Sandra Stocker, BSc

User Engagement in Chatbot Systems

With Focus on User Onboarding

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dr. techn. Johanna Pirker, BSc

Institute for Interactive Systems and Data Science
Head: Univ.-Prof. Dipl-Inf. Dr. Stefanie Lindstaedt

Graz, September 2019

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Abstract

The term "chatbot" refers to software applications that use natural language in a dialog fashioned way to interact with humans. Even though the first chatbots was already published decades ago the rise of chatbots started in 2016. The complexity of language comprehension, but also language generation has been underestimated for a long time. Nowadays language understanding engines based on machine learning as well as whole frameworks are provided to reduce the development effort of chatbots, like Microsoft's Bot Framework. Aspects like the selection of an appropriate framework, a suitable chatbot architecture, the design of a good language model and the definition of the chatbot personality are main aspects to consider. Many of the available text- and speech-based chatbots suffer from low user acceptance and high drop-off rates. Users that leave the conversation within the first few minutes of interaction tend to not use the system again. Many users still prefer to talk to humans over talking to machines, as many of them rarely get in touch with this technology. A main focus was therefore, the identification and implementation of suitable onboarding and engagement mechanisms to minimize these problems. The applied strategies had been selected carefully, as not all available onboarding and user engagement mechanisms are suitable for every business case. To measure their impact a survey was conducted that compares an engaging chatbot (including onboarding mechanisms) and a chatbot without these mechanisms. Hereby it was possible to prove that user engagement as well as user acceptance has increased for participants interacting with the engaging chatbot. Also, the appearance of the chatbot itself was perceived as more attractive and aesthetically appealing. After the interaction with the chatbots more participants consider using these systems in the future. Nevertheless, users have to get in touch with chatbots more frequently to increase trust and user acceptance for this technology.

Kurzfassung

Der Begriff „Chatbot" bezieht sich auf Softwareanwendungen, welche natürliche Sprache nutzen um mit dem Benutzer in einen Dialog zu treten. Auch wenn die ersten Chatbots bereits vor einigen Jahrzehnten entwickelt wurden, gewann der Chatbot erst ab dem Jahr 2016 an Popularität. Die Komplexität der Sprache, sowohl bei der Generierung als auch beim Verständnis, wurden lange Zeit unterschätzt. Heutzutage erfolgt die Spracherkennung in vielen Fällen auf Basis von maschinellem Lernen. Komponenten zur Sprachverarbeitung sowie umfassende Entwicklungs-Frameworks werden bereits zur Verfügung gestellt, um das Erstellen eines Chatbots zu erleichtern. So beispielsweise auch das Microsoft Bot Framework. Die Auswahl eines passenden Frameworks, eine durchdachte Architektur, die Entwicklung eines ausgereiften Sprachmodells sowie das Festlegen der Chatbot Personalität sind entscheidende Aspekte, die bedacht werden müssen. Viele der Text- und Sprachbasierten Chatbots leiden unter einer geringen Benutzerakzeptanz und hohen Absprungrate. BenutzerInnen, welche die Kommunikation innerhalb der ersten Minuten abbrechen, verwenden das Programm üblicherweise nicht noch einmal. Die meisten AnwenderInnen bevorzugen noch den Kontakt mit realen Personen, da der Umgang mit der Technologie für viele noch unbekannt ist. Das Ziel der Arbeit liegt somit in der Herausarbeitung und Implementierung von passenden Strategien, um den BenutzernInnen den Einstieg in die Technologie zu erleichtern und sie dazu zu motivieren, das Programm auch langfristig zu nutzen. Nicht alle Strategien sind für jeden Anwendungsfall geeignet, deshalb ist es wichtig, diese sorgfältig auszuwählen. Um den Einfluss der angewendeten Strategien zu evaluieren wurde eine Studie durchgeführt. Hierbei wurden zwei Chatbots miteinander verglichen. Der erste Chatbot beinhaltete nur wenig Elemente um den Benutzer zu unterstützen und zu motivieren, wohingegen der zweite Chatbot mehrere dieser Mechanismen inkludierte. Im Rahmen der Studie konnte festgestellt werden, dass dieser „verbesserte" Chatbot sowohl

die Akzeptanz der BenutzerInnen als auch deren Motivation den Chatbot zu nutzen positiv beeinflusste. Zusätzlich haben sich die Strategien auf die Wahrnehmung der AnwenderInnen ausgewirkt. So wurde er zum Beispiel auch als ansprechender empfunden. Um die Verwendung von Chatbots im Allgemeinen zu etablieren, müssen BenutzerInnen vermehrt mit der Technologie in Kontakt gebracht werden.

Contents

List of Figures

Listings

List of Tables

1. Introduction

Since users often describe navigating traditional websites to gather information as a difficult task, chatbots are a suitable alternative to overcome these problems. If the user needs some information it is possible to ask the chatbot for it instead of searching through multiple web pages (Drift, SurveyMonkey, Salesforce, & myclever, 2018). Furthermore, a chatbot responds in real-time and is available 24/7, whereas customer service hotlines typically have specific business hours. Nowadays it is not only possible to communicate with chatbots over a text-based interface, but also using voice-based interfaces becomes more and more popular (Klopfenstein, Delpriori, Malatini, & Bogliolo, 2017) and allow integration into platforms like Alexa and Cortana. Even though the popularity of chatbots only started to rise around 2016, the topic has already been around for several decades. During the chatbot history various approaches have been developed, raising from simple pattern matching dialogue systems (Weizenbaum, 1966; Epstein, 1992) to sophisticate chatbot architectures using machine learning and neural networks that are common nowadays (Fadhil, 2018). Natural language understanding is a complex task for computers and mobile devices (Goldberg, 2017). Scientists underestimated the complexity of the comprehension as well as the generation of human language for a long time (Hill, Ford, & Farreras, 2015). Also, the way how people interact with digital devices has changed in recent years and the expectations of users concerning language understanding and language processing are increasing (Klopfenstein et al., 2017). In 2016 the MIT Technology Review (Knight, 2016) highlighted conversational interfaces as one of ten breakthrough technologies. Even though until now thousands of text- and speech-based chatbots have been developed many of them struggle with high user drop-off rates as well as problems concerning language understanding and user acceptance. According to Debecker (2017) for some brands this user drop-off rate can go up to 40% after the first few messages. The user attention span is limited. Hence, user engagement and usability are crucial aspects to consider during the development process of a successful chatbot. To motivate the user to communicate with the chatbot for a longer period can be hard. Especially since engagement strategies developed for traditional systems like apps or websites are not often applicable to chatbot systems. Nevertheless, even if user engagement is integrated within the chatbot many users still might prefer talking to a real person over talking with a chatbot. The average user acceptance is still not high but it is already improving in some fields of customer service (Debecker, 2017; Garcia, 2018). Therefore, an effective onboarding process is crucial for the success of the system. It does not matter how well the chatbot is designed to fulfil the given user tasks if users do not start to interact with the system. Users have to be convinced to communicate with chatbots instead of relying on traditional information sources like websites or help desk employees. According to Petersen, Thomsen, Mirza-Babaei, and Drachen (2017) the onboarding phase, which refers to the first few minutes of the user interacting with a system, is crucial. Users that stop interacting with the system within this phase are unlikely to ever return.

1.1. Goals and Objectives

The purpose of this work is to extract and evaluate strategies to help increase user acceptance and user engagement within chatbot systems. In particular, the main focus of this thesis is the identification of successful onboarding strategies. The process consists of the planning, design, implementation and evaluation of an engaging chatbot. This includes:

- Collect, evaluate and categorize engagement and onboarding strategies for traditional systems as well as chatbot systems from the literature
- Examine the target group and conduct a focus group analysis to gather information about user perception and expectations
- Extract suitable onboarding and engagement strategies for the field of operas and theatres
- Design a chatbot architecture and conversational flow that involve these onboarding and engagement strategies
- Implement the chatbot as text-based as well as voice-based interface

- Evaluate how integrated onboarding and engagement strategies affect the user experience.

1.2. Methodology and Structure

The structure of the thesis consists of four main chapters: Background, Design, Implementation and Evaluation. Figure 1.1 displays the four chapters and highlights the important aspects covered within these chapters as well as what outcome each chapter provides that can be used in the following chapter. Chapter 2 is based on literature research and is responsible to provide background knowledge and strategies that are further processed to design decisions within the next chapter (Chapter 3). The implementation of the chatbot (Chapter 4) is based on these design decision. The last final output of the thesis are the findings gained by evaluating the developed chatbot (Chapter 5).

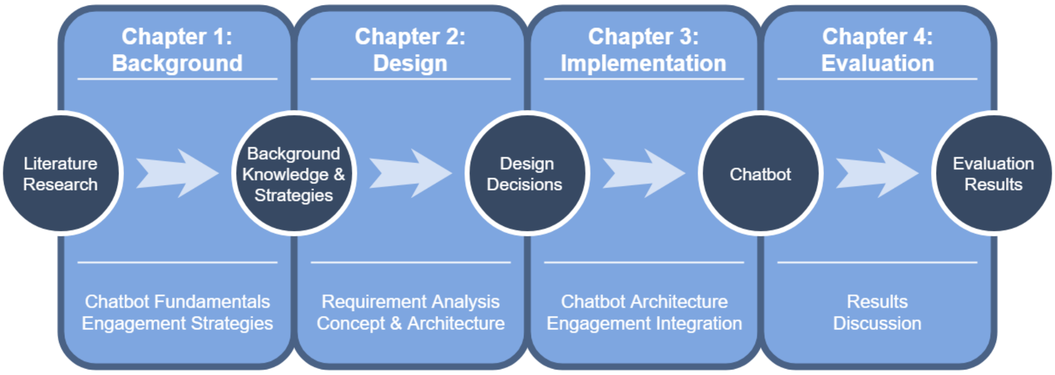

Figure 1.1.: The figure represents the four consecutive chapters of the thesis. Each chapter has some core aspects that are covered and the output of each chapter is then used as input for the subsequent one.

Within the first part (**Chapter 2**) theoretical background knowledge is used to introduce the reader to the fields of chatbots and user engagement. The purpose of this chapter is to ensure a deeper understanding of both topics and describe some fundamental aspects that are used in later chapters. After a general description of chatbots and their current state, basic components of a chatbot are identified and characterized. Nowadays there are already many tools available that support the developer with the implementation of a chatbot. As examples, the frameworks of the Tech Giants Facebook (Wit.ai), Google (Dialogflow) and Microsoft (Microsoft Bot Framework) are described shortly. Challenges that might arise when developing a chatbot are also part of this chapter. To understand the idea behind user engagement a brief description of the topic as well as the engagement process is provided. There already exist engagement strategies for traditional systems like applications or websites that can also be adapted and applied to be suitable for chatbot systems. Additionally, also engagement strategies in chatbot systems are discussed. At the end of the chapter, a summary of possible engagement strategies is provided.

Chapter 3 focuses on the requirement and the design of the chatbot. The first part of this chapter deals with the requirement analysis of the stakeholders. This includes the evaluation of the requirements for the customer and developer as well as for the future users of the chatbot. A target group analysis, as well as focus group interviews with people of different ages, are conducted. The obtained results are evaluated and user requirements are extracted. The stakeholder requirements are then used to identify the functional and non-functional requirements of the system. The next section focuses on the concept and architecture of the chatbot. This includes the selection of a suitable framework and the design of the language model. Also, the chatbot personality is identified.

The third part of this thesis (**Chapter 4**) covers the actual implementation of the chatbot. The setup of the framework, the language understanding engine and the question-answer catalogue are handled at the beginning of this chapter. This is followed by content integration. It deals with the logic behind the conversation as well as the preparation of suitable content to form the responses. The last aspect of this chapter focuses on the integration of interactivity, onboarding and user engagement strategies within the chatbot.

Within the last chapter (**Chapter 5**) the evaluation of the developed chatbot is covered. In this regard, a survey with two independent groups is conducted. One group is evaluating a chatbot without onboarding and with only basic user engagement elements and mechanisms. The other group interacts with the chatbot that includes several of the strategies described in this thesis. The evaluation of the chatbots consists of a pre-questionnaire, an interactive experiment and a post-questionnaire. In the chapter, the process of the survey, the used material and the participants are described. Subsequently, the results are evaluated and discussed.

2. Background

Chatbots are known under a variety of different names, ranging from intelligent virtual assistants, digital assistants or language interfaces (Dale, 2016) to conversational agents (Serban et al., 2017), dialogue systems or chatterbots (Ciechanowski, Przegalinska, Magnuski, & Gloor, 2019). Chatbots have been a very popular topic in recent years. Dale (2016) refer to a chatbot as *"any software application that engages in a dialogue with a human using natural language"*. Other papers (Abdul-Kader & Woods, 2015; Schumaker, Ginsburg, Chen, & Liu, 2007) described them similar: *as computer program that mimics intelligent conversation by using natural language*. Within the last decades, the main type of communication used by chatbots was written text, but due to the improvements in speech recognition in recent years also voice-based bots gain more and more popularity. Since many chatbot systems suffer under high drop-off rates of users after the first few messages a user-centred design is required to motivate him or her to try out and keep the user engaged with the system. The following chapter provides background knowledge concerning chatbots in general as well as user engagement in traditional applications but also covers some chatbot related strategies.

2.1. Chatbots

To interact with a chatbot the user is provided with a text- or speech-based interface that allows her or him to use natural language to communicate. Even though the idea of interacting with machines in a dialogue fashioned way was already addressed by Alan Turing in 1950, researches in this field were not very successful for decades. In short, the Turing Test invented by him dealt with the idea of a machine that cannot be distinguished from a human during extensive questioning. If the investigator is not able to tell which of the two test subjects (one person and one machine) was not human, the Turing Test is passed. The investigator is only allowed to use a monitor and a keyboard during the test and is located in another room (Hill et al., 2015). Since 1990 an annual contest - the Loebner Prize competition - takes place to find the chatbot with the most human-like behaviour during a Turing Test (Trazzi & Yampolskiy, 2018). *ELIZA* (Weizenbaum, 1966), developed at the beginning of the 1960s, was one of the first and most well-known chatbots. Simple pattern matching and a template-based response scheme were used to simulate a conversation with Rogerian psychotherapists. The program used the statements of the questioner, detected keywords and rephrased the statement to encourage the person to continue talking (Saygin, Cicekli, & Akman, 2000). Even though ELIZA was not able to pass the Turing Test, the approaches used can also be found in various later bots (Saygin et al., 2000). In literature, there are various approaches of chatbot architectures available ranging from simple pattern matching dialogue systems (Weizenbaum, 1966; Epstein, 1992) to sophisticated chatbot architectures using machine learning and neural networks (Fadhil, 2018). Another approach was applied for JABBERWACKY[1] in 1997. The idea behind JABBERWACKY was to create a chatbot that can learn from the users' behaviour and language style during the conversation and applies this knowledge to the dialogue with other users. Abdul-Kader and Woods (2015) compared chatbots developed between 1991 and 2014 and outlined the significant improvements. They highlighted chatbots until then were mainly focused on one specific target group and for future research, a focus on general-purpose dialogue system is required. Scientists have been underestimating the complexity of human language, concerning not only language comprehension but also its generation, for decades (Hill et al., 2015). In 2016 virtual assistants or chatbots were the most popular topic within the field of language technologies. The intelligent voice assistants of the Big Four - Siri[2] (Apple), Cortana[3] (Microsoft), Alexa[4] (Amazon) and the Google Assistant[5] are the most well-known examples in this field. But

[1] http://www.jabberwacky.com/
[2] https://www.apple.com/siri/
[3] https://www.microsoft.com/en-us/cortana
[4] https://www.amazon.de/b?node=12775495031
[5] https://assistant.google.com/

except these, there are thousands of other text- and speech-based chatbots used for various purposes, ranging from weather information to complex booking processes or just used for simple question-and-answer dialogues. The rise of chatbots is changing the way how people interact with their digital devices (Dale, 2016). The MIT Technology Review (Knight, 2016) refers to conversational interfaces as one of ten breakthrough technologies of the year 2016. William Meisel categorizes the large amount of chatbots in two main classes (Dale, 2016):

1. General personal assistants
2. Specialized digital assistants

General personal assistants have no predefined scope in which they operate. They can cover a wide range of services to help the user to achieve specific goals or are used just for entertainment. Examples for general personal assistants are Siri, Alexa and Cortana (Shum, He, & Li, 2018; Sarikaya, 2017). The term "specialized digital assistants" refers to chatbots that cover a more focused scope. They are designed to fulfil a certain task within a predefined scope. This can, for example, be a flight booking assistant (Dale, 2016; Shum et al., 2018). Even though the fundamental technology on which chatbots are based on is still the same as in the time of Eliza, the way people use their digital devices has changed a lot. SMS and Messaging Platforms have emerged, and most people use messaging-capable devices to communicate with others. Since chatbots can now be integrated into many of the big messaging platforms (like Facebook Messenger and Skype) the prerequisites for the success of chatbots are given (Dale, 2016). According to a survey (Drift et al., 2018) conducted with 1051 U.S. citizens in 2018 the three most frustrating problems with traditional online experiences are:

- Sites are hard to navigate
- Answers to simple questions cannot be found
- Basic details about a business are hard to find

The same survey also questioned the potential benefits chatbots will provide, which resulted in 24-hour service and instant responses as main advantages, as well as getting answers to simple questions and easy communications. Chatbots are an opportunity to overcome the problems of traditional online experiences by providing a real-time, on-demand communication platform.

Current State and Global Trends

Handling of natural language is a complex task for computers and mobile devices. At the beginning of natural language processing, rule-based methods were applied, but since the 1990s statistical approaches have been dominant in this field (Goldberg, 2017). The statistical method uses large quantities of empirical data to build its language model, whereas the rule-based approach was based on rules and a predefined vocabulary (Hirschberg & Manning, 2015). The continuous progress of researches in the fields of Artificial Intelligence, Natural Language Processing, and text-to-speech technologies influence how companies interact with their customers. Chatbots are already used on many websites to answer frequently asked questions (Pandita, Bucuvalas, Bergier, Chakarov, & Richards, 2018). Even though at the beginning of chatbots their main goal was to mimic human conversation users nowadays should be aware of the fact that they are talking with a machine (except certain exceptions). For most chatbots, it is better to not pretend to be a human (Klopfenstein et al., 2017). Nevertheless, the expectations of users concerning language understanding and language processing are increasing. In recent years a lot of frameworks and technologies have emerged that try to help simplify the development of a chatbot like Hubot[6] or Wit.ai. Many chatbots nowadays are hosted in the cloud and since 2014 many messaging platforms like Kik and Telegram enable the developer to integrate their chatbots into the messaging platforms. Application Programming Interfaces (APIs) are provided for this purpose and provide services like authentication and messaging as well as user interface (UI) elements, for example, buttons or images. Bot directories (lists of bots available to use within messaging platforms) are already established within popular messaging platforms (Klopfenstein et al., 2017). Also, the way how chatbots and users interact with each other has changed. Xu, Liu, Guo, Sinha, and Akkiraju (2017) categorize two types of user requests, informational request, and emotional requests. While the goal of the first one is to gather information concerning a topic or problem, the second one refers to an expression of emotion or opinion. Social chatbots have emerged in recent years to cover

[6]https://hubot.github.com/

emotional requests. They take the emotional level of the conversation into account and build an emotional connection to the human (Shum et al., 2018). Nowadays chatbots do not only provide text, they rather can be enriched with a variety of different data types like images, videos or emoticons. Also, markdown is available on many platforms to style the content. In former years chatbots where only able to process textual input whereas today speech is another popular communication channel. Text-based chatbots can be integrated into messaging platforms like Facebook Messenger or Slack, whereas voice-based chatbots, on the other hand, can be integrated into virtual assistants like Amazons Alexa or Microsofts Cortana (Klopfenstein et al., 2017; Paikari & van der Hoek, 2018).

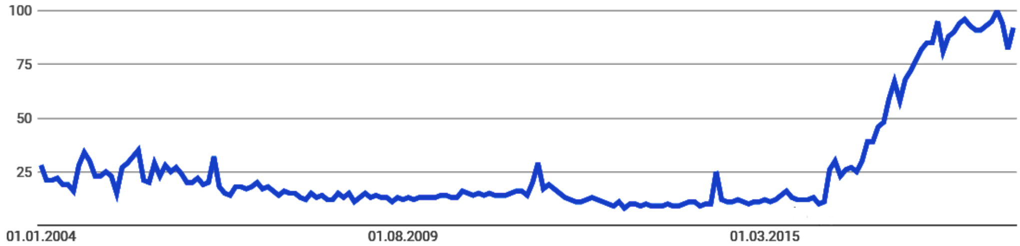

Figure 2.1.: Google Trend Analysis on "Chatbot" Search Requests from January 2014 until January 2019

As visible in Figure 2.1 the Google Trend Analysis[7] shows a rapid increase in search requests concerning chatbots from 2016 until 2019. Many online articles describe 2016 as "the year of the chatbot" and the hype around them has not stopped yet. According to the study "Global chatbot market 2025" of Grand View Research (2017) chatbots are still a rising trend. The research estimated that from 2016 to 2025 the revenue generated in the global chatbot market will rise from 190.8 million to 1250 million U.S. dollars worldwide. This is a rise of more than 600%. Also, Gartner's Hype Cycle (as displayed in Figure 2.2) concerning digital marketing and advertising published in December 2018 outlines the continuously growing interest in the topic of Artificial Intelligence. Also, conversational marketing is mentioned. It includes technologies that enable a company to interact with

[7]https://trends.google.com/trends/

the customer in a dialogue fashioned way, like chatbots. According to Gartner conversational marketing has almost reached the Peak of Inflated Expectation phase (“Gartner Hype Cycle,” 2018). This means success stories for conversational marketing exists, but many companies struggle with this new approach.

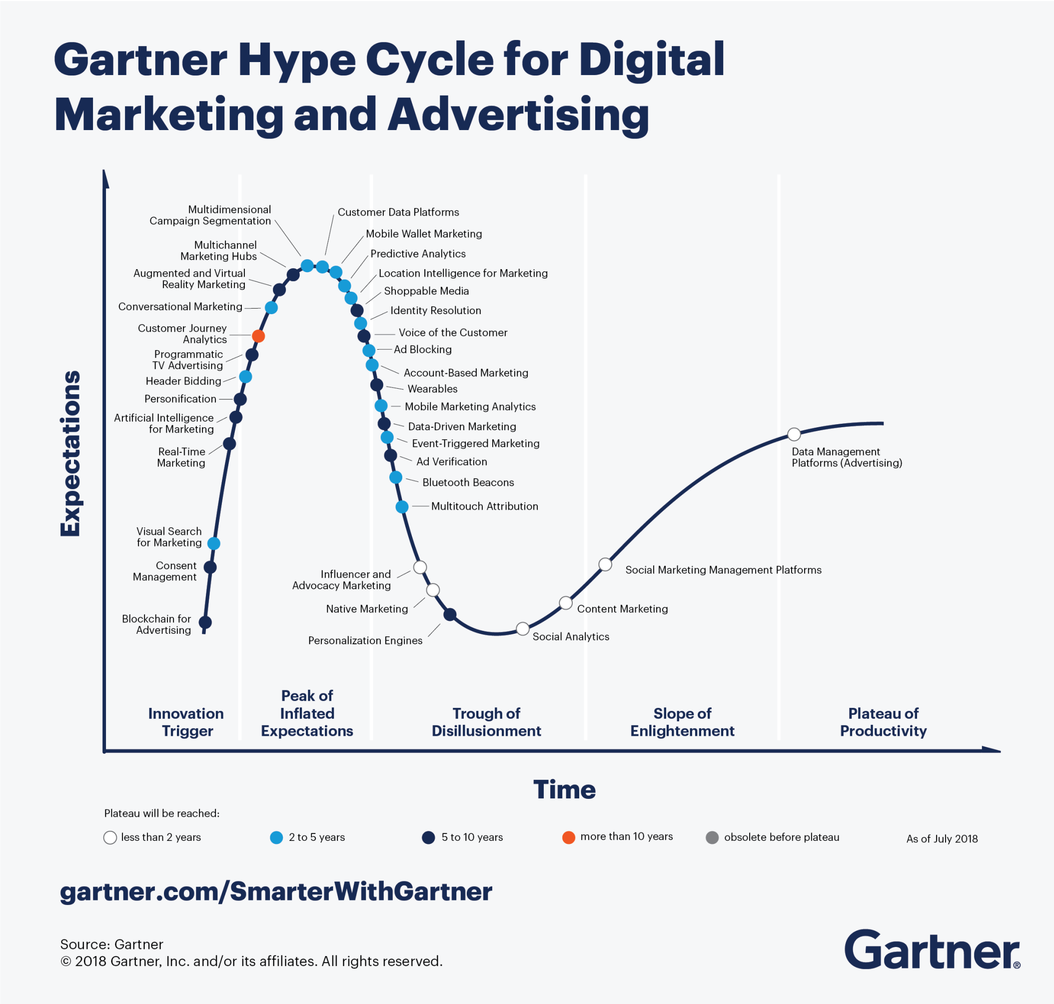

Figure 2.2.: Gartner’s Hype Cycle for Digital Marketing and Advertising illustrates that conversational marketing has almost reached the Peak of Inflated Expectations (“Gartner Hype Cycle,” 2018)

2.1.1. Components of Chatbots

Even though there exist many different chatbot architectures within various in recent papers similar components of a chatbot architecture are described, even though they refer to them with different names. Kompella (2018) lists, as shown in Figure 2.3, NLU, Dialog Manager and Message Generator as general components of a chatbot. Additional to these components there might also be a Channel Connector involved, that enables the chatbot to send and receive messages to/from different platforms. Abdul-Kader and Woods (2015) used the terms "Responder" (interface between chatbot and user), "Classifier" (preparation of input, transformation into logical components) and "Graphmaster" (information storage, pattern matching) to describe their architecture. Rahman, Al Mamun, and Islam (2017) describes the classification part as two separate modules (the intent and the entity recognition module) to process the user's input. The response selector chooses the most suitable response candidate which is then sent to the user. Since the mentioned architectures are very similar, in this thesis only the architecture of Kompella (2018) is described in more detail.

