Masaryk University
Faculty of Informatics

# Chatbot for Laundry and Dry Cleaning Service

Master's Thesis

**Bc. Jakub Kříž**

Brno, Spring 2017

Masaryk University
Faculty of Informatics

# Chatbot for Laundry and Dry Cleaning Service

Master's Thesis

**Bc. Jakub Kříž**

Brno, Spring 2017

*This is where a copy of the official signed thesis assignment and a copy of the Statement of an Author is located in the printed version of the document.*

# Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Bc. Jakub Kříž

**Advisor:** Carlos Ruiz Moreno, Ph.D., RNDr. Adam Rambousek, Ph.D.

# Acknowledgement

I would like to thank both my advisors, Carlos Ruiz Moreno, Ph.D. and RNDr. Adam Rambousek, Ph.D., for their valuable advices, great supervision and time they invested in me and the thesis. I would also like to thank Adrián, the CTO of Mr Jeff, because he helped me with setting up the external server, connecting to Mr Jeff's internal system and a lot of other technical tasks. For helping me with evaluations, translations and many other things needed to complete the chatbot, I would like to thank Mr Jeff's employees. Last but not least, I would like to thank my family for their patience and support.

# Abstract

This work presents a chatbot for laundry and dry cleaning service. At first, a chatbot on Facebook Messenger is introduced. This chatbot was used to acquaint interactions with users and to bring new customers. Based on the experience with the first prototype, the In-chatbot, which is believed to increase customer retention and decrease load on call centers, is created. It is integrated into an existing application of Mr Jeff company to offer users a simplified process of creating and changing orders. In addition, users can use the chatbot to contact a customer support from within the chat. To see the chatbot in a wide context, the In-chatbot is then compared to one chatbot from the Czech Republic and one international chatbot.

# Keywords

chatbot, Mr Jeff, In-chatbot, Jeff, laundry and dry-cleaning, Facebook Messenger, Android, Python

# Contents

# 1 Introduction

In the current world of services, we can see a huge shift from what services looked like in the past and what they look like now. The changes are spread from sharing economy companies, such as Uber or Airbnb, to companies like SpaceX with its reusable rockets. Although, this is just a beginning, since we are, for example, very close to use autonomous cars to travel from one place to another on a daily basis.

Because today's world is changing dramatically, the technology behind the services is being changed quite fast as well. According to Gartner [1], the technology trends for 2017 include artificial intelligence and advanced machine learning, virtual and augmented reality, intelligent applications, intelligent things and conversational systems. Virtual personal assistants, like Apple's Siri, Google Now, Microsoft's Cortana or Amazon's Echo, already make everyday tasks easier. In addition to virtual personal assistants, another type of intelligent applications that can enhance user experience and make our lives better are chatbots. [2]

Chatbots can change the way how services are provided. Instead of using websites or installing another new applications, users could simply order a service via a chat interface. For example, when ordering a new laptop from an e-commerce, instead of going through the whole process on the website, a user could simply type: I want to order a new laptop. This is more natural and close to real conversations. What is more, it can also be faster than a regular process in the current user interfaces. Especially, when buttons are used to replace the typing.

My aim in this work is to create a chatbot for Mr Jeff company. The chatbot will be integrated into an existing Android application focused on a full laundry and dry cleaning service. By providing an additional functionality, the chatbot will simplify the current processes and enhance the interaction with users. To achieve this, the chatbot will be using Mr Jeff's internal system to get relevant data.

In Chapter 2, a chatbot is defined and the history of such systems is introduced. Then, Chapter 3 follows with an insight into the Mr Jeff company and its service of laundry and dry cleaning. Chapter 4 shows how the first prototype was designed, implemented and which platform was used. It also includes examples of interactions with users which was important for designing the main chatbot. In Chapter 5, the In-chatbot is described from the design phase through its implementation to the evaluation. Chapter 6 contains a comparison between two successful chatbots and the In-chatbot.

# 2 Chatbots

Many chatbots have been developed so far and they differ in what they do and how they do it. To clarify what a word chatbot refers to in the context of this work, its definition will be provided. Then, the history and different types of chatbots will be introduced.

There are many words refering to chatbots. These words include conversational agents, conversational systems, chatterbots, chat robots or simply bots. [3] These terms are often used as equivalents. For purposes of this work, a chatbot is defined as:

> *A conversational agent based on rules and/or artificial intelligence that simulates a real conversation by using a natural language to communicate with users.* [4] [5]

A conversational agent is:

> *"A self-contained, and concurrently-executing object, possessing internal state and communication capability."* [6]

## 2.1 History of chatbots

To find the ancestor of chatbots, we have to go back to 1960s when Joseph Weizenbaum from the Massachusetts Institute of Technology published the ELIZA program. ELIZA was meant to simulate a psychotherapist based on keywords matching. Thus, the program was later classified as the first rule based system. Despite the fact that the rules were very simple, many users of the program thought they were chatting with a human therapist. Hence, it is said it passed the Turing Test. [7]

ELIZA was then followed by several other systems. One of which was A.L.I.C.E., an abbreviation of the Artificial Linguistic Internet Computer Entity created by Dr. Wallace in 1995. A.L.I.C.E. is described as:

> *"an Artificial Intelligence (AI) natural language chat robot based on the experiment specified by Alan M. Turing in 1950."* [8]

It was developed as a free software that used a pattern matching techniques to have conversations with its users. A.L.I.C.E. as a program was based on the Artificial Intelligence Markup Language (AIML). In addition to AIML, there was a human supervisor, the botmaster, who was needed in the learning process. The program won the Loebner Prize three times at the annual Turing Test contests. [8]

## 2.2 Rule-based Chatbots

Both ELIZA and A.L.I.C.E are chatbots which are based on pattern and/or keywords matching and thus, their functionality is limited to the defined knowledge base. The keywords matching can be defined from very simple rules using just if-then statements to a very sophisticated knowledge base. One of the sophisticated tools is the AIML used by the A.L.I.C.E. chatbot. Together with the AIML containing the rules, a lot of different modules can be defined to check the spelling, parse the input, avoid loops in a dialog and many more. Still, the problem with the keyword matching approach is a lack of understanding of the user's utterance and dependency on a big amount of pattern data. [9] [10]

Between ELIZA and A.L.I.C.E., PARRY was created as an attempt to simulate a paranoid patient. It was innovative in trying to understand a context and topic or in using a dialogue management. [11] Later in 2003, Sofia appeared as a chatbot that was designed to help college students with mathematical problems using mathematical definitions and general knowledge. Its brain was full of plain text files translated by Perl scripts into a understandable form for chatbots. It was communicating with computer algebra systems to solve the problems. By trying to automate the learning process, the chatbot approached chatbots based on an artificial intelligence. [12]

## 2.3 Chatbots Driven by Artificial Intelligence

In contrast to the keywords approach, chatbots with an engine that uses an artificial intelligence are becoming a trend thanks to a big shift in machine learning in the last few years and involvement of big companies like Google, Facebook or Apple. These chatbots are able to learn from each interaction by using neural networks, deep learning or any other approach. Consequently, there is no need to have huge pattern data, because the chatbot just needs to be trained on example inputs. The better the inputs, the better the performance of the chatbot. It is more about quality than quantity. [13]

One of these chatbots was Tay released by Microsoft in 2016. Tay was published on Twitter as *Taytweets (@TayandYou)* and aspired to be a revolutionary human-like chatbot with an ability to carry on conversations with hundred of exchanges in length. However, in contrast to ELIZA, it did not gain trust of its users. Since Tay was designed in a way that it learned from interactions with its Twitter followers and adapted to them, developers could not predict its behavior. This led to tweets with offensive words and images including racial, political, and societal issues. Microsoft took a responsibility for not predicting that users could coax it into a verbal misbehavior, so the project was shut within 24 hours after more than 93,000 tweets. The interesting thing is that Tay was the one considered responsible for its inappropriate statements by many people. [14] [15]

Another chatbots which use artificial intelligence include DoNotPay and the chatbot created for Dell service. DoNotPay is a chatbot accessible from the welcome screen of the `http://www.donotpay.co.uk/` website. Generally built to appeal parking fines, the chatbot now provides answers to more legal questions and it is still free of charge. By having more conversations, the chatbot improves in terms of interactions and legal skills. [16] On the contrary, Datasys has developed a chatbot in order to replace call centers. The chatbot serves as an online customer support for Dell company. [17]

# 3 Company Overview

The chatbot that is created in this work arose during my internship as a requirement from a Spanish company branded Mr Jeff. In this chapter, the company will be introduced together with the service it provides. The source of the information is the official page of Mr Jeff. [18]

Mr Jeff is a company focused on a full laundry and dry cleaning service. It was founded by three entrepreneurs Eloi Gómez, Rubén Muñoz and Adrián Lorenzo in August 2015. They wanted to change the sector and to free their customers of doing the washing. Eloi himself comes from a family of laundresses, so he is more aware of the customer needs. By providing flexible hours and personalized approach, they have come with a solution for everyone.

## 3.1 Laundry and Dry Cleaning Service

As can be seen in Figure 3.1, the laundry and dry cleaning service itself is very simple from the perspective of a customer. A customer just needs to follow these steps:

1. make an order in the application, on the website or by phone,
2. hand clothes to Jeff at the preferred place and time,
3. take the clothes from Jeff at the arranged time and place.

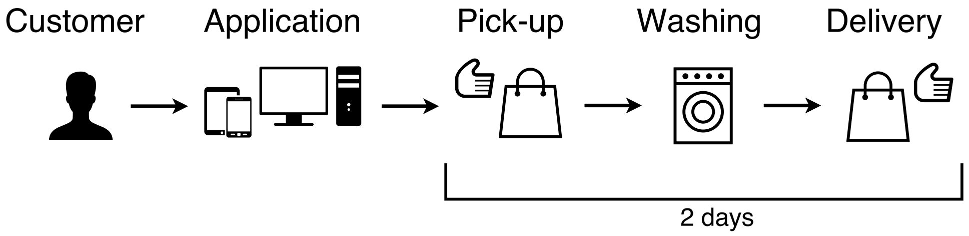


Figure 3.1: Mr Jeff's Service

In Mr Jeff company, Jeff is how delivery boys, who are responsible for clothes pick-up and delivery, are called.

The service is currently available for customers in Madrid, Valencia and Barcelona in Spain and since May also in Mexico City in Mexico. Before making an order, a customer can check whether the service is available in his area by entering a zip code in the application, website or by asking Mr Jeff's customer service via phone.

To choose clothes that will go into an order, a customer can use one of these four options:

- Android application,
- iOS application,
- Mr Jeff's website,
- customer service reachable by phone or WhatsApp.

When using the application, users receive discounts every week. The website on the other hand features a live chat with Mr Jeff's customer service team to answer customers' questions. With both the website or the applications, a customer does not need to register anywhere, the user account is created when the first order is made. In comparison to the previous options, an order made by telephone is the fastest one.

### 3.1.1 Pick-up and Delivery

In terms of the pick-up and delivery of clothes, Jeffs play a significant role. Their task consist in taking customer's clothes to a dry cleaner and returning them in time. Jeffs come with a bag to get the clothes and return with clean, ironed and folded items in a bag of laundry or in a protective cover for dry cleaning items. Mr Jeff guarantees that clothes will be ready withing 48 hours except special garments, but the concrete time of the delivery depends on customers. The pick-up and delivery is done from Monday to Friday in the slots from 14:00 to 16:00 and from 19:00 to 22:00; on Saturdays, the slot is from 10:00 to 12:00. The places of pick-up and delivery can differ as well as times. Customers can modify the pick-up or delivery information according to this scheme:

- until 12:00 for the orders that are delivered or collected from 14:00 to 16:00,
- until 17:00 for the orders that are delivered or collected between 19:00 and 22:00,
- until 00:00 the previous day for the orders that are to be delivered or collected on Saturday.

### 3.1.2 Washing and Dry Cleaning Centers

To provide a quality in washing and dry cleaning, Mr Jeff has contracts with washing and dry cleaning centers with more than 20 years of experience in the sector. These centers are located in the same cities in which the service is available; that is, in Valencia, Madrid and Barcelona.

### 3.1.3 Subscription

For those, who have to do the washing frequently, Mr Jeff offers a monthly subscription.

The subscription includes washed and folded laundry and shirts that are washed, ironed and delivered sheathed on a hanger. The only items excluded are items for dry cleaning, carpets, curtains, furs and duvets. The service consists of washing in water, drying in a dryer and ironing of shirts. The clothes are picked-up the day chosen by a customer and delivered weekly with the next pick-up. Time slots are the same as for regular orders.

To know how many clothes are remaining, a customer can use the application containing the remaining visits, kilograms of laundry and the number of shirts left. In addition, customers are informed by an email on a regular basis, so they know their consumptions.

