user has.

The flow of the conversation should be natural and evident, allowing the user to change the flow when needed.

[5] https://shopbot.ebay.com/



Women's Jeans

Which Bottoms Size (Women's) do you want? Feel free to type your own, too.

0

Great choice! Take a look at these.

Juniors Black Lovesick Size 0 Jeans
$10.00

Aerop
Flare
$13.9

What about work appropriate blouses?

Sure – in which department?

Tops & Blouses

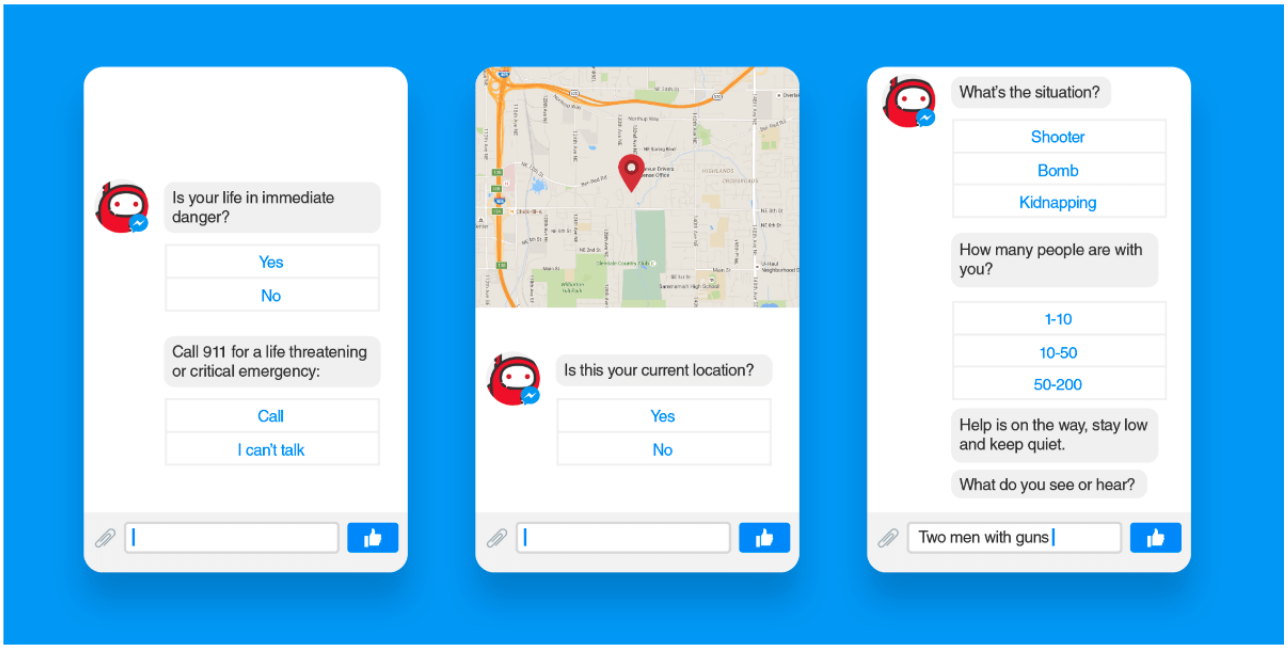
Which Size (Women's) do you want? Feel free to type your own, too.

Figure 2.11: Example of the Best Practice #6 Chatbot's Conversation Flows: ShopBot Ebay managing a conversation flow change

2.2.7 Best Practice #7 Chatbot's Communication Limitations

Chatbots should have a limited script of the possible scenarios (cf. Fig. 2.6). If a user wants to go through an unknown topic for the chatbot, it is expected to notify the user that this question goes beyond its knowledge. To set boundaries designers need to not only consider what the chatbot can do, but also what it cannot do. This problem is addressed by using a default answer. A default answer is the message that the chatbot will display when the conversation does not follow the defined flow. In the default answer, the chatbot needs to acknowledge that the user's input goes beyond its knowledge and if it is appropriated to offer the user to contact a person.

An example of this is when a user asks Poncho, the weather chatbot, something



Is your life in immediate danger?

Call 911 for a life threatening or critical emergency:

Is this your current location?

What's the situation?

How many people are with you?

Help is on the way, stay low and keep quiet.

What do you see or hear?

Two men with guns

Figure 2.12: Example of the Best Practice #6 Chatbot's Conversation Flows: Rescue.io chatbot leading the conversation flow

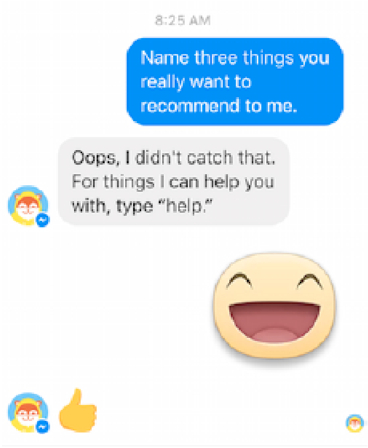
that it was not created to answer, Poncho replies with: "Oops, I didn't catch that. For things I can help you with, type *help*" [27] (cf. Fig. 2.13). Chatbots need to gracefully establish limits to users that go beyond the chatbot's knowledge.

Category III: Chatbot Features

2.2.8 Best Practice #8 Chatbot's Personality

Personality is defined as the combination of characteristics or qualities that form an individual's distinctive character [3]. **A chatbot's personality is what makes a chatbot unique. By adding extra information to the dialogue and having a matching language, the chatbot can have a distinctive personality.** However, designers should be careful not to include too much additional information to the point it makes the user feel bored or annoyed. There has to be a balance between giving just the relevant information and adding extra information (e.g., chitchat) to make the chatbot's conversation more engaging. If designed correctly, the chatbot personality is an important part to make the user more engaged with the interface.

The chatbot's personality should take into account the target audience for which



8:25 AM

Name three things you really want to recommend to me.

Oops, I didn't catch that. For things I can help you with, type "help."

Figure 2.13: Example of Best Practice #7 Chatbot's Communication Limitations: Poncho chatbot acknowledging its limits

it was designed. The personality should match the users, therefore, how friendly, sarcastic or humorous the chatbot is should depend on the people who will use the chatbot. **It is very important that the design of the personality considers the chatbot audience and the purpose the chatbot is trying to support.**

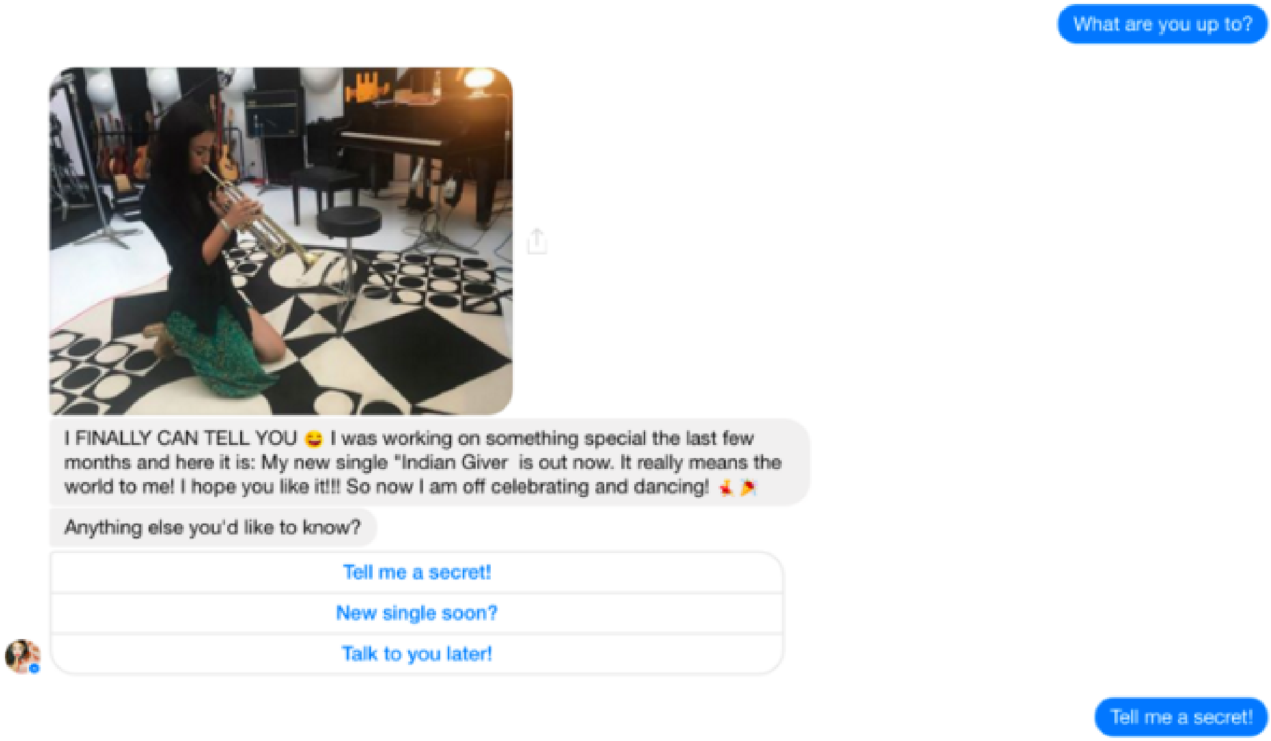
Users like to interact with chatbots in a human way: in other words, users like to have the feeling of being in communication with a person rather than a chatbot [27]. For this reason designers should consider strategies to engage users in the conversation.

A clear example of how to get user engagement is Aura's bot.[6] Aura Dione is a Danish singer who is using a chatbot to be in contact with her fans. The chatbot's designer uses the personality of the singer to provide a specific personality to the chatbot. (An example of Aura's bot can be seen in Fig. 2.14.) Therefore, the tone of the chatbot is like an artist, and fans can communicate with it and ask extra information about the singer. Fans have the feeling of interacting with the singer, and this leads to user engagement and retention [33].

According to Scott [27], there is a difference between the content (*i.e., relevant information to help the user*) and the medium (*i.e., the chatbots personality*). Users need to be entertained and helped.

Another important consideration to take into account is to design the chatbot with an specific persona in mind such that the user gets the feeling the chatbot has an age, heritage, friends and occupation. All these specifications enable designers to

[6] http://m.me/aurapower



What are you up to?

I FINALLY CAN TELL YOU 😆 I was working on something special the last few months and here it is: My new single "Indian Giver is out now. It really means the world to me! I hope you like it!!! So now I am off celebrating and dancing! 💃🎉

Anything else you'd like to know?

Tell me a secret!

Figure 2.14: Example of the Best Practice #8 Chatbot's Personality: Aura's Bot and its personality of the singer Aura Dione

have a clear definition of whom the chatbot will resemble.

To summarize, the chatbot's success may be defined by how designers balance between entertainment and guidance. Therefore, the difference between chatbots that are being used and those that are not, is how compelling and pleasant the chatbot experience is.

2.2.9 Best Practice #9 Chatbot's Help

The help provided by the chatbot should be straightforward for users and it should give a possible solution when the user gets stuck in the conversation. Users should have a specific script about how to use the chatbot. Commonly, users only need to type the word "*help*" or the command "*/help*" in order to get additional support from the chatbot. When an error has occurred, the chatbot should automatically ask users if they need any help. In addition, some chatbots have the feature of "what to ask?" This allows the user to ask the chatbot about its conversational topics (capabilities), and after selecting that option, the chatbot will display a list of the possible topics the user may ask about. This feature allow users to understand what can be asked in the conversation and allow designers to educate people about how to use the chatbot.

In addition to regular human conversational elements, **chatbots may also understand command expressions.** These statements are commands that users write to interact with the chatbot using domain-specific knowledge.

How the help option is requested by users, depends on the kind of expressions that the designer specified in the best practice #5 Chatbot's Language. If designers allowed the use of command expressions, users can use expressions like */help*, if not, users may only use natural language expressions such as "tell me what to ask?"

2.2.10 Best Practice #10 Chatbot's Memory

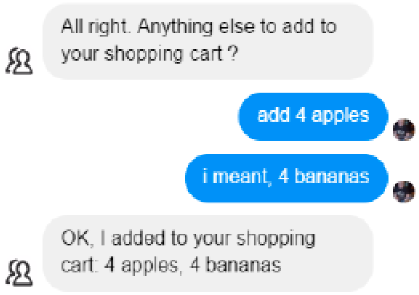
Chatbots should be built with the ability to keep track of the user's interests and preferences, even if they change over time *(situational knowledge).* Conversational agents can save the user's habits and preferences in the different sessions that the user has with the chatbot [31]. As an example of this, Poncho can save the user's current location. So that next time the user asks for a weather broadcast, Poncho already knows which location to look at, saving the user time and keystrokes. These actions allow the user to have a customised experience.

In order to make the chatbot more efficient, designers should save information that is relevant to the user's context. According to Abowd et al. [4], "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves."

Designers can get the user's context from the information users are sharing in the conversation with the chatbot. Additionally, designers can also use sentiment analysis to get the user's context. According to Kamath [15], "Sentiment analysis applies natural language processing techniques and computational linguistics to extract information about sentiments expressed by authors and readers about a particular subject."

2.2.11 Best Practice #11 Making Changes on the Fly

It is normal that users make mistakes when they are choosing between different options in an interface. **It is very important that chatbots have the ability to**



All right. Anything else to add to your shopping cart ?

add 4 apples

i meant, 4 bananas

OK, I added to your shopping cart: 4 apples, 4 bananas

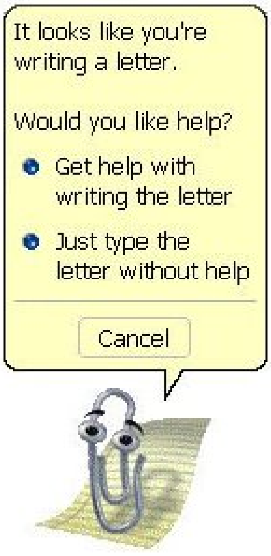
Figure 2.15: Example of failing the Beast Practice #11 Making Changes on the Fly: A chatbot without the undo capability

undo or redo user's responses. In that case, users should know that they are able to undo an unwanted change triggered by a misinterpreted message or a mistaken click [27]. Figure 2.15 shows an example of a chatbot without the undo capability. In this example, the user made a mistake while adding items to the shopping cart. Because the chatbot does not have the ability to undo, the user has to either, start the conversation from the beginning or buy all the items even if they are not needed. This situation could make users feel annoyed or frustrated.