Natural Language Understanding (NLU) is a subfield of Natural Language Processing (NLP). NLP, also known as computer linguistics, refers to computational techniques to handle human language. It covers learning, understanding and producing of human language and is part of the field of computer science (Hirschberg & Manning, 2015). Researches in the field of NLU focus on the complex task to enable machines to read and understand human language. When extracting the meaning of natural language its inherent complexities have to be overcome (Navigli, 2018). Navigli (2018) describes text understanding as more than just the processing of strings. According to him: *"Text implies knowledge of concepts of the real world, it requires further reasoning, it arouses emotions."* Natural Language Understanding is the main and most complex part of a chatbot. NLU is responsible to extract information from the user input. This information includes the purpose of the user request as well as possible parameters (intents and entities). To enable the NLU to provide a good intent and entity extraction result, training data in the form of sample conversations has to be prepared and the language model has to be trained. The more paths of a conversational flow are covered the better the predicted result. This process is called interactive

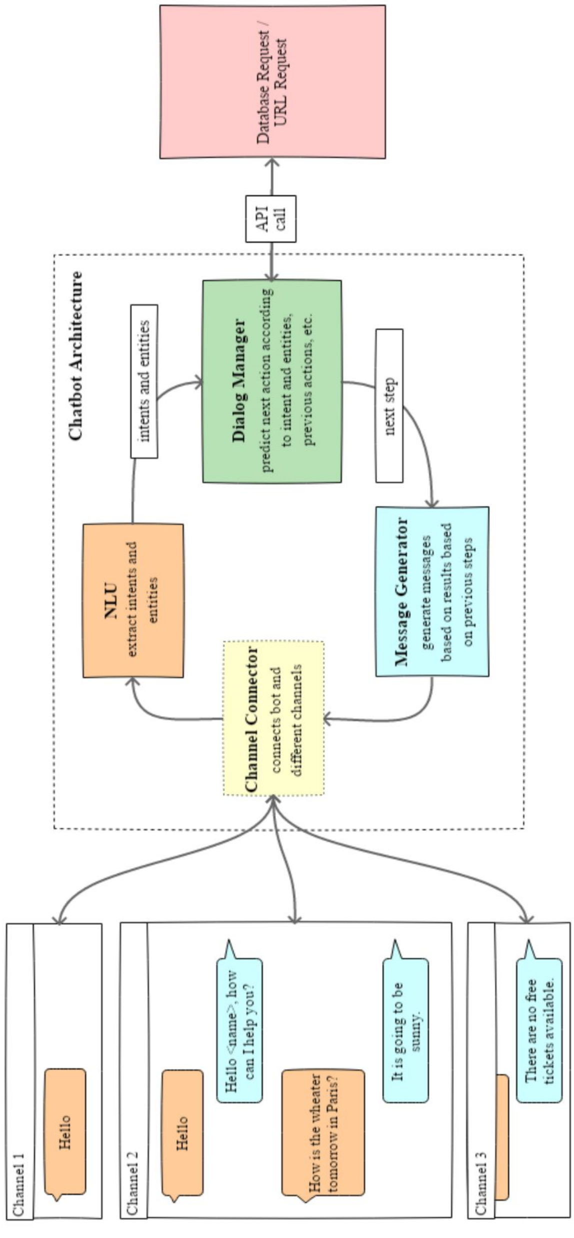

Figure 2.3.: Chatbot Architecture adapted from Kompella (2018)

learning.

One task of the **Dialog Manager** is to keep track of the previous conversation to predict appropriate responses for future requests. Also, intents and entities extracted by NLU take an effect on the response prediction. Based on previous conversations the dialogue manager keeps track on persistent information that can improve the predicted reply. The dialogue manager handles the logic that is necessary to transform the given input into a valid output by predicting the next action like responding to the user, retrieving some data from the database or making an API call to enrich the responses with external data.

The **Message Generator** is responsible to display the appropriate response to the user. It uses predefined templates that are selected based on the results of the dialogue manager. Placeholders within the templates are replaced by the data that is retrieved by the dialogue manager. After the valid response is constructed it is passed to the user.

The **Channel Connector** is a part of the chatbot architecture that has not necessarily required. It is responsible to retrieve the user input from one platform and pass it to the chatbot as well as returning its response to the correct platform. A channel is a distribution channel that allows a chatbot to communicate with users of different platforms like Facebook Messenger or Slack. Messages of the chatbot are retrieved and converted to a valid message for the desired platform and vice versa. Therefore it is possible to develop one chatbot and integrate it within many different platforms without any modification(Kompella, 2018).

Nowadays there already exist tools that help the developer to create chatbots more easily. Especially when dealing with NLU it is convenient to be able to not reinvent the wheel but instead rely on provided services and adapt them to own needs. A basic distinction between the available categories of tools and a description of three available frameworks are provided within the next chapter.

2.1.2. Tools and Frameworks

Since chatbots have been the trending topic in 2016 and still are on their rise a huge variety of different tools and frameworks have emerged to ease the development for users with and without programming skills. Rahman et al. (2017) classifies the available platforms into three different categories:

1. **Nonprogramming Chatbots**
   Basic chatbots can be programmed without any deeper coding or machine learning knowledge or Natural Language Processing expertise. The developers do not have to deal with the underlying technology. Mohanoor (Mohanoor, 2018) refers to this kind of chatbots as Conditional Logic bots. They use scripted conversation and pattern/keyword matching for communication. To interact with this kind of bot the user is typically provided with buttons or list selections. Examples for this type of platforms are Chatfuel[8] and ManyChat[9].

2. **Conversation-Oriented Chatbots**
   Chatbots categorized as conversation-oriented chatbots use Artificial Intelligence Markup Language (AIML) to transform user input into chatbot responses (Rahman et al., 2017). AIML is a type of XML and is used to store patterns and responses during user conversations. The ALICE system (Artificial Linguistic Internet Chat Entity) is an example of this type of platform and provides about 24,000 patterns (Wallace, 2003). Even though AIML can be used to customize chatbots, it is not suitable for long conversations due to its limitations. Therefore AIML chatbots are mainly used for chitchat (Shum et al., 2018).

3. **Platforms by Tech Giants**
   Google, Amazon, Facebook, IBM, and Microsoft also developed platforms to build chatbot systems. The frameworks of the tech giants try to understand what the user wants (Mohanoor, 2018). They provide built-in Natural Language Understanding and can be used to build more complex conversation flows.

[8]https://chatfuel.com/
[9]https://manychat.com/

The following section covers the platforms of Facebook (wit.ai), Google (Dialogflow) and Microsoft (Microsoft Bot Framework) in more detail.

Wit.ai

In April 2016 Facebook launched Wit.ai[10] as a tool to build bots that can be addressed within its messenger platform. Facebook's vision for Wit.ai is the empowerment of developers by providing an open and extensible natural language platform ("About Wit.ai – Medium," n.d.). Over 180.000 developers make use of Wit.ai to integrate natural language understanding in their applications and devices. The Natural Language Understanding (NLU) is improved with every interaction and all developers can profit from this progress. The main channel for Wit.ai chatbots is the Facebook Messenger, but it is also possible to integrate the chatbot into own websites or apps. The frameworks natural language skills can be used for chatbots, mobile apps, home automation, wearable devices as well as hardware. Wit.ai is trained by examples. This means example user input is inserted into the User Interface of Wit.ai and the tool will expand these examples by additional possible phrases. The more examples are used to train the language understanding engine the better the chatbot will understand the user requests. An entity is a piece of information that is extracted from the user message. Trait entities are used to detect the intent of a user's request. An example of a trait entity is a weather request. Additionally, to trait entities also keyword entities and free text entities are provided. Keywords can be used to detect entities that belong to a predefined list whereas free text entities extract substrings of the message. The more valid examples are given the better Wit.ai will predict the entities. Wit.ai provides the possibility to create open or private apps. Some components of open apps like Intents, Entities, and validated expressions are accessible to the community. To develop chatbots with Wit.ai the platforms Node.js, Ruby and Python are available. For other platforms, the HTTP API has to be used. The framework supports over 50 different languages, including the main languages like German, English, and Spanish, but also a variety of African languages. Wit.ai is free for private as well as for commercial usage without any rate limitations (Wit.ai, n.d.).

[10] https://wit.ai/

Dialogflow

Dialogflow[11], former api.ai, is another tool to develop voice and text-based interfaces. Api.ai was launched by Speaktoit in September 2014 as a natural language understanding platform (WIEREMA, 2014). Two years later, in 2016, Google bought the company and renamed it to Dialogflow. Googles reason for this purchase was to help developers building apps for the Google Assistant, a virtual assistant powered by AI. The main targets are mobile and smart home devices. The newest version of Dialogflow (Version 2) SDKs and underlying APIs are available in following programming languages: Node.js, Python, Java, Go, Ruby, C# and PHP. Dialogflow handles the natural language understanding for the developer. An agent is used to handle the user input and transform it into structured data that can be processed to return a suitable response. Intents are used to define how inputs and responses should be mapped. An intent typically consists of the following parts: training phrases, action and parameters and responses. Training phrases should be provided to simulate possible user inputs. The natural language engine expands these sentences and phrases with additional similar phrases to be able to detect the users intent more accurately. Further training enables a more robust model. Dialogflow allows specifying entities that are later detected as parameters. Dialogflow extracts these entities from the user input and enables further processing of them within a custom logic called "fulfilment". Here the developers can include the logic to customize the chatbot's responses. Responses can be text- or speech-based as well as visual. Sending a reply is achieved either by the build-in response handler or by calling the "fulfilment" logic to process the input and return a valid response to Dialogflow. Fulfilment is just a simple Webhook that enables the user to call web server endpoints and return their responses to the user. Additionally, follow-up intents can be specified to construct a conversational flow just by using the Dialogflow graphical user interface. The framework makes use of Googles machine learning expertise and runs on the Google Cloud platforms with all its advantages like its scalability. As already mentioned, it is mainly used to build apps for the Google Assistant. Nevertheless, it also supports a huge variety of other channels. These channels are Facebook, Slack, Twitter, Viber, Twilio, Telegram, Skype,

[11]https://dialogflow.com/

Kik, LINE, Cisco Spark, Cisco Tropo. Additionally, it provides the possibility to export the required data, like Intent Schema and Sample Utterances, to integrate it into an Alexa Skill as well as import the data of the Alexa Skill into a Dialogflow chatbot. The same can be done with Microsoft Cortana. At the moment there are 20 different languages supported by Dialogflow including English, Spanish, Chinese, and German. Dialogflow is available in two different editions. The standard edition is available for free. It fulfils the requirements for the majority of developers. Except for Text or Google Assistant, all provided features have a maximum request limitation. Toll-free phone calls are only available in the Enterprise Edition. This edition uses a "pay as you go" model. Text requests cost $ 0.002 or $ 0.004 depending on the selected Enterprise Edition. All features are available unlimited and the user pays a small amount of money per request (Dialogflow, n.d.).

Microsoft Bot Framework

The Microsoft Bot Framework[12] was first announced at the Build 2016 in San Francisco (Mayo, 2017) and can be used by developers to build their chatbots. The Microsoft Bot Framework consists of two main components, the *Bot Builder SDK* and the Cognitive Services. Additionally, the chatbot can be easily connected to different channels if it is deployed on the Microsoft Azure platform by using the "Bot Connector Service". With "*Azure Bot Services*" an integrated environment is provided to build, deploy and connect a chatbot. Templates enable users to set up their chatbots within a few seconds. Azure Cognitive Services can be used to build intelligent chatbots without the requirement of having deeper knowledge concerning the topics AI, data science or machine learning. The cognitive services include APIs, SDKs, and services to enable the user to add features concerning the fields language, speech, vision, knowledge and search. Available services are for example the language understanding intelligent service (LUIS[13]) and the Bing Spell Check service, but also text analytics and text-to-speech/speech-to-text capabilities. Microsoft LUIS is available in 13 different languages including German, English, and Spanish. It is responsible for the Natural

[12]https://dev.botframework.com/
[13]https://www.luis.ai/

Language Understanding (NLU) of the chatbot and makes use of the concept of Entities and Intents. To build a LUIS language model a developer can use predefined domains with predefined Intents and Entities or create a custom model by defining own Intents and Entities. Intents are used to specify the user intention. This means they represent goals a user wants to reach with his or her input. This input is called ”utterances” within the LUIS domain. An intent can, for example, be booking a flight or reserving a table. To be able to improve the recognition of the user intention, a variety of sample utterances for each intent should be provided. This active training is important for the underlying machine-learning intelligence of the language understanding service. Additionally, to the concept of intents also entities are a main part of the LUIS language model. Entities can be described as keywords or phrases within an utterance that should be detected. An entity generally is used to represent a class or collection of similar items like places, dates or people. Utterances can contain entities, but they do not have to. Another service that can be integrated into the developers' chatbots is the QnA Maker service[14]. The user can define multiple questions and answer pairs that can be accessed by using the provided API. This service also makes use of Natural Language Processing to improve the prediction of the users desired answers. Besides the concept of intents and entities, the developer is provided with the possibility to use patterns or phrase lists. Phrase lists represent words or phrases that belong to the same class and have to be handled similarly. It can be used to define synonyms or add app vocabulary that is required for the scope of the chatbot. By using ”Azure Bot Services” the user can take advantage of the ”write once run anywhere” methodology. This means the chatbot is developed once and can be provided across multiple channels. At the moment Email, GroupMe, Facebook Messenger, Kik, Skype, Slack, Microsoft Teams, Telegram, SMS, Twilio, Cortana and Skype for Business are supported. By using the Direct Line API it is also possible to integrate the chatbot into own websites and apps. To build a chatbot the developers are provided with the Bot Builder SDK. It is currently available for C#, JavaScript, Java, and Python and is available for free. ”Azure Bot Services” is free for Standard channels. Premium channels are free for up to 10,000 messages/month and cost € 1.265 per 1,000 messages after this limit is exceeded (Microsoft, 2019a).

[14] https://www.qnamaker.ai/

Table 2.1 summarizes the most important aspects of the frameworks. The compared features are the launch year, machine learning capabilities, the used concept, the existence of prebuild and composite entities, prebuild domains, the supported languages, SDKs provided to develop the chatbot, as well as the pricing models and available distribution channels.

	Wit.ai	**Dialogflow**	**Microsoft Bot Framework + LUIS**
Launch Year	2013 (2016 by Facebook)	2014 (2016 by Google)	2015
Machine Learning	Yes	Yes	Yes
Concept	Intent & Entities	Intents, Entities & Actions	Intents & Entities
Prebuild Entities	Yes	Yes	Yes
Prebuild Domains	No	Yes	Yes
Composite Entities	No	No	Yes
Supported Languages	50+	20	13
SDKs	Node.js, Ruby, Python	Node.js, Python, Java, Go, Ruby, C#, PHP	C#, JavaScript, Java, Python
Pricing Model	Free	Standard for Free or Enterprise as pay as you go	10,000 transactions free then pay as you go
Channels	Facebook Messenger, Custom App & Website (using API)	Facebook Messenger, Slack, Twitter, Viber, Twilio, Telegram, Skype, Kik, LINE, Cisco Spark, Cisco Tropo, Alexa, Cortana, Custom App & Website (using API)	Email, GroupMe, Facebook Messenger, Kik, Skype, Slack, Microsoft Teams, Telegram, SMS, Twilio, Cortoana, Skype for Business, Custom App & Website (using API)

Table 2.1.: Enhanced comparison of Wit.ai, Dialogflow and Microsoft Bot Framework + LUIS

2.1.3. Challenges and Limitations

Even though the topic of chatbots is very popular in recent years and the available tools for the creation of chatbots have improved, there are still some challenges a developer has to overcome to create a chatbot that users want to talk with.

Natural Language Understanding

Hill et al. (2015) mention issues not only concerning the understanding of words and phrases but moreover regarding the huge variety of possibilities how words can be combined to express certain meanings. Alpana (2017) describes the different ways of how users interact/write with the chatbot can be a challenge. This does not only refer to the way of texting (short sentences, long sentences, keywords) but also the user's language (usage of slang, misspelling, abbreviations). A chatbot has to be trained to react to these differences between the users to provide good answers. Even though NLU nowadays can categorize synonyms correctly in most cases, the mixing of local languages, abbreviations, and slang words is still hard to handle. Chatbots should be able to defer between questions to answer and general phrases or chit-chat like "thank you" or "ok". It is not appropriate if the chatbot tries to lookup information in the database if the user asks questions like "How are you?" (Alpana, 2017). Additionally, language support for NLU can be a problem. Many languages like English, German, Spanish and French already have good NLU support, whereas the correct understanding of the meaning when using less used languages can arise problems.

User Acceptance

A survey of May 2018 in the United States targeting internet users (Garcia, 2018) dealt with the main challenges of chatbot usage. According to this study about half of the interviewed audience sees the chatbot preventing the user from talking to a real person as a big concern. The high amount of irrelevant responses is also considered as a huge problem. Directing the user to FAQ pages and long response times are listed as causes that prevent customers from using a chatbot. According to a study of Pega (2017) that analysed the acceptance of Artificial Intelligence chatbots worldwide in the year 2017 there are already some fields of customer service where chatbots gain more and more acceptance. The leading field in this respect is Online retail service. Here already 34% of customers prefer talking to chatbots over talking to a live customer service representative, whereas within the governmental sector only 10% of users prefer chatbots. In other fields like health care and telecommunications, every fourth person favours communication with chatbots. Users expect chatbots to react in a proper way to benefit from the conversation. If the user feels dissatisfied with the chatbots replies several times he or she likely stops using it (Alpana, 2017). Additionally, also the limited user attention span can be challenging. It is hard to develop chatbots that can hold the user's attention since she or he is expecting very fast response times. To prevent user frustration chatbots should forward the dialogue to a real human in case they are not able to provide the desired information (Gurwani, 2018). Since user acceptance is not always given and has to be earned it is very important to apply user engagement strategies during the development of a chatbot to improve the acceptance of the user.

2.2. User Engagement

Lehmann, Lalmas, Yom-Tov, and Dupret (2012) describes engagement as *"the quality of the user experience that emphasizes the positive aspects of the interaction [...] and so being motivated to use it.* According to them, users invest time, attention and emotion into engaging systems and therefore an engaging system design should be one main goal when developing a new website or application. Kim, Kim, and Wachter (2013) summarized the findings of Pagani and Mirabello (2011) on the term "engagement" as *state of being involved, occupied, retained, and intrinsically interested in something*. According to O'Brien and Toms (2008) the user engagement process can be split up into four phases: point of engagement, the period of engagement, disengagement, and reengagement. The **point of engagement** refers to the reason why user interaction started. This can have multiple causes like personal interest, information retrieval or social motivations. Also, aesthetics can be a point of engagement. This step includes elements that attract users attention and initiate engagement. In this paper, this step is also referred to as **onboarding** process. The paper of Renz, Staubitz, Pollack, and Meinel (2014) describes onboarding as *"the sum of methods and elements helping a new user to become familiar with a digital product"*. Onboarding strategies should help the user to smoothly get engaged in a digital product. The time, while a user is engaged, is referred to as **period of engagement**. The user engagement relies on the interest of the user for the interaction. Important aspects to keep the user engaged include the aesthetics of the system provided feedback and information as well as the level of user control during the interaction with the system. The next phase is called **disengagement** and refers to the point at which the user wants to stop the interaction (for example when losing interest) or the engagement is dropped. The engagement can decrease due to the system (like usability issues) or external factors (like interruptions or distractions). Nevertheless, the state of disengagement has not to be the end of the engagement process. If the previous experience with the system was perceived as positive the user is likely to return to use the system again. The motivation behind this can be the fun provided, rewards or discovering of new information. This is called **reengagement**. Reengagement can happen multiple times during the users' interaction with the system. To mention is also that barriers can prevent the user from being or becoming engaged. For example, this can be caused by poor usability. The result is **nonengagement** of the user while interacting with the system. O'Brien and Toms (2008) additionally identified attention, aesthetics, interest, challenge control, motivation, novelty and feedback as influencing attributes of engagement. Even though not every strategy of traditional systems can be applied on chatbots, many of them can be adapted to support the chatbot engagement process. Therefore this chapter does not only analyze engagement strategies for chatbots, but also traditional systems. In addition to describing engagement strategies, in general, a focus is set on onboarding strategies.

2.2.1. Engagement in Mobile Apps and Websites

User engagement is a major challenge in most of the digital systems nowadays ranging from simple websites to applications for the various kinds of digital devices like mobile phones, smart TVs or wearable devices. Asimakopoulos, Asimakopoulos, and Spillers (2017) point out, that every third owner of a fitness tracker stops using the device within the first year. For people downloading activity tracking apps onto their mobile phone, this period is even shorter. After two weeks 62% of users did not utilize the application anymore. Competition with others and additional information like step count or burned calories are used to increase user engagement. Self-efficacy and motivation are aspects that are outlined as an integral part of the user engagement in this field and also feedback can influence the engagement level of the user (Weston, Morrison, Yardley, Van Kleek, & Weal, 2015). Boiano, Bowen, and Gaia (2012) highlight the importance of suitable content for the required platform. Not only the quality of the provided information but also the used media has to be evaluated. Texts targeting mobile devices should be short or at least divided into short paragraphs. Longer texts are suitable for tablets or computer screens but not for the small screens of mobile devices. Images and Videos on mobile devices suffer from the small screen size and quality loss, but they can help to get the user emotionally involved and engaged. Also, Gamification is about engaging the user at an emotional level. It is a common approach to increase user engagement within applications. Deterding, Khaled, Nacke, and Dixon (2011) define Gamification as "*the use of game elements and game-design techniques in non-game contexts*". For example, a leaderboard or badges can be used to motivate the user to use the application more frequently. Showing the progress of achieving a goal is an example of how the user can be engaged at an emotional level (Werbach & Hunter, 2012). To a large extent, the approach is based on intrinsic motivation as this engages the user at a deeper level (Burke, 2014). Intrinsic elements are challenges, curiosity and fantasy (Garris, Ahlers, & Driskell, 2002). In contrast to extrinsic motivation, which is described as "*doing something because it leads to a separable outcome*", intrinsic motivation is referred to as "*doing something because it is inherently interesting or enjoyable*" (Ryan & Deci, 2000). If applied right, Gamification is an effective tool to change behaviours, develop skills or encourage innovation (Burke, 2014). The design of the user interface for mobile applications as well as websites should strive for simplicity and intuitiveness. Buttons and links are often hard to use correctly with fingers on mobile devices. To provide good accessibility for all systems the clickable area of these elements can be enlarged (Boiano et al., 2012). O'Brien and Toms (2008) compared researches concerning the theories of flow, aesthetics, play and information interaction to get a deeper understanding of engagement. They defined aesthetics, effective appeal, challenge, and feedback as engaging attributes. Additionally, also sensory appeal and motivation influence the level of user engagement. Nevertheless not every attribute has to be present for an engaging experience, this depends on the field of application. They identified engaging attributes for the field of video games, educational applications, online shopping, and web search.

Onboarding

As already mentioned it is not only important to engage users to use the system, but even more important to engage the user to start using the system in the first place. Petersen et al. (2017) stated that the first few minutes of the user interacting with an application, the point of engagement, are crucial. Users that stop using the system within this phase are unlikely to return. According to van Drongelen and Krishnaswamy (2017) the appearance of the mobile application should be interesting at first sight and should explain why the user should use the application by providing at maximum four benefits. According to them, this can be achieved with an introduction view, for example by providing an introduction story and "call-to-action"-elements (like buttons with explanation texts). Renz et al. (2014) suggest the use of common user interface and user experience patters as they are already known by most users and therefore no explanation is needed. Apps following the same design pattern are easier to handle for new users. If new design patterns are applied, a further explanation might be necessary. Also, video tutorials are mentioned as suitable onboarding strategy, depending on the field of the application. Gamification, as described earlier, can be a successful onboarding mechanism too.