The subscription is offered in two variants, individual and students. The individual variant offers 20 shirts and 15 kg of laundry for 65 € a month. In contrast to the individual variant, the variant for students costs 45 € and includes 20 kg of laundry and 15 shirts per month.

# 4 Chatfuel Chatbot

Before starting to work on the chatbot, it was required to know the users' behavior, interactions and get an initial evaluation. Therefore, the first prototype was created as a chatbot deployed on Facebook Messenger. It was named simply Chatfuel Chatbot due to the platform used and it was evaluated and enhanced continuously until sufficient information was obtained.

## 4.1 Purpose

The main purpose of the Chatfuel Chatbot was to get new customers by providing users with a basic information about Mr Jeff, its service, products and subscription.

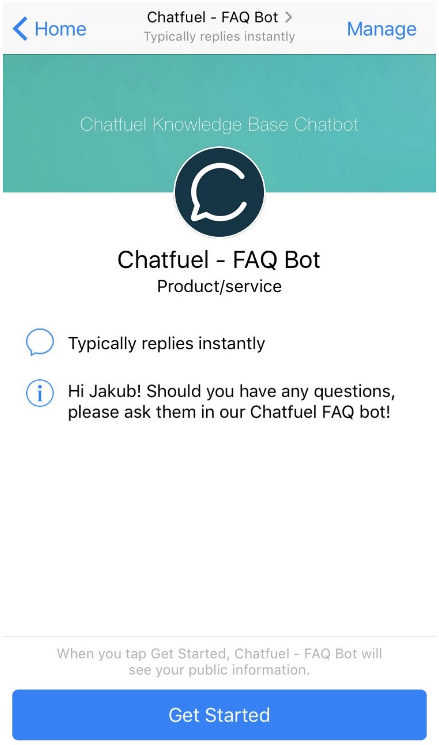
Mr Jeff had a significant number of followers on Facebook and the easiest way to build a chatbot was to use an existing platform and deploy the chatbot on Facebook page of the company. So, users could chat with the chatbot and a required feedback on interactions could be obtained.

## 4.2 Platform

Not to build everything from scratch, the Chatfuel platform was chosen, because it provided an easier interface and more features to build a chatbot than other platforms.

The company started with the goal *"to make bot-building easy for anyone."* [19] They started in 2015 with the Telegram instant messaging service and now they are focused mainly on Facebook Messenger. [19]

To build a chatbot on this platform, no programming skills are required and the platform is free of charge. Furthermore, an artificial intelligence engine is used to enhance the communication with users. The only requirement when creating a chatbot for Facebook Messenger is to have an existing Facebook account, that will be used to login to Chatfuel Dashboard, and to have a Facebook page associated to that account. As a result, many chatbots from various fields have been



Home | Chatfuel - FAQ Bot | Manage

Typically replies instantly

Chatfuel Knowledge Base Chatbot

Chatfuel - FAQ Bot

Product/service

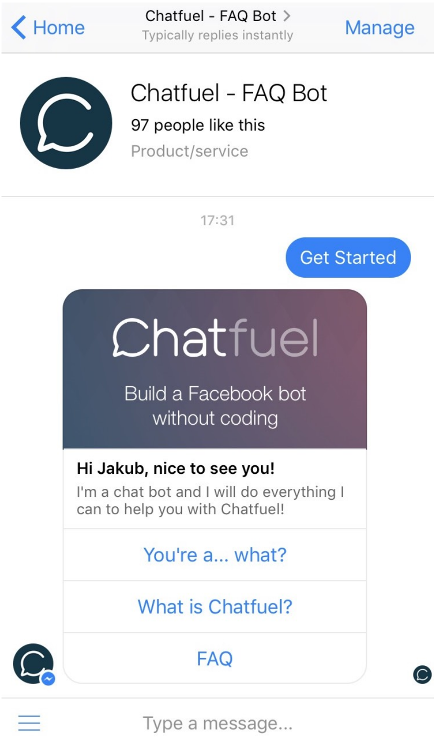
Typically replies instantly

Hi Jakub! Should you have any questions, please ask them in our Chatfuel FAQ bot!

When you tap Get Started, Chatfuel - FAQ Bot will see your public information.

Get Started

Figure 4.1: Get Started Button



Home | Chatfuel - FAQ Bot | Manage

Typically replies instantly

Chatfuel - FAQ Bot

97 people like this

Product/service

17:31

Get Started

Chatfuel

Build a Facebook bot without coding

Hi Jakub, nice to see you!

I'm a chat bot and I will do everything I can to help you with Chatfuel!

You're a... what?

What is Chatfuel?

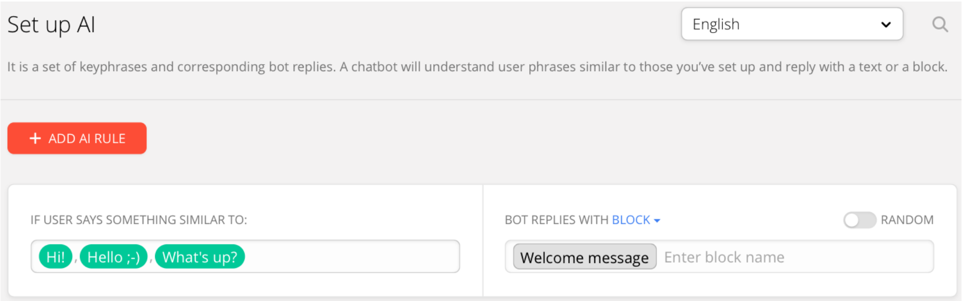
FAQ

Type a message...

Figure 4.2: Welcome Message Block

developed using the platform. This includes chatbots for Adidas, ABC News, MTV, Uber, TechCrunch and others. [19]

The basic building tool in the platform is a block which consists of one or more message cards that are sent together to a chatbot user. Blocks are linked together using buttons and can be grouped into groups. When the chatbot is created, there are already two blocks «Welcome message» and «Default answer». The welcome block is intended for new users and appears when the chat is started by tapping «Get Started» button (see example in Figure 4.1). An example of an existing welcome block can be seen in Figure 4.2. The welcome block is created with a link to the main menu, but the link can be deleted. The block containing the default answer is used when a user sends a message that is not recognized by the artificial intelligence. It is also possible to start a live chat session with a Facebook page administrator when the message is not recognized. Within a block, there are text



Set up AI

English

It is a set of keyphrases and corresponding bot replies. A chatbot will understand user phrases similar to those you've set up and reply with a text or a block.

IF USER SAYS SOMETHING SIMILAR TO: Hi!, Hello ;-), What's up?

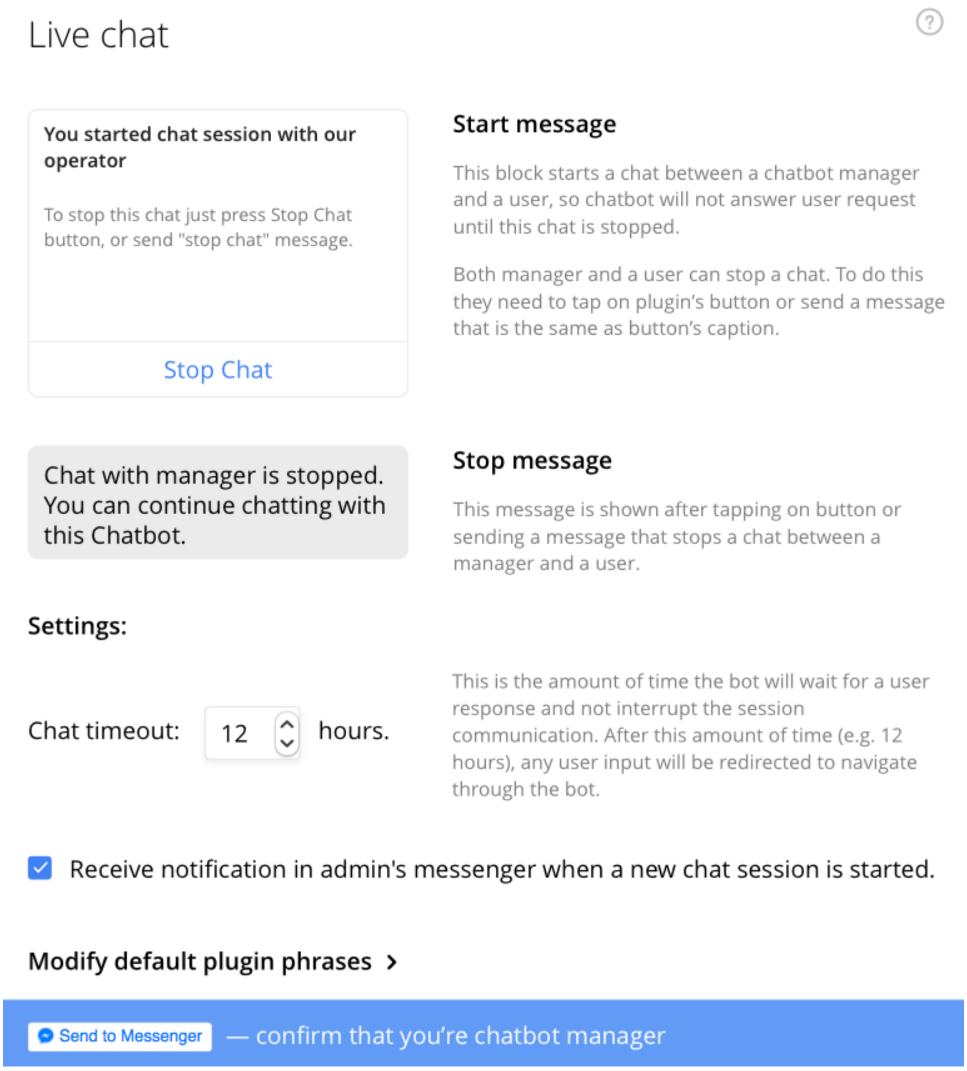
BOT REPLIES WITH BLOCK: Welcome message

Figure 4.3: An example of AI Setup

cards, image cards and gallery cards. A text card can only contain text with a length up to 320 symbols and a maximum of 3 buttons. An image card supports most types of images and also GIFs. A gallery card consists of two required fields: title and one of image, subtitle or buttons. Titles are allowed to have up to 40 symbols, subtitles up to 80 symbols and there can be up to 3 buttons in each card. A card can contain up to 9 items in a slider. A gallery item may consist of an image, title, subtitle, URL and up to 3 buttons. GIF images are not supported in galleries. [20]

Another important part of the platform is the artificial intelligence accessible via AI setup tab in the dashboard, which can be seen in Figure 4.3. Here, developers can set the expected phrases which do not need to be exactly the same as the sentences a user writes. Hence, it saves a lot of effort for developers, because they do not need to write every possible sentence for a specific answer. When the triggering phrases are defined, it is required to assign a block that will be shown as an answer or a text that will be shown immediately. [20]

Beside blocks, cards and the artificial intelligence, there are various plugins available. From Google search to JSON API and Live Chat. JSON plugin can be used, for example, to generate different cards from JSON files which are placed in an external server. Live Chat (see Figure 4.4 for more information), on the other hand, allows users to talk to a human operator instead of the chatbot. The chat is then stopped either by the user or after a specified time. When a user enters the live chat, a push notification together with a link to the conversation is sent to the chatbot manager. [20]



Live chat

You started chat session with our operator

To stop this chat just press Stop Chat button, or send "stop chat" message.

Start message

This block starts a chat between a chatbot manager and a user, so chatbot will not answer user request until this chat is stopped.

Both manager and a user can stop a chat. To do this they need to tap on plugin's button or send a message that is the same as button's caption.

Chat with manager is stopped. You can continue chatting with this Chatbot.

Stop message

This message is shown after tapping on button or sending a message that stops a chat between a manager and a user.

Settings:

Chat timeout: 12 hours.

This is the amount of time the bot will wait for a user response and not interrupt the session communication. After this amount of time (e.g. 12 hours), any user input will be redirected to navigate through the bot.

☑ Receive notification in admin's messenger when a new chat session is started.

— confirm that you're chatbot manager

Figure 4.4: An overview of the Live Chat plugin

Since the first prototype was developed, new features have been introduced including:

- Quick Replies,
- User Variables,
- Chatbot entry points,
- broadcasting history and statistics,
- typing block,
- Chat Room,
- Location plugin.

## 4.3 Analysis and Design

At first, the requirement was to deploy a simple chatbot that would provide basic information about Mr Jeff and its service. Over time, new requirements were added and current were changed, for example, due to the limitations of the platform.

The first diagram showing the basic flow of the first prototype was created with an emphasis on the subscription and can be seen in Figure 4.5. It showed the possible options for customers to follow. When a conversation was started, a customer could order a subscription on Mr Jeff's website, see other available options via menu or order a particular product via the website. The menu offered these options: an introduction of Mr Jeff, listing of categories of products and a live chat with a human operator. This was the main functionality the first prototype offered in its early stage.