During a conversation, chatbots should provide hints when a user makes a mistake. In the example presented in Figure 2.15, when the user typed "I meant" it is a clear indication that the user made a mistake. Therefore, words as "meant" gives designers a clue of when to use the undo capability.

Another approach is when the chatbot directly asks users if the information provided is accurate. The Poncho chatbot has the ability to ask users if the information provided was appropriate; if not, users can clarify it.

Chatbots should express error messages in a straightforward way. According to Armstrong [9], chatbots should be able to express that they did not understand and ask the user to repeat their message in a different way. If necessary the chatbot should give different answers when it did not understand the user in order to avoid the user having to deal with the same message over and over again. After several error messages in a row, the chatbot should allow the user to either report the problem or provide feedback.



It looks like you're writing a letter.

Would you like help?

● Get help with writing the letter
● Just type the letter without help
Cancel

Figure 2.16: Example of the Best Practice #12 Social Intelligence: Microsoft Office Assistant Clippy and its random interruptions

2.2.12 Best Practice #12 Social Intelligence

Another relevant challenge that designers need to keep in mind is **giving the chatbot social intelligence. Therefore, the chatbot should be sensitive to the user's preferences.** According to Siegel et al. [32], "Social intelligence can be defined as skills that enable an individual to understand the impact of emotions on behavior and thinking, to regulate emotions and behavior, to understand the importance of emotions in others, and to understand social interactions and engage in adaptive ways with others in social situations."

Chatbots should be helpful without being annoying. An example of an annoying chatbot is the Microsoft Office Assistant "Clippy the Paperclip" (cf. Fig. 2.16). Some of the most well-known of Clippy's problems were the random interruptions to users asking them if they need help writing a letter, moving and winking at odd intervals, and distracting users [19]. According to Shneiderman [30], "The anthropomorphic styles are cute the first time, silly the second time, and an annoying distraction the third time."

Another relevant definition to this best practice is Social context. According to Villegas et al. [36], "Social context emerges from the interrelation among individual human and group context. Samples of this relational context are affiliations, colleagues, and customers."

Chatbots help users by “chatting” with them and being guides for certain tasks that the user needs to perform. Chatbots that have the ability to save user preferences should be careful not to share personal information with other users. An example of this is Xiaoice, a chatbot developed by Microsoft. Xiaoice is an artificially intelligent software program designed to chat with people. This chatbot can be used for individual and group conversations and there is evidence that Xiaoice has revealed personal information gathered during one-to-one sessions in group conversations [26]. In best practice #18 there is further description of how chatbots should manage user’s data and privacy concerns.

In summary, designers should keep in mind that the chatbot should help users at an appropriate time and give relevant information to help them, without compromising sensitive information.

2.2.13 Best Practice #13 Chatbot’s Documentation

Chatbots should be able to be used without the need for a manual. How to use a chatbot should be a simple conversational flow. However, there are some users that like to deeply understand how the interface works and detect its limitations. Also, it should include a concrete list of what steps can be carried out and the critical points of the chatbot [27]. Besides, the chatbot should have a concrete list of the conversational scenarios that are included.

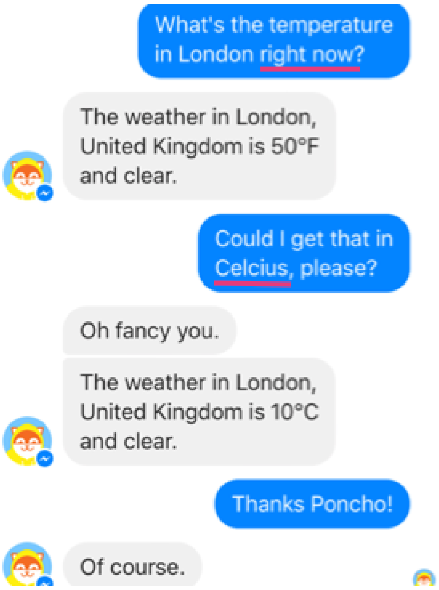
Finally, the chatbot should provide an easy way to access the documentation. One example of this is when the user types slash “/” and a list of the supported commands of the chatbot appears on the screen [1].

Category IV: Human Factor Concerns

2.2.14 Best Practice #14 Conversational and Situational Knowledge

Conversations need to follow certain rules and to have a semantic order for the sake of having a clear understanding between the user and the chatbot. Therefore, in order to have a good conversation, the chatbot and the user should share conversational and situational knowledge.

***Conversational Knowledge* is when the user is able to follow the mean-**



What's the temperature in London right now?

The weather in London, United Kingdom is 50°F and clear.

Could I get that in Celcius, please?

Oh fancy you.

The weather in London, United Kingdom is 10°C and clear.

Thanks Poncho!

Of course.

Figure 2.17: Example of the Best Practice #14 Conversational and Situational Knowledge: Poncho chatbot replying to a request to present information in a different format

ing of the chatbot's remarks, while the chatbot is able to select a response that fits coherently into what the user is saying. Therefore, it is vital to understand how chatbots and users follow each others' thoughts. Conversational Knowledge is the elements in the conversation that do not change over time.

Figure 2.17 shows the user asking Poncho to provide the temperature in Celsius rather than Fahrenheit degrees: in this example, the temperature units are the shared conversational knowledge. The user request makes sense in the conversational context, and the chatbot is able to make the temperature conversion.

Situational Knowledge is another important part of having a mutual understanding in a conversation. According to Reichman [25], people need to share Situational Knowledge in order to understand and follow a conversation because it involves previous knowledge about the topic that is being discussed.

In a user-chatbot conversation, Situational Knowledge **is when the chatbot is situation-aware regarding the user context.** Therefore, Situational Knowledge change over time according to the user's context. Figure 2.17 shows a user asking "What is the temperature in London right now?" Giving that Poncho and the user share the same situational knowledge about the current time, the provided forecast will be according to the time when the user asked the information.

2.2.15 Best Practice #15 User's Control

Users are accustomed to and like having control over the software. For instance, when users have the ability of doing more customizations in an interface, they tend to feel more engaged in using the interface because it considers their preferences. Therefore, it is extremely important to make users feel that they have control over the chatbot interface. According to Maes [31], "users must be able to turn over control of task to conversational agents but users must never feel out of control."

Chatbots should not make users believe that they are powerless over the interface; chatbots are there to help users have a better experience using the interface. Chatbots should help users to understand the interface and have better results. According to Shneiderman [31], it is "necessary to give the users the feeling of being in control and therefore they can be responsible for the decisions they make."

2.2.16 Best Practice #16 User's Input

The input of this best practice came from the user's perspective. **There are several ways that the chatbot may receive the input from a user, by clicking an image in a carousel, clicking buttons or by directly typing a message in a text box, among others.** And designers need to have a clear understanding about when is a better option for users to click rather than type.

When chatbots need to ask a closed-ended question, providing a yes or no button facilitates the user to answer a question. When users need to type an open-ended answer, designers need to keep in mind that users will not type a large amount of text, especially those that are using the chatbot through a mobile phone. Therefore, all the questions should be intended on having a short answer; it is also difficult for the chatbot to understand the user's intent from a long paragraph.

2.2.17 Best Practice #17 User's Recognition and Recall

According to Scott [27], **users do not like to read a large amount of text: *"they will read the first message and then their eyes glaze over. They skim the rest of the text*.** The principle of least effort set forth by Zipf, also specifies that people use shortened words and expression in speech to obtain the maximum communication by using the least cost [40]. **Therefore, chatbots should keep**

their messages short and give the users a small number of options, in order to avoid users getting lost or feeling overwhelmed. Short messages and small number of options will allow to have a minimal design and a better user comprehension.

In addition, users do not remember details of the options given by an interface. When users avoid reading a large amount of text, they may misunderstand the chatbots message and finish with an unsuccessful result. The design of the chatbot dialogue should only have relevant information to help users with their tasks and prevent a dull interaction with the chatbot.

2.2.18 Best Practice #18 Ethics: Data Security and Privacy

Technology has reached to a point in which we should beware of the type of information we are sharing with an interface. Chatbots are not different; users need to be careful about the data they share with a chatbot. **Chatbot's designer must provide the chatbot with a *code of conduct*. Chatbots need to tell users "how their data is processed in the background when talking to them"** [22] **and what measures the designers took to protect users' privacy and data.**

Users need to understand that when they are interacting with a chatbot, it is not a private and personal conversation, even though they may have expectations of privacy. Interactions are being recorded on servers, data is being saved into a database, and conversation records are being used to understand the user and save their preferences. All this information is being gathered in order to improve the performance of the chatbot. According to Ondrisek [22], "All transactions are logged by the platforms servers, which monitor and log the communication between the user and the bot, therefore the bot's platform provider has complete access to the unencrypted conversation." Here is an example of how some chatbots are dealing with these privacy concerns. According to Yao Baogang [18], regarding Xiaoice, the chatbot that provides entertainment and companionship, "We don't keep track of user conversations with Xiaoice, we need to know the question, so we store it, but then we delete it. We don't keep any of the data. We have a company policy to delete the user data."

Designers should treat user's data ethically, and inform the user if their information is being shared with third parties.

Chapter 3

Applying Best Practices in the Design of a Chatbot

This chapter describes how I used the best practices from Chapter 2 to design a chatbot to support the Shift customer service. Shift is a web application created by Redbrick, a technology company in Victoria, BC.

According to Cui et al. [12], "Customer service plays an important role in an organization's ability to generate income and revenue. It is often the most resource-intensive department within a company, consuming millions of dollars a year to change the entire perception customers hold."

Customer support is not a simple task. Staff need to have accurate knowledge about the information given to the users, and at the same time be professional and friendly. Even when customers are giving negative feedback, staff need to behave appropriately and try to deal with the complaints in a professional and friendly manner.

Another important aspect of customer service that is hard to accomplish is providing service 24 hours a day, seven days a week. This is difficult to achieve because employees are typically hired for an eight-hour schedule. International companies also have to deal with customers in different time zones, making the customer service process slow and expensive.

Redbrick, the software company for which the chatbot was designed, was founded in 2011, and develops software for consumers and businesses. It has four main products; Shift, Assembly, Bedface and Deskmetrics.

Shift is a desktop email client application that has been on the market for 16 months. Shift enables users to improve their email productivity and combine their

GMail, Outlook and Microsoft 365 accounts in one application. In addition, Shift also allows extensions and apps in the same environment. Therefore, users can be more efficient by switching between emails, apps and extensions, creating an organized work environment.

Shift users can get help, give feedback or report a bug by submitting a HappyFox ticket. HappyFox is a support tool that allows companies like Redbrick to manage their customer support. Users need to wait until a Shift representative replies to the ticket, and usually the business hours are from 9:00 to 17:00. Therefore, a user who sends a message at night or in a different time zone, needs to wait until a staff member replies.

A great solution for all the challenges associated with customer service are chatbots. Chatbots can deal with repetitive tasks such as answering questions and providing customers a 24/7 service. Companies like Redbrick can use fewer staffing resources by adding a chatbot to their customer support. Staff will manage only the tasks that the chatbot is not capable of dealing with, making their jobs more efficient and less time-consuming.

One example of a chatbot used for customer support is Super Agent [12]. Super Agent replies to customer questions based on a questions and answers engine that contains the information regarding the company's product.

The ultimate goal for a customer support chatbot is to help users get the assistance they need by answering customer questions in the most accurate way. Another reason to create a chatbot for customer support is the strong desire that the Shift team has to be innovative and adopt new ways to interact with customers.

3.1 Shiftbot Customer support

3.1.1 Shiftbot Goal

Shiftbot is the chatbot I designed and implemented to be an extension for Shift's customer support. Shiftbot provides users 24/7 customer support: any time a user has a question, feedback, or would like to report a bug, they can do it using Shiftbot. The chatbot can provide the existent canned answers for supporting Shift. Shiftbot answers all the repetitive and simple questions that the users have. In addition, the chatbot can give a full description of the different features of Shift. In case the chatbot is not able to answer a question, the user is given the option of submitting a ticket, and then a Shift representative contacts the user and gives them the relevant answer.

3.2 How to use Best Practices During the Design of a Chatbot

The process used to create the text based chatbot Shiftbot is displayed in Figure 3.1. The white boxes represent one step of the process and a thick arrow is used to represent the output. The top of the figure represents the steps of phase III, and the bottom of the figure displays the output from the steps.

First, it was necessary to create a user persona to have a clear idea of the person that will use the chatbot. For this, I conducted an analysis of demographics and interviewed the Shift marketing coordinator.

Once the user persona was defined, I started the design process. In the second step, I designed the chatbot with the features the user persona would need in the chatbot and I matched the personality of the chatbot to the personality of the possible users to make it more appealing and easy to use. In order to have an efficient chatbot design process, the 18 best practices from Chapter 2 were used to help design the chatbot.

The third step is the chatbot implementation. The tool used to create Shiftbot was Chatfuel.

The fourth step was designed to get feedback on the chatbot. After several tests with users, I used the log files to identify gaps in the chatbot performance by having an iterative reviewing process. Here the purpose was to address all the issues that were not contemplated in the chatbot scenarios. After this review, I updated the chatbot according to the new design specifications, and the users re-tested the chatbot.

The best practices were revised and updated three times in the course of this study: after step two, three, and four.