Even though not every engagement and usability strategy of traditional websites and apps applies to chatbots there are a lot of ideas that can be taken and applied on chatbots too. For the onboarding phase within chatbot systems following ideas are extracted:

- design a nice-looking interface to attract the user
  - make use of common user interface (UI) and user experience (UX) patterns
  - focus on simple and intuitive interfaces
- try to engage the user on an emotional level
  - use Gamification elements
  - use media to support the content (if appropriate)
- motivate the user to make use of the system
  - providing additional ("nice-to-have") features that are not part of the core functionality of the system
  - use an "introduction view" to explain how the system works
  - include "call-to-action" components, for example buttons
  - use video introductions if appropriate

To keep the user's attention not only during the onboarding phase but for a longer period also following aspects should be taken into account:

- ensure accessibility of buttons and links for different devices (if required)
- prepare content suitable for the given platform
- give good feedback for the user
- ensure a high quality of the provided information

2.2.2. Engagement in Chatbot Systems

According to Debecker (2017) a huge problem for chatbots is the high drop-off rate of the users. Since some brands even experienced a drop-off rate of 40% after the first few messages, user engagement should be considered as an integral part of the chatbot design process. The following enumeration contains engagement strategies described in the literature, that have been collected and categorized:

1. **Onboarding**
   Følstad and Brandtzæg (2017) describe a chatbot interface as *"blank canvas"*. The provided content and visual aids are based on user input. Even though a chatbot nowadays can provide text, video, images, gifs and audios these features are mostly just provided on a user request. Therefore onboarding is a significant factor for good usability, especially when dealing with new users. The user has to understand not only how to use the chatbot but ideally also why the chatbot is worth using. The first impression of the chatbot counts (Botanalytics, 2017). Luger and Sellen (2016) describe the inexperienced user as one main challenge concerning conversational interfaces. People have to get used to the new technology to make the systems successful. According to them the level of satisfaction and trust are influenced by the time invested by a person. As soon as the user identified the scope covered by the program and practised its usage, the interaction results in higher satisfaction and trust. A playful entry, like finding 'Easter eggs', can help the user to get a better understanding of the scope of the system and familiarize her or him with the conversational interface. Playful interactions often result in a low drop-off rate. In this playful phase users typically are more likely to forgive failure, but this ends after the amusement passes (Luger & Sellen, 2016; Følstad & Brandtzæg, 2017).

2. **Usability**
   According to Peras (2018) chatbot usability is defined by efficiency and effectiveness, which refers to the effort and time needed for a user to reach one desired goal. User satisfaction, on the other hand, is described as *the users pleasure arising from the comparison of their expectations and chatbot performance* (task completeness, promptness,

and appropriateness). A minimum requirement of a chatbot is that the user can fulfil tasks, that are covered by the scope of the chatbot, smoothly. These so-called happy paths have to work since otherwise, the bot is useless (Lee, 2018). A chatbot designed to fulfil a variety of different needs might not be the best choice when dealing just within a certain domain and specific requirements. Considering the context, content, and kind of interaction can have a positive impact on the user's success to get the desired information (Lockton, 2010; Fadhil, 2018). Users that do not know what the system can or cannot do are often overwhelmed or frustrated by the limited tasks they can fulfil successfully (Luger & Sellen, 2016; Følstad & Brandtzæg, 2017). If a chatbot is designed to communicate in a logical order the user is more likely to achieve her or his goal and prevent frustration due to dead-ends. The suitable logic to apply often depends on the topic as well as the purpose of the chatbot (Fadhil, 2018). Edge cases that the chatbot is not designed to cover have to be evaluated. If the value gained from responding to an uncovered user request is high and the effort to develop is low this case should be incorporated. Users are very likely to talk to chatbots (and computers in general) in the same way they also would communicate with other persons. Therefore, the chatbot also has to react well to off-context requests (Fadhil & Schiavo, 2019). If a user request targets a topic that the chatbot is not designed for it is important to inform the user about the chatbots scope instead of a plain "error" message. Good feedback can help to get the system work. Otherwise, the user might try to rephrase the same question multiple times and gets frustrated. To be able to recognize additional use cases of the chatbot it can be useful to save these requests. Furthermore, to cover unpredicted requests some strategies to simulate understanding are suggested. Giving the chatbot a *personality* might help overcome inappropriate questions. *Small talk* (or chitchat) can be used to establish an emotional connection between the user and the program as well a avoid silence. If the chatbot is *directing the conversation* it can keep the conversation with little effort. Also *failures* like typing errors or rephrases make the conversation more human-like (Maria João Pereira, Coheur, Fialho, and Ribeiro, 2016; Maria Joao Pereira and Coheur, 2013; Katkute et al., 2017). To improve the user satisfaction level real test users of the target audience should be included during the development process. This provides the opportunity to already detect some possible edge cases (Lee, 2018). Even though usability is a core aspect of user engagement also other factors have to be considered to provide an overall experience of a product (Fadhil, 2018).

3. **Content & Design**
Design patterns that are successfully applied to graphical user interfaces do not work for chatbot user interfaces (Fadhil, 2018). According to Følstad and Brandtzæg (2017) the overall chatbot design has to move from *explanatory* to *interpretational*. This means instead of informing the user about accessible content and features the design should focus on understanding the users' needs and the best way how to fulfil them. Since the chatbot can make use of visual elements in most of the available channels this is a good way to keep the user engaged. Visual elements can be for example images, videos or gifs. Debecker (2017) reported a decrease in the user drop-off rate of about 19 % when using visual elements during the conversation. Fadhil and Schiavo (2019) summarizes several research papers that outline the importance of simplicity in interaction. It is recommended to move from general conversation to specific topics or requests. Complicated conversation flow paths should be avoided since these are error-prone. Autocomplete lists or button lists can help users to reach their goal faster and with less effort. Additionally, this also has the advantage of minimizing the risk of dead-ends. Chatbots use the concept to communicate in a chat-based manner. Therefore the messages should be kept short. Their purpose is to encourage users to interact with the chatbot and not overwhelm them by an immense amount of text. An excessive amount of text will lead to users that lose their interest in the conversation and drop-off Debecker (2017). According to Lee (2018) another important aspect to establish a high user satisfaction is the used language. If the chatbot acts within a formal environment the language should be adapted to this situation. Debecker (2017) suggest, as, in real marketing, the chatbot should be directed to a specific target group. Chatbots that continually provide relevant and interesting information without overwhelming the user with the provided data have a good chance to keep the users' interest offer a long period (Fang et al., 2018). To keep the user engaged a chatbot it is good to contain personalization ele-

ments like using the user's name during the conversation. In a survey conducted by Portela and Granell-Canut (2017) participants felt more engaged if the chatbot was able to remember parts of the conversation (like the participant's name) but also when it asked personal questions. Also Lockton (2010) mentions that a user is more engaged in a system if it somehow mirrors or mimics the user's mood or behaviour like *ELIZA* did. Nevertheless, the chatbot has to be designed to use personalization tokens only when it is appropriate but never overwhelm the user with the amount of collected user data (Debecker, 2017).

4. **Error Handling**
   Users might try to get certain information multiple times even though the chatbot replies that it does not know the answer. To prevent user frustration in this matter an escalation scenario can be implemented that redirects a user to a real human or at least provide some information to contact a real human (Debecker, 2017). If a chatbot relies on external services unpredictable errors may occur. The developer should make sure to test these services regularly to prevent issues concerning these services. Also, the user has to be informed about these unpredictable errors by a suitable message to prevent a decrease of the user satisfaction level (Lee, 2018).

5. **Reengagement**
   Reengagement within a chatbot system does not only refer to engage the user again after she or he stopped using the bot but rather also is used to motivate the user to continue communicating with the system during a conversation. Shevat (2017) as well as Fadhil and Schiavo (2019) describe notifications as an efficient tool to engage the user to start communicating with a chatbot again. Nevertheless, the timing of the notification is important. This can, for example, be the release of a new feature or a new product. Another possibility is to use information from old conversations to remind the user of the chatbot. If the timing or message is not well-picked notifications can have a contrary effect on the user and decrease the positive opinion on the gained value through the chatbot. While a user is communicating with the chatbot continual reengagement is still important. To motivate the user to continue the conversation buttons at the end of a message as

well as questions can help to keep the dialogue going.

Table 2.2 summarizes the research results of each of the discussed aspects and provides a checklist to integrate user engagement into a chatbot. Additionally, it aligns the previously defined engagement categories with the user engagement process steps described within the introduction of this chapter. Engagement strategies described within the category Onboarding are applied during the initial step of the engagement process, referred to as the point of engagement. The second step, the period of engagement, covers the engagement strategies outlined within the categories Usability, Content & Design as well as Error Handling. Most strategies listed within the sections of Usability and Content & Design also have an impact on the point of engagement. Nevertheless, as they should be present during the whole period of user interaction (from first user interaction until the end of the conversation) they are categorized within the period of engagement phase. The aspects of disengagement and nonengagement are not covered in this table as they occur if the chatbot fails to get or keep the engagement of the user. Nevertheless, the user can be reengagement step by applying engagement strategies listed.

2.3. Summary

Chatbots, also known under the term virtual assistants, digital assistants or language interfaces have been very popular in recent years (Dale, 2016). To communicate with the chatbot a user has to enter natural language into the provided text- or speech-based interface (Hill et al., 2015). Siri, Cortana and other general virtual assistants can answer a broad variety of questions concerning various topics (Sarikaya, 2017; Shum et al., 2018). Nevertheless, most chatbots are designed to operate within a predefined (limited) scope (Dale, 2016; Shum et al., 2018). Chatbots are a useful tool to overcome problems of traditional websites, like websites that are hard to navigate and desired information that is difficult to find. A chatbot can help to overcome these issues, as the user can just ask for the required information. An additional benefit can be found in the real-time and on-demand service (Drift et al., 2018). During nearly 60 years of chatbot history

Engagement Process Step	**Engagement Category**	**Engagement Strategies**
Point of Engagement	Onboarding	• first impression counts • outline the benefits of using the chatbot • provide additional features that are only available by using the chatbot • use ”call-for-action”-elements • familiarize the user with the chatbot/technology • strive for a playful entry
Period of Engagement	Usability	• ensure that core tasks are working • limit the scope • design logical conversation flow • evaluate edge cases & recognize additional use cases • provide good feedback (e.g. for errors) • minimize the effort to use the chatbot
	Content & Design	• focus on understanding users’ needs • focus on relevant/interesting information • use personalized elements • try directing the conversation • use Gamification elements • use adequate language • use visual aids (e.g. images) if appropriate • keep messages short (do not overwhelm user) • prepare content suitable for the given platform • enhance chatbot with small talk capabilities • design a simple, intuitive and nice-looking interfaces • design for simplicity in interactions
	Error Handling	• consider ”escalation scenario” • minimize unpredictable errors (e.g. external services) • respond with suitable error messages
Reengagement	Reengagement	• use notifications • add buttons or questions to the end of the message (”call-for-action”-elements)

Table 2.2.: User Engagement Strategies for Chatbots within the Engagement Process

various designs of chatbot architectures have been developed, raising from simple pattern matching dialogue systems (Weizenbaum, 1966; Epstein, 1992) or learning by chatting (JABBERWACKY) to sophisticated chatbot architectures using machine learning and neural networks (Fadhil, 2018). Many chatbot architectures in recent years describe components like Natural Language Understanding, Dialog Manager, Message Generator and an optional Channel Connector (Kompella, 2018). To develop a chatbot there are numerous different tools and frameworks available. Some of them do not require coding knowledge, but these tools are often not suitable for complex chatbot tasks. Wit.ai, Dialogflow, and Microsoft Bot Framework are examples of frameworks that can be used to build chatbots by using a programmatic approach. Even though the concept of chatbots is not new scientists have struggled with the complexity to understand and generate human language for decades. Natural Language Processing, especially Natural Language Understanding is still a main challenge within the chatbot development process (Hill et al., 2015). Each user communicates differently, concerning the communication style (like short/long sentences) but also the used language (like abbreviations or slang) (Alpana, 2017). Also, the acceptance of users can be a problem as in most aspects people still prefer to communicate with real humans instead of using chatbots or apps. The chatbot, therefore, has to fulfil high requirements to be accepted as alternative (Pega, 2017). To enhance user acceptance it is important to engage the user to make use of this technology. Since traditional design patterns for graphical user interfaces do not apply to chatbots the developer has to switch from an explanatory to an interpretational design (Fadhil, 2018; Følstad & Brandtzæg, 2017). O'Brien and Toms (2008) define five aspects of the engagement process: the point of engagement, the period of engagement, disengagement, reengagement, and nonengagement. Successful onboarding of the user, especially of new users, is crucial to motivate the user to interact with the system. Therefore it is important to apply methods and mechanisms so that also new users can smoothly interact with the system. This does not only include actual onboarding strategies to gain the user's attention, but also provide good usability and suitable content design to prevent losing the user's attention within the first few minutes of interaction. If the chatbot was able to get the users interest this is called the point of engagement. After this first engagement, it is important to keep the user motivated to use the program (period of engagement). Therefore good usability and well thought out content and design are required in this phase. Engagement strategies for this aspect can be a well-designed user interface and logical conversation flow, but also focusing on the core tasks and provide the desired information suitable for the target audience. If a problem arises that the chatbot is not able to solve it is important to provide good feedback for the user to understand the problem or handoff to a human to prevent user frustration. A user can be disengaged through external factors, but also due to dropping interest or usability issues. If this happens it is important to reengage the user, for example with additional questions from the chatbot or through notifications. The worst case that can happen is that the user is not motivated at all to use the system. This is called nonengagement.

3. Requirements and Design

The customer is responsible for the management of a variety of different theatres as well as the ticket sale of the offered performances in these theatres. A way to simplify information retrieval for the customer and provide an easy way to support the customer 24/7 was needed. Chatbots have been identified as a good tool not only to help the users to find the required data but also as a marketing tool to demonstrate innovation. The design and development of a user-friendly chatbot have to be in the focus of attention.

3.1. Requirement & Stakeholder Analysis

Developing an application for a customer always leads to different requirements of the involved stakeholders and all should be covered as best as possible. The stakeholders regarding the development of the chatbot include the customer, the developer as well as the end-users of the application. Within the following chapter, the requirements of all parties involved are described. To understand exactly who the future end-users are a target group analysis is provided. Additionally, focus-group interviews are conducted to collect information about the user requirements and expectations of different age groups.

3.1.1. Customer Requirements

The initial requirements of the customer where roughly outlined before the actual development of the chatbot started. The main goal of the customer was a tool, that enables visitors of their website to retrieve information easily by asking questions and getting the appropriate responses. The chatbot implementation is used to create an innovative program, not only to simplify the search process for users but also to use the program as a marketing tool. The customer requires an application that can search within all the provided data and delivers it to the end-user in a dialogue fashioned way. The first milestone includes an only text-based chatbot, but for the second milestone, the integration of the chatbot with Alexa is required. To minimize the development effort (and therefore the costs) the chatbot logic should be used by the text-based, as well as the speech-base chatbot interface. Since the integration of chatbots within major messaging platforms is already very common, the customer also requires the integration of the chatbot into the Facebook Messenger. The colour scheme used with the chatbot should be red and grey, as these are the colours of the company website. The scope that has to be covered by the chatbot includes information concerning the repertoire of the different theatres (like date of the performance, duration, place of performance) as well as knowledge of the respective cast of a performance. The user should also be provided with the possibility to search for specific performances by date, location or category. Additionally, also a question and answer catalogue has to be provided to cover general information concerning the different theatres as well as frequently asked questions These include questions like cheap parking spots or information concerning ticket cancellation or special offers. The customer needs to be able to enhance this catalogue continuously to be able to deal with more and more user requests. Even though an advanced escalation scenario (forwarding of the communication between the chatbot and the user to an employee of the customer support) was planned in the beginning, this feature was cancelled during the development process and replaced by providing contact information for the user to reach out to the customer service by their own.

3.1.2. Developer Requirements

When developing an application it is always easier to operate within a field of existing knowledge. Therefore the requirements of the developer included fundamental knowledge of the required programming language as well as concerning the used bot framework. The selected framework should be easy to use and it should be able to smoothly train the underlying language model. A usable interface and a well structured and complete documentation can improve the usability of a development framework tremendously. Therefore these aspects, as well as a supporting community, are important aspects to consider. Additionally, the performance of the selected tool and the provided feature is decisive for the selection of an appropriated tool. The developer requires as much of the desired features to be already implemented to speed up the development process and focus on the actual development of the chatbot. Another requirement of the developer is an appropriate performance of the language model predictions.

3.1.3. User Requirements

Also, the third stakeholder involved in the chatbot development process, the user, has own requirements. The main purpose of the planned chatbot is to simplify the information gathering process for visitors to the opera house and different theatres. Social factors like gender, age, education, religion or race can influence the perception of a user (Reuband, 2018). Therefore it is important to identify the target audience to provide a suitable and engaging tool at the end of the development process.

Target Group Analysis

Reuband (2018) conducted a survey to determine the social composition of visitors to cultural institutions in Germany. As shown in Table 3.1 the share of female visitors is higher than the share of male visitors for performances in the opera house as well as in the theatre. Considering the age of the audience it is noticeable that the older the age group the more percent of the total visitor share is assignable to it. Only 7% of the audience of an opera performance is below thirty, whereas the amount of visitors between 45 and 59 is already 24% and even continues to increase to 52% for visitors older than 59. Similar behaviour is visible for the audience of the theatre. Even though there is a decrease of visitors within the age between 30 and 44, in general, the share of the audience increases the older the age group.

The average age of a visitor of the opera house is 56 and for a visitor of the theatre, the average age is 49. Also, a relation between the level of education and the visits to opera and/or theatre performances can be derived. For example, persons with the highest level of education of lower secondary school only contribute to 11% of the audience of the opera house and only 5% of the visitors of a theatre performance. Graduates of a university, on the other hand, represent 39% of opera visitors and even 45% of the audience of a theatre performance.

	Opera House	**Theater**
Gender		
• male	42	39
• female	58	61
Age		
• < 30	7	20
• 30-44	16	14
• 45-59	24	27
• > 59	52	40
• Average Age	56	49
Education		
• Lower Secondary School	11	5
• Secondary School	21	18
• Advanced Technical College Certificate	14	11
• School Leaving Examination	16	22
• University	39	45

Table 3.1.: Social composition of visitors of cultural institutions adapted from Reuband (2018) in percent

As a result of these observations, the average visitor of the institutions is female, between 49 and 56 years old and has a high level of education. According to this analysis, it is most suitable to address the target audience in a formal way. Polite and fact-based answers are therefore preferred over funny answers and jokes. As many members of the target audience are not grown up with digital devices the user has to be able to use the chatbot without any prior knowledge. An easy and intuitive interface has to be designed. When using the chatbot for the first time the user has to be informed about the information the tool can provide to get started. During the dialogue, it can be very frustrating if the information already provided within the conversation has to be repeated over and over again. Therefore the chatbot should be able to keep a certain context and answer additional questions to this topic. The provided information has to be presented nicely. Images, videos or other visual aids can support this and have to be included within the textual interface. The speech output has to move from a monotonous voice to a more appealing listening experience, for example by including emphasizes and breaks whenever suitable. Neither the text-based nor the speech-based interface is allowed to overwhelm the user by eternally long sentences. If the chatbot gets stuck, whether because of irrelevant context information or other reasons, the user has to have the possibility to restart the conversation at any point to prevent dissatisfaction with the tool.

Focus-Group Interviews

According to Berkup (2014) people born in different periods have different personalities, viewpoints and values, which influences the expectations and perceptions of technology and its rapid and successive changes within each generation. Therefore, additionally to the provided survey of Reuband (2018) focus-group interviews are conducted in the course of this thesis to identify the differences and possible generation-related issues about the technology of chatbots and it's acceptance within different age groups. Based on the outcome of the interviews strategies are developed that motivate the users to get in touch with this technology. To select the **participants** for the interview first, the boundaries of the different age groups have to be identified. In literature, the generational classes are not always described identically. The classification of generations used within this paper is based on the categorization used within the paper of Berkup (2014) and is listed in Table 3.2. The generational classes consulted for the focus-group interviews

Generation Name	Chronological Generation Classification	Current Age
Traditionalists	1900 - 1945	119 - 74
Baby Boomers	1946 - 1964	73 - 55
Generation X	1965 - 1979	54 - 40
Generation Y	1980 - 1994	39 - 25
Generation Z	1995 - now	< 25

Table 3.2.: Adapted categorization of generational classes by Berkup (2014)

are marked bold. Besides this definition also the classification into "digital natives" and "digital immigrants" is taken into account. Prensky (2001) describes digital natives as people that are born into the digital world whereas digital immigrants had to learn to adapt to the digital environment. For digital natives, the usage of technologies like computers, video games or the internet is normal as they interact and practice the usage form a young age on. Digital immigrants, on the other hand, adopted the new technologies at a later point in their life. So even though they learned to handle the technologies they do not interact the same way digital natives do. Plafrey and Gasser (2008) identified 1980 as the first year of digital natives. Based on this information the generational classes "Generation X" and "Baby Boomers" mentioned earlier are summarized to a single group called "Digital Immigrants" within this analysis. The interview, therefore, takes place with following groups:

1. Generation Z: participants with age < 25 years
2. Generation Y: participants with age between 25 and 39 years
3. Digital Immigrants: participants with age > 39 years

The interview is conducted with three groups and a total amount of 13 participants between 15 and 60 years. In total seven female and six male persons are consulted. The characteristics of each interviewed group are illustrated in Table 3.3.

After the selection of the participants the actual interview **process** can start.

	Generation Z	**Generation Y**	**Digital Immigrants**
Boundaries	$<$ 25 years	25 - 39 years	$>$ 29 years
Gender	2 male, 3 female	2 male, 2 female	2 male, 2 female
Age	15 - 18	26 - 31	56 - 60
Age Mean	16.20 (SD=1.30)	28.25 (SD=2.63)	58.25 (SD=1.71)

Table 3.3.: Characteristics of participants of the evaluation groups

Each interview consists of a paper questionnaire to be filled out by each interviewee individually, as well as an open discussion within the whole group. The questionnaire can be found in Appendix A. After each participants of a group has finished the paper questionnaires the group is encouraged to discuss the questions as well as give their opinion on additional upcoming statements. Table 3.4 summarizes the **results** of all focus-group interviews by categorizing the results into four main categories:

1. What are the general preference concerning events?
2. Why people do use chatbots?
3. Why people do not use chatbots?
4. Which preferences do people have for chatbots in the field of operas and theatre?
5. Which channels do the user prefer?

Generation Z	**Generation Y**	**Digital Immigrants**
$<$ 25 years	**25 - 39 years**	**$>$ 39 years**
1. General Information		
Preferred event type		
Musicals, Concerts and Festivals	Special Events, Concerts	Musical, Theater, Concerts
Preferred place to purchase tickets		
Online	Online and Offline	Offline
Filter Criteria for Events		
Name, Artist, Location, Genre, Date	Location, Artist, Price, Arrival	Name, Date, Location

2. Reasons to use chatbots		
test out the technology	faster responses	fast and exact information
website hard to navigate	simpler for complex search requests	curiosity
information hard to find	easier to get desired information	information hard to find
additional features / information only available with chatbot	fast help for easy problems	
3. Reasons not to use chatbots		
rarely offered on used websites	rarely offered on used websites	rarely offered on used websites
websites often easier to use	websites easier to use in most cases	cumbersome typing (text-based chatbot)
not used to this technology	not suitable for complex/specific problems	fear of invalid information
bad results / hard to get what is wanted	annoying if bad results ("do not understand")	little experience with this technology
low trust in results	low trust into service	
4. Preferences for chatbots in this field		
Greeting		
greeting words	greeting words	greeting words
not too many information	short introduction	only necessary information
information that user communicates with chatbot	description of scope	providing help if needed
friendly		do not use the term "Chatbot" as users might not know what this is

Language		
professional language	professional language	professional language
no colloquial language	no small talk	formal language
short answers	responses as short as possible as long as needed	short answers (not too much information)
only important information	possibility to define how detailed response should be	
Gamification and Easter Eggs		
No	No	No
Other preferences		
critical processes (ticket ordering) with link to website, not within chatbot	definition of scope for better results	as little typing as possible (proposed answers)
fast communication (no long response times)	if no information provide link where information can be found	introduction to see possible way to use the chatbot
easy to close when not needed		design appearance to evoke curiosity
4. Channel preferences		
Facebook (Messenger) is not used	Little trust in information provided within Facebook Messenger	Facebook Messenger and Alexa rarely used

Table 3.4.: Summary of results of the conducted focus-group interviews

During the interviews, it became obvious, that most of the interviewees have rarely get in touch with chatbots. Many of the interview participants stated that they would use the chatbot if they know it provides fast and easy access to the desired information or it offers additional functionalities compared to the website. Nevertheless, as they are not accustomed to the technology few of them would try out the chatbot in the first place. Therefore they are not aware of its advantages. For this reason, the point of engagement or also onboarding of the user has to be a main goal within the chatbot design. Many findings proved the information gathered within the literature review right. People within all three age groups indicate little experience with this kind of information gathering as the main problem. Only a minority of the persons consulted has already used a chatbot. Especially Generation Z declared that most websites they use are well structured and information is easy to find. According to them, in this case, chatbots are only used if they provide additional benefits. This also matches the statements of a majority of the other two age groups. The harder it is to find information on the website the more likely it is that a user starts communicating with the chatbot. Another fear of many interviewees is bad responses. This includes chatbots repeatedly answering with messages like *"I do not understand you!"* or provide wrong information within their responses due to bad performances of the language understanding engine. Some Digital Immigrants also state that they prefer classic website navigation over chatting with the bot, as they do not regularly use the computer and typing is, therefore, a time-consuming task for them. Furthermore, none of the interviewed Digital Immigrants has a voice-assistant like Amazon Alexa at home, to overcome the typing issue with a speech-based interface. While people of Generation Z prefer buying tickets online and Generation Y uses offered offline as well as online options, whereas many Digital Immigrants still purchase their tickets through advanced booking or the box office. The main filters that should be supported for the search of events are the event name, artists, genre, price, date and location. During the open discussion, the importance of a suitable channel selection became clear. The interviewees of Generation Y do not fully trust the information given within the Facebook Messenger and therefore prefer a chatbot on a website, whereas none of the participants of Generation Z even has a Facebook account to use the chatbot within the Messenger. The proposed engagement strategy of hidden "Easter Eggs" within the application, mentioned within Chapter 2.2.1, was dismissed by almost all of the participants. Only one member of Generation Z mentioned, that this feature is not necessary but would be fun to explore. Also, the approach of Gamification (described in Chapter 2.2) was considered as not suitable for the given chatbot by the audience. According to them, there is no meaningful application for this approach within the planned chatbot.