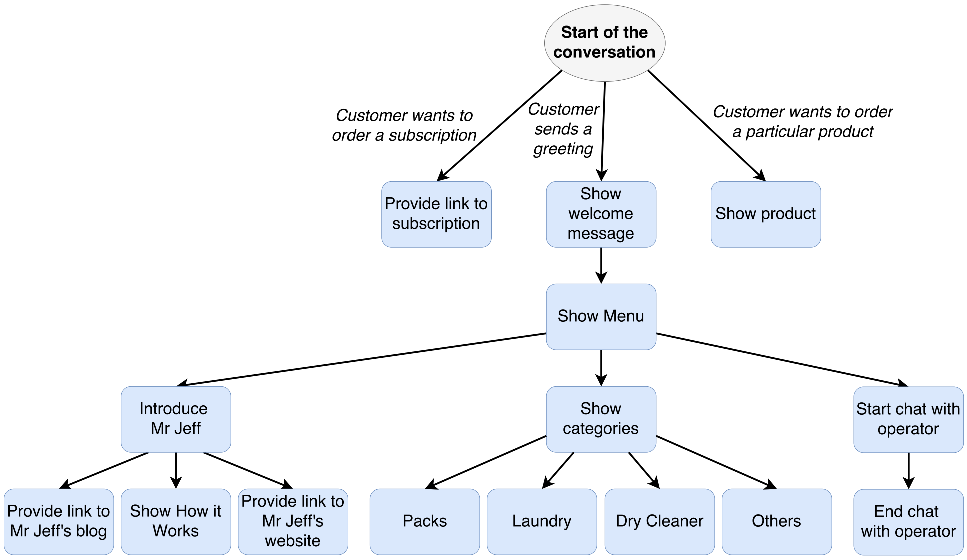

Start of the conversation
Customer wants to order a subscription
Customer sends a greeting
Customer wants to order a particular product
Provide link to subscription
Show welcome message
Show product
Show Menu
Introduce Mr Jeff
Show categories
Start chat with operator
Provide link to Mr Jeff's blog
Show How it Works
Provide link to Mr Jeff's website
Packs
Laundry
Dry Cleaner
Others
End chat with operator

Figure 4.5: The basic flow of the first prototype in its early stage

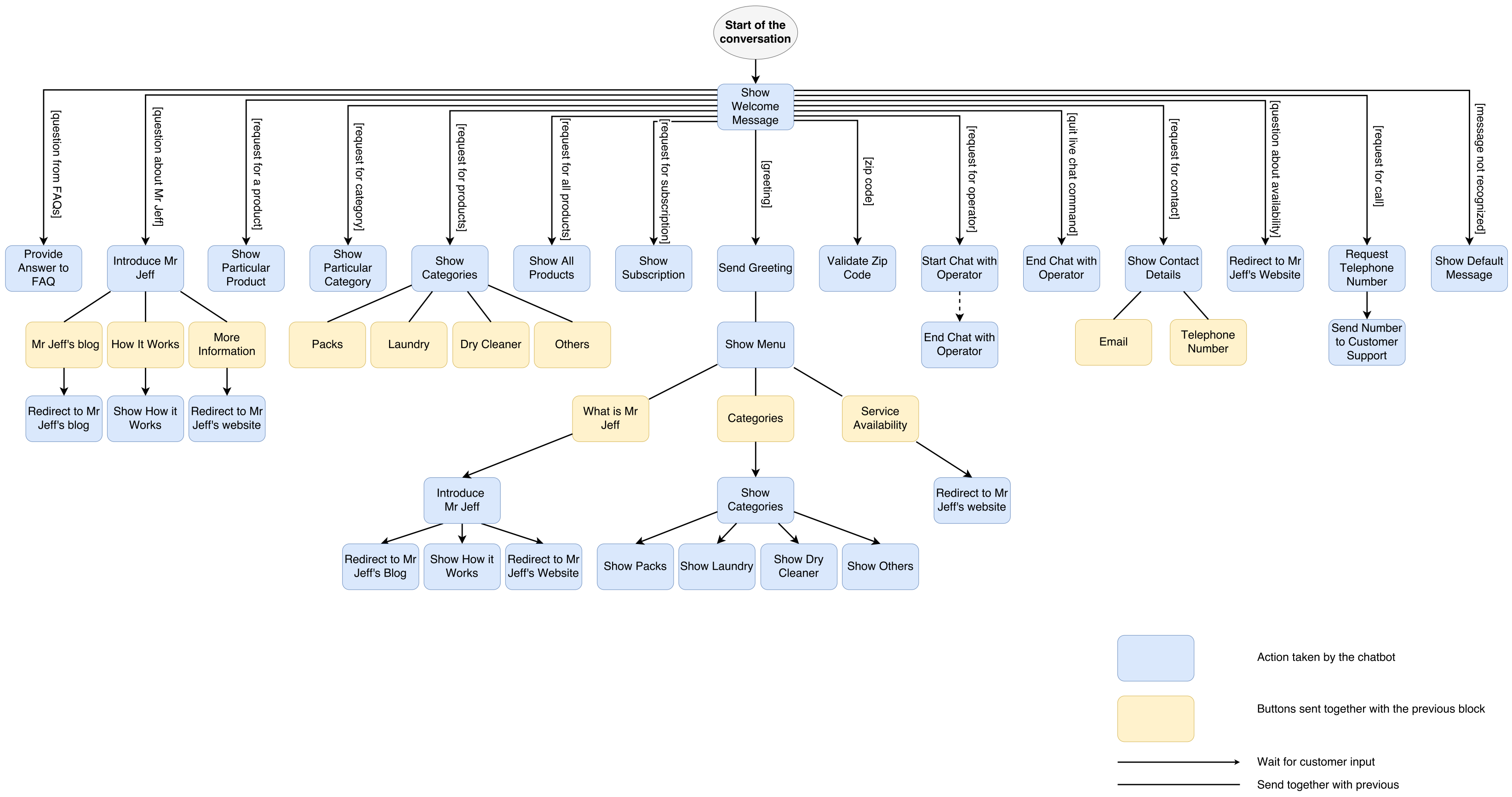


Figure 4.6: The flow of the chatbot conversation

After some iterations, the flow diagram changed in a way it included more options for users to take before the menu was shown. New features were added, such as answering frequently asked questions, providing a contact to Mr Jeff or obtaining user's telephone number to contact him or her later. The overall picture can be seen in Figure 4.6.

## 4.4 Implementation

The prototype was being developed in a testing environment in a way that the chatbot was not connected to Mr Jeff's Facebook page, but it was connected to a temporary page created only for the development purposes. When testing the chatbot, a new conversation was started between the chatbot and the Facebook account associated with the chatbot in the Chatfuel platform.

The key features that were implemented in the first prototype include:

- show products with pictures and price,
- provide information about Mr Jeff with links to its website,
- check service availability with zip code,
- provide a chat with the administrator of Mr Jeff's Facebook page,
- acquire customer's telephone number to contact him in the future,
- give answers to FAQs.

To get information about products together with their images, a JSON plugin provided by the platform and an internal PHP server to store files was used. This was done to ensure an easier future maintenance. The connection to the external server from the platform using JSON files was possible only in one way. This means only HTTP GET requests to the server were made without processing the response from the server. After the request was executed, the gallery was rendered by Chatfuel from the JSON file with a predefined structure. The file could contain one product or a list of products. Hence, files with individual products and files containing whole categories were created. The example JSON file containing the jersey product in Spanish is as follows:

```
[
  {
    "attachment":{
      "type":"template",
      "payload":{
        "template_type":"generic",
        "elements":[
                  {
            "title":"Jersey Estándar",
            "image_url":"http://185.129.248.175/chatfuel/imgs/tintoreria/jersey.jpg",
            "subtitle":"",
            "buttons":[
              {
                "type":"web_url",
                "url":"http://mrjeffapp.com/producto/jersey-estandar/",
                "title":"Pedir por 5,00\u20ac"
              }
            ]
          }
        ]
      }
    }
  }
]
```

In addition to JSON plugin, Live Chat plugin was integrated into the first prototype. This enabled a live chat with a Facebook page administrator within the chat. The live chat was started by writing a sentence similar to the examples defined in the platform, for instance, "start live chat". Then, the messages either sent by the user or by the administrator were not processed by the chatbot until a specific sentence was sent by the user. This sentence, again, should have been similar to the defined examples, so the artificial intelligence behind could assign a correct response block to it.

At first, the chatbot was developed in English, but the only way to provide translations was to create a clone in the platform and change the texts manually. So, the later versions were developed in Spanish. Beside adjusting the texts, the translation was done with the help of the content manager, Elena.

When the chatbot was ready to be deployed, it was connected to Mr Jeff's Facebook page and the followers of the page were informed that this functionality is available.

## 4.5 Evaluation

The Chatfuel Chatbot was being evaluated continuously; however, the required feedback was obtained after the chatbot was deployed on the Facebook page of Mr Jeff.

The first evaluation was made by the marketing department. Afterwards, more people from inside the company were involved. When the chatbot was approved and made public, users who followed the Mr Jeff's Facebook page were informed about the functionality and motivated to try the chatbot. The motivation was a discount voucher which they would receive after communicating with the chatbot for a while. Still, it took some time until there were enough interactions with users, because they were not used to message the Facebook page.

In the meantime, Chatfuel announced a new key feature – analytics. With this tool, valuable information could be acquired, such as:

- daily active users,
- user retention,
- user activity,
- most popular blocks,
- most common phrases.

During the first month, the total number of users stopped at 33. The progress can be seen in Figure 4.7. In addition, Figure 4.8 shows that the biggest user activity was present in the beginning when users were informed via Facebook and motivated with a discount voucher. After that, the activity got more stable at around 2 users interacting with the chatbot a day. The user retention ended on 6 users. In terms of the content, the most popular blocks were Default Answer, Menu and Zip Code. The most popular button, on the other hand, was the Get Started button used to start the conversation with the chatbot. After 5 months, the total number of users increased to 102 and the user retention to 7 users. The most popular blocks were Default Answer, Welcome Message and Zip Code. The most popular user inputs were zip codes that were used to check the availability of the service in an area.