3.3 Best Practices Applied to the Design of a Chatbot

The methodology followed in this chapter was created to the design of a customer support chatbot based on the best practices in Chapter 2. The best practices were

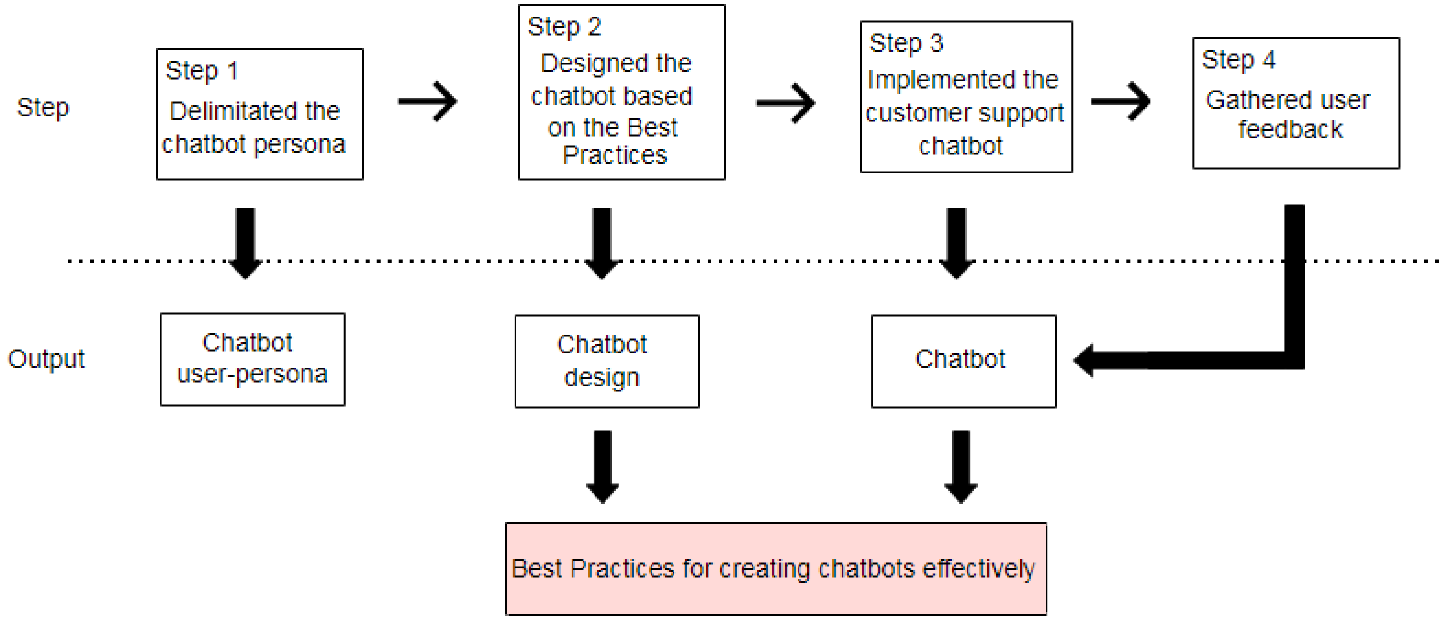

Step
Step 1
Delimitated the chatbot persona
Step 2
Designed the chatbot based on the Best Practices
Step 3
Implemented the customer support chatbot
Step 4
Gathered user feedback
Output
Chatbot user-persona
Chatbot design
Chatbot
Best Practices for creating chatbots effectively

Figure 3.1: Methodology of Phase III: These are the steps carried out to implement the customer support chatbot. In addition, this methodology helped to improve the best practices.

used and considered in a particular sequence to design the Shift customer support chatbot. However, this does not mean that the sequence is the same for all chatbot design. A good chatbot may be designed using all the best practices, but the sequence of how they are used in the design phase may be different.

All the figures of this section were taken from the platform Chatfuel. Thus, all the examples are from the designer's perspective and they are not the final user's view. For this reason, in some of them there are the add button option and the typing element displayed as a tool in the development platform.

For the design and creation of Shiftbot, the best practices were used in the following sequence. Please notice that some of the best practices overlap in the design sequence:

- Best Practice #1 Chatbot's Purpose
- Best Practice #6 Chatbot's Conversation Flows
- Best Practice #2 Chatbot's Conversational Scenarios
- Best Practice #3 Chatbot's Interaction Elements

- Best Practice #4 Chatbot's Conversational Elements
- Best Practice #7 Chatbot's Communication Limitations
- Best Practice #12 Social Intelligence
- Best Practice #14 Conversational and Situational Knowledge
- Best Practice #10 Chatbot's Memory
- Best Practice #18 Ethics: data security and privacy
- Best Practice #15 User's Control
- Best Practice #8 Chatbot's Personality
- Best Practice #5 Chatbot's Language
- Best Practice #9 Chatbot's Help
- Best Practice #16 User's Input
- Best Practice #11 Making Changes on the Fly
- Best Practice #17 User's Recognition and Recall
- Best Practice #13 Chatbot's Documentation

3.3.1 Best Practice #1 Chatbot's Purpose

The goal of Shiftbot is to augment Redbrick's customer support process. Customers can use natural language statements in order to communicate with the chatbot and the chatbot will provide them with the most relevant answers to their requests.

An early meeting with Redbrick's Marketing Coordinator was carried out to gather information about the Shift users' demographics. From this meeting I identified that the Shift users are mostly men (90%) between the ages of 18 to 34. The users are mostly from the US and they are early adopters that have a good understanding of different software technologies. They use Shift in order to become more efficient in their jobs and have a better use of their time.

Figure 3.2 shows two bar charts: the first chart shows the gender and age demographics, and the second bar chart shows the time when Shift's customers use the tool. The choropleth map displays the different regions where Shift users live.

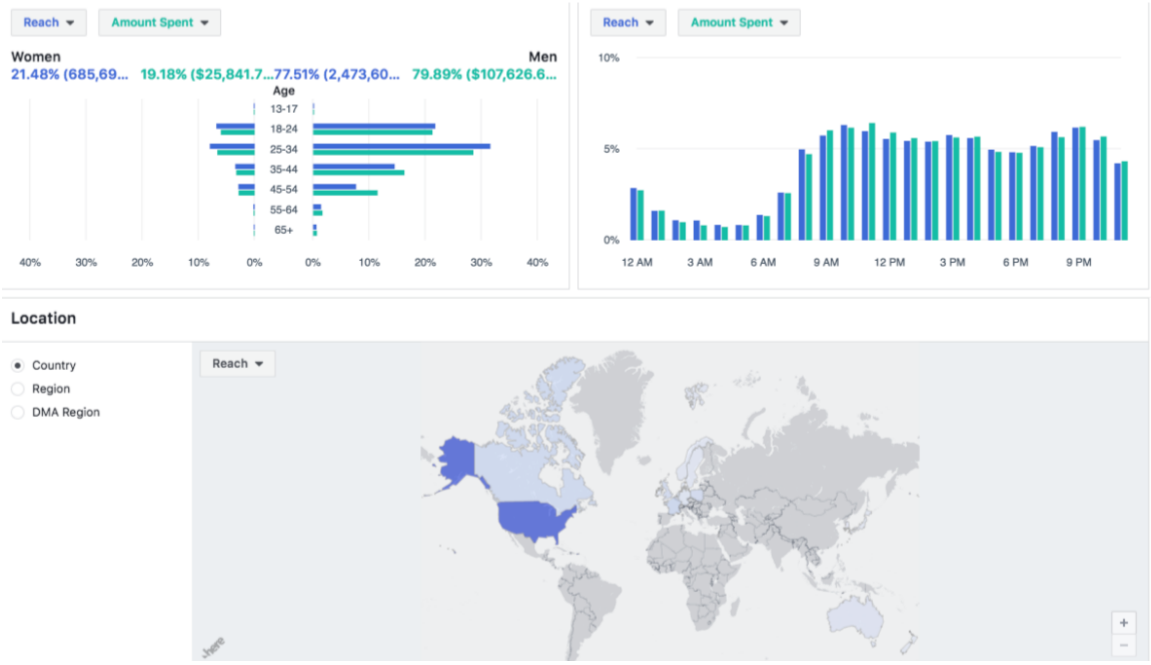


Figure 3.2: Shift Demographics

So, Shiftbot is designed to aim at the Shift's intended user demographics. The chatbot should use words, expressions, and jokes relevant to the target market and should address the most frequent questions asked by them.

The description of the Shiftbot persona is based on the demographics provided by the marketing coordinator. Shiftbot's users are adults aged between 18 and 34 years old, mostly men, and early adopters. They use several social media platforms daily such as Facebook. They are tech-savvy and like to be efficient and fast in their work. Despite, they have knowledge in different applications, the users do not have a general knowledge of software commands, for this reason, the language selected is natural language rather than commands expressions.

"Patrick" the user persona, is a graphic designer that would like to have another option for Shift customer support. He would like to get help at any time, especially at night when he uses Shift to work. He also would like to have access to customer support on weekends. In addition, he finds it very irritating that after having submitted a ticket he remembers another question that he forgot to ask, and then he needs to start the whole process of submitting another ticket. Further details of the user persona "Patrick" can be seen in Table 3.1.

Age:	30
Gender:	Male
Occupation:	Graphic Designer
Status:	Married
Location:	New York, NY
Tier:	Being productive
Innovation adoption:	Early adopter
Motivations:	• Get things done • Efficiency and speed • Achievement
Frustrations:	• Slow process times • Lack of data (information)

Table 3.1: Chatbot Persona

3.3.2 Best Practice #6 Chatbot's Conversation Flows

Figure 3.3 shows the general chatbot flow. This flow does not contemplate errors and therefore, the flow is not being interrupted by an incorrect user input. Red boxes with straight lines are used to specify the user input into the flow, and the blue boxes with dotted lines are designed for the chatbot's input. All the arrows represent flow, and a thin red rectangle represents a decision making point for the user to direct the flow. Finally, a black file represents a link that is being opened in the browser.

3.3.3 Best Practice #2 Chatbot's Conversational Scenarios

The conversational scenarios cover the possible questions a Shift user may ask. The content of the conversational scenarios were gathered from the canned answers of the Shift customer support tool HappyFox. The right-hand side of Figure 3.4 shows the Welcome message scenario, in which the chatbot greets the user and tells them its purpose.

The application of this best practice was iterative, and the scenarios grew as other best practices were applied. After analysing the best practices, I came up with 94 different conversational scenarios (cf. Fig. 3.4) that the user and the chatbot may have.

3.3.4 Best Practice #3 Chatbot's Interaction Elements

The chatbot's interaction elements was designed simultaneously with best practices #4 Chatbot's Conversational Elements. Figure 3.5 shows the different interaction elements used with Shiftbot. There figure shows a GIF element that represents a feeling of joy because the user was able to solve the problem. The typing element,

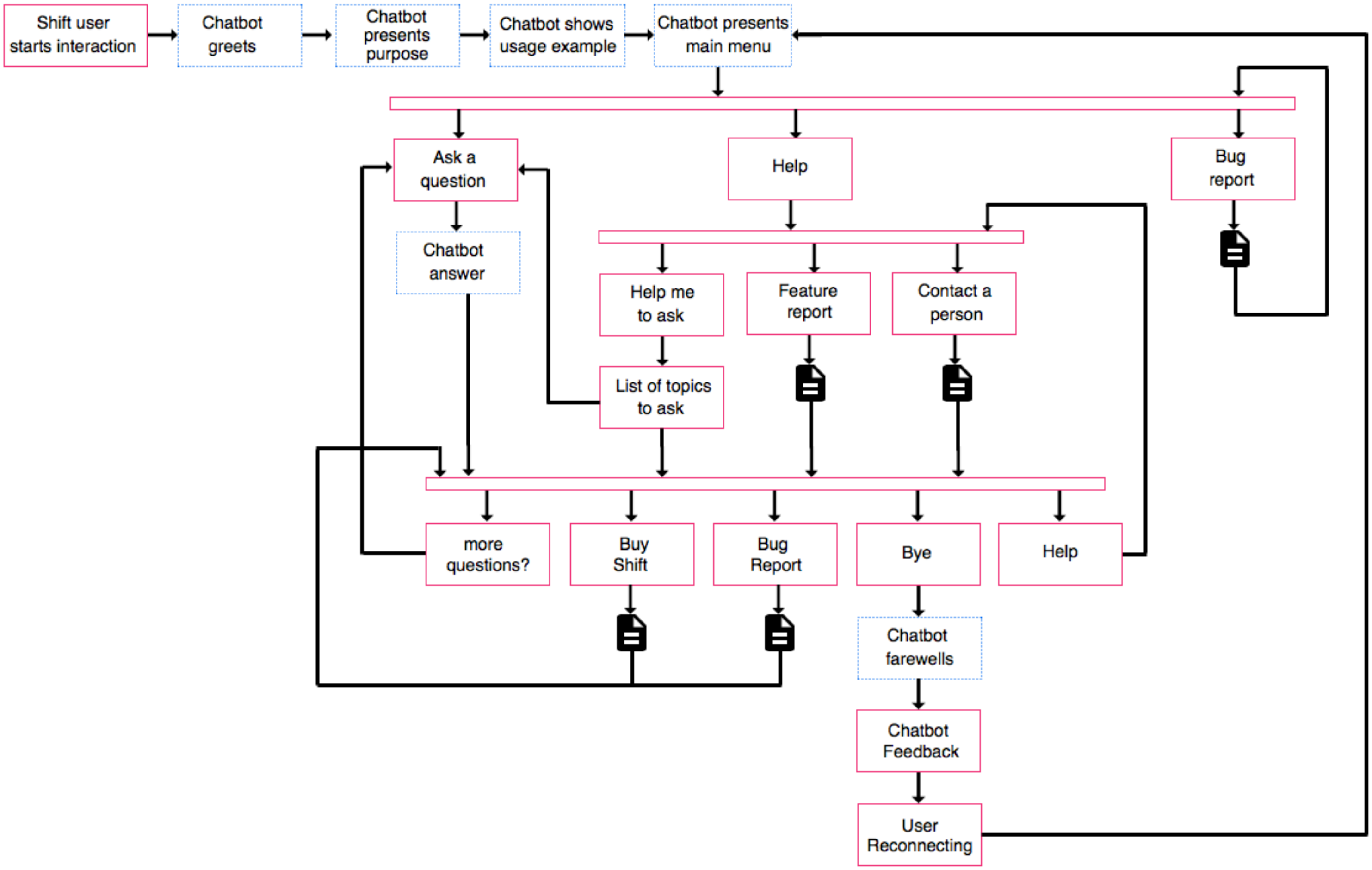
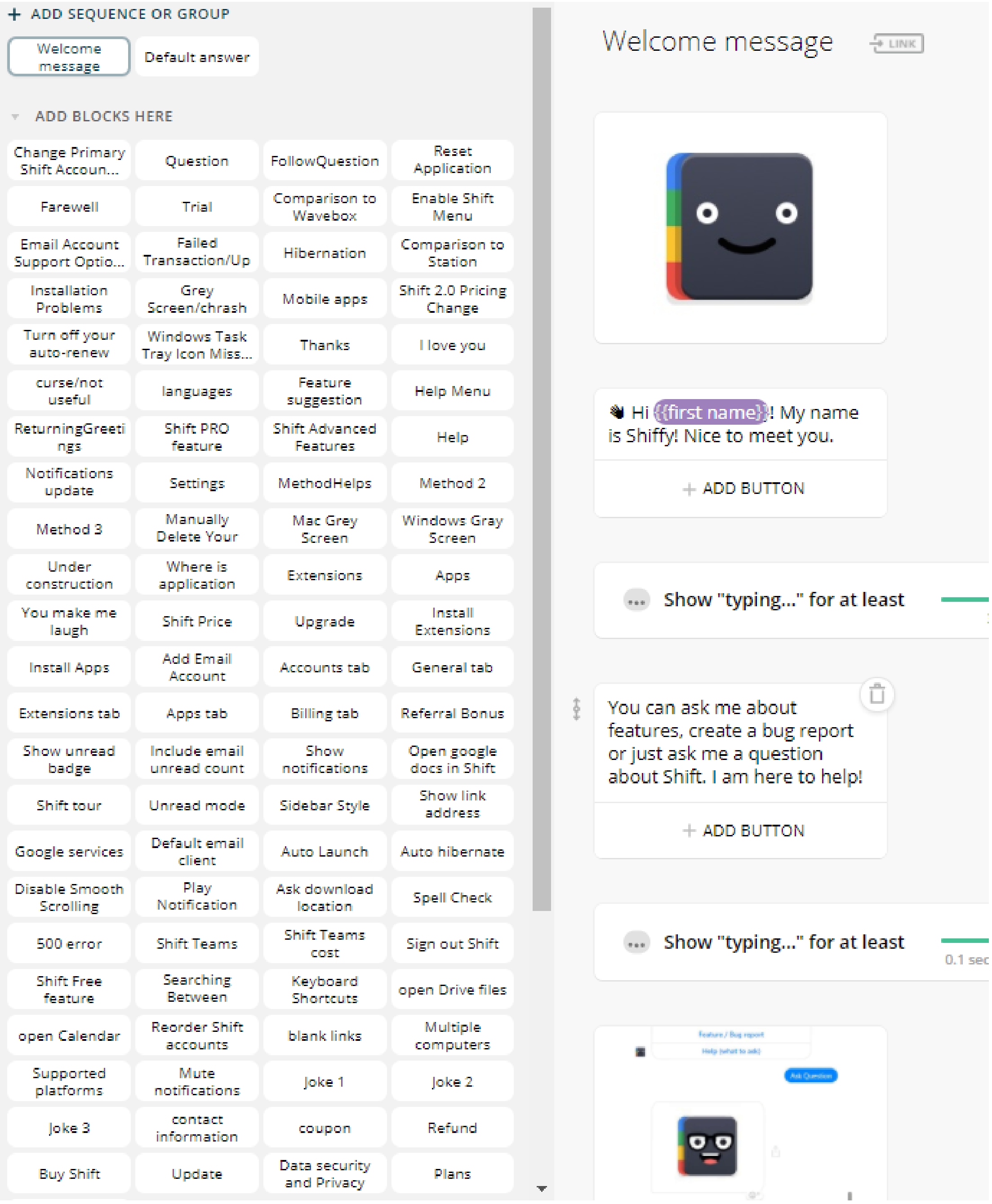