During the interview also different preferences concerning the appearance and behaviour of the chatbot have been extracted. All three groups favour a short introduction message including the greeting and a "call-for-action" question like "*How can I help you?*". While Generation Z wants to be informed that he or she is currently talking with a chatbot, the Digital Natives mention the confusion that could be caused by this term, as many users might not know what a chatbot is. According to them, an explanation is required if the term chatbot is used. The definition of the scope is classified as useful within every group. Nevertheless, Generation Y prefers to be informed about the scope within the initial message, whereas Generation Z, as well as the Digital Immigrants, prefer to see this information when they look for further help. Functionalities to help users to understand how the chatbot works are specially requested by users of the Digital Immigrants. The idea of a short "guided question" tour to involve the user with the technology was approved by them. The preferences concerning the language of the chatbot are nearly the same overall three groups: they prefer a professional and no colloquial language. Jokes are classified to lower the trust in the responses, as the bot is not taken seriously. Only Digital Natives mentioned that they want to be addressed in a formal way as the chatbot is, among other channels, used within the company's website. Small talk capabilities are not important for users. As digital natives are used to getting and process information fast, also the communication with the chatbot is expected to be fast and efficient. Only the required information should be provided, and this without long response times. The group of Digital Immigrants mentioned, that they would try out the chatbot more likely when it evokes curiosity. According to Generation Z critical processes (like actually buying the tickets) should not be handled within the chatbot, but rather a link should be provided that leads back to the website. Generation Y additionally states, that even if no information can be found to a given topic within the scope, the chatbot should reply with a link or a hint where the information might be found. Based on the results of the interviews some requirements concerning the onboarding process of users can be extracted:

- The design of the chatbot appearance has to evoke the curiosity to motivate the user to start using the system.
- A "Guided Conversation" can be used to help people that are not used to the technology to learn how to use a chatbot.

- Users of the page know what they want. "Gimmicks" that do not help the user to reach his or her goal are classified dispensable.
- Especially the initial message should be as short as possible but include information on how the user can get help if required.
- Proposing possible questions helps to begin a conversation.
- Proposed questions as well as provided buttons additionally help users that do not use computers very often to simplify the information gathering process.
- Be aware of unknown words (like the term "chatbot") and avoid using them without further explanation.

Within the next chapter, the gathered requirements are further processed to get functional and non-functional requirements that the developed chatbot has to fulfil after it's release.

3.2. Functional and Non-Functional Requirements

The stakeholder requirements outlined in the previous section combined with the engagement strategies outlined in Table 2.2 are used to obtain functional and non-functional requirements of the chatbot. Gross and Yu (2001) mention that non-functional requirements are often described as system quality attributes that are a significant factor in the success of software. Functional requirements, on the other hand, specify the desired behaviour of the system. The functional and non-functional requirements have been identified and categorized according to the structure of the user engagement process described in Chapter 2. Additionally also the requirements regarding the underlying system are analyzed and listed below. As some requirements do not only belong in the category "Point of Engagement", but are required during the whole "Period of Engagement", they are mentioned in the second category, even though they are also part of the onboarding requirements.

Functional Requirements

1. Point of Engagement
   1.1. The user should be able to get additional instructions if needed

   1.2. The user should be provided with additional functionalities (like complex filter possibilities) that are not part of the website
2. Period of Engagement
   2.1. Usability & Interaction
      2.1.1. The user should be able to ask follow-up questions to previous questions (context-awareness)
      2.1.2. The user should be able to use the chatbot within the companies website, Facebook Messenger and Amazon Alexa
      2.1.3. The user should be provided with good feedback (for example for errors)
   2.2. Content & Design
      2.2.1. The user should be provided with images and videos while using the text-based interface
      2.2.2. The user should be able to get responses for simple small talk questions
   2.3. Core Functionality
      2.3.1. The user should be able to search for upcoming events
      2.3.2. The user should be able to search for details of events (date, time, cast, price, location, and category)
      2.3.3. The user should be able to search for information concerning the cast (role, cast of performance, next play)
      2.3.4. The user should be able to get answers for frequently asked questions
      2.3.5. The customer should be able to easily adapt and enhance the question-and-answer catalogue for frequently asked questions
   2.4. Error Handling
      2.4.1. The user should be able to restart the conversation at any point during the conversation
      2.4.2. The user should be provided with suitable error messages to understand why a request failed
      2.4.3. ~~The user should be forwarded to chat with a human customer service employee if the user struggles for a while or gets frustrated~~[1]

[1] Adapted during development

The user should be provided with contact information (phone number and mail address) of the customer service if the user struggles for a while or gets frustrated

3. Reengagement
   3.1. The user should be motivated to continue the conversation by additional questions at the end of the chatbot's response.
4. Framework
   4.1. Language Support
      4.1.1. The developer should be able to make use of an existing NLU engine and not start from scratch
      4.1.2. The developer should be provided with sufficient support for the German language by the underlying NLU engine
      4.1.3. The developer should be provided with machine learning techniques to improve the underlying NLU engine
   4.2. Integration
      4.2.1. The developer should be able to easily integrate the chatbot into the companies website
      4.2.2. The developer should be able to easily make the chatbot available within the Facebook Messenger of the companies Facebook page
      4.2.3. The developer should be able to easily make the chatbot available as Alexa Skill
      4.2.4. The developer should be able to use a single code base for the text- and speech-based interface
   4.3. Usability
      4.3.1. The customer should be able to define own question-and-answer pairs

Non-Functional Requirements

1. Point of Engagement
   1.1. The user should not be confronted with unknown terms without further explanation
   1.2. The user should get engaged with the system by proposed buttons and questions

1.3. The user should not be distracted with various "Gimmicks" that do not help to achieve the desired goal
1.4. The user should be able to use the chatbot even if he or she never has used a chatbot before
1.5. The user should be able to see the benefits of using the chatbot over collecting information by navigating through the website
1.6. The user should get curious about what the system can offer
1.7. The customer should get a competitive advantage by using the chatbot to demonstrate a high level of innovation

2. Period of Engagement
    2.1. Usability & Interaction
        2.1.1. The user should be provided with responses within 3 seconds
        2.1.2. The user should be provided with a logical conversation flow
        2.1.3. The user should be provided with sufficient results concerning the core tasks of the chatbot
        2.1.4. The user should be able to use the system 24/7
        2.1.5. The user should be able to get desired information easily
    2.2. Content & Design
        2.2.1. The user should be provided with an interface that is based on common user interface and user experience patterns
        2.2.2. The user should be provided with short sentences
        2.2.3. The user should be provided just with the information needed
        2.2.4. The user should be provided with a simple and nice-looking interface
        2.2.5. The user should be provided with emphasized voice responses while using the voice-based interface
        2.2.6. The user should be able to communicate with the chatbot in a dialogue fashioned way
        2.2.7. The user should be provided with adequate responses concerning language and quality of information
        2.2.8. The user should not be overwhelmed by the responses
        2.2.9. The user should be provided with responses that are suitable for the used platform

3. Reengagement
    3.1. The user should be motivated to use the chatbot again

4. Framework
    4.1. Usability
        4.1.1. The developer should be able to build on already existing programming experience
        4.1.2. The developer should be able to get to know and use basic functionalities of the tool within a day
        4.1.3. The learning process of the developer should be supported by good documentation and examples
        4.1.4. The developer should be able to get help from the community if problems arise
        4.1.5. The developer should not have to worry about slow response times

3.3. Concept and Architecture

After the consideration and analysis of all requirements of the different stakeholders and identification of the functional and non-functional requirements of the system, the concept and the architecture of the chatbot had to be planned. The functional and non-functional requirement analysis provided the foundations for this conceptual step. The second part of this chapter deals with the design of the chatbot personality. Before the actual design of the chatbot can be planned the underlying framework has to be selected to be able to identify a chatbot architecture that is easy to implement within the framework's capabilities. After the framework selection, the desired conversational flow has to be planned. In the next step, the intents and entities for the language understanding process have to be defined.

3.3.1. Framework Selection

One fundamental decision concerning the creation of the chatbot was the selection of the underlying (language understanding) framework to be used for development. Even though the most commonly used frameworks provide similar mechanisms to train their language understanding models they differ in many other aspects. Table 3.5 is an enhanced version of the comparison table of Wit.ai, Dialogflow and Microsoft Bot Framework including LUIS presented in Chapter 2.

1. **Language Support**
   The natural language understanding processes of all presented frameworks are based on machine learning and use the concept of intents and entities for this purpose. In contrast to the other two tools, Dialogflow is the only one that additionally provides the opportunity to define actions based on the detected intent directly from the user interface. It, therefore, provides the possibility to define simple behaviour without any coding. Nevertheless, as most of the data needed for the response of the chatbot has to be gathered from a database, this feature does not provide a lot of additional value to the development process. To speed up the development process all three frameworks provide prebuild entities that allow easy extraction of common entities (like date and time). In addition to this Dialogflow and Microsoft LUIS also support entire prebuild domains. By using prebuild domains, build-in intents and entities are added to the language model. The added intents are already trained with several sample utterances. This can speed up the development process of the language model a lot. Wit.ai does not provide a possibility for correcting spelling mistakes and the spell check of Dialogflow is rated as not very sophisticated from some users. Bing Spell Check, which is available as additional service on the Azure Bot Framework Platform performs well. As it is available separate service it also can be integrated into other frameworks, even though the easiest way to use it is with Microsoft LUIS. Wit.ai supports over 50 different languages and therefore by far the most languages of all compared tools. Dialogflow handles twenty languages and Microsoft LUIS thirteen languages at the moment. Even though it is good to have a huge variety of supported languages this is not necessary for the requested chatbot as it only has to support the German language. Therefore all of the provided frameworks are suitable regarding this aspect. All in all **Microsoft Bot Framework with LUIS** has the best coverage of the requirements concerning the language support.

2. **Integration**
   The variety of channels that are supported by the different frameworks varies a lot. Wit.ai of Facebook has the most limited range of supported channels as it only provides integration into the Facebook Messenger by default. Both, Dialogflow and Microsoft Bot Framework, provide a huge variety of supported channels, ranging from SMS (Twilio) and Email support to common messaging platforms like Facebook Messenger, Slack or Telegram as well as speech-based channels like Cortana. All three frameworks provide an API that allows developers to integrate their service into any custom application or website. Even though the supported channels of Dialogflow and Microsoft Bot Framework are very similar, Dialogflow does support all of the required channels (Facebook Messenger, Website and Alexa) for the planned chatbot, whereas the Microsoft Bot Framework does only support Facebook Messenger and Website integration by default. For this chatbot **Dialogflow** is the best option concerning channel integration.

3. **Usability**
   As during the development of an application new features can only be noticed and learned if they are well documented, considering this aspect during the stage of framework selection can ensure that the selected framework does not only provide all the required features but also enables the developer to make use of them. The most advanced documentation can be found for Microsoft Bot Framework. It does not only explain the concepts and possibilities in detail but also provides additional examples for better understanding. Also, the Dialogflow documentation is good to understand and very helpful but does not provide as detailed examples as the documentation of Microsoft. Wit.ai does only provide basic information about how to use the system which makes it harder to get started. All frameworks provide software development kits (SDK) for different platforms. The SDK for Wit.ai is available in the programming languages Node.js, Ruby and Python. C#, JavaScript, Java, and Python are supported by the Microsoft SDK. Dialogflow SDK is supported for Node.js, Python, Java, Go, Ruby, C#, and PHP and therefore provides the widest range of supported programming languages of the compared tools. Since C# is the programming language the developer has the most experience,

Dialogflow or Microsoft C# SDK is preferred. Additionally to the core features and structure of the language understanding frameworks also the average response time, as well as the intent detection precision, are crucial aspects to consider when selecting an appropriate framework for the development of a chatbot. The average response time and intent detection precision have been analyzed within a survey of Intento (Intento, 2017). When investigating the average response time of the frameworks it is obvious that Wit.ai has the slowest average response time. On average it takes the tool 0.96 seconds to respond to the user. Dialogflow and Microsoft LUIS can easily outperform this with an average response time of 0.28 seconds (Dialogflow) and 0.21 seconds (LUIS). All tools have good performance concerning the intent detection precision but also in this regard Dialogflow and Microsoft LUIS perform better than Wit.ai. The customer wants to be able to maintain basic question-answer pairs on its own. Therefore a user interface has to be provided to simplify this process. Microsoft Bot Framework provides this with the additional QnA service. Microsoft Bot Framework is the only tool that provides additional Services that can be easily integrated within the chatbot, like Text-to-Speech or Speech-to-Text processing, automatic translation, the definition of question and answer catalogues or search. Nevertheless, all of these external services are standalone services that can also be integrated within chatbots of other platforms. Additionally, it has to be mentioned, that these extra services also might cause additional costs. The easiest way to use services like Bing Spell Check or the QnA (question-and-answer) service as it is required for the developed chatbot is with Microsoft LUIS. Concerning usability Dialogflow and Microsoft Bot Framework both have some main advantages and disadvantages. As **Microsoft Bot Framework,** is more flexible concerning additional features (as many services can easily be added) and the question and answer catalogue can be maintained by the customer itself this framework seems to be a good choice.

4. **Pricing Model**
   Only Wit.ai offers its services for free. Dialogflow provides two different pricing models, the standard version is available for free whereas by using the enterprise version each request is charged with a small

amount of money ($ 0.002 or $0.004). Microsoft offers its Bot Framework for free but uses a pay as you go model for the language understanding service (LUIS). It provides 10,000 free transactions per month. When exceeding this quota every 1,000 transactions are charged with € 1.265 for text-requests. **Wit.ai** and the standard version of **Dialogflow** are the cheapest options. The pricing of Microsoft strongly depends on the needed amount of transactions.

Result

Wit.ai is not able to keep up with the functionality and usability provided by Dialogflow and Microsoft Bot Framework. Even though the language model is suitable Wit.ai is not to reach the level of usability and channel support of the two other frameworks. Even though it is available for free it is no suitable option for the development of the requested chatbot. Dialogflow and Microsoft Bot Framework are very equal but have both some advantages and disadvantages. While Dialogflow is a bit slower than the Bot Framework of Microsoft it is the only framework that supports Amazon Alexa by default. Additionally, there is a free pricing model for standard usage. Microsoft Bot Framework, on the other hand, provides further services like the spell check engine and the question-and-answer catalogue. Microsoft Bot Framework has a pay as you go pricing model. Nevertheless, as 10,000 transactions (request and response pairs) are free this should not take much into account. As one important requirement of the customer was to be able to update basic question and answer pairs on its own this was the decision-making point towards Microsoft. Here no additional user interface has to be developed and the integration of the QnA service into the chatbot can be easily achieved. So, **Microsoft Bot Framework** was selected as framework.

3.3.2. Language Model Design

Microsoft (Microsoft, 2019e) suggests planning the language model before the implementation by considering the following aspects: domain, intents, utterances, and entities. According to this suggestion, the natural language model was planned.

	Wit.ai	Dialogflow	Microsoft Bot Framework + LUIS
Language Support			
NLU Engine	Yes	Yes	Yes
Concept	Intent & Entities	Intents, Entities & Actions	Intents & Entities
German Language Support	Yes	Yes	Yes
Machine Learning	Yes	Yes	Yes
Prebuild Entities	Yes	Yes	Yes
Prebuild Domains	No	Yes	Yes
Spell Check	No	Yes (not very sophisticated)	Yes (additional service)
Integration			
Website	Yes	Yes	Yes
Facebook Messenger	Yes	Yes	Yes
Amazon Alexa	No	Yes	No (only with REST API)
Usability			
Documentation	Basic	Good	Good + many examples
Community	Yes	Yes	Yes
Experience	Partial	Yes	Yes
Average Response Time	0.96 sec.	0.28 sec.	0.21 sec.
Intent Detection Precision	Good	Very good	Very good
Question-Answer Catalogue	No	No	Yes (additional Service)
Expandability	-	-	Azure Cognitive Services (Spell Check, Text-to-Speech, Speech-to-Text, Translation, QnA Service, Bing Search)
Pricing Model	Free	Standard for Free or Enterprise as pay as you go	10,000 transactions free then pay as you go

Table 3.5.: Result of framework comparison with highlights of suitable frameworks within each category

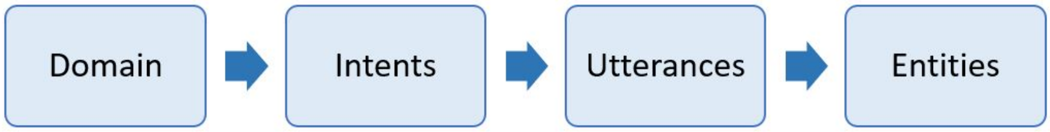

Figure 3.1.: Suggested consideration workflow for the development of a natural language model with Microsoft LUIS presented by Microsoft (Microsoft, 2019e)

Domain

As already mentioned in Chapter 2 the scope of the application, also called domain, that is covered has to be defined. The wider the scope of the chatbot the more complex the request handling. When developing a domain-specific chatbot it, therefore, is better to keep the scope as small as possible. The main purpose of the chatbot that is developed in the course of this thesis is providing information about the organization and it's daughter organizations as well as support the user by providing details concerning plays and casts. In addition to this, the chatbot has to be able to respond to predefined frequently asked questions. Everything else exceeds the scope of the chatbot and hence can be answered with a fallback answer.

Intents

It is suggested by Microsoft *to only define as many intents as one needs to perform the functions of the app*. This is due to the fact that LUIS might not classify the utterances properly if too many intents exist. Too few specified intents, on the other hand, might be too general and overlapping. Therefore the identification and selection of appropriate intents can be hard. Especially since the intent and entity selection of the natural language model is one of the most crucial parts during the design process of the chatbot as the defined intents and entities have a huge impact on the performance of the NLU capabilities of the chatbot. After considering the requirements of the chatbot seven different intents were defined within the planning stage of the chatbot. This structure is outlined in Table 3.6. Additionally to these intents there exists also a **None** intent. This intent is also very important since the developer can define inappropriate or not meaningful utterance. Hence this intent can be used to identify out-of-scope requests. While developing the chatbot it became more and more clear that this defined structure of the beginning was not sufficient for the chatbot requirements.

Intent Name	Description
Greeting	Handles greetings of the user.
Closing	Handles end of a conversation.
HandoffToHuman	Detects request of a user to talk to a human person.
Help	Identification of a request for further guidance.
QnA	This intent is trained to identify frequently asked questions like the parking possibilities, discounts or information concerning the different theatres.
CastDetails	It is used to determine the user intention to get information concerning the cast of a play like a role an actor plays within a performance, all members of the cast for a specific performance or the next date an actor performs within a performance.
EventDetails	The EventDetails intent detects utterances that er related to plays and events. This includes questions concerning the performance date, start and end times as well as content and prices of different plays.

Table 3.6.: Initial Intent Structure

The intent detection was not accurate enough and the correct identification of the appropriate answer, especially concerning the intents *CastDetails* and *EventDetails* was often not given. Therefore the whole intent structure was refined. The problematic intents were partitioned in more granular intent to improve the detection of the correct conversation flow. The other intents mainly remained the same. During the first tests of the chatbot another useful intent was identified, the *ChitChat* intent. It is used to enable the chatbot to answer simple chitchat questions. This intent was included to improve the usability of the chatbot since people like to ask some general questions to get used to the chatbot. Table 3.7 shows transformation of the initial to the final intent structure.

Utterances

In the context of a chatbot, the term utterance refers to user input that has to be interpreted. LUIS uses example utterances for its machine-learned intelligence. Therefore it is important to train each of the specified intents of the previous steps by inserting a broad variety of different example inputs

Initial Intent	**Refined Intent**	**Description**
Greeting	Greeting	Handles greetings of user.
Closing	Closing	Handles end of conversation.
HandoffToHuman	HandoffToHuman	Detects request of a user to talk to a human person.
Help	Help	Identification of a request for further guidance.
QnA	QnA	This intent is trained to identify frequently asked questions like the parking possibilities, discounts or information concerning the different theaters.
None	None	Detect out-of-scope requests
CastDetails	CastGetCast	Detects Request for information concerning the cast of a performance or the name of the actor playing a certain role
	CastGetPlay	Identifies questions concerning performances a specific cast member acts in
	CastGetRole	This intent is trained to get request concerning the role an actor plays
	CastGetTime	Questions categorized within this intent refer to the time a specific cast member acts
EventDetails	EventDetails	Content of performance
	EventLocation	Location a performance or event takes place
	EventPrice	Costs of a ticket for a event or performance
	EventSuggestion	Information regarding upcoming events, performances on a specific date or performances of a specific category
	EventTime	Details concerning start, end and duration of a performance
-	ChitChat	Detects chitchat request of the user.

Table 3.7.: Final Intent Structure

for each of them. Microsoft (Microsoft, 2019d) suggest to include different utterances with the same meaning but different structures (like length, word order, grammar, punctuation, pluralization or stemming). Additionally, the following suggested aspects have been kept in mind during the design process of the chatbot:

- **User Input is not always well formed**
  Different users use different writing styles. While some users might stick to full sentences, others just might feed the chatbot with some keywords. Furthermore, spelling mistakes are common. It has to be decided if an additional tool, like Bing Spell Check[2] should be used or if the language model is also trained with misspelt phrases.
  Considering the suggested approaches the Bing Spell Check seems to be the preferred solution for the designed chatbot. On the one hand, less effort concerning training of the language understanding model is involved. On the other hand, correcting typos and misspellings during prepossessing of the user input helps to limit errors within entities, like the performance name. Since these entities are later used to query the database, correct spelling is necessary to detect suitable data.

- **Language differs between user groups**
  As already mentioned in Chapter 2 it is important to pay attention to the wording and terms used by the target audience, since there can be huge differences between different target groups. Also, the level of domain experience of typical users of the application affects the phrasing of the required training utterances. As the target group analysis displayed a high level of education among the visitors of performances a high language level and a high domain experience is expected.

- **Varying Terminology and Phrases**
  To train the intents successfully, it is important to provide a broad variety of phrases with different word order as well as varying terms. Even though LUIS can identify some synonyms from the context it is better to already use common terms while training the language

[2]https://azure.microsoft.com/en-us/services/cognitive-services/spell-check/

model. There are a variety of different possibilities to query each of the identified intents. Therefore it is necessary to rephrase common sentences and use different terms to help LUIS predict intents and entities more accurately. Since different persons do use different terms and wording the utterances for the required chatbot are not designed by a single user, rather multiple persons are encouraged to participate in the identification process of useful utterances.

- **Punctuation marks are not ignored**
  LUIS is not designed to ignore punctuation marks by default. This is due to the fact that some applications might rely on using specific punctuation marks for the intent or entity detection. If it is not important for the language model of a chatbot, LUIS provides the possibility to define patterns and outline punctuation marks that have to be ignored. The developed chatbot does not have to rely on punctuation marks to distinguish between certain intents or entities. Therefore some common patterns including them have to be added to the pattern list of LUIS.

Entities

While intents are used to predict a specific goal a user wants to achieve with an input, entities are used to extract information from the utterance. A single utterance can contain zero, one or multiple entities and every entity can consist of one or multiple words. Even though entities are not required (in contrast to intents that must be specified) it is highly recommended to make use of this feature. Depending on the use case entities can be used exclusively for a single intent but also be shared among all intents. LUIS distinguishes between machine-learned and non-machine learned entities. It depends on the use case which entity type should be preferred. (Microsoft, 2019c). In the context of this chatbot, only the entity types listed in Table 3.8 are used. Simple entities are based on a machine-learned entity detection method. Therefore they have to be labelled within the provided training utterances to enable LUIS to learn the detection of an unrecognized entity within new user input. To be able to detect the name of the event, cast or role within a given user request this entity type is used. Nevertheless, since names typically have no global underlying schema in common it was

Entity Type	Machine-learned	Purpose
Simple	Yes	Contains a single concept in word or phrase
List	No	List of items and their synonyms extracted with exact text match.
Pattern.any	Mixed	Entity where end of entity is difficult to determine.

Table 3.8.: Adapted list of entity types provided by Microsoft (2019c)

hard for LUIS to recognize all different variations of names. Especially the names of performances can range from a single to multiple words, including names of persons, locations or nearly everything else. Therefore, to support the name detection process for every simple name entity additionally, a pattern entity was provided. With this entity type, it is possible to define placeholders for terms with a variable amount of words within specific sentence patterns. The third entity type used is the list entity. A list entity allows defining lists of synonyms for specific words. This entity type is especially useful for a range of words that never changes. In the context of this chatbot, it is used to identify the category an event belongs to or the location where the event takes place. By using list entities it was possible to define all relevant values and map them to a specific key value that can be extracted from the result of LUIS. For example, the list of categories included the normalized values among other terms the values "Musical", "Kinder- und Jugendprogramm" and "Theater". While these values exactly match the required search terms used for the database query in a later step, additional values like just "Kinderprogramm" or "Jugendprogramm" can be defined as a synonym of "Kinder- und Jugendprogramm". By doing so it is possible to get the required, normalized search term from LUIS even though the user inserted another term from the specified list.