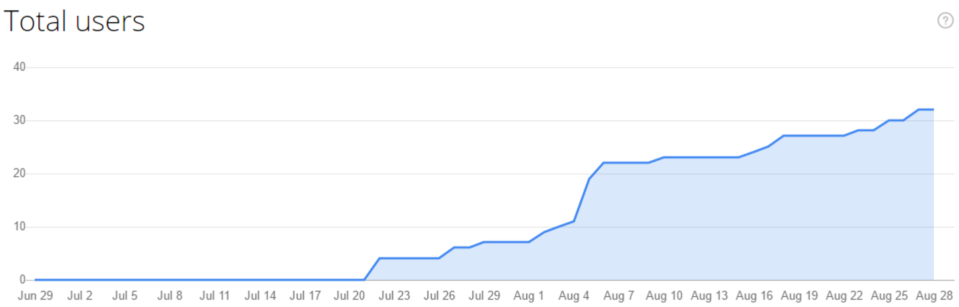


Figure 4.7: Chatfuel analytics: the total number of users

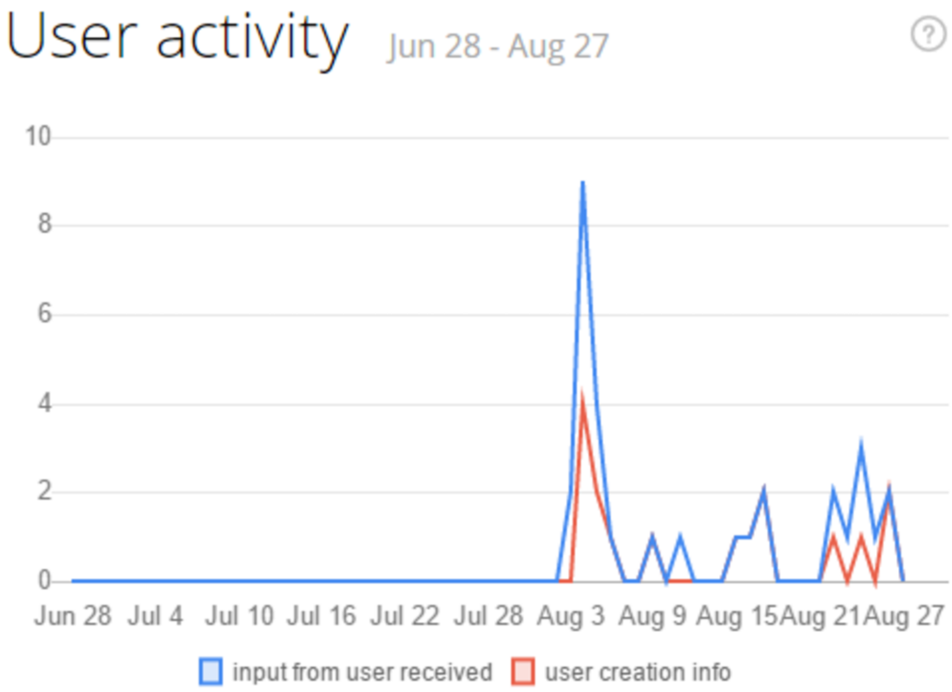
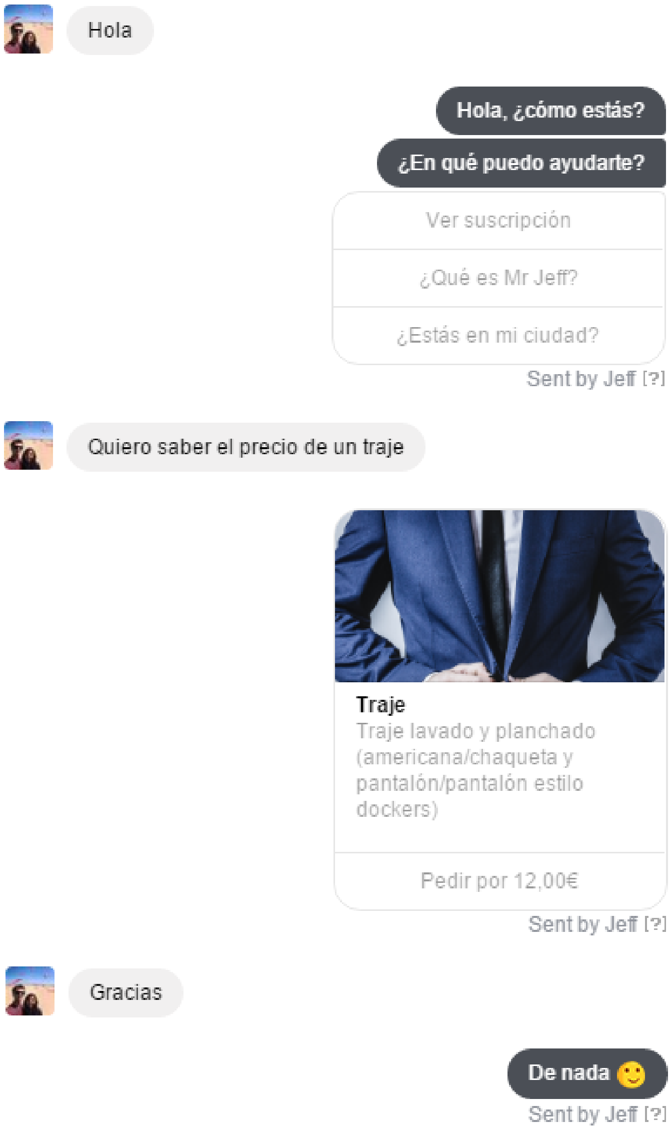


Figure 4.8: Chatfuel analytics: user activity

The interactions with users showed that it was hard to predict the conversation flow when there were many possibilities and users followed paths which were not defined. For example, people often expected human on the other side, because they were not used to chatbots, yet. Moreover, they often started the chat just to get a discount. However, there were some useful interactions acquired. In Figure 4.9, the user asked for a price of a suit and even though the sentence was different from what was defined in the AI Setup, it was correctly recognized and user received information about the product together with a link to Mr Jeff's website. On the contrary, the user in Figure 4.10 asks about hours of pick-up and delivery and it is recognized at the second attempt, even though the word hours was used in both sentences. In both the examples, users seemed satisfied with the responses from the chatbot.

Currently, the Chatfuel Chatbot is disconnected from the Mr Jeff's Facebook page and the administrator of the page is responsible for the communication with customers via Facebook. The reason behind disconnecting the chatbot was mainly the fact that it already met the requirement to provide initial information about interactions and obtain feedback. Furthermore, the chatbot was not able to respond to most of customers' needs over time, because it was not maintained and enhanced further.

To sum up, the chatbot fulfilled the requirement to inform users about Mr Jeff and the service itself. In terms of the interactions, the analytics shows there were not many users interacting with the chatbot. This could be caused by many factors, such as an insufficient promotion of the chatbot, users who were not used to message a Facebook page or users who used different channels. From the conversations, it emerged that the artificial intelligence was not reliable enough and it was hard to ascertain how it worked. In addition, it was important to keep the interactions simple, for example, by using buttons. Furthermore, the limitations of the platform led to a solution with a reduced functionality. However, a valuable feedback was obtained, so the main project could start.



Hola

Hola, ¿cómo estás?

¿En qué puedo ayudarte?

Quiero saber el precio de un traje

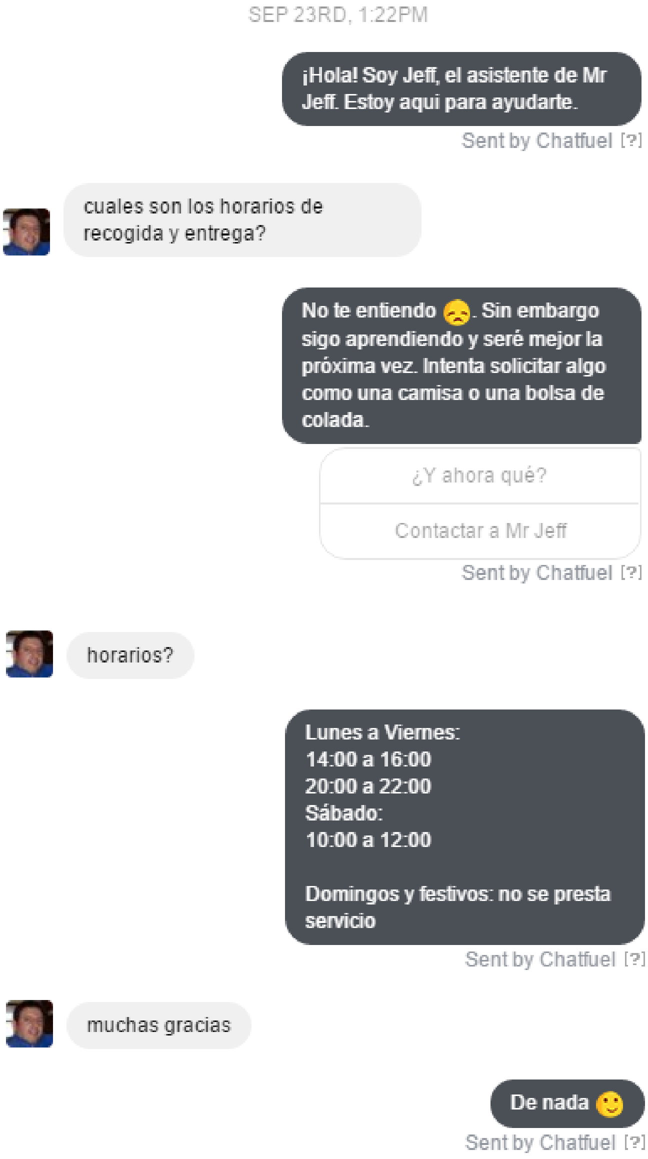
Traje

Traje lavado y planchado (americana/chaqueta y pantalón/pantalón estilo dockers)

Gracias

De nada 🙂

Figure 4.9: An example of the interaction in Spanish



SEP 23RD, 1:22PM

¡Hola! Soy Jeff, el asistente de Mr Jeff. Estoy aqui para ayudarte.

Sent by Chatfuel [?]

cuales son los horarios de recogida y entrega?

No te entiendo 😞. Sin embargo sigo aprendiendo y seré mejor la próxima vez. Intenta solicitar algo como una camisa o una bolsa de colada.

¿Y ahora qué?

Contactar a Mr Jeff

Sent by Chatfuel [?]

horarios?

Lunes a Viernes:
14:00 a 16:00
20:00 a 22:00
Sábado:
10:00 a 12:00

Domingos y festivos: no se presta servicio

Sent by Chatfuel [?]

muchas gracias

De nada 🙂

Sent by Chatfuel [?]

Figure 4.10: An example of the interaction in Spanish

# 5 In-chatbot

Since the interactions and users' behavior was analyzed in the Chatfuel Chatbot, there was a need to design and implement a more sophisticated solution based on that data. Such chatbot was meant to be used in the application for current users and not to attract new customers. It was named the In-chatbot, because many internal and external services were expected to be integrated.

## 5.1 Purpose

The main purpose of the In-chatbot was to increase customer retention and decrease load on the customer support. This was intended to be done by:

- combining multiple communication channels into one,
- providing an added value,
- extending e-commerce and applications functionality,
- guiding customer in the current processes.

## 5.2 Analysis

The In-chatbot was planned to be integrated into different channels, such as Android application or website. In addition, it was obvious that various components of the chatbot would require different external services, not a framework that would include everything together. Due to the complexity, a deep analysis was required.

### 5.2.1 Existing Platforms

Before working on the design of the chatbot, existing frameworks were inspected in order not to reinvent the wheel. In the time the project started, there were already plenty of existing tools that could be useful for building chatbots. Tools varied from messaging and

| | Sinch | Smooch | Quickblox |
|---|---|---|---|
| Complexity | Medium | Low | High |
| Customization | Average | Hard | Easy |
| Documentation | Sufficient | Good | Poor |
| SDK | Android, iOS and Javascript | Android, iOS and Javascript | Android, iOS, Javascript, Windows platform and Blackberry |
| API | – | REST | REST |
| Free Account | 2500 active users per month | 20000 active users per month | 2500 messages per month |
| Additional Features | VoIP calling, SMS and voice verification, SIP Integration, SMS, Video calling | Users management, Authentication, Integrations, Add-ons | Users Management, Video chat, Push notifications |

Table 5.1: A comparison between messaging platforms

conversational platforms through natural language processing engines to bot frameworks. Such tools included Sinch [21], QuickbloxSinch [22], Smooch [23], Messenger Platform [24], wit.ai [25], api.ai [26], LUIS [27], WATSON [28], MS Bot Framework [29], Meya [30] and Howdy Botkit [31].

In the beginning, a messaging platform was needed to create a basic chatbot, so the existing platforms were compared. Because the chatbot was meant to be integrated to existing applications of Mr Jeff, the Messenger Platform was skipped and Sinch, Smooch and Quickblox were compared instead (features available in September 2016).

Given the information in Table 5.1 and personal hands-on experience, Sinch provided the biggest number of various features. Smooch, on the other hand, allowed developers to integrate many well-know external services. Whereas Quickblox offered a complete list of SDKs. In addition, Smooch provided a well documented REST API and SDKs. Sinch was much easier to use when compared to Quickblox, because less lines of code were needed to get it working. Smooch was the easist messaging platform to set up and the platform was also more user friendly in comparison to Quickblox which was a bit hard to get working. In contrast to Sinch, Quickblox provided a REST API for instant messaging, which made it easier to be integrated to various environments. Sinch provided only Android, iOS and Javascript SDK for instant messaging. Smooch provided Android, iOS and Javascript SDKs, REST API and Webhooks. Smooch offered a very limited free usage in comparison to the other two platforms.

## 5.3 Design

Since the beginning, the development of the chatbot was agile meaning that an iterative and incremental approach was followed. Also the prototyping technique was used in the early stage.

At first, the chatbot was designed as a simple application running on the chosen platform where Mr Jeff application already ran. This was intended to test the functionality of the chatbot backend and to display a conversation. The backend, on the contrary, was planned to include the main processing on a separate server. The information about customers, orders and products was meant to be obtained from Mr Jeff's internal server. The same approach is currently used within the applications and the website. Beside the basic functionality, the chatbot was expected to contain many connections to other external services.

### 5.3.1 Components

To see which components would be part of the chatbot, the architecture was discussed and modeled. Then, interactions between components followed.

The architecture uses a client–server model, where clients represent the chat interface and the server is the chatbot itself. From the perspective of layers, the architecture of the In-chatbot includes three main layers: presentation, business and data. As can be seen in Figure 5.1, the business layer contains the core components:

- Messaging Platform,
- API to External Services,
- Analytics,
- Notification,
- Logging,
- Selling,
- Localization,
- Rule-based Engine,
- Driven Conversation Engine.

The presentation layer, on the other hand, contains only UI component. Likewise, the data layer comprises of Products, Orders and Users components.

In contrast to the layers, Figure 5.2 shows a component diagram with a complete list of components. The backend for the In-chatbot communicates with client applications, Mr Jeff Backend and the external Customer Support System. The chatbot backend consists of messaging control, smart engine, management components, localization and internal databases. Particular components, which are already part of the In-chatbot, will be discussed later in Section 5.4.

When the components were defined, it was needed to map them to external services analyzed earlier. Over time, the platforms which were used changed. However, in the current solution, Smooch, WooCommerce [32], Stripe [33] and Zendesk [34] are integrated. Smooch is mapped to the Messaging component, Stripe to the Payment component, Zendesk to the Customer Support component and WooCommerce to the Mr Jeff Backend components.