Figure 3.3: Conversation Flow

showed in the figure below the GIF, is used to let users know that more information is coming and give them time to read the first message. A chatbot designer can configure how long the typing element will last; in the example it takes 3.2 seconds. Finally, several buttons are displayed to allow the user to choose the subsequent conversational flow.

3.3.5 Best Practice #4 Chatbot's Conversational Elements

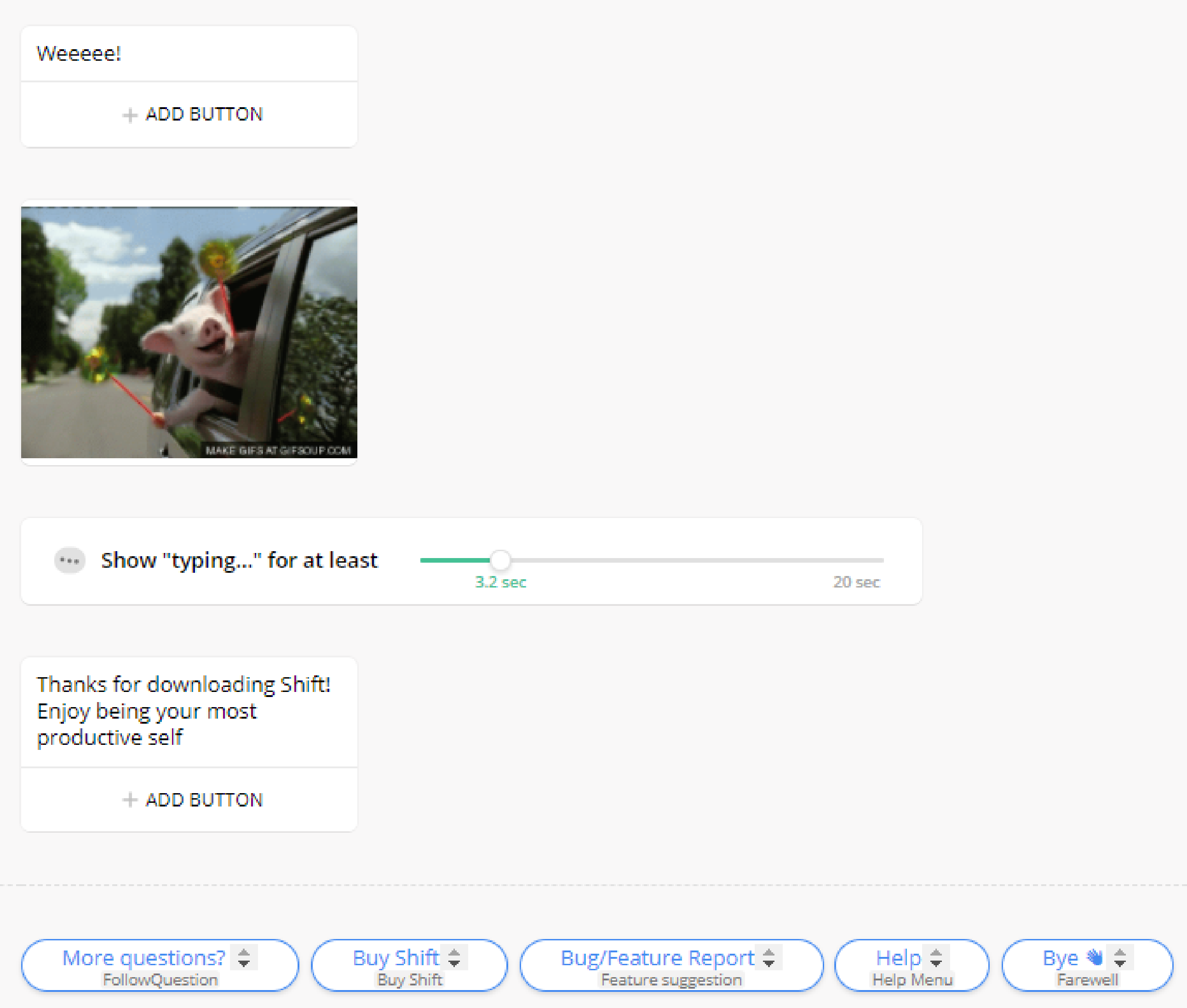
Figure 3.6 shows the Installation Problems' conversational scenario. This figure shows an example of how Shiftbot used an assertion in order to delimit the context of the conversation by saying "Let's get Shift installed for good," thus, the user understands that they will get help installing Shift on their computer. In the same script, Shiftbot



Hi {{first name}}! My name is Shiffy! Nice to meet you.

You can ask me about features, create a bug report or just ask me a question about Shift. I am here to help!

Figure 3.4: Conversational Scenarios



Weeeeee!

Show "typing..." for at least 3.2 sec

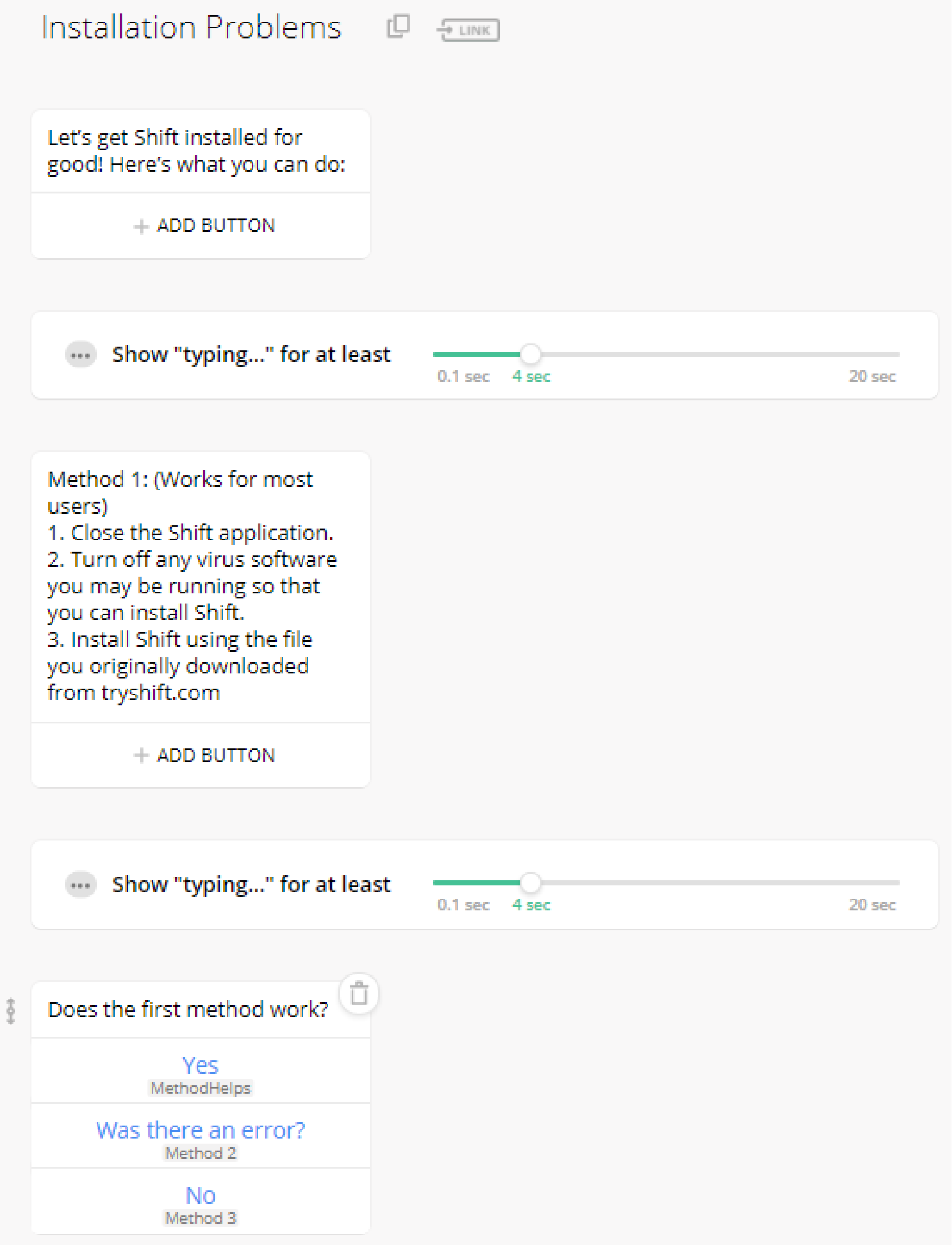
Thanks for downloading Shift! Enjoy being your most productive self

Figure 3.5: Interaction Elements

also has an example of a question that may be answered by a “yes” or “no” (i.e., close-ended question): “Does the first method works?” in which the user can select the option “yes,” “Was there an error?”, or “no.”

3.3.6 Best Practice #7 Chatbot’s Communication Limitations

One of the conversational scenarios that a chatbot must have is the “default answer.” This scenario allows designers to have a response to the user’s input that is not covered



Installation Problems

Let's get Shift installed for good! Here's what you can do:

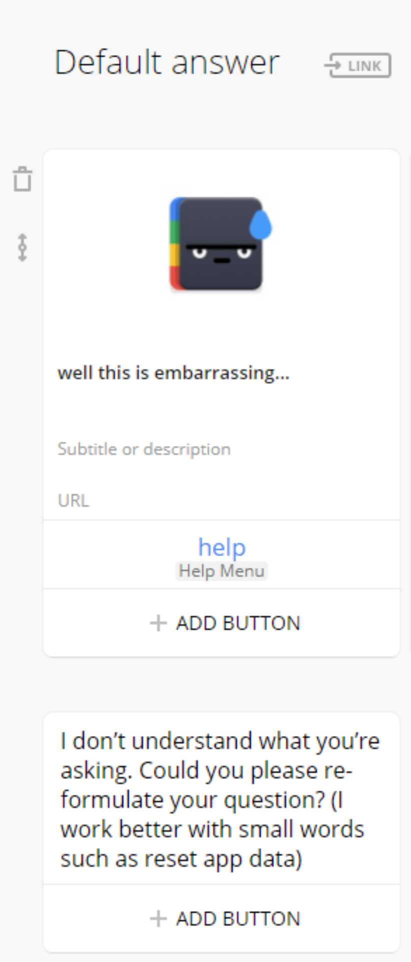
Show "typing..." for at least 4 sec

Method 1: (Works for most users)
1. Close the Shift application.
2. Turn off any virus software you may be running so that you can install Shift.
3. Install Shift using the file you originally downloaded from tryshift.com

Show "typing..." for at least 4 sec

Does the first method work?

Figure 3.6: Conversational Elements



well this is embarrassing...

I don't understand what you're asking. Could you please re-formulate your question? (I work better with small words such as reset app data)

Figure 3.7: Communication Limitation

in the interaction scenarios. Therefore, if a user asks about something unrelated to the chatbot's purpose, the chatbot should gracefully handle those situations with the default script. Figure 3.7 shows the default answer for Shiftbot.

3.3.7 Best Practice #12 Social Intelligence

Social intelligence is an important best practice for a chatbot. The chatbot should display information and interaction elements at the right time and in the correct order to avoid distracting the user and being annoying. And example of this is when Shiftbot displays information and interaction elements in an appropriate amount of time. Figure 3.8 shows a conversational scenario in which Shiftbot displayed an appropriate emoji and message. The conversation is about cancelling the user's Shift subscription, so Shiftbot shows a sad face and the message "We are going to miss you.”

3.3.8 Best Practice #14 Conversational and Situational Knowledge

The conversational knowledge of Shiftbot is all the relevant information about the apps, extensions and features that Shift provides to users, the different subscription plans prices and existing common bugs. An example of this is shown in Figure 3.9. The 500 error message is a known bug on Shift, therefore Shiftbot understands what the users are talking about.

On the other hand, situational knowledge is the user information that changes over time. And example of this is the current subscription plans that the users have, which can be Basic, Pro and Advanced. According to the user’s subscription, Shiftbot should display the appropriate plan description. For example, with an Advanced user, Shiftbot will display all the features that Shift has, while for a basic user, Shiftbot will only display the basic features of Shift.