3.3.3. Chatbot Personality

When considering the design of a chatbot personality, Steele (Steele, 2018) mentions in her online article some main questions that can help to develop a suitable personality:

- **Who is your user and in what situations are they talking to the bot?** Since the audience of theatre and opera performances is typically more formal also the chatbot should interact with them in a formal way. A professional, formal chatbot does not only match the average visitor of an opera or theatre but, according to the focus-group interview, is also the preferred way of interaction for the younger target audience. Polite and fact-based answers are therefore preferred over funny answers and jokes. Many interviewees of the focus-group analysis mentioned, that they have less trust in the provided information if the chatbot is joking. A user typically interacts with the chatbot to get information about upcoming performances or events or to get answers to frequently asked questions (like parking possibilities). The chatbot is therefore used to gather information. Even though the chatbot should be a specialized program concerning the field of opera and the different theatres it should still provide some basic chit chat capabilities as many users try to get in touch with the chatbot by such interactions.

- **What is the goal of your chatbot?** The goal of the chatbot, as already mentioned, is to simplify the information retrieval for customers of the opera and theatres in Graz. The chatbot should help to gather information concerning upcoming plays, cast details as well as frequently asked questions. Additionally, it should demonstrate the willingness to innovate of the customer.

- **How human should you make your bot?** As chatbot nowadays are not yet been developed to such an extent as to convince the user to interact with a real human it is suggested to prevent pretending so. It is better to inform the user that he or she interacts with a machine rather than a human being as the conversation flow sometimes might be confusing. This also corresponds with the outcome of the conducted focus-group interview. The participants also mentioned that they want to be informed whether they talk to a human or a machine. Nevertheless, certain human characteristics like varying the wording of the answers (even if it is the answer to the same question) are good ways to prevent boredom.

- **What gender do you envision your bot as?**
  In literature, giving the chatbot a gender is a highly discussed topic. According to Belmont (2016) *"gendering artificial intelligence makes it easier for us to relate to them, but has the unfortunate consequence of enforcing gender stereotypes."*. As users interact with the conversational interface the same way they interact with humans, gender-specific biases and prejudices influence the user when interacting with the chatbot, too. This is especially the case if the chatbot interacts within a gender-stereotypical domain McDonnell and Baxter (2019). Also Zumstein and Hundertmark (2017) mentions that the trust into female and male chatbots does strongly differ. In technical fields, users tend to show more trust in male chatbots, whereas for service or support request often female chatbots are chosen. Female chatbots are perceived as higher in empathy and warmth. Designing the chatbot to be gender-neutral also is problematic too, as this makes it harder for the user to get an emotional connection to the bot (McDonnell & Baxter, 2019). None of the participants of the conducted focus-group interview mentioned any "gender" preferences concerning the chatbot, but the customer expressed the wish for a female chatbot. As the chatbot's area of application is in the field of customer service and support, the customer's wish matches the suggested gender for this field. The chatbot is therefore envisioned as female. The name "Pamina" was chosen as it is a popular and well-known name within the target group and it is immediately associated with the opera.

Nevertheless, even though Steele (2018) states that the chatbot personality is an important aspect of the design process, the developer should never pay more attention to this aspect than the actual core tasks the chatbot should fulfil.

3.3.4. Design Decisions

After the selection of the framework, the definition of the language model design and identification of the chatbot personality, decisions concerning the appearance and the engagement strategies have to be made.

Appearance

The main colours of the website, where the chatbot will be integrated, are red, white and grey. Therefore the design of the chatbot is adapted to these colours. While chatbot messages are displayed with a grey background, the user input should be highlighted with a red background. The usual position of chatbots on websites is the right bottom corner of the page. Nevertheless, as the website already provides a navigation button to go back to the top of the page at this position, the chatbot's position is moved to the left bottom corner of the page. A red button with a dialogue bubble icon is used to show or hide the chatbot on the website. The appearance of the interface has to be nice and simple. Therefore only necessary elements (message bubbles and user input field) are displayed.

Engagement Strategies

After a careful evaluation of the requirements and the available engagement strategies outlined in this thesis, the following elements and mechanisms are selected that the chatbot should provide:

- Onboarding
  - Chatbot pops up after 5 seconds on the page to get the users' attention.
  - Chatbot provides a short introduction message, including greeting, name of the bot, information that it is not a human, a hint how the user can get further help and a "call-for-action" question.
  - The initial help message outlines the scope of possible questions.
  - The help functionality provides some possible questions to help the user to get started.

  - The help functionality outlines additional features only available within the chatbot (like filtering for different criteria when looking for events).

- Usability
  - The chatbot should only pop up at the first visit or if the user actively clicks on the chatbot button to prevent annoyed users.
  - The core tasks of the chatbot have to be extensively tested by test users throughout the development process to guarantee they work as expected for various user requests (wording and phrasing differences).
  - The chatbot does only handle a limited scope of questions to minimize the training effort as well as limit user frustration due to responses for untrained input.
  - Basic chit-chat capabilities are implemented as users often try out basic questions when first getting in touch with the technology.
  - The conversation flow is structured as well as possible, including proposing follow-up questions where suitable.
  - Buttons and proposed follow-up questions are used to minimize the typing effort for the user.
  - The responses have to be provided within 3 seconds.

- Content & Design
  - Short messages are used to respond to the user.
  - Only important information is provided within the response, but links to further information are provided.
  - As the chatbot is mainly used for information gathering it is not applicable to guide the whole conversation. Nevertheless, if a user wants details for a specific event, the chatbot asks if additional information like content, cast or price is needed.
  - Gamification elements and "Easter Eggs" are not used as they are not required according to the users of the focus-group analysis.
  - The language of the chatbot has to be formal and professional. No jokes are used within the responses as this might lower the trust in the chatbot responses and is not suitable for the given context.
  - As the text-based-chatbot will mainly be used on computers, the

screen size is large enough to show images and videos. Also, the different Card Types provided by the framework are suitable.
  - Emphasises are used to keep the messages for the voice-based-chatbot interesting.

- Error Handling
  - The user has to be informed by a suitable error message if something went wrong, to prevent users that get frustrated as they do get the same error again and again when requesting the desired information.
  - If no information can be provided, the contact data of the help desk is provided.

3.4. Summary

The requirements of the chatbot are identified by evaluating the requirements of the customer, the developer and the user. To specify the user requirements a target group analysis from literature is evaluated as well as focus group interviews are conducted. In the course of the focus group interviews, three different age groups are questioned: Generation Z (age < 25), Generation Y (age 25 - 39) and Digital Immigrants (age > 39). Based on the requirement analysis design decisions are made. This includes aspects like the appearance of the chatbot (colour scheme, position on the page) as well as the chatbot personality and the selected engagement strategies. The chatbot's gender is female and the language to interact with the user has to be formal, professional and fact-based. Basic chit-chat should be supported and the chatbot, in the beginning, the user has to be informed that he or she is not talking with a human. To onboard the user the chatbot should pop-up on the website after several seconds. After a short introduction message including the chatbot's name, a basic introduction and two or three buttons providing possible questions the user should be able to start using the system immediately. A help button should be provided within the initial message to help especially new users to handle the technology. A short description of the scope, special commands (like how to restart the conversation) and possible questions should be mentioned in this context.

The core functionalities have to be tested extensively to ensure that they are work properly. Additional "Gimmicks" should be left out as they might distract the user. The design of the interface has to be simple and intuitive. Buttons and suggested questions are used to simplify the communication between chatbot and user. The messages should be short and long messages should be split up into several messages. It is important to ensure that the provided information is of high quality Also the performance of the chatbot should be good. Responses should be available for the user within a few seconds and the system has to be available 24/7. To prevent user frustration the feedback like error messages have to be clear and understandable. The user has to be able to use the chatbot within the company's website, Facebook Messenger and Amazon Alexa. As development framework, the Microsoft Bot Framework with LUIS as language understanding engine is selected. The criteria evaluated for the selection of the framework are language support, integration, usability and pricing model. Microsoft Bot Framework provides all required language features like a natural language understanding engine with machine learning and an easy way to train the language model. The German Language support is given and it can be easily integrated into several different channels. The usability of the framework is good and the chatbot can be easily extended with additional services like spell checking through the Bing Spell Check and a question-and-answer catalogue (QnA Maker Service). After the definition of the domain and the required intents and entities, the model possible user input has to be collected that can later be used to train the language engine.

4. Implementation

After the process of selecting a suitable framework, the definition of the language model and chatbot characteristics has been completed in the previous chapter the actual implementation of the chatbot can start. Therefore all required components of the chatbot have to be configured and connected. Additionally, the conversation flow has to be implemented and user engagement has to be ensured. The information delivered by the chatbot has to be collected and presented in a suitable way for each channel. To develop the chatbot an iterative development process, as illustrated in Figure 4.1 is used. During this iterative process the architecture, language model and user engagement strategies have been implemented, tested, evaluated and redesigned multiple times. Due to the iterative character, it is possible to measure the results for each iteration and improve the evaluated aspects until the best possible result is achieved.

4.1. Architecture

The architecture of the developed chatbot is based on the presented architecture in Chapter 2. The core architecture consists of the integration of Microsoft LUIS as NLU engine, the Conversationflow Manager, the Response Manager and the Channel Connector. To provide the required knowledge for the chatbot a connection to the question-and-answer catalogue as well as the integration of a knowledge database is implemented. Further steps have to be accomplished to integrate the chatbot into the company's website, Facebook Messenger and Amazon Alexa. Figure 4.2 displays all components of the architecture and outlines how they interact with each other. The architecture of the developed chatbot is based on Microsoft's Bot Framework version 4 and enhanced with Microsoft LUIS

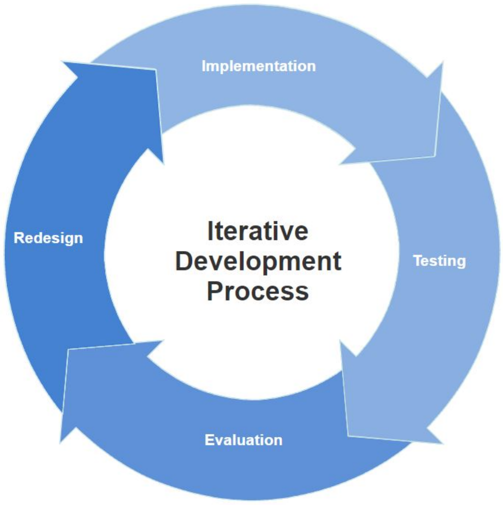

Figure 4.1.: Illustration of iterative development process

as language understanding engine and Microsoft's QnA Maker Service as question-and-answer catalogue.

4.1.1. Core Components

The core components of the chatbot architecture are the language understanding engine Microsoft LUIS, the Conversationflow Manager, the Response Generator and the Channel Connector. A deeper understanding of the tasks of each component and how they work together is provided in the following enumeration:

1. **Microsoft LUIS (NLU)**
   The designed language model for Microsoft LUIS, described in detail within Chapter 3.3.2, is configured through the graphical user interface provided by the service. After the configuration, the language model is trained by adding a large number of utterances for each intent to improve the intent and entity detection of the model. Nevertheless, the training of the model is not finished after these first utterances are added. Moreover, the training of the language model takes place

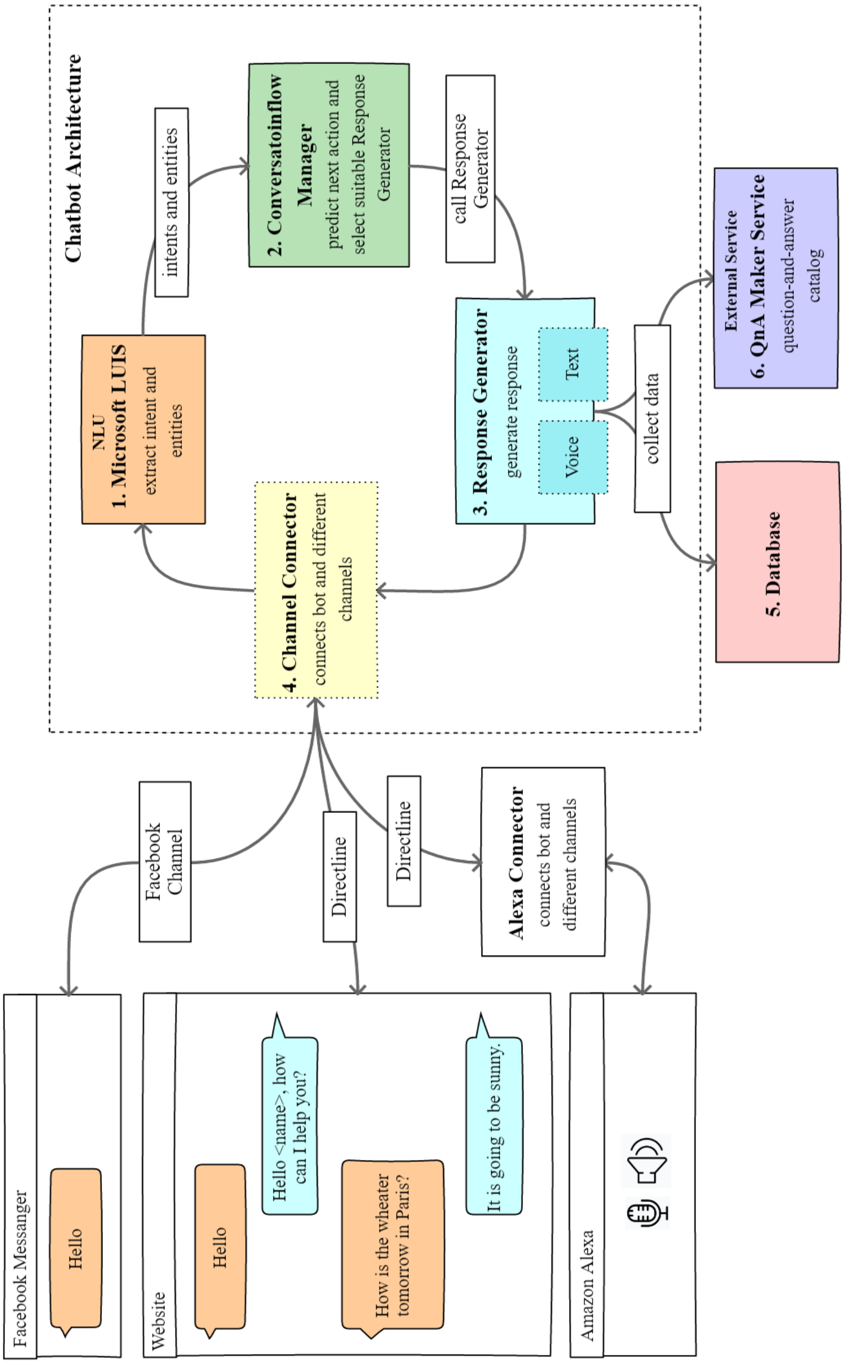

Figure 4.2.: Implemented chatbot architecture (based on presented architecture of Chapter 2)

continuously during the whole development process. As test users are regularly testing the chatbot example utterances are collected which are then used to further improve the language understanding capabilities of the system. Especially messages that are not classified correctly help increase the prediction accuracy immensely. After the setup of the LUIS language understanding service, it can be easily integrated into the chatbot architecture by using the Microsoft Bot Framework SDK. After the configuration of the required Application Programming Interface (API) Key, the LUIS application id and the used Domain (in this case the domain Westeurope is used), LUIS is ready to use within the chatbot. Users make typos. To prevent the need to train the language understanding engine to handle typos correctly the Bing Spell Check was integrated. After the registration of the service within Azure Bot Service, it is possible to add the service to the LUIS configuration within the chatbot system. After this configuration, the user request is processed by the spell check service before the language understanding engine is used to identify the appropriate intent and detect the entities. The intent and entities are returned to the chatbot and can be further processed. The following code snippet shows an example of a LUIS request result. The result includes the following properties:

- **alteredText:** The text after corrections of Bing Spell Check. If no correction is necessary this value is empty.
- **entities:** A list of all entities extracted from the user request with start and end index of the entity within the message as well as the value, type and matching score of the entity.
- **intents:** A list of all intents matching the utterance listed in decreasing order from highest matching score to lowest matching score.
- **text:** Actual user message.

After the LUIS result is received by the chatbot it is further processed. The best scoring intent is saved and the entity values are extracted and prepared to use them as filter values for example within the database.

Listing 4.1: JSON LUIS result for the request *"When are the performances of Tosca?"* (original: *"Wann spielt Tosca?"*)


```
{
    "recognizerResult": {
        "alteredText": null,
        "entities": {
            "$instance": {
                "Event_Name": [
                    {
                        "endIndex": 17,
                        "score": 0.9997958,
                        "startIndex": 12,
                        "text": "tosca",
                        "type": "Event.Name"
                    }
                ]
            },
            "Event_Name": [
                "tosca"
            ]
        },
        "intents": {
            "EventTime": {
                "score": 0.9997124
            }
        },
        "text": "Wann spielt Tosca?"
    }
}
```


2. **Conversationflow Manager**
   The Conversationflow Manager is used to determine the next actions to take according to the intent and entities extracted with Microsoft LUIS as well as information extracted from previous requests. The component supports two different approaches:
   - dialogues
   - single-turn message

   The term dialogue in this context refers to the concept of dialogues of the Microsoft Bot Framework. It enables the chatbot to setup a predefined conversation. This means it is possible to define multiple request-response blocks. Each of these blocks includes a chatbot message (typically a question) as well as the definition of actions according to the subsequent user input. After one block is finished, typically the next block is called. For example, if the chatbot asks the user if she or he wants to get further information of an event, possible answers are *"Yes"*, *"No"* or any other user input. If the answer is "Yes", the chatbot provides suitable information and continues with the next block within the conversation that asks if the user wants to see the cast. Nevertheless, if the answer is *"No"* the chatbot assumes that this event is not of interest for the user and therefore does not continue within this predefined conversation. The same applies if the user input does not match any of the predefined answers. Then instead of continuing with the predefined conversation, it is skipped and instead handles the message with the second mentioned approach, single-turn messages. These messages make use of the intent identified by LUIS in the previous step. The intent is used to define the conversation path. If the intent marks the begin of a predefined dialogue, the first block of the dialogue is called and the following user input is again handled with the dialogue approach. If no dialogue is triggered the chatbot gathers the required information from one of the data sources described later in this chapter. No matter which approach is applied, to send the response back to the user the Response Generator is used. As different channels are supported by different Response Managers the Conversationflow Manager has to identify the channel used by the user to send the message and call the suitable Response Manager.

3. **Response Generator**
   Due to the different output channels, the chatbot should support it was not adequate to implement just one single component to handle voice as well as text messages. Instead, the Response Generator is split up into a generic Response Generator and additional channel-specific Response Generators. The generic Response Generator is used for all responses that do not have to be adapted for the specific channel, like the greeting and closing messages. The channel-specific Response Generators are used to ensure suitable messages for each channel. The channel-specific Response Generators are divided into speech-based and text-based components in the first step but Facebook, for example, does not provide support for customized message cards the text-based

Response Generator is enhanced by a Facebook Response Generator. By splitting up the Response Generator component into these parts following problems are solved:

- Certain phrases or words are not suitable for specific channels: Phrases like *"Use one of the proposed questions or type your question to continue the conversation."* (original: *"Benutzen Sie eine der vorgeschlagenen Fragen oder tippen Sie eine eigene Frage um das Gespräch fortzuführen."*), as shown in Figure 4.3, is improper in the context of voice-based messages as the user does not *type*, but rather *says* what he or she wants. By applying the division between the voice- and text-based Response Generator it is no problem to specify a different wording according to the used output channel.

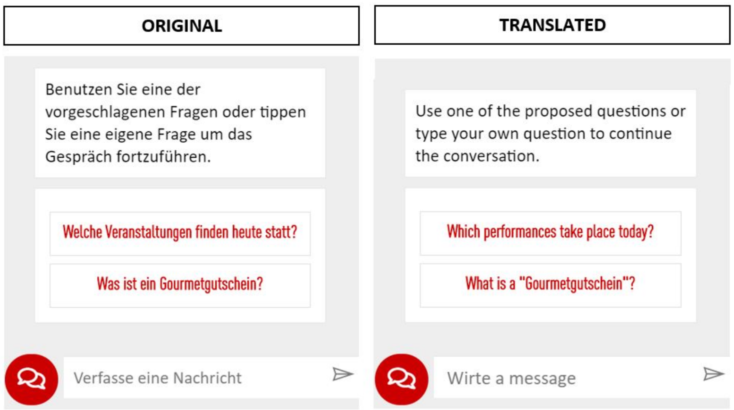

Figure 4.3.: Example of improper wording for voice messages

- Proposed user answers have to be presented differently to prevent confusion:
  The best way to describe this problem is by providing an example: After asking the chatbot for help it is configured to ask if example questions should be proposed. Using a channel that is supporting text messages the configuration of the response is easy as only buttons with possible questions have to be provided. Nevertheless, when sending the same message as voice response

the phrasing is not suitable to be read to the user. By read the button texts out, it sounds like the voice assistant asks questions instead of proposing possible questions. Responses like this have to be rephrased to be suitable to be read out loud. After adding the phrase *"Possible questions are"* (original: *"Mögliche Fragen sind"*) before reading the proposed questions out the user is no longer confused what the chatbot actually wants.

- Images and Videos are not available within voice messages:
  The chatbot cannot refer to images or videos within the context of voice-based messages as the user is not able to see them. This responses have to be adapted to either describe the main information of the image or removed from the answer. Instead of displaying for example the map of possible parking lots it is possible to tell the address or name a well-known place nearby).

- Voice messages should adapted for voice output:
  As humans do not talk in a monotonous voice the user also expects emphasis and breaks from the chatbot while communicating with a voice-based interface. Therefore the messages have to be enhanced with these elements by using Speech Synthesis Markup Language (SSML).

The only way to enhance voice responses is by providing emphasises, whereas text-messages can be enhanced in multiple different ways. Most of the channels support markdown, which is an easy way to improve the style and usability of a message. Text can, for example, be styled bold, as shown in Figure 4.4, and links can be added. Ad-

Figure 4.4.: Example of markdown used to highlight important information like the name of the performance

ditionally, so-called Cards can be sent to the user. Cards can be used to add media or buttons to a message. The Card types used for this chatbot are the Hero Card and the Adaptive Card. The Hero Card can contain a single image, text and buttons. In contrast to Hero Cards, it is possible to fully customise the texts, buttons and images within Adaptive Cards. Even though Adaptive Cards are more complex to create the content can be adapted to fit well into the card which is not always possible for Hero Cards. Unfortunately, they are currently not supported by Facebook Messenger. Therefore Hero Cards are used instead of Adaptive Cards for this channel. Figure 4.5 displays the different appearance of the Hero Card (Facebook Messenger) and the Adaptive Card (Website). The text position is predefined within Hero Cards and can differ depending on the channel. Adaptive Cards are more flexible concerning the style, but are not available in every channel.

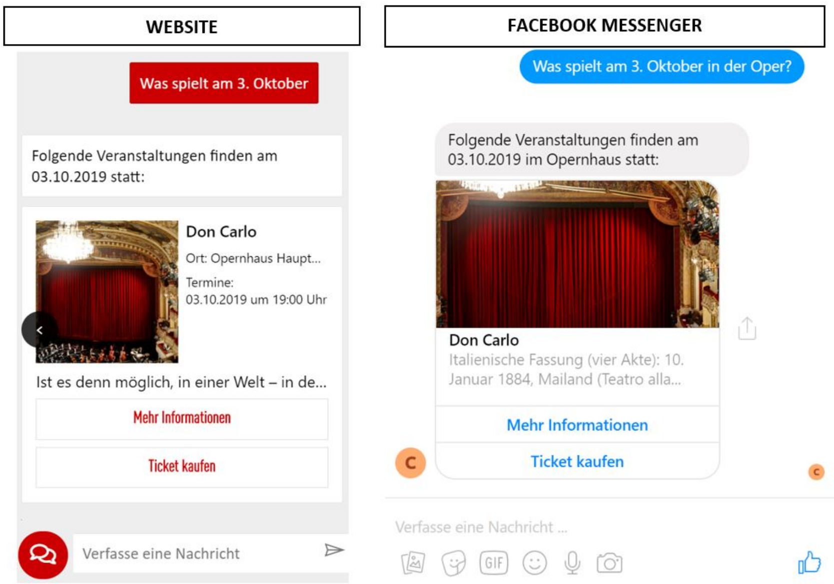

Figure 4.5.: Example of the different appearance of same messages within the chatbot on the website (left) and the chatbot within the Facebook Messenger (right)

4. **Channel Connector**
   The Channel Connector component is used to connect the chatbot

with all the required channels to run at Facebook Messenger, Amazon Alexa as well as the company's website. For this chatbot, the Bot Framework Service is used to fulfil the task. It can transform messages from the bot framework's schema to the channel-specific schema, as long as it is supported by the channel. Not all messages that can be configured by using the Bot Framework SDK are supported in every channel. As already mentioned, for example Facebook Messenger does not support customized message cards but supports standard message cards, so-called Hero Cards. After the message endpoint for the Channel Connector and the name of the chatbot is configured, the developer can activate the required channels. The settings required for the activation are different from channel to channel. The required channels for this chatbot are the Facebook Messenger channel, the Web Chat channel as well as the Directline channel. The Directline channel is used to integrate the chatbot into Amazon Alexa. As the Bot Framework Service does not support Alexa by default, an additional service has to be developed that handles the transformation of bot framework messages to messages for Alexa and vice versa.