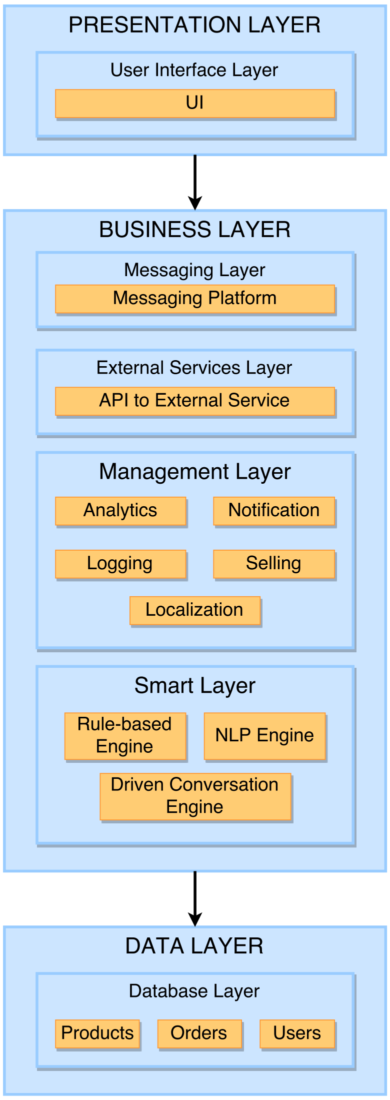


Figure 5.1: A layer diagram showing the In-chatbot architecture

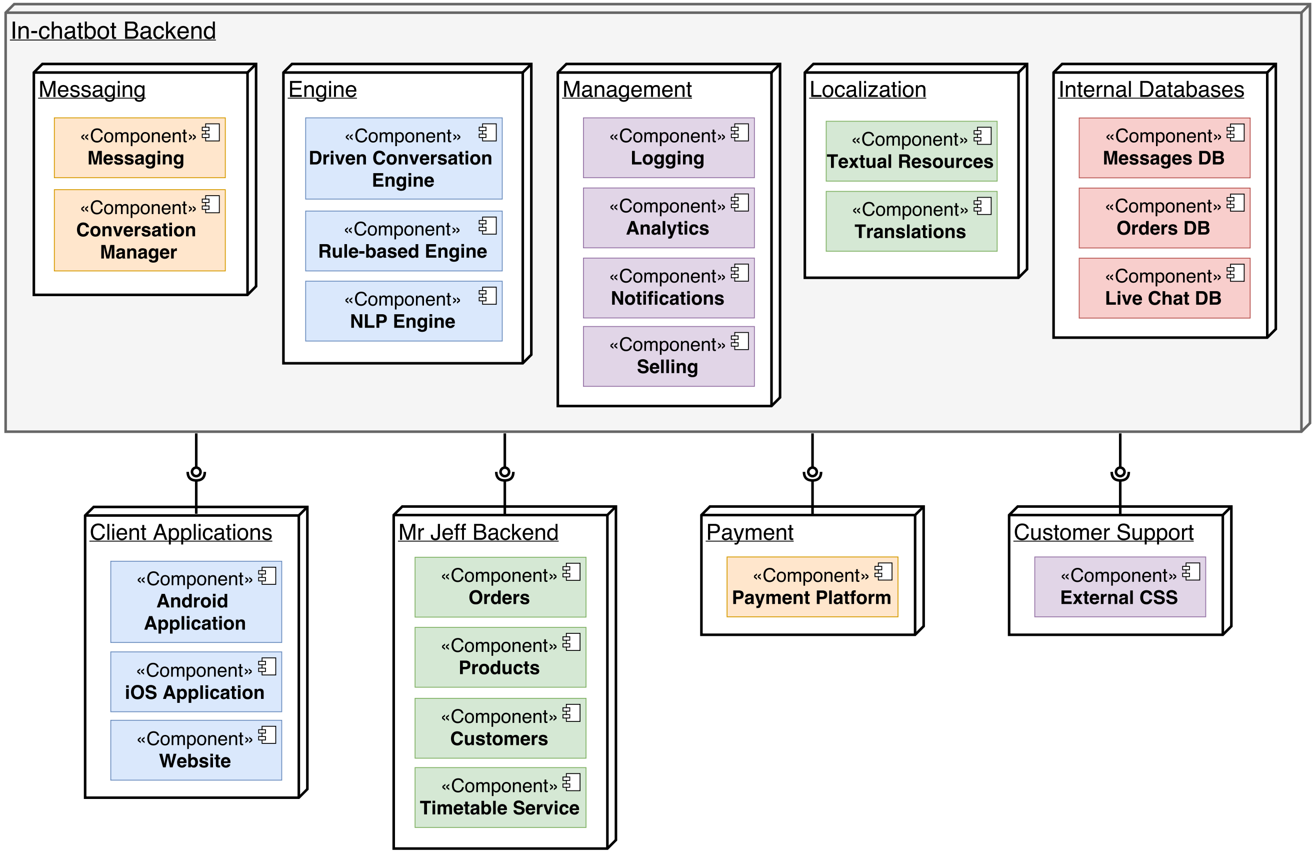


Figure 5.2: A component diagram showing the In-chatbot architecture

Figure 5.3 shows how the particular components communicate with each other in the system. The client application is connected to the backend through the Smooch messaging platform, which also provides a connection to the Stripe payment service. Beside Smooch and Stripe, the backend interacts also with Zendesk and WooCommerce. Within the backend module, the conversation manager is responsible for chat flow including responses to the user from the engine and storing data about interactions into the internal database.

## 5.4 Implementation

### 5.4.1 Prototypes

The first implemented prototype used Quickblox for messaging. Using the Android SDK, a messaging interface was implemented in a simple Android application and the backend script was developed in Python. The Android application worked fine, so every message was shown in the chat. However, it was not easy to set everything up and the backend responded only to the last message in the chat history. The backend was deployed to the Apache HTTP server on AWS (Amazon Web Services) EC2 (Elastic Compute Cloud 2) instance using CGI technology and Flask application server.

Because it was hard to get the chatbot working using Quickblox and it was desired to use an efficient platform, Smooch and Sinch were tried instead. After a sufficient hands-on experience and comparison between these three tools, Smooch was chosen to be used as a messaging platform.

The first acceptable version of the chatbot was built as a separate basic Android application that was later replaced by an integration into the existing Mr Jeff application. The client application used the Smooch Android SDK which provided a chat interface and allowed users to send and receive messages, click on buttons and send pictures. In addition, the backend got more complicated by using a MongoDB database and WooCommerce and Zendesk APIs. The engine, which was used to generate a response from the chatbot, was implemented with a rule-based approach. The features that were provided to users included:

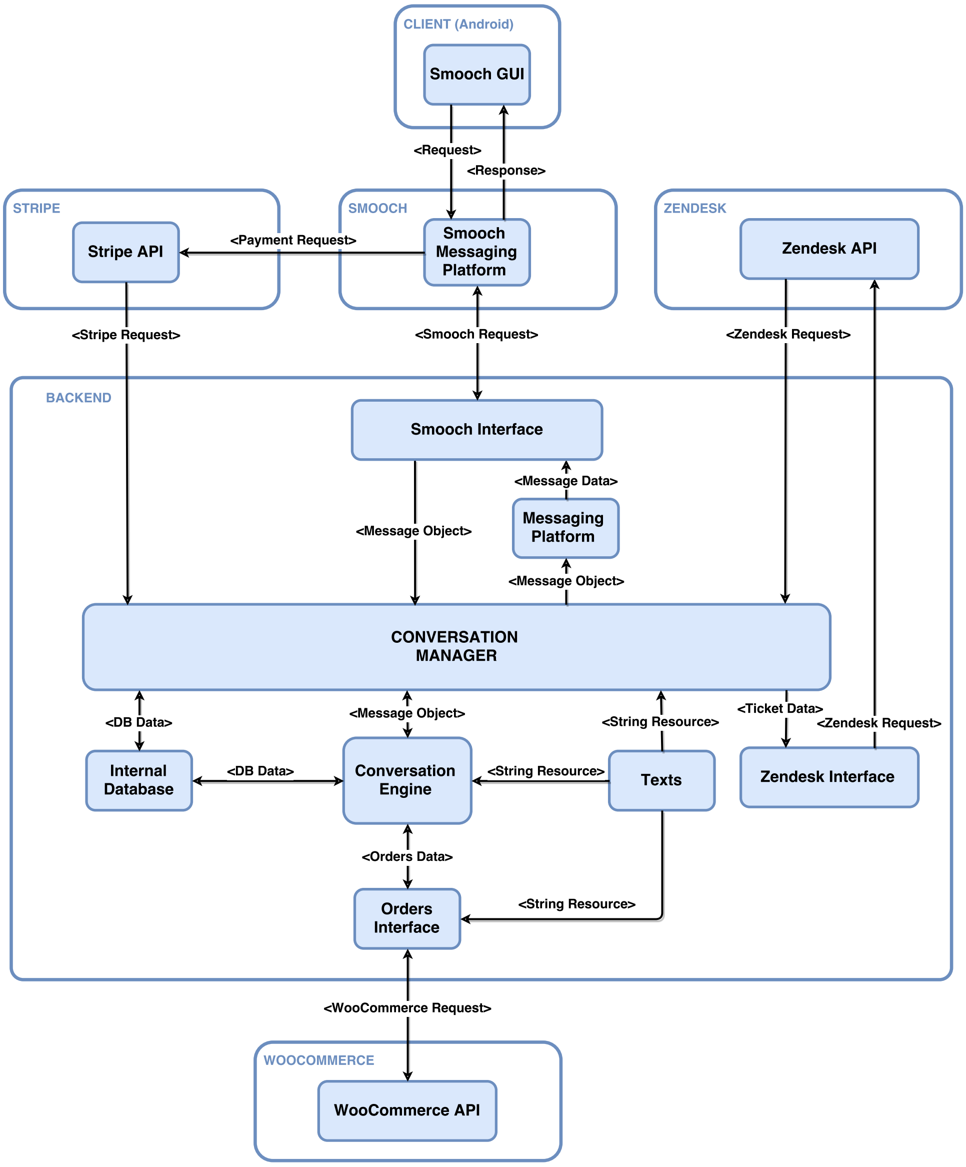


Figure 5.3: Interactions between components

- user login,
- quick reply buttons,
- making new order with a confirmation,
- getting information about the last order (status, products, price),
- visible chat history,
- live chat with customer support,
- sending pictures concerning problems with an order.

The login was connected to the WooCommerce API to obtain information about customers of Mr Jeff. The WooCommerce API was also used for making new orders and getting information about the last order. On the contrary, Zendesk was used to provide a live chat with an operator from the customer support. The functionality more or less copied the application. Users could create a completely new order with chosen products and defined place and time of pick-up and delivery. On the other hand, the order was completed without a payment, because this was not implemented, yet. Products, quantity, dates and addresses were send as buttons or a text and only products and dates were validated. In addition to the application's functionality, the chatbot provided a live chat feature. This was activated either by keywords or by sending a picture of an issue, e.g. dirty clothes. The live chat was an in-chat conversation with the customer support through a Smooch integration. It worked using an YAML file which was later replaced by a database table in MongoDB.

### 5.4.2 Current Status

The idea behind the current version of the chatbot is a minimum viable product. Rather than providing the same functionality as the applications and the website, it offers an added value to its users. Thus, a number of interactions, that are needed before a particular task is completed, is reduced to minimum and the functionality is limited.

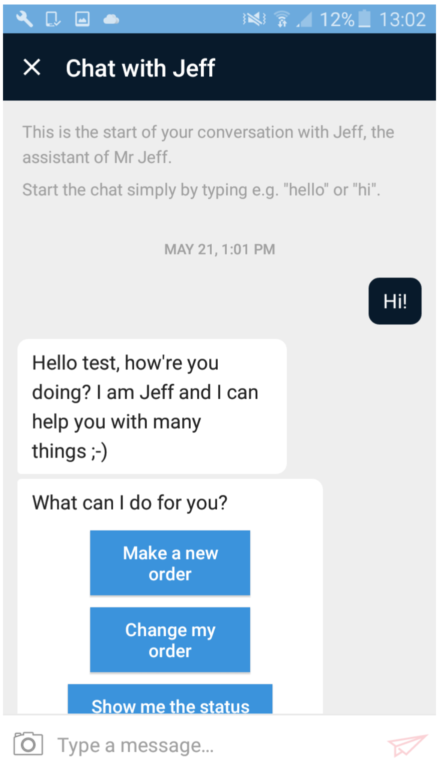
The current version of the application is developed using these technologies:

- Android platform,
- Python programming language,
- Flask application server deployed with CGI,
- Apache HTTP server,
- AWS EC2 instance,
- Amazon Linux,
- MongoDB database.