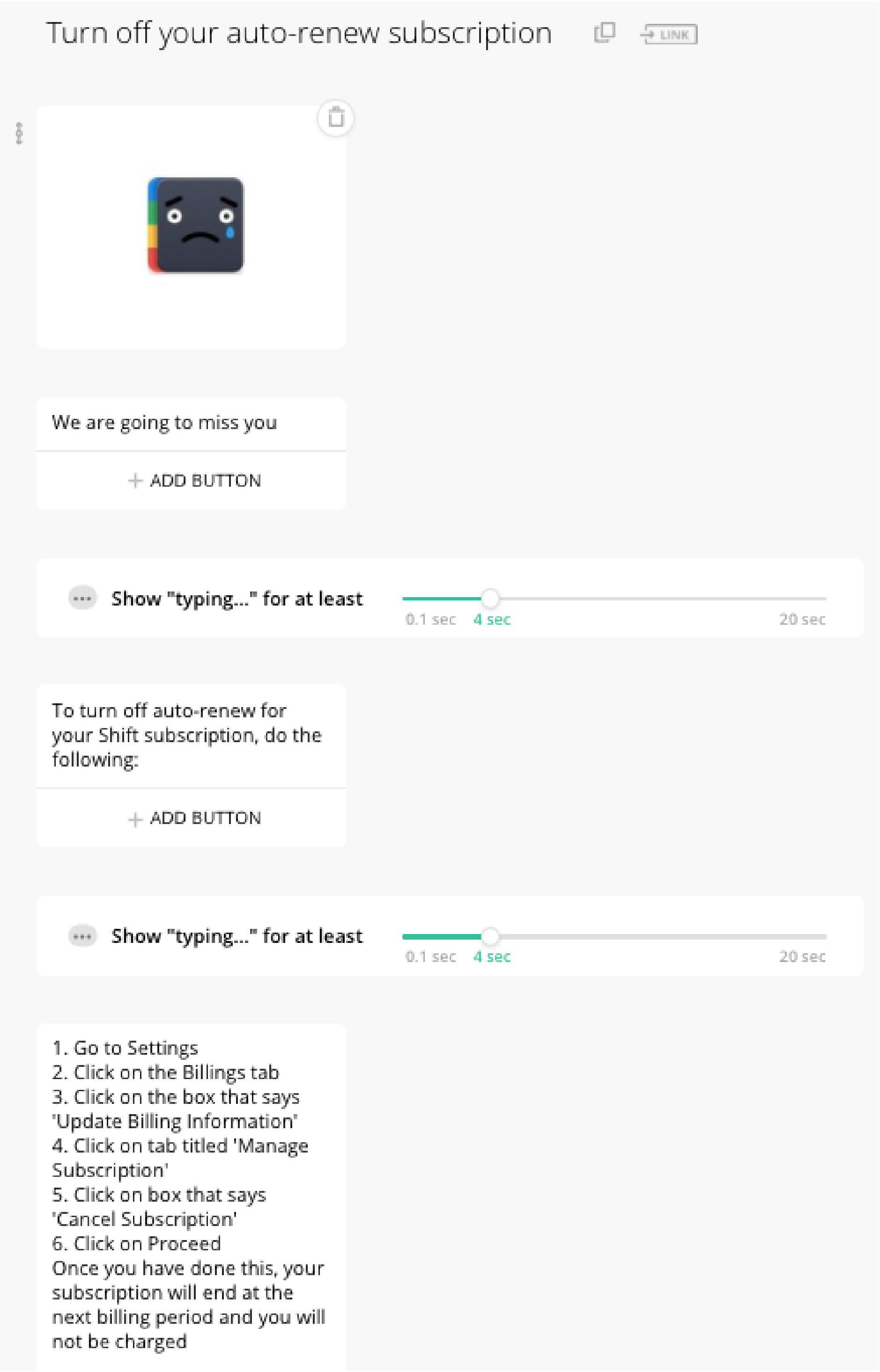
One of the Shiftbot future improvements is to save into variables user’s information such as location, Shift’s application history (subscription date, current subscription plan, coupons used, and email) among others.

3.3.9 Best Practice #10 Chatbot’s Memory

In the design Shiftbot remember user’s preferences of the applications or extensions they already asked about. Shiftbot remembers the user’s current plan, and can provide the relevant information about its features. Because of the project time constraints, this best practice was not implemented in Shiftbot.

3.3.10 Best Practice #18 Ethics: Data Security and Privacy

Designers should consider how the chatbot’s logs are processed. Shiftbot has a conversational scenario that specifies how the chatbot’s logs are used and deleted (cf. Fig 3.10).



Turn off your auto-renew subscription

We are going to miss you

Show "typing..." for at least 4 sec

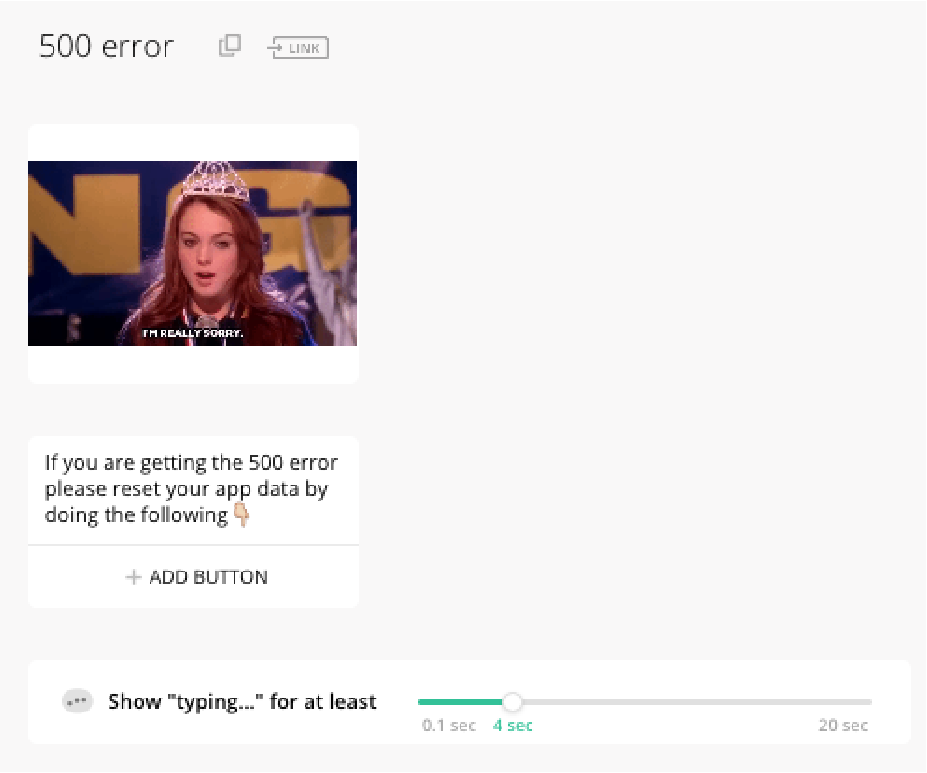
To turn off auto-renew for your Shift subscription, do the following:

Show "typing..." for at least 4 sec

1. Go to Settings
2. Click on the Billings tab
3. Click on the box that says 'Update Billing Information'
4. Click on tab titled 'Manage Subscription'
5. Click on box that says 'Cancel Subscription'
6. Click on Proceed

Once you have done this, your subscription will end at the next billing period and you will not be charged

Figure 3.8: Social Intelligence Example

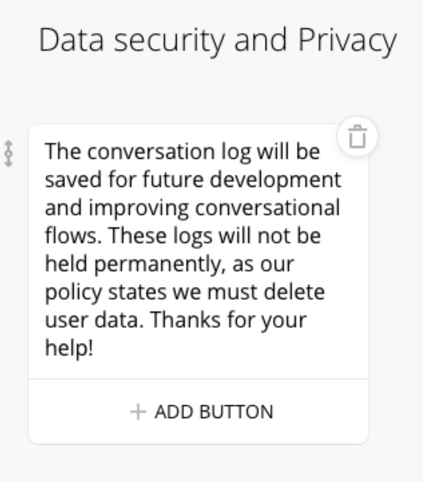


500 error

If you are getting the 500 error please reset your app data by doing the following 👇

Show "typing..." for at least 4 sec

Figure 3.9: Conversational Knowledge Example



Data security and Privacy

The conversation log will be saved for future development and improving conversational flows. These logs will not be held permanently, as our policy states we must delete user data. Thanks for your help!

Figure 3.10: Example of the data security and privacy of Shiftbot

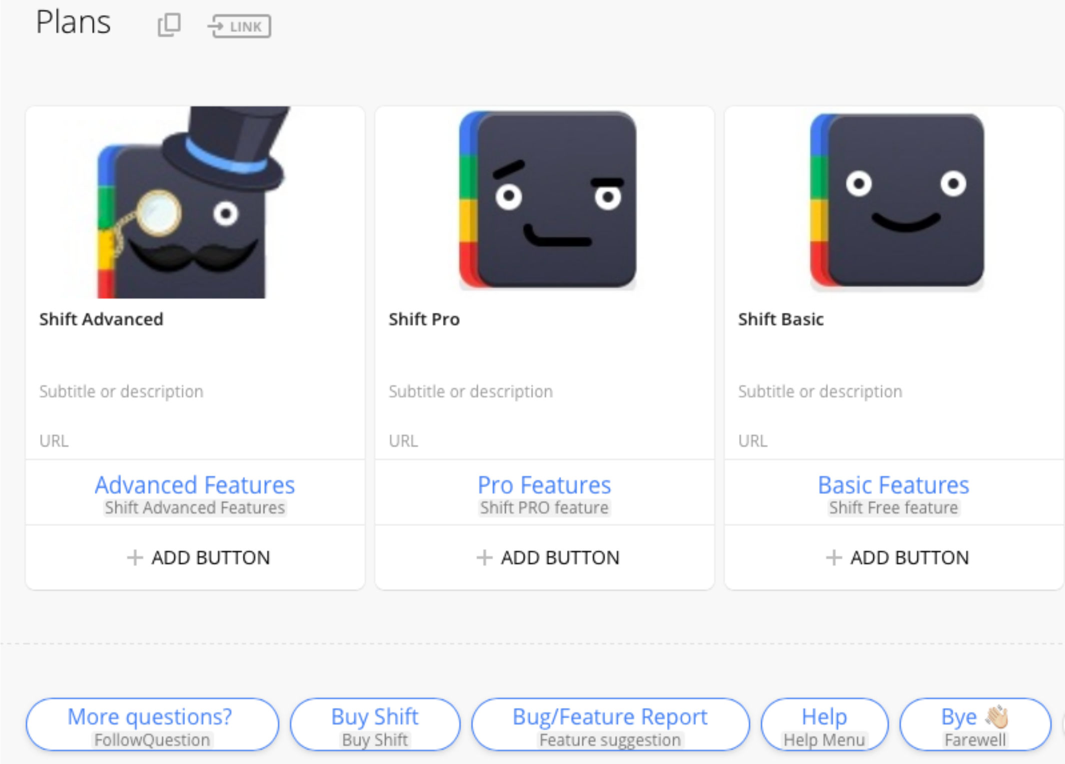


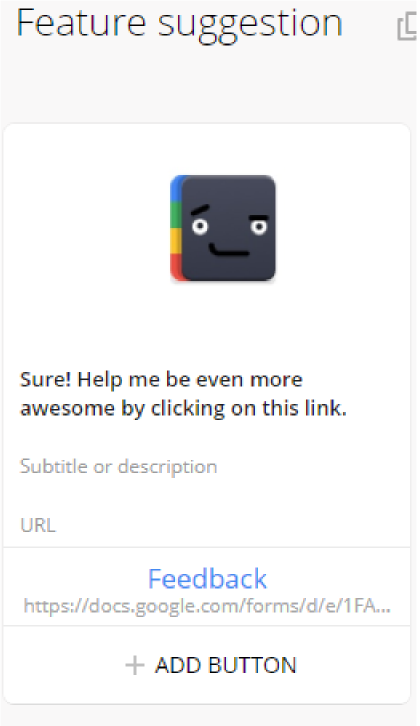
Figure 3.11: Chatbot's Carousel Example

3.3.11 Best Practice #15 User's Control

People using Shiftbot don't feel powerless; users can change the topic of conversation by typing a new question or clicking a button to manage the flow. One example of this is the bottom section of Figure 3.11, in which I show all the possible changes in the conversation. A user may choose to continue asking questions by clicking the "More questions?" button, or can go directly to the buy Shift web page by clicking in the "Buy Shift" button. The user may also click the "Bug Report" button to create a bug ticket, or click the "Help" button if he or she needs further assistance. Finally, clicking the "Bye" button will close the conversation.

3.3.12 Best Practice #8 Chatbot's Personality

The Shiftbot personality was designed to be confident, informative, and fun. This personality match the personality that the chatbot user persona "Patrick" have. Because the Shift demographics shows that users are between ages 18 and 34, Shiftbot is chatty and fun in the conversational scenarios by using jokes and sarcastic comments.



Feature suggestion

Sure! Help me be even more awesome by clicking on this link.

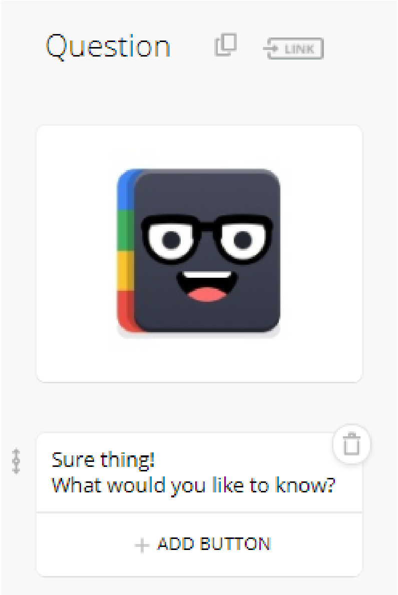
https://docs.google.com/forms/d/e/1FA...

Figure 3.12: Chatbot's Personality

Thus, the chatbots have several GIFs and images that complement the conversations. The chatbot's personality can be displayed by using different interaction elements and language. In Figure 3.12, Shiftbot shows its confident personality by using the phrase "Help me be even more awesome" and having an appropriate image that expresses Shiftbot's feelings when the user wants to give a Shift feature suggestion. Besides, Shiftbot also has jokes included, in which the users can get amused by the chatbot. When a user displays the willing of unsubscribing from Shift, the chatbot shows a crying Shift-emoji.

3.3.13 Best Practice #5 Chatbot's Language

Because Shift is a product for which its users do not use advanced terminology, the language applied to Shiftbot is casual. The language was also designed to match the chatbot user persona "Patrick." Thus, Shiftbot uses common words used by males from 18 to 34 years old. Figure 3.13 shows the chatbot's casual language when it uses the expression "sure thing". This expression is commonly used to informally express that things are fine. Besides, no command expressions are included in the



Sure thing!
What would you like to know?