- Website
  To integrate the chatbot into the company's website the Web Chat channel has to be configured. After adding Web Chat to the list of supported channels a secret is obtained that has to be used within the website to authenticate the Web Chat API requests. To integrate the chatbot into the website Bot Framework Web chat[1] client is used. As publishing the obtained secret within the JavaScript client code would be very insecure a Rest API endpoint is generated to obtain an authentication token for each user. This token is then used to setup the Web Chat client within the website. Additionally, the Web Chat client provides the possibility to adapt the style of the chatbot according to the company's design.

- Facebook Messenger
  The integration into the Facebook Messenger requires two steps, to create a Facebook application and register the channel within the Bot Framework Service. Additionally, a Facebook Page is re-

[1] https://github.com/microsoft/BotFramework-WebChat

quired through which the bot can be accessed. After the Facebook application is configured it has to be connected to the Facebook Page of the company, to enable users to access the bot. The App ID, as well as the App Secret, Page ID and Page Access Token acquired during the setup process of the application, are needed to configure the Channel Connector within the Bot Framework Service. If all data is correct the user is now able to communicate with the chatbot through the Facebook Messenger.

- Amazon Alexa
  As Amazon Alexa is not supported by default this is the most complex part of the integration. First of all, it is necessary to create a skill for Alexa. Additionally, a service has to be developed that is responsible for translating Alexa messages to Bot Framework messages and vice versa. The last step is the connection of all three components, the Alexa skill, the connector service and the chatbot.

  a) Alexa Skill
  To create an Alexa skill, first of all, an invocation name must be configured. This name is required if the user wants to access the skill. As Alexa also works with intents and entities to detect what the user requests, this language model is required too. Nevertheless, as the chatbot relies on LUIS to detect intents and entities the language model for Alexa is simply configured to forward all user input to a single intent, named ”phrases”. The last configuration needed is the endpoint that communicates with the Alexa skill. This endpoint is configured to the connector service described in the next section.

  b) Connector Service
  The task of the developed connector service is to react to Alexa events like OnLaunch (new user access skill) or OnIntent (the intent was detected) as well as sending messages to and receiving them from the chatbot by using the DirectLine channel. Whenever the ”phrases” intent is detected, the

service forwards the user request to the chatbot. The chatbot processes the request and returns the response which is then converted into a suitable message for Alexa and forwarded to the user. As Alexa provides some specific intents like Cancel, Stop, Help and Home by default, also these intents have to be handled within the code of the service.

c) Channel Registration
As already mentioned, DirectLine is used to connect the connector service with the chatbot. DirectLine is used to connect the chatbot with client applications. After the channel is activated within the Bot Framework Service the obtained secret has to be configured within the connector service to ensure authenticated DirectLine API calls.

4.1.2. Content Integration

Even if a chatbot can understand and respond to a user request it is useless if it is not able to provide any information. Therefore another important aspect of a chatbot system is the integration of the content. For this chatbot, three different data sources are available: a database for dynamic requests, a question-and-answer catalogue for frequently asked questions and some links that are used to enhance answers with links to further information. These data sources are described in the section below.

5. **Database**
The core information, the information concerning performances and casts, is saved within a database hosted on Azure SQL Server. The information within the database is provided by the customer and refreshed once a day to keep information like cancelled events or changed casts up-to-date. The database is an abstraction of the original complex database structure of the customer and provides the relevant information within two tables:
    a) Termine
    The table "Termine" (English: Events) covers all relevant information concerning the performances and events that take place.

This includes the event name, date, start- and end-time, price, description and other relevant data.

b) Abendbesetzungen
Within the table "Abendbesetzung" (English: Cast) contains information concerning the performers, their role names, a link to the performance as well as other related data.

The relation and all properties of these two tables can be seen in the provided UML database model (Figure 4.6). The database is connected to the logic of the chatbot and information is retrieved whenever necessary to fulfil a user request. As within the first step of the user message processing the entities have been extracted from the request and transformed into filter values it is now possible to use them as for the database queries. The user can filter for following values within for events:

- name
- date
- start- and endtime
- location
- category

The cast information can be filtered by:

- cast name (first name and/or last name)
- role name
- event

Additionally, when searching for information it is always possible to use filters from the event table for cast information and vice versa. This is because the tables are related to each other. The event table has a zero-to-many relationship with the cast table. This means one event can have zero or multiple cast table entries that are related to it. This enables complex searches and provides the user with the possibility to customize the user query, which is not possible when simply navigating through a website.

6. **QnA Maker Service**
As the customer wants to be able to maintain and manage simple

Figure 4.6.: UML Databes Model

frequently asked question by himself it is important to provide a user interface to fulfil this task easily. The QnA Maker Service of Microsoft provides this possibility as a simple and easy to use graphical user interface is provided that allows to edit and delete existing information as well as to add new information to the knowledge base (KB). The initial KB is setup with provided frequently asked questions from the customer service centre of the company and enhanced continuously during tests with real users that take place regularly within the development process. Therefore, fundamentals concerning the requested information are already available at the launch of the chatbot. The QnA Maker Service is designed to be easily integrated into the Microsoft Bot Framework architecture. To configure the service only the identification number of the knowledge base, the hostname and the endpoint key of the service are required. Additional settings like the minimum threshold value as well as the maximum number of received results can be defined. As for certain requests many similar responses are expected the maximum amount of results returned by the service is limited to 10. This enables further processing and delimitation of the results according to custom needs within the code. Each result is delivered with a confidence score. This score indicates the degree (0 - low, 100 - high) how well the query matches results within the KB. To filter unrelated questions it is possible to define a minimum threshold that has to be reached for results to be returned to the chatbot. According to a guideline of Microsoft (2019b) results with a confidence score below 30 typically do not answer the question of the user whereas a confidence score indicates a high probability that the user gets the desired answer. Nevertheless, the threshold for the service is configured to the value of 30. Even though the KB is not able to identify a concrete answer, this threshold allows answers that are likely to be related to the topic and might provide some useful information for the user. The initial question-and-answer catalogue was provided by the customer. It included questions like parking possibilities, opening hours or discounts. Many predefined questions are similar to each other, especially information concerning certain information (like opening hours or discounts) of the different theatres. Additionally, also chit chat is defined by using this question-and-answer catalogue. To ensure good results following problems have to be solved:

- Detect the best result
  The best way to ensure good answers to the users' questions is to have a look at the provided confidence score of each result set received from the service. The higher the confidence score the better the result. A score of 100 is the highest possible value and identifies a nearly perfect match. If an answer with this confidence score is received it can be directly forwarded to the user at this is typically the requested information. Nevertheless, it is not always easy to determine the best answer of all the provided results of the service as the confidence score might be very low or similar for multiple responses. If no clear answer is identified multiple filters are applied to nevertheless get a good response for the user. If no single best answer is detected in the first step it is checked if a certain topic is requested. How this is done is covered within the next bullet point. In case no specific topic is requested the result set is filtered by a more restrictive threshold to detect if this narrows down the possible answers. If this is not sufficient a check concerning the score difference between the answers is applied to identify a possible best match among these answers. If the result of this limitation is still not meaningful enough the results are further limited by a minimum acceptable threshold (60). After all these filters are applied there is either an identified answer that is forwarded to the user, multiple possible options that have to be further clarified (as described later) or no result at all that can be sent to the user.

- Detect related content
  As similar content is not always requested with similar phrases and is therefore hard to detect by the language engine of the QnA Maker service another possibility is provided to narrow the provided results down to a single best answer. Therefore, it is possible to define meta information for each question-and-answer pair within the catalogue. This information can later be used to filter the result set for similar content. If a user is looking for all kinds of discounts available for certain ticket LUIS is trained to detect the topic of the QnA request as an entity. After the QnA Maker service has delivered appropriate matches for the user

query this entity can be used to filter the result set for all answers with this topic within their meta information. This enables similar content to be detected, even if the wording is not similar.

- Handle missing theatre information
  As a large number of the questions within the catalogue are related to one of the theatres the service can't identify the correct answer if no theatre is defined within the request. For example, the question *"What are the opening hours?"* (original: *"Wie sind die Öffnungszeiten?"*) varies from theater to theater. If it is not explicitly specified for which theatre the information is requested, multiple answers are returned (in this example the opening hours for all theatres) and the confidence score of these results are very similar. To deal with this problem two solutions are applied:
  a) Checking for context information
     Depending on the previous conversation it might be possible to extract the missing information from the given context. For example, if the user previously asked for upcoming events at the opera house it is likely that a question concerning the opening hours after that also targets the opera house. An example is shown in Figure 4.7.
  b) Asking for clarification
     Nevertheless, if no previous conversation exists or it is not possible to extract any information from it another approach has to be applied. To identify the user's intention it is necessary to ask for clarification. Therefore, related questions from the knowledge base are proposed to the user. She or he is then able to select a suitable question or try to get the information by using a different wording if no option is suitable. Figure 4.8 illustrates an example.

The knowledge base is not only used for frequently asked questions but also enhanced with common chit chat that the chatbot should cover. This includes for example questions like *"How are you?"* (original: *"Wie geht es dir?"*) or replies to messages like *"You are wrong"* (original: *"Du liegst falsch"*). As chit chat answers might be among the top-scoring matches for frequently asked questions it is important to

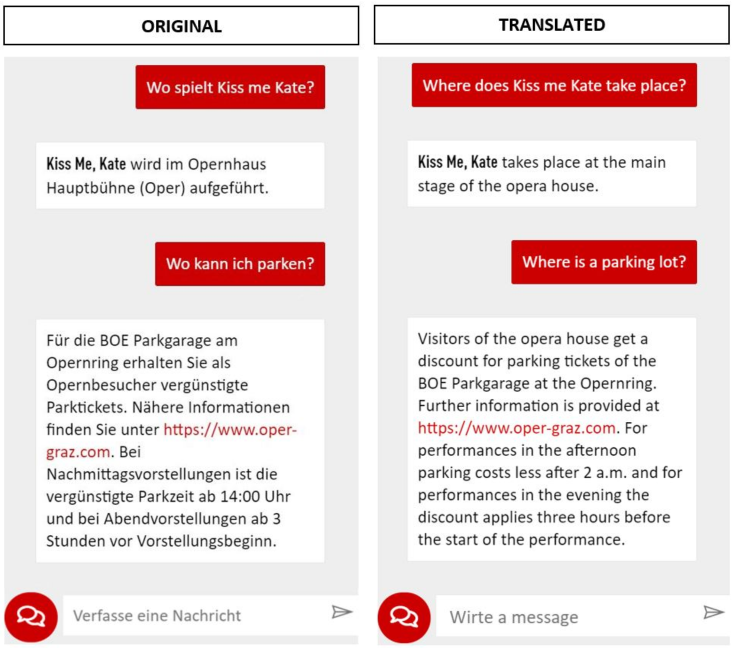

Figure 4.7.: Example of context specific questions: As the performance takes place at the opera, a parking house nearby the opera is proposed

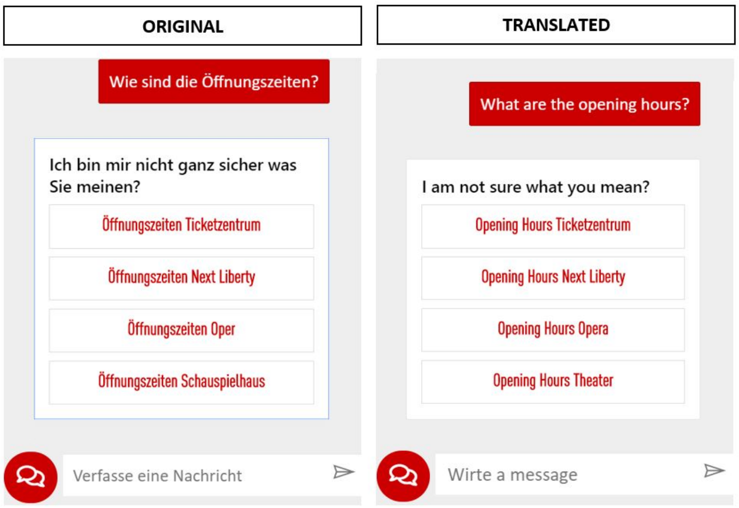

Figure 4.8.: Example of clarification request: If the chatbot cannot identify the answer as some similar options are found, the user is provided with these options to select the desired one

remove them from the answer list. Additionally, when dealing with chit chat messages it is not appropriate to propose possible answers to the user. Therefore, the message with the best score is selected and sent to the user.

7. **Integration of existing sources**
   As the chatbot is not only developed to simplify information gathering for the user but also as a marketing tool for the customer it is important to integrate already existing content too. This means the chatbot should not only answer questions but also provide links to websites of the theatres as well as campaigns of the customer. This includes, for example, the campaign of a free ride on public transports or the voucher for a dinner before or after a performance. By providing these additional links the customer expects a rise in awareness for these campaigns. Leading from the chatbot to the websites of the theatres should also increase their visitor numbers.

Even though it is very important to have a working chatbot this is not enough to motivate the user to communicate with the chatbot for a longer period or come back after the first communication. Therefore the integration of user engagement strategies is essential for the success of the chatbot.

4.2. Integration of Onboarding and User Engagement

Even though numerous onboarding and engagement strategies have been encountered in literature and summarized in Chapter 2 it is not always easy to put the theory into practice. Many elements and mechanisms applied to engage the user during the whole period of engagement are also crucial within the first few minutes of user interaction, the onboarding phase. The following section describes how the strategies are applied to the chatbot.

1. **Point of Engagement**
   As the first impression is very important the initial chatbot message is used to introduce the user to the chatbot, as shown in Figure4.9.

Additionally, the message explains the scope of the chatbot and offers a help button as well as a possible question to help the user to get started.

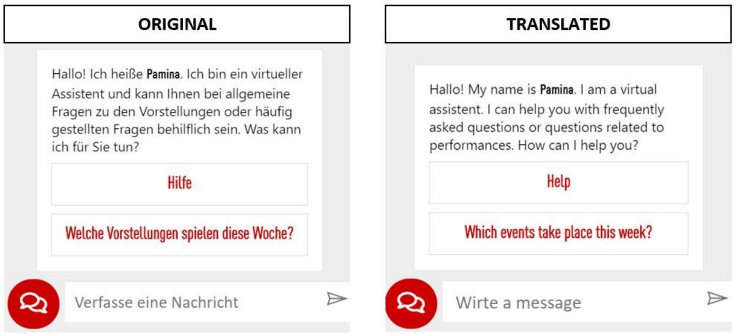

Figure 4.9.: Example of initial chatbot message

Many users are new to the concept of chatbots and therefore not used to this technology. The help function is integrated to support the user with the first interaction with the chatbot. If the user requests help not only a detailed definition of the covered scope is given, but also example questions are delivered (as shown in Figure 4.3. This questions simplify the start of the conversation and provide and demonstrate how requests can be phrased. Additionally, within the help message, the user is encouraged to frame own questions to continue the conversation. Users tend to get confused if words or phrases are used they are not familiar with. As proposed by a user of the focus group the term "chatbot" is therefore not used within the initial message. The chatbot rather refers to itself as a virtual assistant, as this term is more familiar to them. To engage the user to interact with the chatbot it appears as a popup after 5 seconds on the website. Therefore users are informed that there is an alternative to navigating through the whole website to get information. This is not only a good way to gain the user's interest but also a suitable approach for reengagement. To keep the user engaged during the conversation, images and buttons (or emphasises for voice-based communication) are used.

2. **Period of Engagement**

   - Usability
     As already mentioned, the scope of the chatbot is limited to questions concerning upcoming events, cast details and frequently asked questions concerning the different theatres. Additionally, basic support for chit chat requests is provided. Real test users are included during the whole development process and their suggestions are evaluated and implemented. One suggestion was, for example, the rephrasing of messages for voice assistants. As there is no predefined conversation flow that the user must follow it is not easy to simplify the usage for example by providing buttons. Nevertheless, if the user is interested in performance follow-up questions are proposed. For example, if the user read the content of the play he or she might also be interested in the cast. Therefore a message is delivered to ask the user if he or she wants to see cast details. The user is now able to get further information by simply clicking a button. An example of this approach can be seen in Figure 4.10. These buttons simplify the conversation for users and engage them to continue with the dialogue. If the chatbot was not able to find any suitable information it responds with a proper message. This message does not only include the hint that for example no event was found but also mentions the date, location and category (if given) that have been used as a filter for this request. This was due to the fact that sometimes old filters have still been active when they where no longer needed or new filters were not recognized. By displaying applied filters the user can react if something is not as expected.

   - Content & Design
     Images and videos are an important factor to keep the conversation interesting, but also buttons and markdown are used to improve the user experience. Additionally, whenever possible the messages are kept short to prevent overwhelming the user with too much information. Also, the actual information that

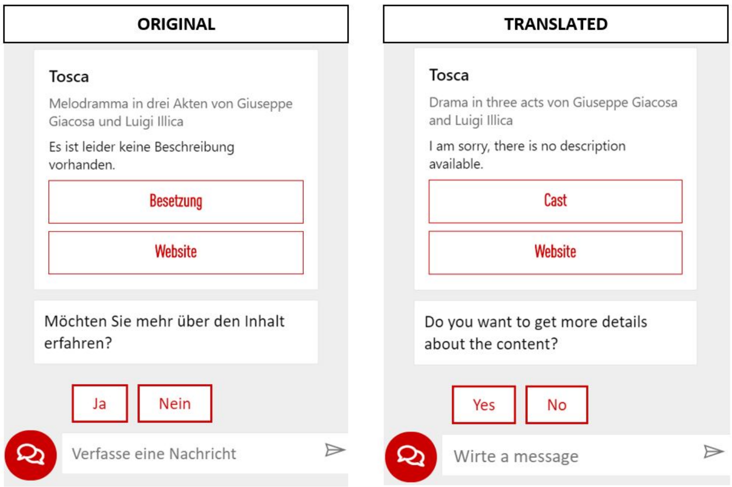

Figure 4.10.: Example of additional buttons displayed to simplify the communication

is delivered is kept as short as possible to still answer the user request. Furthermore, different Response Managers are used to adapting the responses to the requirements of each platform and only deliver content that is suitable for it. This approach provides the possibility to make use of special features or layouts to improve the responses. To integrate the chatbot into the company's website, the cooperate colours, red and grey, are applied to buttons and text bubbles of the chatbot. The interface is kept clean and simple. For the chatbot at the company's website only the input field, a send button and the message bubbles are displayed. Therefore the user does not get confused or distracted and can focus on achieving the desired goal.

4.3. Summary

The implementation of the chatbot is done by using Microsoft Bot Framework. An iterative development process is followed, including the implementation, testing, evaluation and redesign phases that were repeated until the chatbot's performance was sufficient. The chatbot architecture consists of the core components (NLU, Conversationflow Manager, Response Generator and Channel Connector) and the knowledge bases (database and question-and-answer catalogue). Microsoft LUIS is used as a natural language understanding engine. After the configuration of this service, the engine is trained with the language model designed in the previous chapter. The intents and entities extracted with LUIS are forwarded within the chatbot architecture to the Conversationflow Manager. This component determines the next actions based on the provided input (intent and entities). It decides if the message is part of a multi-turn dialogue or just a simple single-turn message. Additionally, this component selects the appropriate Response Manager (text-based or speech-based) to return the message in an appropriate format. The selected Response Manager than takes the provided entities and uses them to identify a suitable response by querying the available knowledge bases and transforming them into an appropriate format. The final message is then delivered to the Channel Connector Component, which then forwards the message to the user by using the appropriate channel (Facebook Messenger, Website or Alexa). To be able to provide the chatbot for Alexa a so-called Alexa Skill has to be configured and as the Bot Framework does not support Alexa by default a connector service to connect the chatbot with Alexa is developed. The knowledge bases used to get the desired information for the responses are a database that contains all event and performance specific information as well as the QnA Maker Service of Microsoft. The Microsoft service is used to predefined question-and-answer pairs that can be queried with the user input of the chatbot. During the development process, several challenges are mastered, including missing support for some language features within Microsoft LUIS, compatibility issues for messages across the supported platforms and the selection of the most appropriate response among similar options. As not only the core functionality is important for the chatbot, but rather the users that interact with the system onboarding strategies are applied. To gain the user's attention the chatbot is configured to pop-up after the user has been five seconds on the company's website. The initial message is then designed to give a short description of the chatbot and provide buttons for the help functionality as well as a possible question. The help functionality is very important as this should help new users to learn how to interact with the chatbot. Therefore the covered scope is explained in more detail and possible questions are outlined. The usability of the chatbot does not only have to be present within the first few minutes of interaction (the onboarding phase) but rather must be there during the whole period of engagement. This includes basic chit-chat capabilities, a well-trained language model, well-formulated responses (also for errors) and buttons and links to simplify the interaction for the user. Images and videos are integrated as they are easy tools to engage the user. The messages are kept as short as possible without reducing the quality of the responses. To optimize the responses for the different channels and handle channel specific characteristics (like not supported or additionally supported functionality) the Response Manager is used.

5. Evaluation

Chatbots suffer from a lack of user acceptance and a many users stop interacting with the bot after a few messages Debecker (2017). People that are not used to the technology prefer traditional information sources like websites or help desks and avoid using chatbots Pega (2017). Additionally, at the beginning of the rise of the chatbots, many of them had suffered from bad performances concerning usability and language understanding. Even though user acceptance is increasing in some fields (like customer service), onboarding strategies and user engagement are required to develop successful chatbots. Therefore appropriate elements and mechanisms extracted from these strategies were applied to the developed chatbot. To identify whether and in which degree these strategies help users to get to know the technology faster and measure their engagement during the interaction a user survey was conducted that is evaluated in this chapter.

5.1. Methodology

The evaluation of the onboarding and engagement strategies was conducted within two groups. The first group (Group A) is confronted with a chatbot with nearly no onboarding and just simple engagement elements. The second Group (Group B) is supplied with the chatbot that involves onboarding and engagement elements and mechanisms. The study is conducted with each participant separately and the participant and the supervisor are in the same room during the whole experiment. The procedure is the same for all participants of both groups:

1. Short introduction and explanation of the topic

2. Pre-questionnaire to evaluate the experience and expectations of the participant
3. Handout of the task description
4. Experiment: Interaction of the participant with the chatbot (either chatbot A or chatbot B)
5. Post-questionnaire to evaluate onboarding and user engagement of the tested chatbot

5.1.1. Participants

The evaluation of the chatbot is conducted with 12 participants, divided into two groups: Group A and Group B. The allocation of the groups is random and is not influenced by gender, age or technical experience of the participant. Hereby, the term "technical experience" refers to the daily usage of computers. All participants are between 20 and 60 years old and in total 8 male and 3 female are consulted. The gender and age distribution, as well as the average age and the amount of technically experienced participants, can be seen in Table 5.1

	Group A	**Group B**
Gender	5 male, 1 female	3 male, 3 female
Age	20 - 59	20 - 60
Age Mean	37.17 (SD=16.78)	36.00 (SD=17.39)
Technical experience	4 experienced, 2 inexperienced	3 experienced, 3 inexperienced

Table 5.1.: Characteristics of participants of the evaluation groups

5.1.2. Material

To evaluate the effect of user engagement and onboarding strategies on the user's experience while using the chatbot the participants are provided with a pre-questionnaire, a task description and a post-questionnaire. For a better evaluation concerning the effect of the strategies, each group is provided with a different chatbot, chatbot A and chatbot B.

Pre-Questionnaire

The pre-questionnaire is used to gather general information about the participant like gender, age and technical experience. Also, it serves to identify the experiences and expectations of the participants. It is evaluated whether or not the user already interacted with a chatbot and how he or she would rate their satisfaction concerning usability and accuracy of the results by using the Likert scale between low satisfaction (1) and high satisfaction (5). Additionally, open questions are used to gather the user expectations concerning language, appearance and functionality. It is also evaluated which channels the participant prefers using the chatbot. The pre-questionnaire can be found in Appendix B.

Task Descriptions

Users typically visit this website or use the Alexa Skill to either get some information or buy a ticket for one of the performances. This means, they have a certain goal in mind that they want to reach. Also, by watching the test users consulted during the iterative development process it became clear, that it is hard for the user to interact with the chatbot without any goal he or she wants to achieve. Therefore, the user is provided with three tasks he or she has to fulfil with the help of the chatbot during the experimental part of the evaluation process. The provided task descriptions can be found in Appendix C. The tasks are derived from the core functionality of the chatbot and are the same for both groups. In the beginning, the users can familiarize themselves with the chatbot. The first task requires the users to find an event or a performance they want to visit. After a preferred performance is chosen the users have to find additional information related to it, like the duration, price or cast. Within the last exercise, the user has to pretend to order some tickets for this performance. After the user has fulfilled these tasks the experimental part is finished and the user can continue with the post-evaluation.