Based on a deeper knowledge of Android operating system, the Android platform has been chosen. Whereas, Python has been selected because of its simplicity and the available support from Carlos, the advisor. Since Flask in a combination with CGI has seemed to be the easiest option to deploy Python application on a server, it has been used in the work. Amazon AWS has already been used by the company, so the EC2 instance with a default Amazon Linux operating system and a pre-installed Apache HTTP server has been a natural decision. On the contrary, MongoDB has been chosen, because it has worked well with JSON files that have been used with most REST API services.

A client application is implemented as a simple integration into the existing Android application used by Mr Jeff's customers. This integration uses a Smooch SDK to provide a chat interface to users in a form of a complete GUI. The GUI has been customized as much as possible to conform to the look of the Mr Jeff application. The chatbot functionality is only available to the authorized users. To find out whether a current user is authorized or not, the InchatbotAuthorizationTask calls the chatbot API using the HTTP POST method. The chatbot is accessible in menu through the "Talk to us" section. In Figure 5.4, a chatbot GUI can be seen. In contrast, Figure 5.5 shows how the default layout of Mr Jeff's application looks like.

The server part of the chatbot is much more complex in comparison to the client application. For a better development process, three environments are used: development, test and production. These environments are separate folders with different URLs on the server. In



Chat with Jeff

This is the start of your conversation with Jeff, the assistant of Mr Jeff.

Start the chat simply by typing e.g. "hello" or "hi".

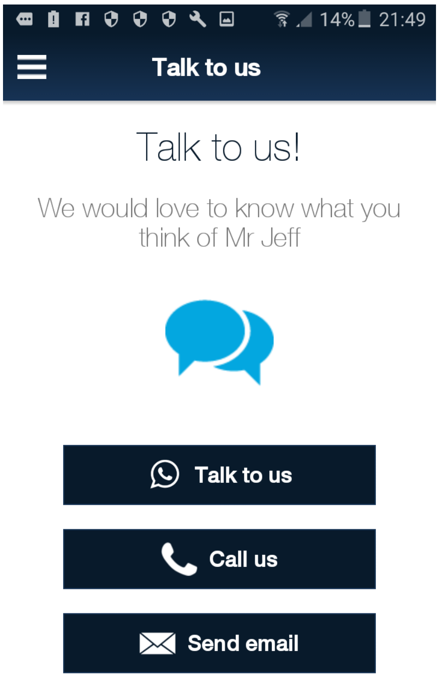
MAY 21, 1:01 PM

Hi!

Hello test, how're you doing? I am Jeff and I can help you with many things ;-)

What can I do for you?

Figure 5.4: In-chatbot Example



Talk to us!

We would love to know what you think of Mr Jeff

Figure 5.5: Default Layout

addition, the environments influence which Smooch, Woocommerce, Zendesk and Stripe account is used in communication with the In-chatbot backend. This information is defined in a configuration file. All three environments include three different routes:

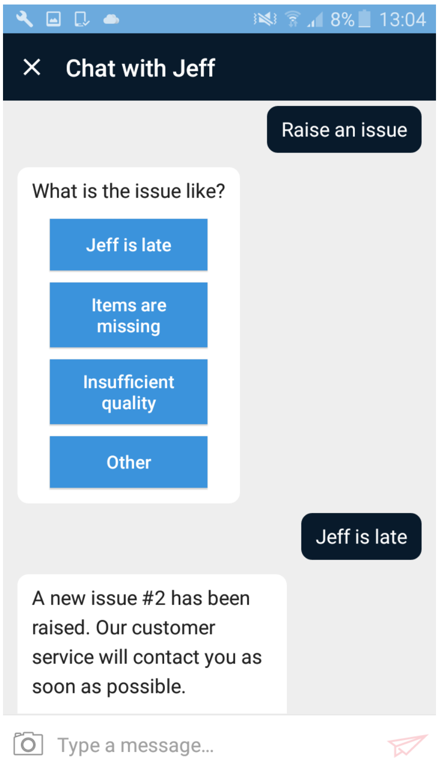
- /api,
- /authorizedusers,
- /authorize.

API route is used for the communication with client applications. The route for authorized users stores users into the database of authorized users. The authorize route decides whether to authorize the current user in the client application to use the chatbot or not. The decision is made based on the contents of the database of authorized users. Internally, the requests coming from Smooch or Stripe are preprocessed, so they fit the uniform structure defined in the backend. There are four types of requests defined:

- text,
- button,
- image,
- payment.

The particular components of the In-chatbot backend conform to what was outlined in the previous section. Conversation Manager is responsible for managing the whole conversation flow by processing every incoming request. At first, a message from a user is stored in Chat Messages database table. Then, a response from the engine is obtained and sent back to the client application. For payments, the flow is simpler, so only a payment confirmation is sent and stored in Chat Messages database table. When the live chat feature is activated, messages are stored in Chat Messages database table and forwarded to the client application. In contrast, Messaging Platform works as a middle layer between the external messaging platform and Conversation Manager. It allows to obtain the whole conversation and send a textual message or a message with buttons. Then, the engine component is divided into Rule-based Engine and Driven Conversation Engine. Rule-based Engine produces responses to textual responses based on keywords. It recognizes greetings and textual commands related to the live chat feature. In addition, it creates a random default message when the keyword is not recognized. Whereas Driven Conversation Engine processes button clicks and returns textual messages with buttons. To obtain and process information, it communicates with Order Interface, Current Orders database table, Live Chat Tickets database table and Zendesk Interface. Internal databases, on the other hand, are used to store interactions or to store temporary data needed across different sessions. For internal databases, MongoDB is used. The databases are named after environments: test_db, development_db and production_db. Chat Messages, Current Orders and Live Chat Tickets are defined as database tables that are called collections in MongoDB. In Chat Messages, all the interactions are stored. Current Orders are used to store temporary data about an order. Likewise, Live Chat Tickets serves as a temporary store of ticket information and live chat activation indicator. Most of the mentioned components need to access textual resources that are handled in Textual Resources component. Currently, only Spanish and English is available. For localization purposes, such as time and date formatting, the Flask Babel Python package is used.

Beside the core functionality, the backend needs to communicate to external systems as well. All the external services are called via a defined interface class that abstracts different API calls. The most important integration is Orders Interface and Orders API that communicates with the Woocommerce backend containing all the orders, products and customers. It uses a uniform representation of data defined in Order, Customer and Product classes. Then, there is Smooch Interface, Smooch API and Smooch Helper. Smooch Interface provides a middle layer between the API and the messaging component. Smooch API is used for API calls and Smooch helper transforms the requests coming from Smooch to the uniform structure further used in the program. To receive payment information and send a confirmation and/or receipt, the Stripe request is processed. A connection with customer support system is implemented as API calls to Zendesk platform. This allows users to report an issue by creating a ticket with



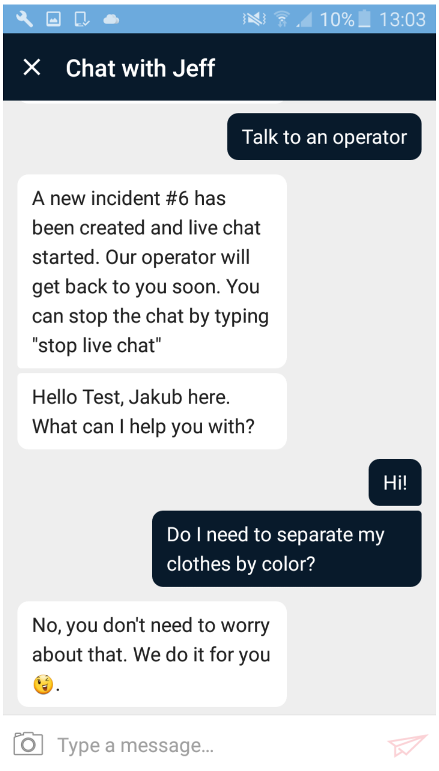
Raise an issue

What is the issue like?

Jeff is late

A new issue #2 has been raised. Our customer service will contact you as soon as possible.

Figure 5.6: Reporting an issue



Talk to an operator

A new incident #6 has been created and live chat started. Our operator will get back to you soon. You can stop the chat by typing "stop live chat"

Hello Test, Jakub here. What can I help you with?

Hi!

Do I need to separate my clothes by color?

No, you don't need to worry about that. We do it for you 😘.

Figure 5.7: Talk to operator

a selected category (Jeff is late, Items are missing, Bad quality, Other) or talk to an operator immediately within the chat. An example of raising an issue can be seen in Figure 5.6. The live chat example can be seen in Figure 5.7. Timetable Service, on the contrary, provides data about available hours and dates. It calls the external PHP script placed on one of Mr Jeff's servers.

The backend also includes an authorization service used to store users, which are authorized to use In-chatbot, into the Authorized Users collection in MongoDB database. Authorization service enables to add users to the authorized group or to find out whether a user is a part of the group or not. This approach allows to incorporate the chatbot functionality into the running application without users noticing. Then, it can be switched on by just adding a user into the group without installing updates.

An important part of the implementation are also unit tests. Some of the tests use Mock library to patch methods that use external services. Currently, there are 63 tests testing 14 classes.

In comparison to the first acceptable prototype, there were some functionality added and some enhanced. For example, the connection to the Zendesk system has been changed in a way that the integration offered by Smooch has been replaced by a direct API call from the backend. This has enabled a fully controlled communication. Beside that, a direct payment using Stripe has been added. The overall functionality has been reduced in order not to copy the application, but to simplify interactions. Consequently, users are only allowed to create an order based on the previous one with an option to change the date and time. When changing the current order, users are, likewise, able to change the time or date of an order. This is all done in a few clicks and thus, it is faster and more comfortable for users. Currently, the chatbot functionality let users to:

- use buttons to communicate with the chatbot,
- create, change or monitor an order,
- pay directly,
- report an issue,
- chat with customer support,
- chat in Spanish or English based on local settings.

## 5.5 Evaluation

All the versions of the chatbot have been evaluated continuously by the advisor Carlos and the marketing team. When the first completed version of the In-chatbot was released, more people from Mr Jeff were involved and the first evaluation begun.

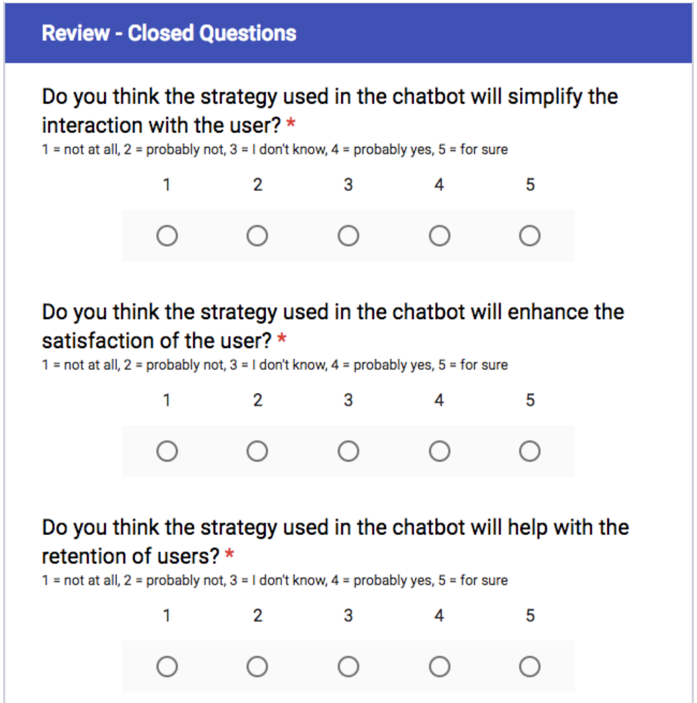
### 5.5.1 First Evaluation

In comparison to the current version, the version of the chatbot included in the first evaluation lacked the functionality of raising an issue and talking to an operator. For evaluation purposes, a questionnaire was created on Google Forms. The participant was at first asked to download the Android application containing the chatbot. Then, the task was to create a new account in the application and send the email of the account to me. Afterwards, a new order was created, because the functionality of the chatbot was based on previous orders. There were also instructions on which credit card to use when paying for an order. The main part of the questionnaire contained open and closed questions. Closed questions can be seen in Figure 5.8 and were used to obtain a feedback on whether the solution helped to simplify interactions with users, enhance users' satisfaction and increase retention of users. On the contrary, Figure 5.9 shows open questions allowing participants to contribute with their opinions on what important parts were needed to be added, what functionality they missed, what did they think about graphical elements and any other suggestions.

There were 3 tester participating on the first evaluation. They answered that the interaction is probably simplified in comparison to the application. Regarding to an enhanced satisfaction, the major answer was I don't know. The opinions on an increased retention of users differed from probably not to probably yes. The participants missed this functionality:

- extension of the conversation options,
- more basic answers,
- redirect the user to support when chatbot does not know the answer,
- other things related to the service, not just with the order.

As improvements, providing a telephone number to customer service and including other options in the menu were mentioned.