Figure 3.13: Chatbot's Language

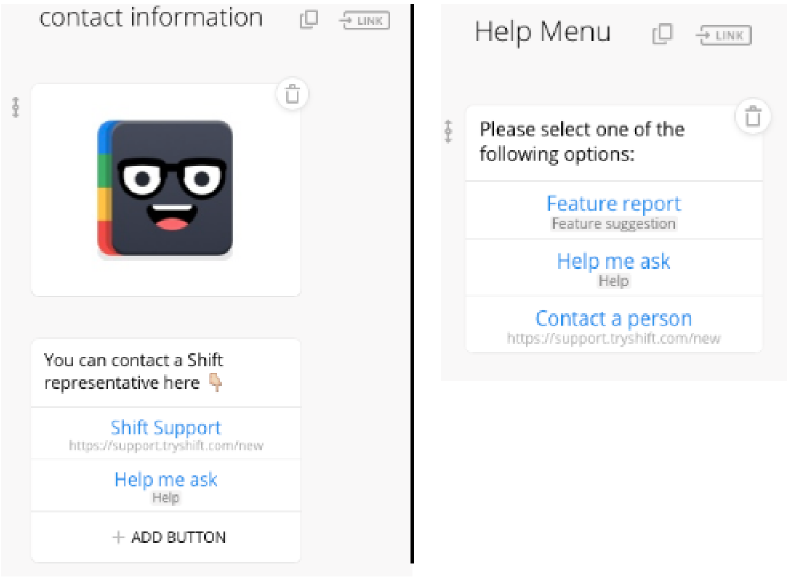
conversations, as the Shift demographics shows that there are not a large number of people that use them. Thus, users can use regular natural language expressions. Shiftbot works better with short sentences, however, this can be improved in the future to include more elaborate conversations such as long questions and a more varied use of words.

3.3.14 Best Practice #9 Chatbot's Help

There are two ways a chatbot works as intended: the first one is when the flow of the conversation is going smoothly and the second one is when the conversation is not flowing well and the chatbot recognizes this and offers help to the user.

Shiftbot offers help in two ways. The first one is when the user types the word "help". In this case, Shiftbot offers the user options to either contact a Shift representative or to explore the "help me ask" option, in which the user can review a complete list of topics to ask the chatbot (left-hand side of Fig 3.14).

The second way that Shiftbot provides help is with the static menu, in which the user can select the option "Help". This option offers the user three alternatives: "Feature report", "Help me ask", and "Contact a person" (right-hand side of Fig 3.14).



You can contact a Shift representative here 👇

Please select one of the following options:

Figure 3.14: Chatbot's Help Example and Help Menu Example

3.3.15 Best Practice #16 User's Input

Shiftbot has several options to get information from the user, such as text fields, buttons, links and carousels. Figure 3.11 shows a carousel that contains the three different Shift purchase plans for the user to select. Links to the features of each plan are included in the carousel. Finally, at the bottom of the Figure, there are five different buttons the user can use to change the flow of the conversation.

3.3.16 Best Practice #11 Making Changes on the Fly

The undo feature can be handled in different ways in the chatbot design. If the chatbot sells products the bot should identify words such as "I meant" or "I have made a mistake", among others. On the other hand, because Shiftbot is not oriented to sell merchandise, users can always re-type what they want to say if they make a mistake. Therefore, if the user asks something erroneous they can always re-type it and this will not break the flow of the conversation with the chatbot.

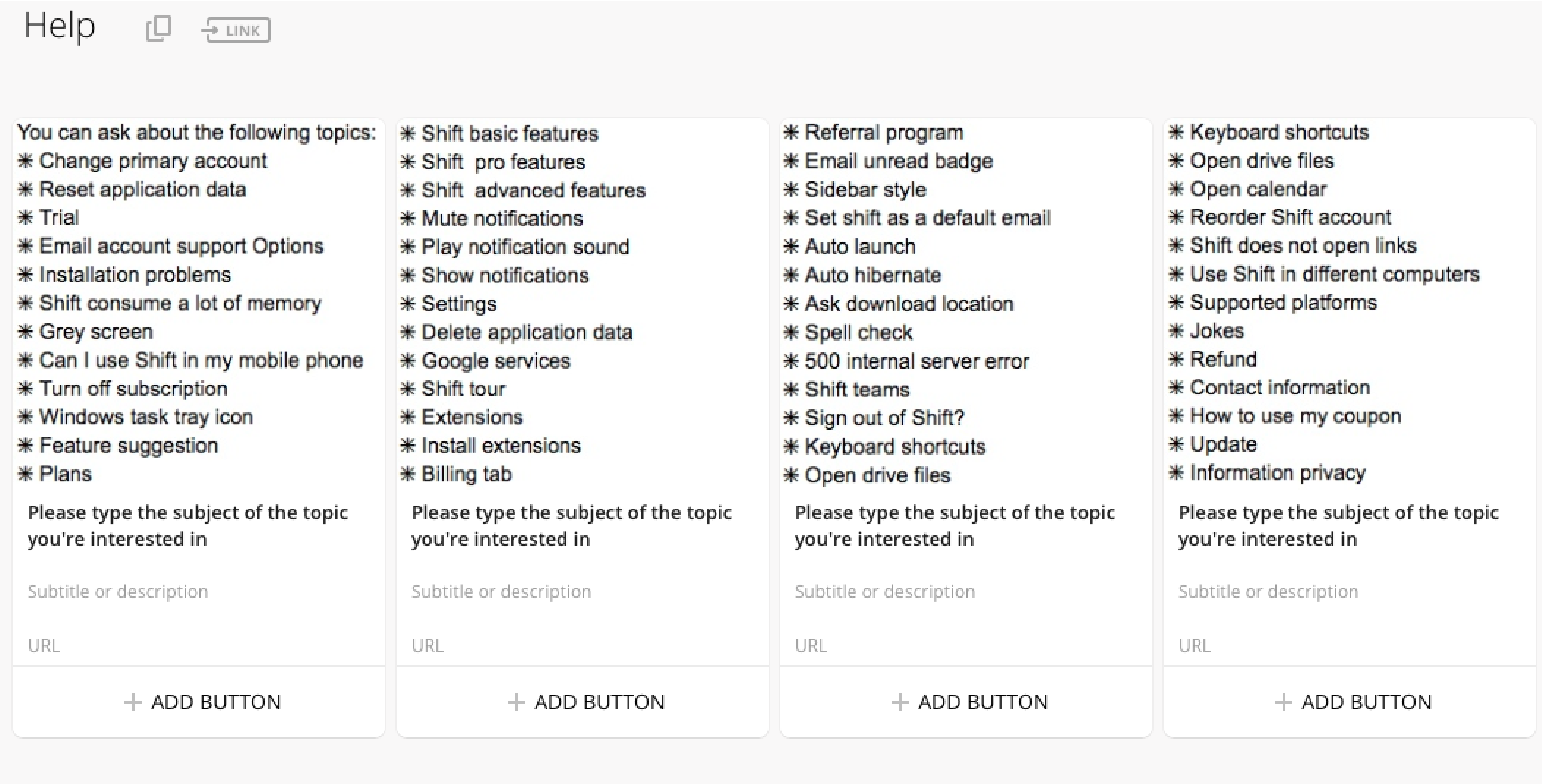
3.3.17 Best Practice #17 User's Recognition and Recall

All the Shiftbot messages are designed to be short. In cases where the message is longer than usual, several text boxes are used. It is also necessary to keep in mind that the user needs time to read the provided message, so to allow for extra reading time the typing interaction element is used and the duration of the element is set accordingly to the length of the message.

As a general rule a designer should include the smallest number of options possible to meet the chatbot's requirements. In this case, Shiftbot has a static menu with five options, which can be seen in Figure 3.5.

3.3.18 Best Practice #13 Chatbot's Documentation

Designers should document all the conversation scenarios so that users can see which topics they can chat about with the bot. Figure 3.15 shows the possible topics that a user may ask Shiftbot. Having this list allows users to see what they can ask of the chatbot.



You can ask about the following topics:
✳ Change primary account
✳ Reset application data
✳ Trial
✳ Email account support Options
✳ Installation problems
✳ Shift consume a lot of memory
✳ Grey screen
✳ Can I use Shift in my mobile phone
✳ Turn off subscription
✳ Windows task tray icon
✳ Feature suggestion
✳ Plans

Please type the subject of the topic you're interested in

✳ Shift basic features
✳ Shift pro features
✳ Shift advanced features
✳ Mute notifications
✳ Play notification sound
✳ Show notifications
✳ Settings
✳ Delete application data
✳ Google services
✳ Shift tour
✳ Extensions
✳ Install extensions
✳ Billing tab

Please type the subject of the topic you're interested in

✳ Referral program
✳ Email unread badge
✳ Sidebar style
✳ Set shift as a default email
✳ Auto launch
✳ Auto hibernate
✳ Ask download location
✳ Spell check
✳ 500 internal server error
✳ Shift teams
✳ Sign out of Shift?
✳ Keyboard shortcuts
✳ Open drive files

Please type the subject of the topic you're interested in

✳ Keyboard shortcuts
✳ Open drive files
✳ Open calendar
✳ Reorder Shift account
✳ Shift does not open links
✳ Use Shift in different computers
✳ Supported platforms
✳ Jokes
✳ Refund
✳ Contact information
✳ How to use my coupon
✳ Update
✳ Information privacy

Please type the subject of the topic you're interested in

Figure 3.15: Chatbot's Documentation Example

3.4 Chatbot Implementation

After the design process of Shiftbot, the implementation part followed. In here, I research about the platform on which the chatbot is hosted and the platform for the development of the chatbot that were more suitable for the chatbot's purpose and Shift.

According to Shevat [29], the platform in which the chatbot is hosted is where the chatbot will be used; it is the platform where the user will interact with the chatbot. Thease platforms for chatbots include Slack and Facebook Messenger among others. Both Slack and Facebook Messenger have unique features that made the selection of the platforms easier, because the platform is selected based on the chatbot's requirements.

Slack is a great platform for teams as it supports one-to-one and team conversations. This means the type of information that the chatbot conveys in one-to-one sessions can be different from the conversation in teams. This characteristic adds a new level of complexity to the design of a chatbot because the chatbot must use social intelligence (Best Practice #12) to avoid over-sharing sensitive information.

Facebook Messenger offers only one-to-one conversations when a chatbot has been added. So it mimics the customer support process beautifully. In this process users send a ticket about their Shift question and a Shift representative replies to the ticket. The customer support process requires just one-to-one conversations. For this project the selected host platform was Facebook because it supported the required customer support features that Shift needed. Besides, according to the Shift marketing coordinator, one of the most commonly used platforms by Shift users is Facebook Messenger.

Next was the selection of the development platform, which is the tool that designers use to create the chatbot. For the development of Shiftbot I considered two development platforms: Chatfuel and Manychats. Both platforms have a free basic plan and do not require programming skills, so the development of the chatbot is quite straightforward in either case. Also, both platforms support Facebook.

The difference that I found crucial in choosing Chatfuel over the other platform was the Natural Language Processing (NLP) algorithm that it has. In order to have a successful match in ManyChats between what the user is typing and what the chatbot understands, the key words should be exactly the same. In other words, if the user has a typo, the chatbot will not understand and reply with the default answer (Best

Practice #7 Communication Limitation) breaking the flow of the conversation.

On the other hand, Chatfuel has a combination of their own NLP algorithm and a third party to verify results. Thus, if a Shiftbot user makes a typo, the chatbot will be "smart" enough to understand what the user's message is. In addition, Chatfuel allows advanced custom coding, which could be useful in the future if the chatbot were to require design changes.

The chatbot has some limitations due to time constraints, such as not being able to save user's data into variables. Another Shiftbot limitation is the use of words that the chatbot accept in order to provide information. This may be improved by gathering feedback, from the Shiftbot's logs, about the common words used by the Shiftbot's users.

There is another kind of limitation in Shiftbot that is related to the NLP algorithm that Chatfuel has. The algorithm does not work so well with plural words. One example of this is when a conversational scenario was trigger by the word "joke." When one of the testers used the plural word "jokes," Shiftbot reply with the conversational scenario of "I do not understand that, could you please re-phrase it?" After this, I have to review the list of words that trigger the conversational scenarios in order to include plural words.

3.4.1 Chatbot Development

The following is a list of features currently implemented in Shiftbot:

1. Can provide help at any time of the day (24/7).
2. Replies to as many questions as users desire.
3. Provides accurate information about Shift features, Apps and extensions.
4. Allows users to give feedback about Shift.
5. Allows users to report a bug about Shift.
6. Allows users to view a complete list of what they can ask.
7. Provides information regarding how user data (logs) is being used and deleted.

The implementation of Shiftbot followed the Shiftbot's design previously described. The first section to create was the Welcome message (Best Practice #2

Chatbot Conversational Scenario refer to Appendix A Fig A.1) and the default answer (Best Practice #7 Chatbot Communication Limitation cf. Fig 3.7). Once those two main conversational scenarios were ready, I proceeded to create the following 94 base scenarios.

Due to time constraints, the saving data feature is not supported, which affects the following best practices: #10 Chatbot's memory, #12 Social Intelligence, #14 Conversational and Situational knowledge, and #15 User's control.

Future development of Shiftbot will include the following features:

1. Have the ability of saving data so that the expressions and information provided by the user is associated with the user's preferences.

2. Support situational knowledge in order to remember the user's current subscription plans.

3. Allow users customize Shiftbot.

4. Send notifications to users about new features.

5. Send upgrade coupons to Free or Pro users.

3.4.2 Evaluation

A preliminary test of the chatbot was carried out with the Shift's Director of Technology and the Shift's Director of Products. Their feedback informed changes to the design and implementation of the chatbot. The most important change requested was to include more conversational scenarios. When the meeting was carried out, the chatbot had 42 conversational scenarios. These flows addressed questions about the most common user support issues. However, it was suggested to include conversational scenarios to describe the features and functionalities of Shift. A minor suggestion was to include one or two informal conversations to amuse chabot users.

In addition, there was a discussion about what other platforms could be used for Shiftbot in the future. As Slack is the more used platform by Shift users, it would be ideal to have the chatbot on Slack, however, this is not supported with the current development platform, Chatfuel.

Furthermore, several tests were also carried out with the Shift marketing team, and with Redbrick's developers and employees. After the tests, all the logs and feedback were used to improve Shiftbot.