Chatbots

For the evaluation of the onboarding and user engagement strategies, two different chatbots are provided. Group A interacts with a plain chatbot with little to no onboarding and engagement elements, whereas Group B communicates with the chatbot that was developed with a focus on onboarding and user engagement. Figure 5.1 displays the initial message of chatbot A and Figure 5.2 represents the same message from chatbot B.

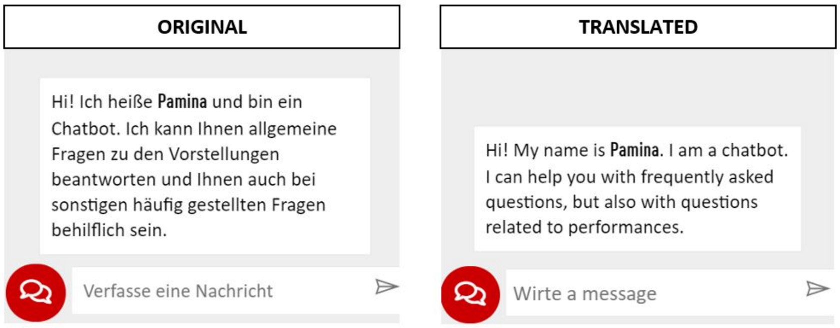

Figure 5.1.: Introduction message delivered by chatbot A

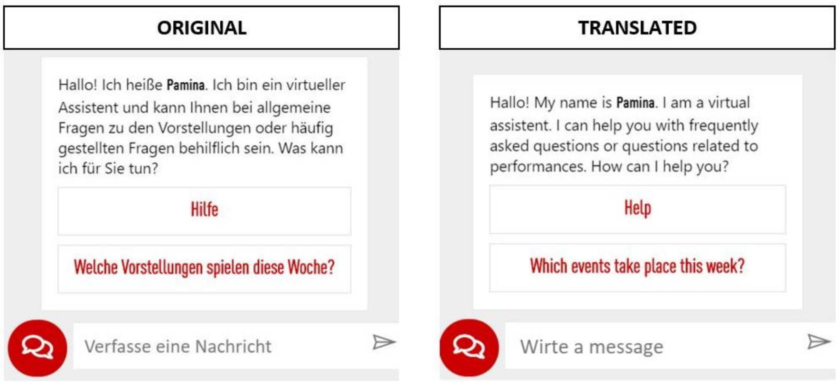

Figure 5.2.: Introduction message delivered by chatbot B

The main onboarding strategies added for chatbot B are:

1. popup to attract the user's attention
2. advanced introduction message

3. "call-for-action" buttons
4. help functionality with a detailed explanation of the chatbot's scope
5. further explanation and proposing some questions to show how to interact with the chatbot
6. propose additional question concerning information a user might want to know about events and performances during the conversation
7. enhance responses with buttons and links to further information

An example for the engagement strategy of "call-for-action" elements that are used within chatbot B can be seen in Figure 5.3, whereas the same message of chatbot A is displayed without any further elements. The message delivered by chatbot A is visible in Figure 5.4.

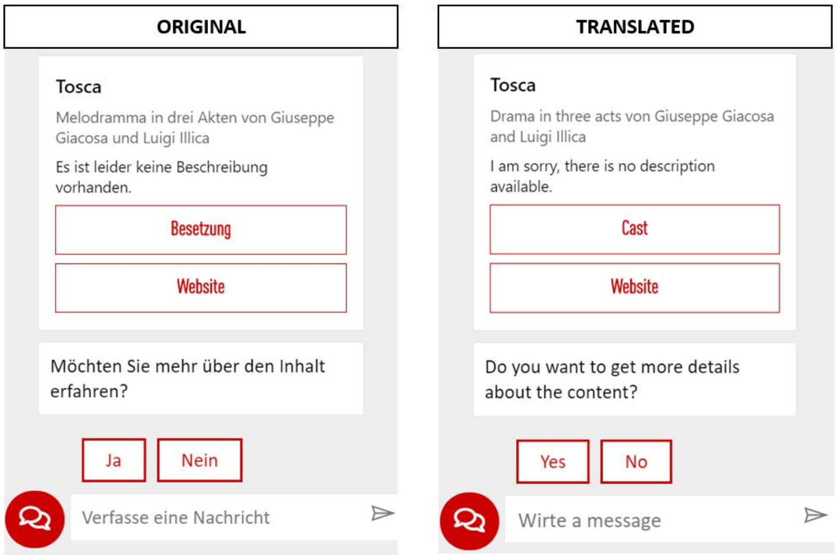

Figure 5.3.: Introduction message delivered by chatbot B

Even though the bots differ concerning the applied strategies and mechanisms, the overall structure, appearance and delivered information is the same for both chatbots. While chatbot A focuses on the information, chatbot B additionally provides further help and enhances the responses with helping elements like buttons or asking additional yes/no questions to support the user to get all the details needed.

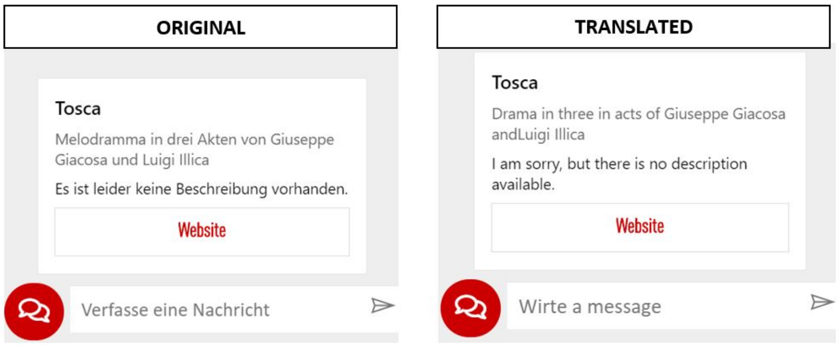

Figure 5.4.: Introduction message delivered by chatbot A

Post-Questionnaire

The post-questionnaire is used to evaluate the user experience of the experimental part of the evaluation process. Therefore the onboarding phase and user engagement are evaluated. To identify the user engagement within the post-questionnaire, the User Engagement Scale (UES) provided by O'Brien, Cairns, and Hall (2018) is used. They define a scale to measure user engagement that has already been used in various digital domains. As already mentioned in Chapter 2.2 they identified six dimensions of engagement (aesthetic appeal (AE), focused attention (FA), novelty (NO), perceived usability (PU), felt involvement (FI) and endurability (EN). As the scale was later adapted by combining novelty, felt involvement and endurability to a single factor, called reward factor (RW), this adapted scale is used for the evaluation of the chatbots. UES consists of 31 items. To evaluate the items the Likert scale between strongly disagree (1) and strongly agree (5) is used. Additional to the UES 16 questions have been identified to assess the implemented onboarding strategies. The used questionnaire for the post-evaluation can be found in Appendix D.

5.1.3. Limitations

Due to a time frame constraint, the survey was conducted with only a small set of participants. As only twelve people are consulted, the results can just be treated to give a tendency toward a possible increase of user engagement and simplification of user interaction. The results cannot be used to identify the degree of impact concerning these aspects. Even though a tendency concerning the influence of gender or age is derived from the results, further investigation has to be done to assess the validity of these assumptions.

5.2. Results

The results provided in this section are the combination of answers from the pre-questionnaire, the supervisor's perception of the experiment as well as the results of the post-questionnaire of Group A and Group B. All twelve participants handed in valid answers and no answer was missing. In the following sections, the results of these three aspects are presented.

5.2.1. Pre-Questionnaire

Within the pre-questionnaire, seven out of the twelve consulted persons stated that they already have used at least one chatbot. Five of them mentioned that they became aware of the chatbot as it was popping up at a page they where using. Two of the participants mentioned that they used the chatbot due to recommendations of friends and one person became aware of them through internet videos. The participants rated the satisfaction concerning the responses provided by the tested chatbots as moderate (M=2.83; SD=1.47). One person did not answer this question and stated that the experience was too short to rate the received messages. Also, the satisfaction concerning the usability of the chatbots was perceived as moderate (M=2.71; SD=1.38). Four of the participants that did not use a chatbot until this experiment explained that they did not notice chatbots on the websites they used. Only one person stated that low trust in complex results is the reason why the technology is not used. Additionally, one male participant mentioned that he does not want to use a chatbot at all. When rating the likelihood a participant would use a chatbot to get an event or performance-related information the results have been widely spread. While three participants would try out a chatbot for this purpose, three other participants find it very unlikely that they would use such chatbot. Interesting is, that three out of four female participants tend to try out a chatbot in this field (M=3.80; SD=1.78), whereas most male participants tend to ignore the chatbot (M=2.00; SD=1.16). The channel that is most likely to be used with chatbots is the website. Eight participants would use a chatbot on a website, while only one considers Amazon Alexa as a communication platform and only two participants would use the Facebook Messenger for this purpose. Most participants expect correct and relevant answers to their questions. Additionally to this information also the expectations of the chatbot concerning language, structure and appearance are asked. According to the participants, the design of the chatbot should be simple and intuitive. The chatbot should pop up on the page, but an additional icon should be provided to show and hide the chatbot whenever wanted. The language is expected to be clear, grammatically correct and factual. Two participants mentioned that the chatbot should support multiple languages.

5.2.2. Experiment

During the experiment, it was intended that the supervisor does not answer questions about how to use the chatbot. Nevertheless, two participants of Group A needed the help of the supervisor to start the conversation with the chatbot. Interesting was, that as they did not know how to interact with the chatbot they immediately started looking for the desired information on the website itself. Users of Group B have been provided with a button to ask for further help within the initial chatbot message. Half of the participants of this group made use of this functionality. Interesting is the huge variety of used phrases to ask the chatbot for the same information. While some participants used keywords to get the information, like *"Musicals October"* (original: *"Musicals Oktober"*), others relied on full sentences, for example, *"What performances can I see in October?"* (original: *"Welche Vorstellungen finden im Okotber statt?"*). Most participants were able to get the desired information immediately or after a small number of tries without the help of the supervisor. For a majority of the participants, collecting information for further details was easier. Especially for users of chatbot B, as the chatbot asks if further details should be delivered by itself. A problem that became clear during the experience was that certain phrases are not understood by the chatbot and the language model has to be further trained. Additionally, some questions were asked that request information that the chatbot does not have. For example, the chatbot can list all members of a cast for most performances but does not know which of them play main characters as this information is not available within the chatbot's data sources. In this regard, more suitable error messages are required. Nevertheless, all participants of both groups were able to fulfil all the required tasks. The duration of the experiment was between 4 minutes (technical experienced user using chatbot B) and 17 minutes (technical inexperienced user using chatbot A).

5.2.3. Post-Questionnaire

Within the post-questionnaire, the user engagement score, as well as the performance of the onboarding elements and mechanisms, are evaluated.

User Engagement Scale (UES)

Due to the small number of participants, the evaluation of the User Engagement Scale (UES) was not as straight forward as expected. To assess the reliability of the data Cronbach's α was calculated. It is used to measure the internal consistency of the dimension. While the reliability of the data was adequate for all dimensions of the evaluated data of Group B, the reliability of dimension FA of Group A was not sufficient at all (< 0.5). This means data collected within this dimension of Group A may have limited applicability. The inconsistency is caused by the question FA.3 *"I blocked out things around me when I was using the chatbot."*. When dropping this question a consistency of 0.748 would be reached, which is sufficient. The overall reliability of the data collected from Group A was excellent and the reliability of the data of Group B was adequate. To have accurate values it was necessary to reverse the items PU.1 to PU.6, Pu.8 as well as RW.3 since these values evaluate negative instead of positive aspects. Table 5.2 displays the reliability for all four dimensions of both groups, including the mean and standard derivation of each dimension. Considering the mean score within all dimensions it is apparent that the mean scores of the dimensions

FA and RW have slightly increased (+0.266 and +362), whereas the mean score of PU increased even more (+0.565) and AE shows the most significant improvement (+0.834). Figure 5.5 displays the differences between the mean score values of participants of Group A and Group B.

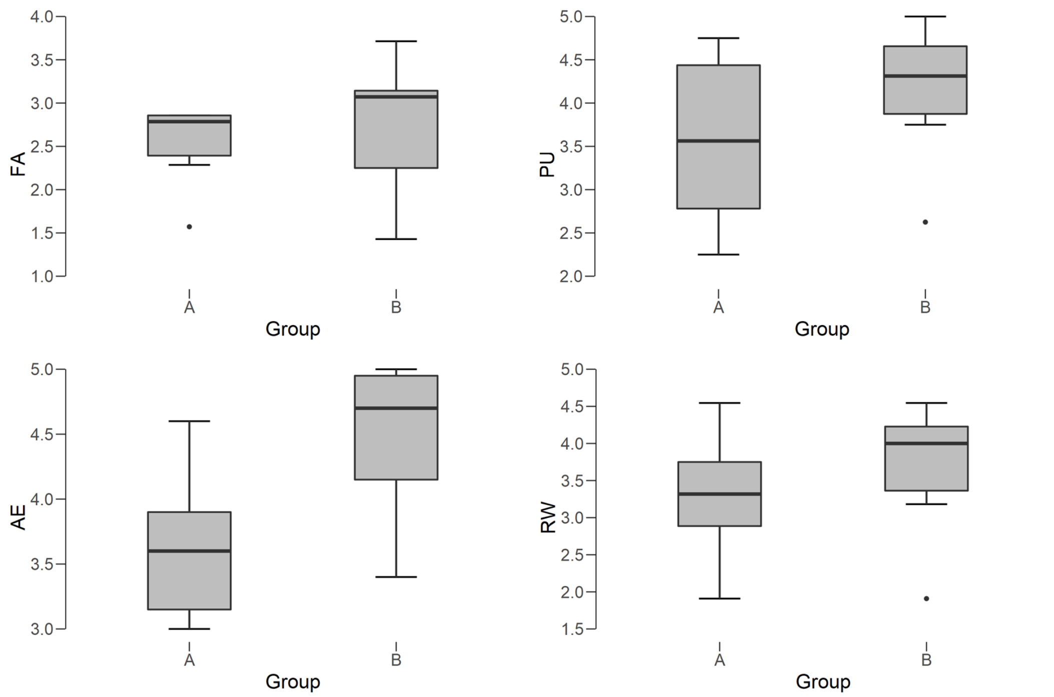

Figure 5.5.: The figure displays the boxplots of mean scores of the evaluated user engagement dimensions: focused attention (FA), perceived usability (PU), aesthetic appeal (AE) and reward factor (RW). Group A was confronted with the chatbot without any onboarding and only simple user engagement mechanisms, whereas Group B interacted with a chatbot including many of theses strategies.

Interestingly, the chatbot appearance that is evaluated within the dimension of aesthetic appeal has been the same for both chatbots, except of additional buttons provided by chatbot B. Nevertheless, chatbot B was perceived as more attractive (Group A: M=3.50; SD=1.05 | Group B: M=4.50; SD=0.84), aesthetically appealing (Group A: M=3.67; SD=0.812 | Group B: M=4.67; SD=0.52) and more appealing to the visual senses (Group A: M=3.33; SD=0.52 | Group B: M=4.33; SD=0.52). Also the dimension of perceived usability (PU) is higher for chatbot B. The scale of the user engagement score

Dimension	Group	Mean	SD	Cronbach's α
FA	Group A	2.524	0.690	0.348
	Group B	2.738	0.460	0.833
PU	Group A	3.56	0.235	0.930
	Group B	4.125	0.478	0.760
AE	Group A	3.633	0.321	0.810
	Group B	4.467	0.139	0.933
RW	Group A	3.288	0.578	0.932
	Group B	3.652	0.565	0.899
Overall Reliability	Group A			0.938
	Group B			0.815

Table 5.2.: The table displays the Scale Reliability Statistics of Group A and Group B by using Cronbach's α. The mean value displayed, is the average score of the participants of the group for the specific dimension. This score is based on the Likert scale between 1 (strongly disagree) and 5 (strongly agree).

is from 0 (no engagement) to 20 (high engagement). The score is gained by calculating the average score for each dimension of each participant and then adding these values. None of the participants of both groups had an engagement level below 10. While chatbot A only managed to reach an engagement score over 13 for two participants, the user engagement score of only one participant of chatbot B was below this mark. The average overall engagement score for chatbot A was 13.00, whereas chatbot B managed to reach a score of 14.98. Interesting is also the results concerning the age group of the participants. The classification used to categorize the generational classes is the same as already used within the focus group analysis of Chapter 3: Generation Z (< 25 years), Generation Y (25 - 39 years) and Digital Natives (> 39 years). For chatbot A as well as chatbot B Generation Z was rating the highest user engagement score (chatbot A: 15.43; chatbot B: 16.94). Participants of Generation Y and Digital Natives obtained a very similar score for both chatbots. Nevertheless, as the number of participants of each generational class was very small (less or equal to three participants) the validity of these assumptions require further investigations.

Onboarding Performance

The second part of the post-questionnaire is used to measure whether or not different onboarding strategies have influenced the user experience.

- **Simplification of Interaction**
  When comparing the evaluation results of both groups the positive influence of the onboarding strategies on the outcome can be seen. While on average participants of Group A rated the level of provided help as okay (M=3.167; SD=0.75), most persons of Group B described it as good or even excellent (M=4.167; SD=1.17). As shown in Figure 5.6 concerning the statement *"I quickly learned how to interact with the chatbot."* this aspect was also rated better for chatbot B. Group A had an average score of 3.50 (SD=1.05), whereas Group B reached an average score of 4.67 (SD=0.52). Additionally, most participants of Group

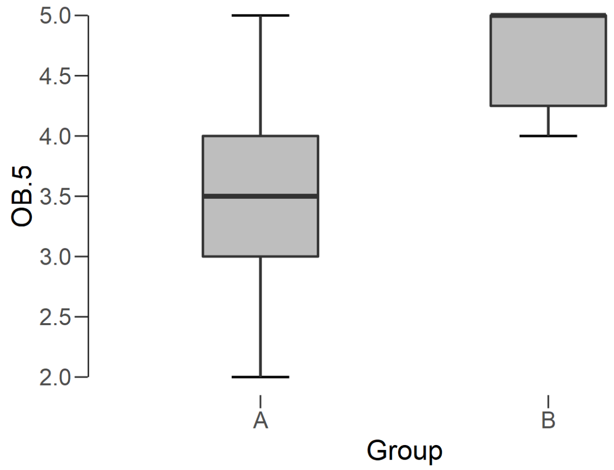

Figure 5.6.: Boxplot of statement 5 of the Onboarding (OB) evaluation: assessment of the usage of the chatbot was quickly learned

B stated that the communication was simple (M=4.33; SD=1.21) and no help of another person was needed (M=5.00; SD=0.00). For the questioned person of Group B the chatbot was okay or easy to use (M=3.50; SD=0.837) and most of them also did not need additional help by another person (M=4.33; SD=0.82). All consulted persons of

Group B mentioned that they had no problems to get the required information (M=4.83; SD=0.41), whereas this was not so easy for participants of Group A (M=3.17; SD=0.75).

- **Content & Design**
  The length of the chatbot answers was the same for both chatbots. Participants of Group B considered the length of the messages as okay (M=3.83; SD=1.60), whereas Group A was satisfied with their length (M=4.17; SD=0.75). The introduction message of chatbot B (M=4.17; SD=0.75) is preferred over the initial message of chatbot A (M=3.83; SD=0.41). Only one participant of Group B had missed some functionality, whereas this was true for 5 out of 6 persons of Group A. All participants of both groups had trust in the results. Also, the language of the chatbot was perceived as appropriate and the answers were clear and understandable for all of them.

- **Interest**
  Participants of both groups have been interested in the functionality provided by the chatbot (Group A: M=4.00; SD= 0.89 — Group B: M=3.83; SD=1.6). 4 out of 6 participants of Group B and 3 participants of Group A consider using the chatbot in future to get event-specific information. Nevertheless, only half of each group consider using chatbots in general to get information in the future.

5.3. Discussion

Overall the results of the evaluation confirm the positive influence of suitable onboarding and engagement strategies on the user experience. Participants using chatbot B had less issues with the human-computer interaction and most of them experienced the fulfillment of the tasks as simple. Also the time needed to get the desired information (especially for new users) can be reduced with suitable onboarding strategies and a simplification of the whole conversation (by providing buttons and proposing questions). During the pre-questionnaire, most of the participants who did not have used a chatbot until the experiment have not come across them or were not aware of them while surfing the internet. The usage of the "popup" effect to make the user aware of the chatbot on the page can be registered as a successful approach, as 5 out of the 7 participants that used chatbots already noticed the chatbot due to this. According to the results, females are more likely to interact with provided chatbots. To get the user engaged into using the system sufficient help is crucial. When users do not know how to initiate the conversations with the chatbot they start to look for the information somewhere else. Even though most of the participants did not remember the initial message within the post-questionnaire and had to look it up again, the message had an impact on the user experience. Especially including the help function as a button within the initial message of chatbot B supported the user within the first few minutes of interaction. Even though the length of the messages was rated to be okay, some participant stated that they would have preferred less information, especially concerning the description of the performance. A participant also remarked that as on the website already are a lot of images and headlines the chatbot did not stick out when looking on the page. The experiment also confirmed the importance of sufficient training of the language model to cover the broad variety of different phrases that can be used to ask for the same information. Therefore it is important to collect possible user input from as much test users as possible and continue the training of the language model even after publishing the chatbot. Additionally, if a question is within the scope of the chatbot but the exact information is not available in any data source consulted by the chatbot a suitable message must be delivered to the user. As the experiment shows, if the "error" message is not specific enough the user might rephrase the same matter several times and gets annoyed as no valid response is delivered. Based on the evaluated results it can be said that onboarding affects user engagement. The user engagement increased for all evaluated areas: aesthetic appeal, focused attention, novelty, perceived usability, felt involvement and endurability. Interesting is, that onboarding also influences the perception of the chatbot concerning attractiveness, aesthetically appeal and appeal of visual senses. By applying onboarding strategies the confusion level and frustration level of the user was reduced, and the user got less annoyed. Nevertheless, even though after the experiment eight out of the twelve participants are more likely to start using a chatbot for event-specific information in the future (compared to their answers before the experiment), many of them are still sceptical.

6. Conclusion

In this thesis, a broad variety of different onboarding and user engagement strategies are described. Many of these elements and mechanisms have been used for the development of an information-gathering chatbot for theatres and the opera. A first evaluation proved that the applied strategies have improved the user experience and have simplified the first interaction with the chatbot for new users but also more experienced ones. During the research and development process of the thesis, problems arose and mistakes have been made. The lessons learned from these experiences are listed in the section below.

6.1. Lessons Learned

During the thesis, several important aspects and considerations have evolved that are listed below to help future researches to avoid the mistakes made.

Scope-Awareness

Due to the wide field of operas, theatre plays, concerts and other kinds of events that are covered by the chatbot the user can easily get confused about what questions the chatbot can and cannot answer. To minimize the frustration level due to out-of-scope requests and to help the user to get the conversation started it is reasonable to inform the user about the scope of the chatbot at the beginning of the conversation. Therefore the user knows which questions the bot can answer. A predefined scope (in contrast to a general-purpose scope) further simplifies the complexity of the language understanding requirements of the chatbot. As chatbots in most cases have to deal with specific problems of a certain domain the chatbot doesn't need to be able to handle requests concerning unrelated topics.

Interactivity

Interactivity is an integral part to establish user engagement within the chatbot. Systems that only answer to user requests without any kind of self-initiating messages do not motivate the user to interact with the system. Nevertheless, if like for this chatbot no predefined conversation structure can be established this might be a challenge. During the design of the chatbot, it is therefore important to evaluate which additional information a user might be interested in. ”Call-for-action” elements can then be used to encourage the user to continue the conversation.

Language Support

When selecting a language understanding engine (like in this thesis the Language Understanding Service of Microsoft) it is important to check beforehand how well the required language is supported. In most language understanding engines the quality of the language understanding results for the English language is very good, but for other languages the outcome of the provided extraction results should be verified beforehand. Even though LUIS provided good language understanding capabilities of the German language and Microsoft continuously has improved them during the thesis, the missing support of some fundamentals lead to problems during the development. For example date and time entities have been incorrectly categorized for several months within the development.

Extensive Training of the Language Model

In addition to the lesson learned concerning the language support of the NLU engine it has also been mentioned, that extensive training of the language model is crucial to get good predictions of the detected intent and entities. Already within a small scope, like within the evaluation process, the wording and interaction style differs a lot among the users. Here it became obvious that certain words and phrases were not used to train the language model, which resulted in bad responses. Therefore, even if the chatbot was already released, the language model has to be further trained to improve the results.

Suitable Channel Selection

As discovered within the focus-group analysis, a suitable channel selection for the developed chatbot is crucial. It has to be examined not only which channels are used by users of the target group, but also if the information provided within a certain channel is considered as trustworthy. Another aspect to consider is the usability of the channel. Persons that are not used to interacting with computers regularly might prefer speech-based over text-based channels, as this avoids time-consuming typing for this unpracticed users.

Involve Target Users

Even though the awareness of chatbot technology has increased in recent years, most people have not or just rarely get in touch with this technology. Therefore it is important to include users of the target group within the development process. They can help to understand the problems users are faced when using the chatbot for the first time, as well as develop strategies to overcome these issues. The more diversity there is within these test users, the more possible problems might be detected. An additional advantage is, that the requests of the test users can be used to further train the language model and improve the language understanding capabilities of the chatbot.