Review - Closed Questions

Do you think the strategy used in the chatbot will simplify the interaction with the user? *

1 = not at all, 2 = probably not, 3 = I don't know, 4 = probably yes, 5 = for sure

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

Do you think the strategy used in the chatbot will enhance the satisfaction of the user? *

1 = not at all, 2 = probably not, 3 = I don't know, 4 = probably yes, 5 = for sure

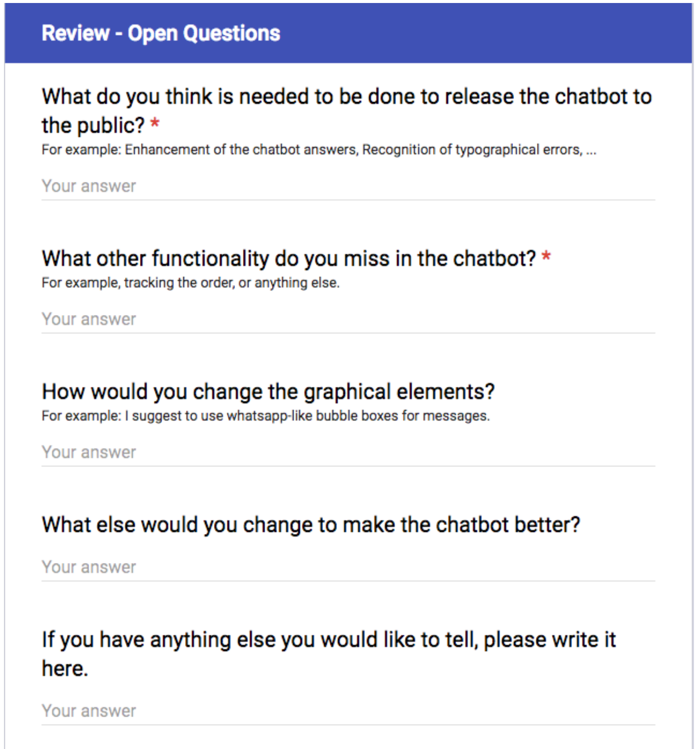
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

Do you think the strategy used in the chatbot will help with the retention of users? *

1 = not at all, 2 = probably not, 3 = I don't know, 4 = probably yes, 5 = for sure

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

Figure 5.8: Closed Questions



Review - Open Questions

What do you think is needed to be done to release the chatbot to the public? *

For example: Enhancement of the chatbot answers, Recognition of typographical errors, ...

Your answer

What other functionality do you miss in the chatbot? *

For example, tracking the order, or anything else.

Your answer

How would you change the graphical elements?

For example: I suggest to use whatsapp-like bubble boxes for messages.

Your answer

What else would you change to make the chatbot better?

Your answer

If you have anything else you would like to tell, please write it here.

Your answer

Figure 5.9: Open Questions

Some of the mentioned improvements and requirements are already fulfilled in the current version. In particular, a connection to the customer support system has been added and thus, more options in the menu are available.

# 6 Comparison With Existing Chatbots

There are already plenty of chatbots available on various platforms, such as Facebook, Kik or within a standalone application. The In-chatbot is compared with two successfull chatbots in this chapter. These chatbots differ in country of origin, release date, platform and many other factors.

## 6.1 AXA Assistance CZ/SK

The first chatbot to be compared is called AXA Assistance CZ/SK or simply AXA chatbot. Based on the DoNotPay chatbot desscribed in chapter 2, the idea of a chatbot, which arranges a travel insurance for customers via Facebook Messenger, has arisen. It has been developed in cooperation of AXA Assistance, an insurance company, and Pragonauts, a company focused on connecting computers and humans. Rather than on the process itself, Pragonauts have focused on making the chatbot firm and enhancing the interactions. [35] [36]

In some aspects, the AXA chatbot is very similar to the In-chatbot. This includes:

- a minimum viable product (MVP) approach,
- a friendly informal language,
- a creative content,
- contact support within the chat,
- buttons as a primary source of communication,

However, the functionality of the support itself is different. In contrast to the connection to Zendesk support system in the In-chatbot, the support here is done by chatting with the administrator of the Facebook page similarly to how it was in the Chatfuel Chatbot. When the support option has been chosen, it has taken one day to receive an answer and a message from the chatbot has been received as well during the conversation with the support. Buttons are slightly different as well, because they disappear after being clicked while buttons in the In-chatbot can be clicked repeatedly.

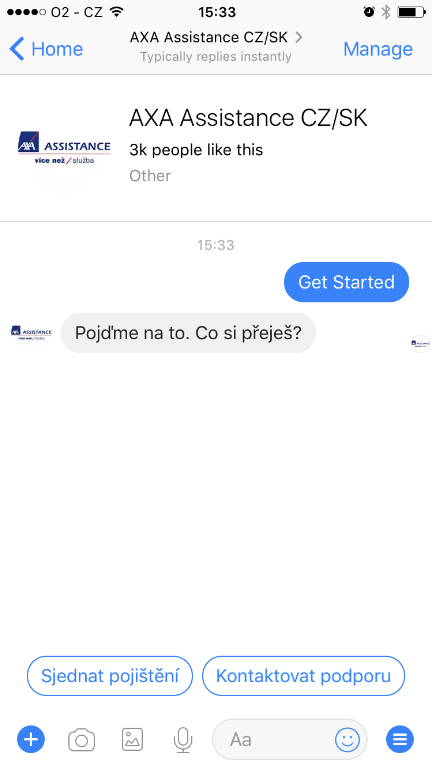
Unlike the In-chatbot, there is no personalization in the greeting and the chatbot uses no personality itself. The initial screen is shown in Figure 6.1. Thanks to the possibility offered by Messenger Platform, the menu is a graphical element and user does not need to ask for it or see it with most of the messages as a button as it is done in the In-chatbot. As the technology behind the chatbot, a Node.js server has been chosen, which is different from the Flask application server used in the In-chatbot backend. Regarding an integration of external services, the AXA chatbot just sends all the information collected by the chatbot to the website of AXA assistance where the order can be completed. The website can be seen in Figure 6.2. The In-chatbot, on the other hand, uses a lot of external services to let user do everything within the chat.

In addition to the functionality of the In-chatbot, the AXA chatbot contains more graphical elements, such as a menu or a carousel of products. The look of the menu can be seen in Figure 6.3 and the carousel in Figure 6.4. Moreover, the AXA chatbot uses a paraphrasing to assure a user that the right information has been entered.

## 6.2 Lemonade

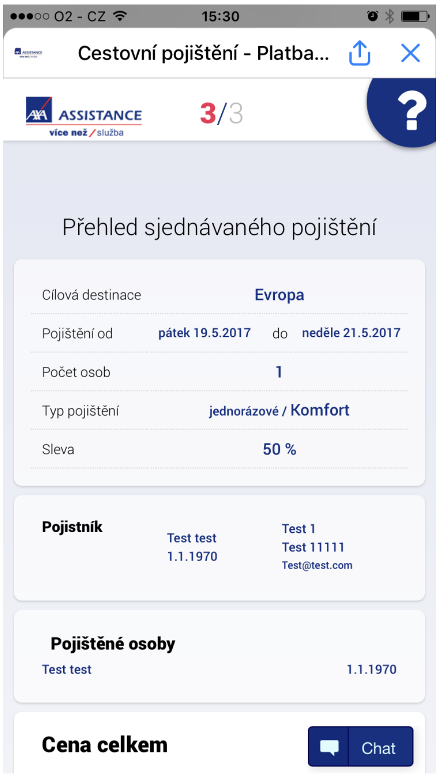
Lemonade is a chatbot developed by the Lemonade insurance agency, a startup located in New York. It is a peer-to-peer insurance company focused on insurance for homeowners and tenants in New York. They have used chatbots and artificial intelligence to automate the process and replace brokers and papers. Based on two real employees, the chatbot uses two personalities: Maya and Jim. [37] [38]

Likewise the In-chatbot, Lemonade uses a friendly informal language and a personalized greeting. Users are also able to use buttons to communicate with the chatbot. In contrast to the In-chatbot, Lemonade does not provide a customer support option within the chat. It also does not follow the minimum viable approach, because the chatbot is a primary tool used to communicate with customers. When summarizing the order, chat environment is left and a new page with tabs appears as shown in Figure 6.6.



Get Started

Pojďme na to. Co si přeješ?



Přehled sjednávaného pojištění

| | |
|---|---|
| Cílová destinace | Evropa |
| Pojištění od | pátek 19.5.2017 do neděle 21.5.2017 |
| Počet osob | 1 |
| Typ pojištění | jednorázové / Komfort |
| Sleva | 50 % |

**Pojistník** Test test 1.1.1970 Test 1 Test 11111 Test@test.com

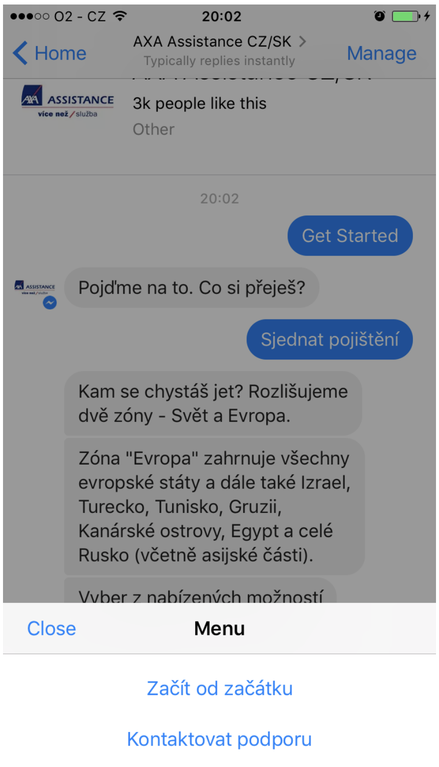
**Pojištěné osoby**

Test test 1.1.1970

**Cena celkem**

Figure 6.1: Conversation Start

Figure 6.2: Order Summary



Get Started

Pojďme na to. Co si přeješ?

Sjednat pojištění

Kam se chystáš jet? Rozlišujeme dvě zóny - Svět a Evropa.

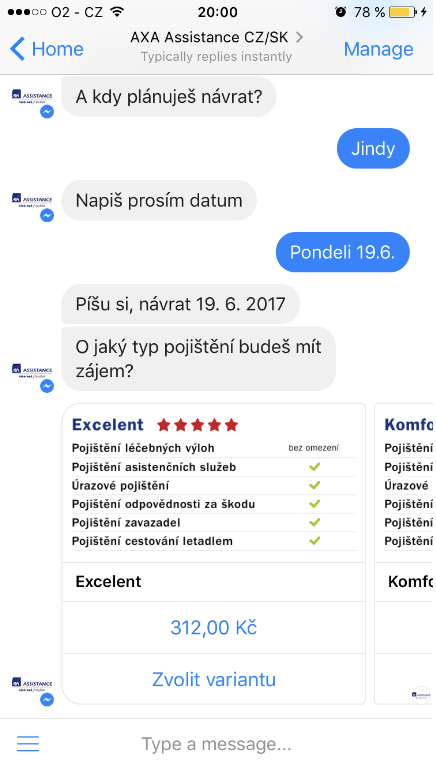
Zóna "Evropa" zahrnuje všechny evropské státy a dále také Izrael, Turecko, Tunisko, Gruzii, Kanárské ostrovy, Egypt a celé Rusko (včetně asijské části).

Vyber z nabízených možností

Začít od začátku

Kontaktovat podporu

Figure 6.3: Menu Element



A kdy plánuješ návrat?

Jindy

Napiš prosím datum

Pondeli 19.6.

Píšu si, návrat 19. 6. 2017

O jaký typ pojištění budeš mít zájem?

**Excelent** ★★★★★

| | |
|---|---|
| Pojištění léčebných výloh | bez omezení |
| Pojištění asistenčních služeb | ✓ |
| Úrazové pojištění | ✓ |
| Pojištění odpovědnosti za škodu | ✓ |
| Pojištění zavazadel | ✓ |
| Pojištění cestování letadlem | ✓ |

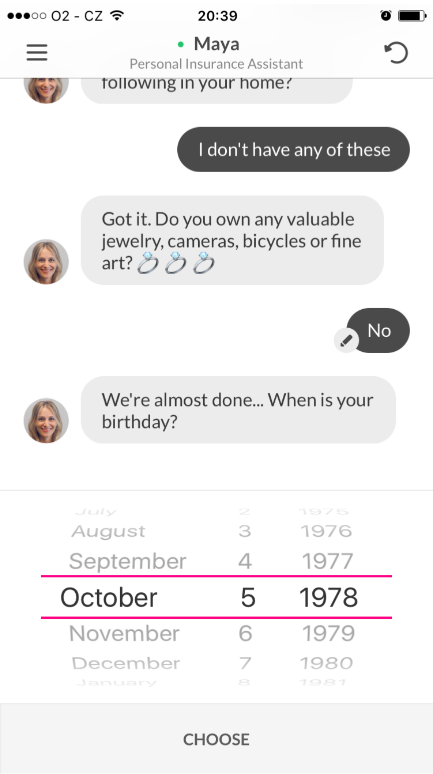
Excelent

312,00 Kč

Zvolit variantu

Komfo

Figure 6.4: Carousel Element



●●●○○ O2 - CZ 20:39

Maya
Personal Insurance Assistant

following in your home?