The best practices were also updated after the user's feedback. One example of this was the best practice #9 Chatbot's Help, a tester suggested that having a "what to ask?" option will help users to know what the users can ask to the chatbot. Also, after this feedback, the best practice #13 Chatbot's Documentation was updated to include the advice of including a complete list of the chatbot conversational scenarios.

Chapter 4

Evaluation of the Best Practices Through Interviews

4.1 Context

This chapter describes the evaluation that was carried out for the best practices which consisted of two interviews with Redbrick's employees (referred here as interviewee 1 and interviewee 2). Both of them had previous experience in chatbot creation.

In order to conduct the interviews, ethics approval was obtained from the Human Research Ethics Board of the University of Victoria. The two interviewees signed an explicit written informed consent to the use of data in an anonymous way.

Interviewee 1 is a media buyer whose job responsibilities consist of creating and managing advertising strategies and campaigns for Redbrick. Interviewee 1 reported having created a chatbot named "Automaatic" that provides automotive news and information about cars. The development platform used for this chatbot is ManyChat and the platform in which the chatbot is hosted is Facebook. Interviewee 1 has a basic programming background.

Interviewee 2 is a software architect whose job responsibilities include establishing and articulating the architectural vision for Redbrick's systems, tuning the architecture to ensure systems are scalable and available, among others. The chatbot created by interviewee 2 is named "Pongbot" and is used at Redbrick to track the company's ping pong tournaments. "Pongbot" also manages weekly statistics about the ranking positions of the players, selects the competitors, and displays GIFs and sentences related to the score of each tournament. The development platform used to create

"Pongbot" is Python, Pychallonge and an API from http://challonge.com. The host platform is Slack.

In my opinion, both chatbots are well designed: they have a defined purpose and are implemented accordingly. However, there is room for improvement. The first chatbot "Automaatic" can be improved by adding some of the best practices, such as #2 Chatbot's Conversational Scenarios, #8 Chatbot's Personality, and #10 Chatbot's Memory. All these best practices will help the chatbot to improve user engagement by adding extra information in conversational scenarios and acknowledging the user's preferences.

The second chatbot, "Pongbot," already saves data (best practice #10 Chatbot's Memory). However, this information can be used in some of the conversational scenarios for making remarks on the users' rankings. Specifically, best practice #14 Conversational and Situational knowledge may be improved: after a tournament, the ranking of the users changes and Pongbot can use the new information to make additional remarks in the conversation. In addition, adding the best practice #8 Chatbot's Personality will increase the agreeability of the chatbot among users.

4.1.1 Evaluation Methodology

The evaluation consisted of three phases for both interviewees. The first phase consisted of a pre-interview questionnaire sent via email before the evaluation interview. (cf. Appendix A Fig A.2 and A.3). In the second phase, I interviewed each participant about the chatbot best practices, and the third phase consisted of a questionnaire sent via email after the interview (cf. Appendix A Fig A.4 and A.5).

The evaluation was independent for each interviewee. Each interviewee filled the pre-interview questionnaire form before the evaluation interview of the best practices was carried out. This evaluation lasted around 45 minutes, in which the interviewee was asked to describe each of the best practices and was strongly encouraged to ask questions and give feedback. After the description of the best practices, the researcher asked follow-up questions such as "Do the best practices make sense?", "Does the classification of the best practices make sense?", and "Is something missing in the classification or in the definition of the best practices?" After concluding the interview, the post-interview questionnaire link was sent to the interviewee (cf. Appendix A Fig A.4 and A.5).

The pre-interview questionnaire was created to gain an understanding of the background of the interviewees, such as job description, job responsibilities, educational background and chatbot knowledge. The post-questionnaire was created to give interviewees another channel in which to provide feedback after the interview. The post-interview questionnaire included questions such as “How likely is it that they use the best practices to create a chatbot?”

4.1.2 Evaluation Results

The results of the evaluation interview revealed that the best practices, and the classification are relevant for chatbot design. For some best practices further explanations were needed, however, after this, the interviewees agreed that the provided definitions of the best practices were correct.

The only change that was made after the user feedback was to rename the Best Practice #12, Social Intelligence (it was named Social and Emotional Intelligence before this). The feedback received from the interviewee 2 was that the word “emotional” led the reader to think about empathy and this best practice relates to being careful about the information provided by the chatbot, not how the chatbot approach’s to users.

Another comment received in the meeting evaluation was for best practices #14 Conversational and Situational Knowledge. The interviewee 2 said, “To me that almost sounds like two separate things (best practices), not one.” The reason why this best practice has Conversational and Situational Knowledge together, is because both of them are dealing with the context of the conversation between an specific user and the chatbot.

The other best practices were well received by the interviewees. After the interviews, the relevant feedback was applied to have more accurate best practices. Both interviewees agreed that the best practices will allow them and others chatbot designers to have a better chatbot’s design process.

Chapter 5

Conclusions

> Design is not just what it looks like
> and feels like. Design is how it works.
>
> *Steve Jobs*

The main contribution of this Master's project was to came up with a list of best practices that provide designers with a structure of how to design a script-based chatbot effectively.

Every one that wants to implement a chatbot needs to understand that the design phase will establish the success of their implementation process. Having a good design will allow designers to have a solid base to create a chatbot, and the implementation will be smooth and hopefully painless. In addition, a good design may also improve the chatbot use experience.

Knowing the final users of the chatbot is crucial for a good design phase, and the more information the designer has about the user, the more likely it is that the chatbot design will meet to the user's needs and preferences.

Designers need to understand that the design of a chatbot is an iterative process, which means the chatbot will need to be updated from time to time, new flows may be added to address new issues or to explain new features. Also, additional responses should be added from time to time in order to prevent the user be bored of having the same responses all the time. Furthermore, the chatbot design should be regularly revised using the best practices. The best practices can help in this iterative process to ensure the design of the chatbot behaves as intended.

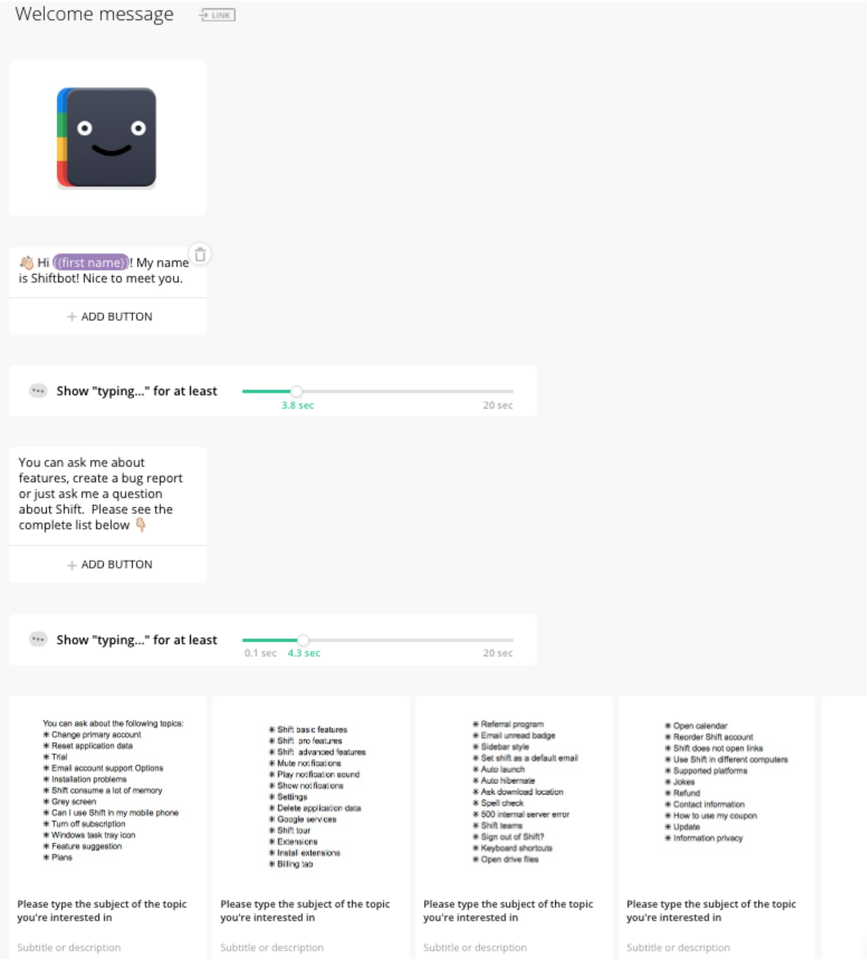
Although the best practices were created considering the chatbot design process, they can also be used to evaluate already implemented chatbots.

After this research I hope these best practices will allow designers to have a more joyful chatbot creation and users will have a better experience using the resulting chatbots.

For future work, this Master's project can be improved by adding additional best practices in order to address different types of chatbots, such as speech-based chatbots. In addition, more evaluations can be carried out to test the efficiency of the best practices. A chatbot designer can use the best practices to design a new chatbot from scratch.

Appendix A

Additional Information



Hi {first name}! My name is Shiftbot! Nice to meet you.

Show "typing..." for at least 3.8 sec 20 sec

You can ask me about features, create a bug report or just ask me a question about Shift. Please see the complete list below 👇

Show "typing..." for at least 0.1 sec 4.3 sec 20 sec

You can ask about the following topics:
✱ Change primary account
✱ Reset application data
✱ Trial
✱ Email account support Options
✱ Installation problems
✱ Shift consume a lot of memory
✱ Grey screen
✱ Can I use Shift in my mobile phone
✱ Turn off subscription
✱ Windows task tray icon
✱ Feature suggestion
✱ Plans

Please type the subject of the topic you're interested in

Subtitle or description

✱ Shift basic features
✱ Shift pro features
✱ Shift advanced features
✱ Mute notifications
✱ Play notification sound
✱ Show notifications
✱ Settings
✱ Delete application data
✱ Google services
✱ Shift tour
✱ Extensions
✱ Install extensions
✱ Billing tab

Please type the subject of the topic you're interested in

Subtitle or description

✱ Referral program
✱ Email unread badge
✱ Sidebar style
✱ Set shift as a default email
✱ Auto launch
✱ Auto hibernate
✱ Ask download location
✱ Spell check
✱ 500 internal server error
✱ Shift teams
✱ Sign out of Shift?
✱ Keyboard shortcuts
✱ Open drive files

Please type the subject of the topic you're interested in

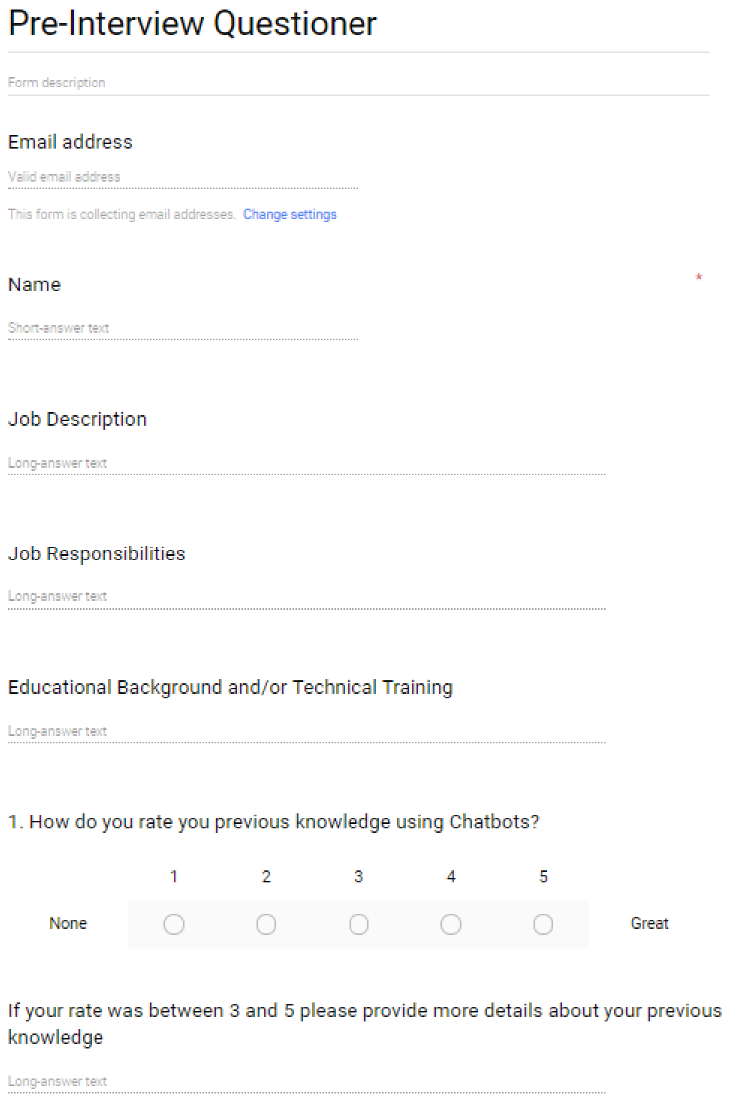
Subtitle or description

✱ Open calendar
✱ Reorder Shift account
✱ Shift does not open links
✱ Use Shift in different computers
✱ Supported platforms
✱ Jokes
✱ Refund
✱ Contact information
✱ How to use my coupon
✱ Update
✱ Information privacy

Please type the subject of the topic you're interested in

Subtitle or description

Figure A.1: Welcome message



Pre-Interview Questioner

Form description

Email address

Valid email address

This form is collecting email addresses. Change settings

Name *

Short-answer text

Job Description

Long-answer text

Job Responsibilities

Long-answer text

Educational Background and/or Technical Training

Long-answer text

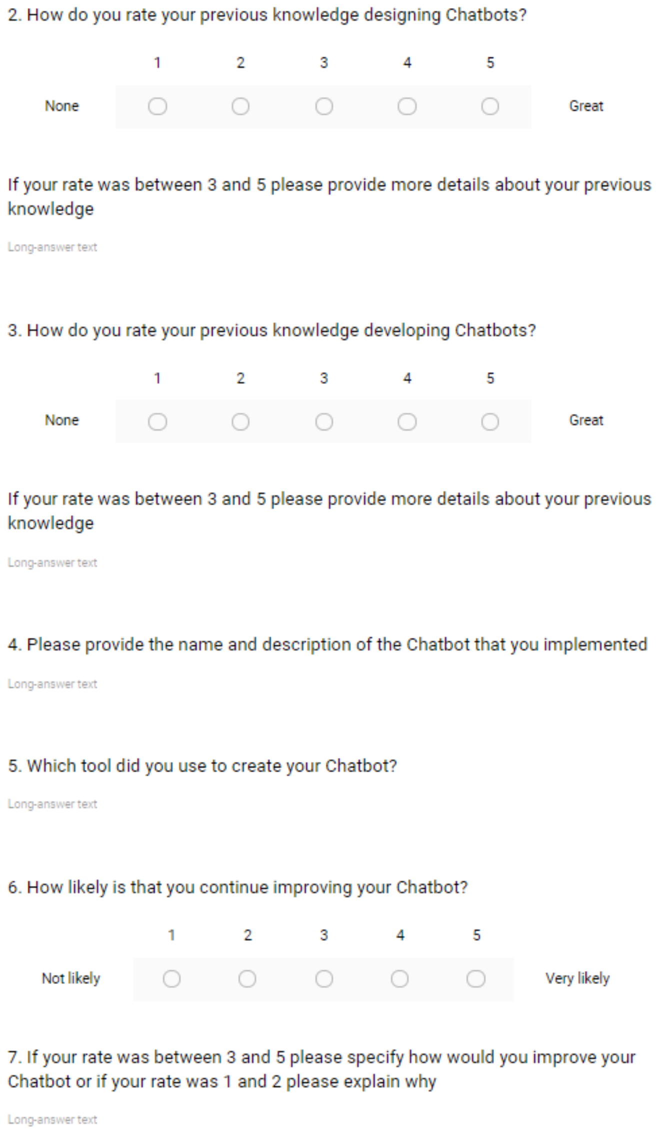
1. How do you rate you previous knowledge using Chatbots?