Strategies and Context

A broad variety of different onboarding and engagement strategies have been identified and described in this thesis. Nevertheless, by questioning the interviewees of the focus-groups, as well as the test users during the development process, it became clear, that not all of the strategies are suitable for every business case. For example, areas that are perceived as formal and serious should not try to onboard the users by making use of jokes or Gamification elements, as this might lead to less trust in the answers of the bot. As wrongly applied strategies might have an adverse effect, the selection of the strategies has to be considered carefully.

6.2. Limitations & Future Work

The first step of the chatbot development included a basic application that can help the user to find event- and performance-related information as well as frequently asked questions more easily. During the testing phases of the development process and the evaluation of the latest version of the chatbot further improvements have been suggested by the participants:

- Further reduce the length of the messages
- Recommend events (especially if no event was found for the given filter)
- Identify a way to make the chatbot stick out when looking at the page with a lot of images and content

Besides these aspects, it is important to raise the awareness of chatbots in general, to establish the technology in the day-to-day life of the people. Even if the chatbot manages to get the attention of the user in the first place it is still hard to overcome the low user acceptance and encourage the user to interact with the chatbot. The onboarding strategies described in this thesis can help some users to overcome the resistance of some users but the technology has to establish so that more people make use of it. Even though a broad variety of possible onboarding strategies are identified and described within this thesis, not all of the elements and mechanisms are perceived as suitable for every business case. As mentioned within the result evaluation of the focus-group analysis, strategies like making the chatbot funny, Gamification or "Easter Eggs" are not perceived as suitable within the field of operas and theatres. Therefore, only strategies suitable for the given field have been evaluated. The impact of other mentioned elements and mechanisms to improve the onboarding process and user engagement have to be evaluated within a future work.

6.3. Summary and Outlook

Chatbots, also known as virtual assistants, digital assistants or language interfaces, are no new concept. The first chatbot was already developed decades ago and even though nowadays there are already thousands of text- and speech-based chatbots available all over the world, the technology still suffers from high drop-off rates and low user acceptance. The high complexity of language understanding and generating were underestimated for a long time. When the hype around chatbots started in 2016, many users were disappointed by their performances. Nevertheless, they are a good way to overcome issues of traditional websites, like information that is hard to find and websites that are very complex to navigate. Chatbots can simplify the human-computer interaction, but the user first has to accept this "new" technology. Even though chatbots for example within the field of customer support have already managed to increase the user acceptance, many people still prefer talking to a real human. To encourage the user to make use of chatbots onboarding strategies for chatbots are therefore crucial. Additionally, the user has to be encouraged to use the system by the applied user engagement elements and mechanisms. Even though strategies used for traditional website and mobile applications are not always applicable to chatbots, many of them can be adapted or used as a base to develop chatbot-specific concepts. In this thesis, several possible approaches for onboarding as well as for user engagement, are outlined. This ranges from suitable help messages to buttons and other "call-for-action" elements as well as appropriate responses. In this work, a chatbot was developed by using Microsoft Bot Framework in conjunction with Microsoft LUIS as language understanding engine and the QnA Maker service as question-and-answer catalogue. The developed chatbot involves many of the outlined onboarding and engagement strategies described in this thesis. Within a conducted evaluation it was possible to show that a chatbot developed to involve onboarding and engagement strategies tends to reach better user engagement and a significant improvement concerning the simplification of the information gathering process (compared to chatbot without these mechanisms). Additionally, the evaluation indicates that the perceived appearance was influenced by the applied strategies, too and the overall usability increased. To measure the user engagement the 31 items of the User Engagement Scale (UES) were used. Additionally, 16 questions have been provided to identify the influence of onboarding strategies on user experience. Even though it was possible to show that well-considered onboarding and engagement elements improve the overall user experience, another outcome of the survey was that many users are still sceptical concerning the usage of chatbots. As chatbots are still not common, even within large companies' websites, many participants stated that they did not get in touch with this technology before this experiment. Nevertheless, as only a small number of participants was consulted, further investigations are needed to validate statistical significance. Additionally, as not all described elements and mechanisms to improve onboarding and user engagement have been considered as suitable for the given field of operas and theatres, left out strategies have to be evaluated in future work. To increase the user acceptance of chatbots, the technology has to be more present on websites and they have to implement strategies to encourage the user to try out and continue using the system. In the survey, it was possible to show that most of the participants are more likely to use the developed chatbot (or a chatbot on another website) than before the experiment. The more possibilities the user has to get in touch with the technology the easier it is to convince him or her to use a chatbot.

Appendix

Appendix A.

Focus-Group Interview: Questionnaire

2019 Chatbot Onboarding Strategies:

Alter:
Geschlecht:

Veranstaltungen:

Welche Art von Veranstaltung besuchen Sie gerne?

Wo kaufen Sie Tickets für solche Veranstaltungen?

Wie finden Sie Tickets auf der Webseite?

Nach welchen Kriterien würden Sie suchen/filtern?

Chatbot

Verwenden Sie Chatbot Systeme?

Warum / warum nicht?

Wie würden Sie gerne von einem Chatbot begrüßt werden?

Welche Frage würden Sie als erstes stellen?

Wie soll der Chatbot aussehen?

Welche Informationen wünschen Sie sich vom Chatbot zu bekommen?

Würden Sie einen Chatbot einer klassischen Webseite vorziehen?

Wie wahrscheinlich würdest du einen Chatbot verwenden, um Veranstaltungstickets zu kaufen? (Skala 1-5: 1.. sehr unwahrscheinlich – 5 .. sehr wahrscheinlich)

Appendix B.

Evaluation: Pre-Questionnaire

2019 – Chatbot Evaluation - Prequestionnaire

Info:

Alter:
Geschlecht:
Identifikationscode[1]:
Gruppe:

Erfahrungen:

Hast du bereits einen Chatbot benutzt?

Falls ja:

- Wie sind Sie auf den Chatbot aufmerksam geworden?
- Wie zufrieden waren Sie mit den Ergebnissen? (Scala 1 – 5; 1 sehr unzufrieden – 5 sehr zufrieden)
- Wie zufrieden waren Sie mit der Bedienbarkeit (Scala 1 – 5; 1 sehr unzufrieden – 5 sehr zufrieden)

Falls nein, warum nicht?

Wie wahrscheinlich würden Sie einen Chatbot verwenden, um veranstaltungsspezifische Fragen abzuklären? (Scala 1-5; 1.. sehr unwahrscheinlich – 5 sehr wahrscheinlich)

Erwartungen:

Welche Erwartungen haben Sie an den Chatbot (Funktionalität, Sprache, Aussehen)?

Auf welche Funktionalität können Sie nicht verzichten?

Auf welchen Kanälen würden Sie den Chatbot benutzen?

- Webseite:
- Facebook:
- Amazon Alexa:
- Sonstige:

[1] Bitte denken Sie sich eine beliebige Zahlen- und Buchstabenkombination (Länge 5 Zeichen) zur Identifikation aus und verwenden Sie diesen Code auch zur Identifizierung bei der Nachevaluierung (2. Fragebogen)

Appendix C.

Evaluation: Task Description

2019 – Chatbot Evaluation - Tasks

Aufgabenstellung:

1. Suchen Sie eine Veranstaltung im Oktober, die Sie interessiert.
2. Versuchen Sie zusätzliche Informationen zu der Veranstaltung herauszufinden (z.B. Inhalt, Besetzung, Dauer, etc.)
3. Versuchen Sie eine Karte für diese Vorstellung zu kaufen (HINWEIS: Die Karte muss nicht tatsächlich gekauft werden. Die Aufgabe endet beim Klick auf den Link „Karte kaufen“. Hier wird noch kein tatsächliches Ticket bestellt.)

Appendix D.

Evaluation: Post-Questionnaire

2019 – Chatbot Evaluation – Postquestionnaire

Identifikationscode[2]:

		Trifft nicht zu	Trifft eher nicht zu	Teils-Teils	Trifft eher zu	Trifft zu
FA.1	Ich habe mich in der Arbeit mit dem Chatbot verloren.					
FA.2	Ich war so auf das Experiment fokussiert, dass ich die Zeit vergessen habe.					
FA.3	Während der Benutzung des Chatbots habe ich die Dinge um mich herum ausgeblendet.					
FA.4	Während der Benutzung des Chatbots habe ich die Welt um mich herum vergessen.					
FA.5	Währen der Benutzung des Chatbots verging die Zeit wie im Flug.					
FA.6	Ich bin in diese Aufgabe vertieft.					
FA.7	Während des Experimentes lies ich mich mitreißen.					
PU.1	Die Verwendung des Chatbots hat mich frustriert.					
PU.2	Die Verwendung des Chatbots war für mich verwirrend.					
PU.3	Während der Benutzung des Chatbots fühlte ich mich genervt.					
PU.4	Während der Benutzung des Chatbots fühlte ich mich demotiviert.					
PU.5	Die Verwendung des Chatbots war schwierig.					
PU.6	Diese Erfahrung war anspruchsvoll.					
PU.7	Ich hatte das Gefühl die Kontrolle über die Situation zu haben.					
PU.8	Ich konnte einige Aufgaben mit dem Chatbot <u>nicht</u> erfüllen.					
AE.1	Der Chatbot war ansprechend.					
AE.2	Der Chatbot war schön gestaltet.					
AE.3	Die Bilder bei den Nachrichten des Chatbots haben mir gefallen.					
AE.4	Der Chatbot hat die visuellen Sinne angesprochen.					
AE.5	Der Aufbau vom Chatbot hat mich angesprochen.					
RW.1	Die Verwendung des Chatbots hat sich ausgezahlt.					
RW.2	Ich werte diese Erfahrung als Erfolg.					
RW.3	Die Erfahrung verlief nicht so wie ich mir das gedacht hätte.					
RW.4	Die Erfahrung hat sich gelohnt.					
RW.5	Ich würde den Chatbot meiner Familie und Freunden empfehlen.					
RW.6	Ich habe den Chatbot aus Neugier weiter benutzt.					
RW.7	Der Inhalt des Chatbots hat meine Neugier angeregt.					
RW.8	Die Erfahrung hat mich gefesselt.					
RW.9	Ich fühlte mich in die Aufgabenstellung eingebunden.					
RW.10	Die Erfahrung hat mir Spaß gemacht.					
RW.11	Das Experiment hat mich interessiert.					
OB.1	Der Chatbot hat von Begin an meine Neugier erweckt.					
OB.2	Ich war neugierig, was der Chatbot alles kann.					

[2] Verwenden Sie hier den Code, den Sie auf dem 1. Fragebogen angegeben haben.

OB.3	Die Begrüßung fand ich ansprechen.					
OB.4	Die angebotene Hilfe war ausreichend.					
OB.5	Ich lernte schnell wie der Chatbot verwendet werden kann.					
OB.6	Die Länge der Nachrichten war angenehm.					
OB.7	Ich hätte mir zusätzliche Funktionen gewünscht.					
OB.8	Die Kommunikation war einfach.					
OB.9	Ich konnte den Chatbot ohne Hilfe von anderen benutzen.					
OB.10	Die gesuchten Informationen konnte ich einfach bekommen.					
OB.11	Die Ausdrucksweise des Chatbots war passend für den gegebenen Context.					
OB.12	Ich habe den Antworten des Chatbots vertraut.					
OB.13	Ich würde diesen Chatbot zur in Zukunft verwenden, um Informationen zu Veranstaltungen zu erhalten.					
OB.14	Ich konnte Informationen schneller als auf der Webseite finden.					
OB.15	Die Antworten des Chatbots waren klar verständlich.					
OB.16	Wenn ein Chatbot auf einer Webseite angeboten wird, werde ich ihn in Zukunft verwenden.					

Bibliography

Abdul-Kader, S. A. & Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, *6*(7).

Alpana. (2017, May). Challenges with chatbots — not just technical – chatbots life. https://chatbotslife.com/challenges-with-chatbots-not-just-technical-ecb39612422f. (Accessed on 01/27/2019).

Asimakopoulos, S., Asimakopoulos, G., & Spillers, F. (2017). Motivation and user engagement in fitness tracking: heuristics for mobile healthcare wearables. In *Informatics* (Vol. 4, *1*, p. 5). Multidisciplinary Digital Publishing Institute.

Belmont, V. (2016, March). Bots and gender – chatbots magazine. https://chatbotsmagazine.com/bots-and-gender-4ed9865fe2f2. (Accessed on 04/28/2019).

Berkup, S. B. (2014). Working with generations x and y in generation z period: management of different generations in business life. *Mediterranean Journal of Social Sciences*, *5*(19), 218.

Boiano, S., Bowen, J. P., & Gaia, G. (2012). Usability, design and content issues of mobile apps for cultural heritage promotion: the malta culture guide experience. *arXiv preprint arXiv:1207.3422*.

Botanalytics. (2017, September). How sure improved chatbot onboarding to raise retention & engagement. https://chatbotsmagazine.com/how-sure-improved-chatbot-onboarding-to-raise-retention-engagement-bf8d1f0596cf. (Accessed on 01/27/2019).

Burke, B. (2014). *Gamify - how gamification motivates people to do extraordinary things*. Gartner, Inc.

Ciechanowski, L., Przegalinska, A., Magnuski, M., & Gloor, P. (2019). In the shades of the uncanny valley: an experimental study of human–chatbot interaction. *Future Generation Computer Systems*, *92*, 539–548.

Dale, R. (2016). The return of the chatbots. *Natural Language Engineering*, *22*(5), 811–817.

Debecker, A. (2017, February). 9 rules to maximise chatbot retention. https://blog.ubisend.com/optimise-chatbots/maximise-chatbot-retention?utm_source=medium&utm_medium=alexdebecker-benchmarking-engagement&utm_campaign=e8. (Accessed on 01/27/2019).

Deterding, S., Khaled, R., Nacke, L., & Dixon, D. (2011, January). Gamification: toward a definition. (pp. 12–15).

Dialogflow. (n.d.). *Dialogflow docs*. (Accessed on 02/24/2019).

Drift, SurveyMonkey, Salesforce, & myclever. (2018, January). State of chatbots report 2018. https://www.statista.com/study/52442/the-2018-state-of-chatbots-report/. (Accessed on 01/20/2019).

Epstein, R. (1992). The quest for the thinking computer. *AI magazine*, *13*(2), 81.

Fadhil, A. (2018). Can a chatbot determine my diet?: addressing challenges of chatbot application for meal recommendation. *arXiv preprint arXiv:1802.09100*.

Fadhil, A. & Schiavo, G. (2019). Designing for health chatbots. *arXiv preprint arXiv:1902.09022*.

Fang, H., Cheng, H., Sap, M., Clark, E., Holtzman, A., Choi, Y., . . . Ostendorf, M. (2018). Sounding board: a user-centric and content-driven social chatbot. *arXiv preprint arXiv:1804.10202*.

Følstad, A. & Brandtzæg, P. B. (2017). Chatbots and the new world of hci. *interactions*, *24*(4), 38–42.

Garcia, K. (2018, August). U.s. challenges chatbot usage 2018. https://www.statista.com/statistics/918007/challenges-using-chatbots-according-internet-users/. (Accessed on 01/27/2019).

Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: a research and practice model. *Simulation & gaming*, *33*(4), 441–467.

Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, *10*(1), 1–309.

Grand View Research. (2017, August). Global chatbot market 2025. https://www.statista.com/statistics/656596/worldwide-chatbot-market. (Accessed on 01/27/2019).

Gross, D. & Yu, E. (2001). From non-functional requirements to design through patterns. *Requirements Engineering*, *6*(1), 18–36.

Gurwani, M. S. (2018, October). 4 biggest challenges in chatbot development and how to avoid them. https://insights.daffodilsw.com/blog/4-biggest-challenges-in-chatbot-development-and-how-to-avoid-them. (Accessed on 01/27/2019).

Hill, J., Ford, W. R., & Farreras, I. G. (2015). Real conversations with artificial intelligence: a comparison between human–human online conversations and human–chatbot conversations. *Computers in Human Behavior*, *49*, 245–250.

Hirschberg, J. & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266.

Intento. (2017, November). State of the machine translation by intento. https://www.slideshare.net/KonstantinSavenkov/state-of-the-machine-translation-by-intento-november-2017-81574321?next_slideshow=1. (Accessed on 02/24/2019).

Katkute, K. et al. (2017). *Designing user engagement with text-based chatbots* (Master's thesis).

Kim, Y. H., Kim, D. J., & Wachter, K. (2013). A study of mobile user engagement (moen): engagement motivations, perceived value, satisfaction, and continued engagement intention. *Decision Support Systems*, *56*, 361–370.

Klopfenstein, L. C., Delpriori, S., Malatini, S., & Bogliolo, A. (2017). The rise of bots: a survey of conversational interfaces, patterns, and paradigms. In *Proceedings of the 2017 conference on designing interactive systems* (pp. 555–565). ACM.

Knight, W. (2016). 10 breakthrough technologies 2016: conversational interfaces - mit technology review. https://www.technologyreview.com/s/600766/10-breakthrough-technologies-2016-conversational-interfaces/. (Accessed on 02/01/2019).

Kompella, R. (2018, February). Conversational ai chat-bot — architecture overview – towards data science. https://towardsdatascience.com/architecture-overview-of-a-conversational-ai-chat-bot-4ef3dfefd52e. (Accessed on 01/27/2019).

Lee, D. (2018, August). Why bots fail: importance of usability testing – chatbots life. https://chatbotslife.com/https-bit-ly-2mu3vez-482f41f2b6cc. (Accessed on 01/27/2019).

Lehmann, J., Lalmas, M., Yom-Tov, E., & Dupret, G. (2012). Models of user engagement. In *International conference on user modeling, adaptation, and personalization* (pp. 164–175). Springer.
Lockton, D. (2010). *Design with intent: 101 patterns for influencing behaviour through design*. Equifine.
Luger, E. & Sellen, A. (2016). Like having a really bad pa: the gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 chi conference on human factors in computing systems* (pp. 5286–5297). ACM.
Mayo, J. (2017). Programming the microsoft bot framework: a multiplatform approach to building chatbots. Microsoft Press.
McDonnell, M. & Baxter, D. (2019, April). Chatbots and Gender Stereotyping. *Interacting with Computers*, *31*(2), 116–121. doi:10.1093/iwc/iwz007. eprint: http://oup.prod.sis.lan/iwc/article-pdf/31/2/116/29109106/iwz007.pdf
Microsoft. (2019a). *Azure bot service - bot service — microsoft docs*. (Accessed on 02/24/2019).
Microsoft. (2019b, May). Confidence score - qna maker - azure cognitive services — microsoft docs. https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/concepts/confidence-score. (Accessed on 05/19/2019).
Microsoft. (2019c, March). Entity types - language understanding - azure cognitive services — microsoft docs. https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-concept-entity-types. (Accessed on 04/03/2019).
Microsoft. (2019d, January). Good example utterances - language understanding - azure cognitive services — microsoft docs. https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-concept-utterance. (Accessed on 03/28/2019).
Microsoft. (2019e, February). Plan your app - language understanding - azure cognitive services — microsoft docs. https://docs.microsoft.com/en-gb/azure/cognitive-services/luis/luis-how-plan-your-app. (Accessed on 03/21/2019).
Mohanoor, A. (2018, March). Chatfuel vs dialogflow (api.ai) - mining business data. https://miningbusinessdata.com/chatfuel-vs-dialogflow-api-ai/. (Accessed on 02/02/2019).

Navigli, R. (2018). Natural language understanding: instructions for (present and future) use. In *Ijcai* (pp. 5697–5702).

O'Brien, H. L., Cairns, P., & Hall, M. (2018). A practical approach to measuring user engagement with the refined user engagement scale (ues) and new ues short form. *International Journal of Human-Computer Studies*, *112*, 28–39.

O'Brien, H. L. & Toms, E. G. (2008). What is user engagement? a conceptual framework for defining user engagement with technology. *Journal of the American society for Information Science and Technology*, *59*(6), 938–955.

About Wit.ai – Medium. (n.d.). https://medium.com/wit-ai/about. (Accessed on 04/03/2019).

Gartner Hype Cycle. (2018, December). Retrieved August 18, 2019, from https://www.gartner.com/smarterwithgartner/3-insights-from-gartner-hype-cycle-for-digital-marketing-and-advertising-2018/

Pagani, M. & Mirabello, A. (2011). The influence of personal and social-interactive engagement in social tv web sites. *International Journal of Electronic Commerce*, *16*(2), 41–68.

Paikari, E. & van der Hoek, A. (2018). A framework for understanding chatbots and their future. In *Proceedings of the 11th international workshop on cooperative and human aspects of software engineering* (pp. 13–16). ACM.

Pandita, R., Bucuvalas, S., Bergier, H., Chakarov, A., & Richards, E. (2018). Towards jarvis for software engineering: lessons learned in implementing a natural language chat interface.

Pega. (2017, April). Ai chatbot acceptance by service worldwide 2017 — statistic. https://www.statista.com/statistics/717098/worldwide-customer-chatbot-acceptance-by-industry. (Accessed on 01/27/2019).

Peras, D. (2018). Chatbot evaluation metrics. In *Economic and social development (book of proceedings), 36th international scientific conference on economic and social development–* (p. 89).

Pereira, M. J. [Maria Joao] & Coheur, L. (2013). Just. chat-a platform for processing information to be used in chatbots.

Pereira, M. J. [Maria João], Coheur, L., Fialho, P., & Ribeiro, R. (2016). Chatbots' greetings to human-computer communication. *arXiv preprint arXiv:1609.06479*.

Petersen, F. W., Thomsen, L. E., Mirza-Babaei, P., & Drachen, A. (2017). Evaluating the onboarding phase of free-toplay mobile games: a mixed-method approach. In *Proceedings of the annual symposium on computer-human interaction in play* (pp. 377–388). ACM.

Plafrey, J. & Gasser, U. (2008). *Born digital*. Basic Books.

Portela, M. & Granell-Canut, C. (2017). A new friend in our smartphone?: observing interactions with chatbots in the search of emotional engagement. In *Proceedings of the xviii international conference on human computer interaction* (p. 48). ACM.

Prensky, M. (2001). Digital natives, digital immigrants part 1. *On the horizon*, *9*(5), 1–6.

Rahman, A., Al Mamun, A., & Islam, A. (2017). Programming challenges of chatbot: current and future prospective. In *Humanitarian technology conference (r10-htc), 2017 ieee region 10* (pp. 75–78). IEEE.

Renz, J., Staubitz, T., Pollack, J., & Meinel, C. (2014). Improving the onboarding user experience in moocs. *Proceedings EduLearn*.

Reuband, K.-H. (2018). Das kulturpublikum im städtischen kontext. wie sich das publikum der oper von anderen kulturpublika unterscheidet. In *Oper, publikum und gesellschaft* (pp. 143–191). Springer.

Ryan, R. M. & Deci, E. L. (2000). Intrinsic and extrinsic motivations: classic definitions and new directions. *Contemporary educational psychology*, *25*(1), 54–67.

Sarikaya, R. (2017). The technology behind personal digital assistants: an overview of the system architecture and key components. *IEEE Signal Processing Magazine*, *34*(1), 67–81.

Saygin, A. P., Cicekli, I., & Akman, V. (2000). Turing test: 50 years later. *Minds and machines*, *10*(4), 463–518.

Schumaker, R. P., Ginsburg, M., Chen, H., & Liu, Y. (2007). An evaluation of the chat and knowledge delivery components of a low-level dialog system: the az-alice experiment. *Decision Support Systems*, *42*(4), 2236–2246.

Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., . . . Ke, N. R., et al. (2017). A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Shevat, A. (2017). *Designing bots: creating conversational experiences*. " O'Reilly Media, Inc."

Shum, H.-Y., He, X.-d., & Li, D. (2018). From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, *19*(1), 10–26.

Steele, I. (2018, February). Chatbot journey mapping: how to create a chatbot decision tree. https://www.comm100.com/blog/journey-mapping-chatbot-decision-tree-from-scratch.html. (Accessed on 04/03/2019).

Trazzi, M. & Yampolskiy, R. V. (2018). Building safer agi by introducing artificial stupidity. *arXiv preprint arXiv:1808.03644*.

van Drongelen, M. & Krishnaswamy, A. (2017). *Lean mobile app development*. Packt Publishing. Retrieved from https://books.google.at/books?id=7y1EDQEACAAJ

Wallace, R. (2003). The elements of aiml style. *Alice AI Foundation*.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45.

Werbach, K. & Hunter, D. (2012). *How game thinking can revolutionize your business*. Wharton Digital Press.

Weston, A., Morrison, L., Yardley, L., Van Kleek, M., & Weal, M. (2015). Measurements of engagement in mobile behavioural interventions?

WIEREMA, S. (2014, September). Build your own siri: api.ai offers voice integration for all. https://thenextweb.com/dd/2014/09/16/build-your-own-siri-api-ai-offers-voice-integration-for-all/. (Accessed on 01/26/2019).

Wit.ai. (n.d.). Wit — docs. https://wit.ai/docs. (Accessed on 02/24/2019).

Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A new chatbot for customer service on social media. In *Proceedings of the 2017 chi conference on human factors in computing systems* (pp. 3506–3510). ACM.

Zumstein, D. & Hundertmark, S. (2017). Chatbots–an interactive technology for personalized communication, transactions and services. *IADIS International Journal on WWW/Internet*, *15*(1).