	1	2	3	4	5	
None	○	○	○	○	○	Great

If your rate was between 3 and 5 please provide more details about your previous knowledge

Long-answer text

Figure A.2: Pre-interview questionnaire



2. How do you rate your previous knowledge designing Chatbots?

	1	2	3	4	5	
None	○	○	○	○	○	Great

If your rate was between 3 and 5 please provide more details about your previous knowledge

Long-answer text

3. How do you rate your previous knowledge developing Chatbots?

	1	2	3	4	5	
None	○	○	○	○	○	Great

If your rate was between 3 and 5 please provide more details about your previous knowledge

Long-answer text

4. Please provide the name and description of the Chatbot that you implemented

Long-answer text

5. Which tool did you use to create your Chatbot?

Long-answer text

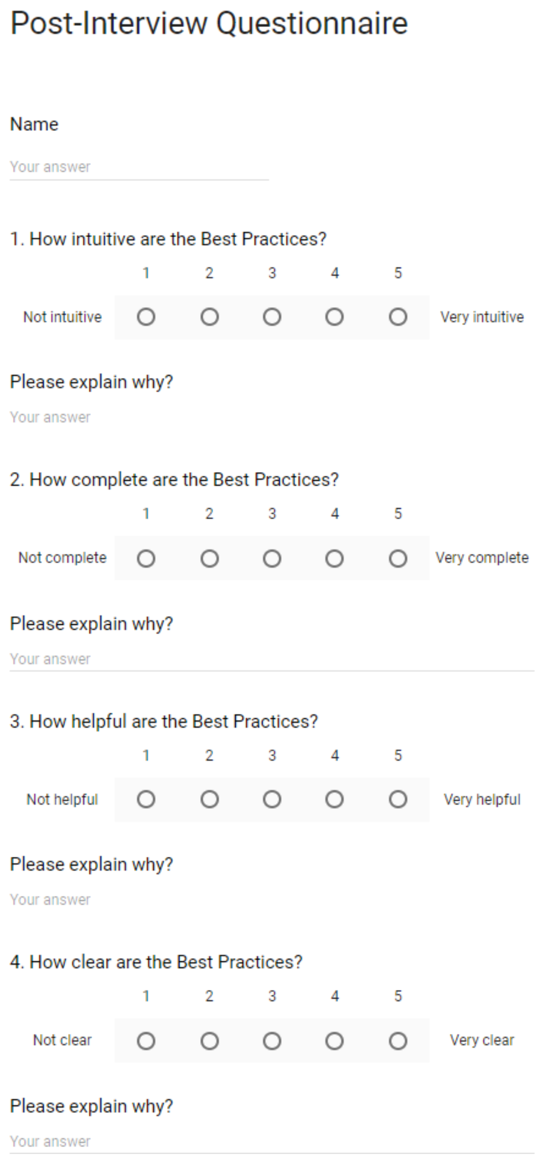
6. How likely is that you continue improving your Chatbot?

	1	2	3	4	5	
Not likely	○	○	○	○	○	Very likely

7. If your rate was between 3 and 5 please specify how would you improve your Chatbot or if your rate was 1 and 2 please explain why

Long-answer text

Figure A.3: Pre-interview questionnaire



Post-Interview Questionnaire

Name

Your answer

1. How intuitive are the Best Practices?

	1	2	3	4	5	
Not intuitive	○	○	○	○	○	Very intuitive

Please explain why?

Your answer

2. How complete are the Best Practices?

	1	2	3	4	5	
Not complete	○	○	○	○	○	Very complete

Please explain why?

Your answer

3. How helpful are the Best Practices?

	1	2	3	4	5	
Not helpful	○	○	○	○	○	Very helpful

Please explain why?

Your answer

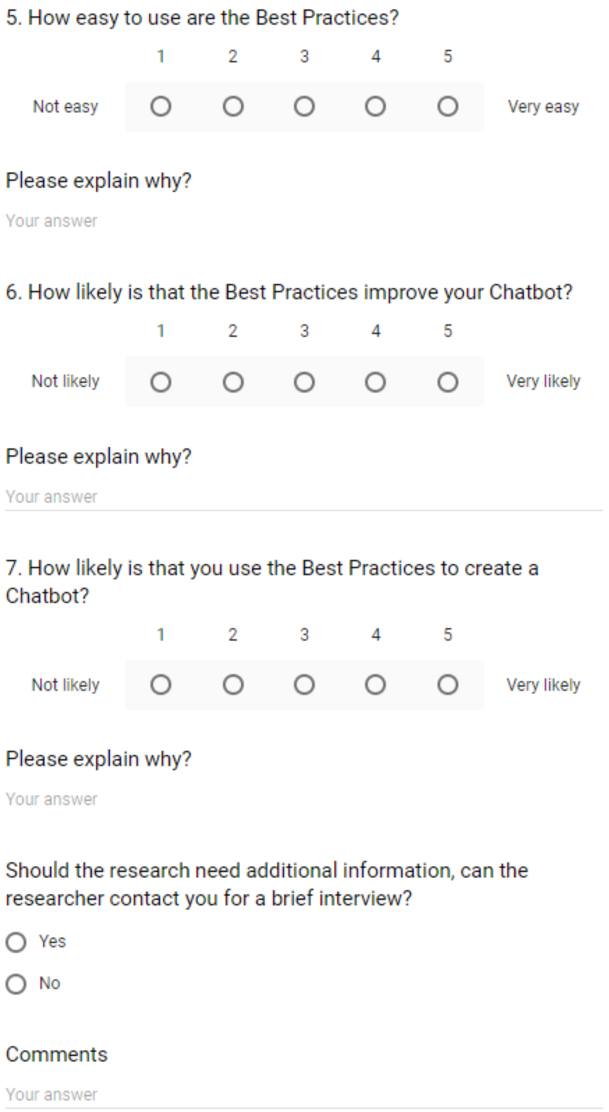
4. How clear are the Best Practices?

	1	2	3	4	5	
Not clear	○	○	○	○	○	Very clear

Please explain why?

Your answer

Figure A.4: Post-interview questionnaire



5. How easy to use are the Best Practices?

	1	2	3	4	5	
Not easy	○	○	○	○	○	Very easy

Please explain why?

Your answer

6. How likely is that the Best Practices improve your Chatbot?

	1	2	3	4	5	
Not likely	○	○	○	○	○	Very likely

Please explain why?

Your answer

7. How likely is that you use the Best Practices to create a Chatbot?

	1	2	3	4	5	
Not likely	○	○	○	○	○	Very likely

Please explain why?

Your answer

Should the research need additional information, can the researcher contact you for a brief interview?

○ Yes

○ No

Comments

Your answer

Figure A.5: Post-interview questionnaire

Bibliography

[1] Bots: An introduction for developers. Telegram Bots https://core.telegram.org/bots, October 2014.

[2] Considering the word chatbot. botnerds http://botnerds.com/chatbots/, November 2016.

[3] *Oxford Dictionaries*. Oxford University Press, 2017.

[4] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307, 1999.

[5] Obaid Ahmed. Designing chatbot conversations. Chatbot's Life https://chatbotslife.com/designing-chatbot-conversations-7fd94b2a78d8, March 2017.

[6] Donald E. Allen and Rebecca F. Guy. *Conversation analysis: the sociology of talk*, volume 200. Walter de Gruyter, The Hague, 1978.

[7] Jason Amunwa. Let's talk about text: A chatbot design guide. Telepathy, 2018.

[8] Prashanti Angara, Miguel Jiménez, Kirti Agarwal, Harshit Jain, Roshni Jain, Ulrike Stege, Sudhakar Ganti, Hausi A Müller, and Joanna W Ng. Foodie Fooderson a conversational agent for the smart kitchen. *Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering*, pages 247–253, 2017.

[9] Gillian Armstrong. Helping your baby bot learn to chat like a grown up bot. Chatbots Magazine https://chatbotsmagazine.com/helping-your-baby-bot-learn-to-chat-like-a-grown-up-bot-99f5170f1c55, August 2017.

[10] Justin Clegg. Rescue.io: A chatbot solution for emergency situations. Square One Labs https://medium.com/square-one-labs/rescue-io-a-chatbot-solution-to-emergency-situations-dd267f174554, April 2017.

[11] Alan Cooper et al. *The inmates are running the asylum:Why high-tech products drive us crazy and how to restore the sanity*. Sams Indianapolis, IN, USA:, 2004.

[12] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. Superagent: A customer service chatbot for e-commerce websites. *Proceedings of ACL 2017, System Demonstrations*, pages 97–102, 2017.

[13] Martin J Eppler. A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing. *Information visualization*, 5(3):202–210, 2006.

[14] Sean M Falconer and Margaret-Anne Storey. A cognitive support framework for ontology mapping. *The Semantic Web*, pages 114–127, 2007.

[15] S Sowmya Kamath, Anusha Bagalkotkar, Ashesh Kandelwal, Shivam Pandey, and Kumari Poornima. Sentiment analysis based approaches for understanding user context in web content. *IEEE*, pages 607–611, 2013.

[16] Anna Lurchenko. Conversational UX design. All facebook messenger bots interactions. Chatbots Magazine https://chatbotsmagazine.com/cheat-sheet-all-facebook-chatbot-interactions-4b14e4e00178, February 2017.

[17] Alexander Maedche, Achim Botzenhardt, and Ludwig Neer. *Software for people: fundamentals, trends and best practices*. Springer Science & Business Media, 2012.

[18] John Markoff and Paul Mozour. For sympathetic ear, more chinese turn to smartphone program. The New York Times https://www.nytimes.com/2015/08/04/science/for-sympathetic-ear-more-chinese-turn-to-smartphone-program.html, July 2015.

[19] Robinson Meyer. Even early focus groups hated clippy. The Atlantic https://www.theatlantic.com/technology/archive/2015/06/clippy-the-microsoft-office-assistant-is-the-patriarchys-fault/396653/, Jun 2015.

[20] Sneha Mishra. How to communicate effectively? BUSINESSTOPIA http://www.businesstopia.net/blog/how-communicate-effectively, 2017.

[21] Oksana Oboishchyk. Smart notes for building a featured bot. Chatbots Magazine https://chatbotsmagazine.com/smart-notes-for-building-a-featured-bot-7a7832c25b8d, May 2017.

[22] Barbara Ondrisek. Privacy and data security of chatbots. Medium https://medium.com/@electrobabe/privacy-and-data-security-of-chatbots-6ab87773aadc, October 2016.

[23] Kara Pernice. Carousel usability: Designing an effective ui for websites with content overload. Nielsen Norman Group https://www.nngroup.com/articles/designing-effective-carousels/, September 2013.

[24] Nicole M Radziwill and Morgan C Benton. Evaluating quality of chatbots and intelligent conversational agents. *arXiv preprint arXiv:1704.04579*, 2017.

[25] Rachel Reichman. *Getting computers to talk like you and me: discourse context, focus, and semantics.* MIT Press Cambridge, MA., 1985.

[26] Katharine Schwab. The challenge of designing a chatbot with manners and the flip side: making one that isn't creepy. Co.Design https://www.fastcodesign.com/3064710/the-challenge-of-designing-a-chatbot-with-manners, April 2016.

[27] Kevin Scott. Usability heuristics for bots. Chatbots Magazine https://chatbotsmagazine.com/usability-heuristics-for-bots-7075132d2c92, May 2016.

[28] Robert Sens. Designing a chatbot conversation. Behance https://www.behance.net/gallery/37453869/Designing-a-Chatbot-UX-Design-Process-Case-Study, May.

[29] Amir Shevat. *Designing Bots Creating Conversational Experiences.* O'REILLY Media, May 2017.

[30] Ben Shneiderman. Looking for the bright side of user interface agents. *Interactions*, 2(1):13–15, 1995.

[31] Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *Interactions*, 4(6):42–61, 1997.

[32] Daniel J Siegel. *Pocket guide to interpersonal neurobiology: An integrative handbook of the mind.* WW Norton & Company, 2012.

[33] Spectrm. Your chatbot's personality is the key. Chatbots Magazine https://chatbotsmagazine.com/your-chatbots-personality-is-the-key-683a2051fb7d, February 2017.

[34] Tanya Stivers. An overview of the question–response system in american english conversation. *Journal of Pragmatics*, 42(10):2772–2781, 2010.

[35] Mary Treseler. Designing bots. O'REILLY https://www.oreilly.com/ideas/designing-bots, December 2016.

[36] Norha M. Villegas and Hausi A. Müller. Managing dynamic context to optimize smart interactions and services. *The Smart Internet: Current Research and Future Applications*, pages 289–318, 2010.

[37] Andrew Walenstein. *Cognitive support in software engineering tools: A distributed cognition framework.* PhD thesis, Citeseer, 2002.

[38] Amelia Wong. How chatbots can help your users without you there. Chatbots Magazine https://chatbotsmagazine.com/how-chatbots-can-help-your-users-without-you-there-e82e6814a903, December 2016.

[39] Mariya Yao. Chatbot UX – Does conversation hurt or help? Smashing magazine https://www.smashingmagazine.com/2016/11/does-conversation-hurt-or-help-the-chatbot-ux/, November 2016.

[40] George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology.* Ravenio Books, 2016.