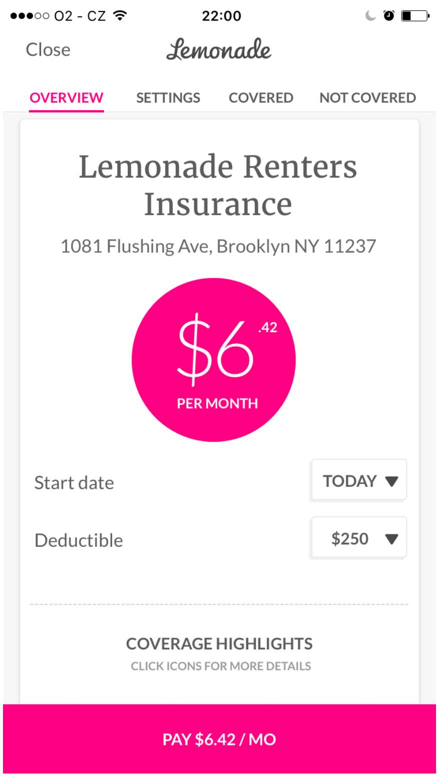
I don't have any of these

Got it. Do you own any valuable jewelry, cameras, bicycles or fine art? 💍💍💍

No

We're almost done... When is your birthday?

Figure 6.5: Date Picker



●●●○○ O2 - CZ 22:00

Lemonade Renters Insurance

1081 Flushing Ave, Brooklyn NY 11237

$6.42 PER MONTH

Start date: TODAY

Deductible: $250

COVERAGE HIGHLIGHTS
CLICK ICONS FOR MORE DETAILS

Figure 6.6: Order Summary

In addition to the In-chatbot, Lemonade uses a lot of various graphical elements, such as map visualization, date picker, data analysis and an animated typing icon. Furthermore, users can go one step back in the conversation flow by clicking on a tiny change icon attached to the previous message. Figure 6.5 shows a date picker element together with previous messages and the change icon. Lemonade also uses a context help, for example, when entering an address.

## 6.3 Summary

Both the chatbots compared to the In-chatbot are similar in the language used, but differs in the technology. The summarizing table 6.1 shows that MVP approach has been used by both the In-chatbot and

| | In-chatbot | AXA Chatbot | Lemonade |
|---|---|---|---|
| Platform | Android | FB Messenger | Android, iOS, web |
| MVP approach | Yes | Yes | No |
| Technology | Python | Node.js | Unknown |
| Artificial Intelligence | No | Unknown | Yes |
| Personality | Jeff | No | Maya and Jim |
| Personalized Greeting | Yes | No | Yes |
| External Services | Yes | Limited | Unknown |
| Customer Support | Yes | Yes | Yes |

Table 6.1: A comparison of In-chatbot, AXA Chatbot and Lemonade

AXA Chatbot. Lemonade is different mainly because it uses an artificial intelligence and the whole application is adapted to the chatbot. Customer support is used in all the compared chatbots. AXA Chatbot has not been assigned a personality like the other two chatbots and it also does not send a personalized greeting. With external services, it is unknown which services are used by Lemonade, but with AXA Chatbot, there is only one connection to an external system.

All the differences and similarities lead to a conclusion that the In-chatbot is the simplest one of the listed chatbots in terms of the functionality, but also powerful in its potential.

# 7 Conclusion

This work shows how chatbots can contribute to an existing laundry and dry cleaning service by enhancing customer satisfaction, simplifying current processes, increasing customer retention and decreasing load on call centers. Based on interactions obtained by the Chatfuel Chatbot, the In-chatbot has been created. During many iterations, a few prototypes have been developed. The current version of the chatbot allows its users to create or change an order based on the previous one and to contact the customer service from within the Mr Jeff's Android application. The chatbot itself uses many external services to provide relevant data to its users.

The first evaluation has already been done. The option to contact the customer support directly, which has been required by some of the testers, has already been included in the current version. However, there is a big need of having more testers involved and start the evaluation from outside of the company as well.

This leads to the future plans with the In-chatbot. At first, an option to change products in the order will be added. Then, another evaluations will run. In terms of bigger changes, usage of artificial intelligence has already been tested with api.ai and will be integrated as soon as there is a feedback on the current solution from the customers. Likewise, a new GUI is being developed, so the limitations of the GUI provided by Smooch will no longer be a problem.

When a more distant future is discussed, an extension to the chatbot is an integration into the iOS application and website. It is also planned to extend the functionality with analytics, push notifications, cross-selling and up-selling.

# Bibliography

1. PANETTA, Kasey. *Gartner's Top 10 Strategic Technology Trends for 2017 – Smarter With Gartner* [online]. Gartner, Inc. and/or its affiliates, 2016 [visited on 2017-03-28]. Available from: `http://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/`.
2. BUTLER, Martin. *Top 10 Strategic Technology Trends for 2017 | Martin Butler | Pulse | Linkedin* [online]. 2016 [visited on 2017-03-29]. Available from: `https://www.linkedin.com/pulse/top-10-strategic-technology-trends-2017-martin-butler`.
3. LUN, Erwin Van. *161 Humanlike Conversational AI Synonyms | Chatbots.org, facilitating the community for professional chatbot developers* [online]. 2011 [visited on 2017-03-31]. Available from: `https://www.chatbots.org/synonyms/`.
4. KERLYL, Alice; HALL, Phil; BULL, Susan. Bringing Chatbots into education: Towards Natural Language Negotiation of Open Learner Models. In: *Applications and Innovations in Intelligent Systems XIV: Proceedings of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Ed. by ELLIS, Richard; ALLEN, Tony; TUSON, Andrew. London: Springer London, 2007, pp. 179–192. ISBN 978-1-84628-666-7. Available from DOI: `10.1007/978-1-84628-666-7_14`.
5. SCHLICHT, Matt. *The Complete Beginner's Guide To Chatbots – Chatbots Magazine* [online]. 2016 [visited on 2017-03-30]. Available from: `https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca`.
6. KAR, Rohan; HALDAR, Rishin. Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements. *International Journal of Advanced Computer Science and Applications*. 2016, vol. 7, no. 11. Available from DOI: `10.14569/IJACSA.2016.071119`.
7. KLÜVER, Jürgen; KLÜVER, Christina. Introduction: Historical Methodical and Conceptual Frames. In: *Social Understanding: On Hermeneutics, Geometrical Models and Artificial Intelligence*.

Dordrecht: Springer Netherlands, 2011, pp. 1–64. ISBN 978-90-481-9911-2. Available from DOI: 10.1007/978-90-481-9911-2_1.

8. WALLACE, Richard S. The Anatomy of A.L.I.C.E. In: *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Ed. by EPSTEIN, Robert; ROBERTS, Gary; BEBER, Grace. Dordrecht: Springer Netherlands, 2009, pp. 181–210. ISBN 978-1-4020-6710-5. Available from DOI: 10.1007/978-1-4020-6710-5_13.

9. FITRIANIE, Siska; WIGGERS, Pascal; ROTHKRANTZ, Leon J. M. A Multi-modal Eliza Using Natural Language Processing and Emotion Recognition. In: *Text, Speech and Dialogue: 6th International Conference, TSD 2003, České Budějovice, Czech Republic, September 8-12, 2003. Proceedings*. Ed. by MATOUŠEK, Václav; MAUTNER, Pavel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 394–399. ISBN 978-3-540-39398-6. Available from DOI: 10.1007/978-3-540-39398-6_56.

10. TINA, Klüwer. From Chatbots to Dialogue Systems. In: *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*. Ed. by PEREZ-MARTON, Diana; PASCUAL-NIETO, Ismael. IGI Global Publishing Group, 2011, pp. 1–22. Available from DOI: 10.4018/978-1-60960-617-6.ch001.

11. MONTERO, Calkin A. S.; ARAKI, Kenji. Enhancing Computer Chat: Toward a Smooth User-Computer Interaction. In: *Knowledge-Based Intelligent Information and Engineering Systems: 9th International Conference, KES 2005, Melbourne, Australia, September 14-16, 2005, Proceedings, Part I*. Ed. by KHOSLA, Rajiv; HOWLETT, Robert J.; JAIN, Lakhmi C. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 918–924. ISBN 978-3-540-31983-2. Available from DOI: 10.1007/11552413_131.

12. KNILL, Oliver; CARLSSON, Johnny; CHI, Andrew; LEZAMA, Mark. An artificial intelligence experiment in college math education. *Preprint available at http://www.math.harvard.edu/~knill/preprints/sofia.pdf*. 2004.

13. TIEN, James M. Internet of Things, Real-Time Decision Making, and Artificial Intelligence. *Annals of Data Science*. 2017, pp. 1–30. ISSN 2198-5812. Available from DOI: 10.1007/s40745-017-0112-5.

14. NEFF, Gina; NAGY, Peter. Automation, Algorithms, and Politics | Talking to Bots: Symbiotic Agency and the Case of Tay. *International Journal of Communication*. 2016, vol. 10. ISSN 1932-8036. Available also from: `http://ijoc.org/index.php/ijoc/article/view/6277`.

15. ALAIERI, Fahad; VELLINO, André. Ethical Decision Making in Robots: Autonomy, Trust and Responsibility. In: *Social Robotics: 8th International Conference, ICSR 2016, Kansas City, MO, USA, November 1-3, 2016 Proceedings*. Ed. by AGAH, Arvin; CABIBIHAN, John-John; HOWARD, Ayanna M.; SALICHS, Miguel A.; HE, Hongsheng. Cham: Springer International Publishing, 2016, pp. 159–168. ISBN 978-3-319-47437-3. Available from DOI: `10.1007/978-3-319-47437-3_16`.

16. LEE, Boyle. *Joshua Browder who created DoNotPay launches robot to give Britons free legal help | Daily Mail Online* [online]. Daily Mail Online, 2016 [visited on 2017-05-17]. Available from: `http://www.dailymail.co.uk/money/news/article-3394093/Joshua-Browder-created-DoNotPay-launches-robot-Britons-free-legal-help.html?ITO=1490&ns_mchannel=rss&ns_campaign=1490`.

17. *Datasys vyvinul inteligentního chatovacího robota pro komunikaci se zákazníky | DATASYS s.r.o.* [online]. Datasys, 2016 [visited on 2017-05-17]. Available from: `http://www.datasys.cz/datasys-vyvinul-inteligentniho-chatovaciho-robota-pro-komunikaci-se-zakazniky`.

18. *Mr Jeff - Tu App Tintorería y Lavandería* [online]. Mr Jeff, 2016 [visited on 2017-04-02]. Available from: `http://mrjeffapp.com/`.

19. *Create a Facebook AI Chatbot Without Coding* [online]. CHATFUEL, 2017 [visited on 2017-04-04]. Available from: `https://chatfuel.com/`.

20. *Chatfuel help: learn how to create chatbot for Facebook Messenger* [online]. CHATFUEL, 2017 [visited on 2017-04-07]. Available from: `https://help.chatfuel.com/facebook-messenger/`.

21. *Real Time Chat | Instant Messaging SDK | Sinch* [online]. Sinch, a CLX Communications Company, 2017 [visited on 2017-05-06]. Available from: `https://www.sinch.com/instant-messaging/`.

22. *QuickBlox Backend: Cloud Communication Backend API As A Service For Mobile And Web Apps* [online]. Quickblox, 2017 [visited on 2017-05-06]. Available from: `https://quickblox.com/`.

23. *Unified Messaging API and Customer Conversation Platform | Smooch* [online]. Smooch Technologies Inc., 2017 [visited on 2017-05-06]. Available from: `https://smooch.io/`.

24. *Messenger Platform* [online]. Facebook, 2017 [visited on 2017-05-06]. Available from: `https://developers.facebook.com/docs/messenger-platform`.

25. *Wit.ai* [online]. Wit.ai, Inc., 2017 [visited on 2017-05-06]. Available from: `https://wit.ai/`.

26. *Conversational UX Platform for products and services - API.AI* [online]. API.AI, 2017 [visited on 2017-05-06]. Available from: `https://api.ai/`.

27. *LUIS: Homepage* [online]. Microsoft, 2016 [visited on 2017-05-06]. Available from: `https://www.luis.ai/home/index`.

28. *IBM Bluemix - Watson* [online]. IBM, 2016 [visited on 2017-05-06]. Available from: `https://www.ibm.com/cloud-computing/bluemix/watson`.

29. *Microsoft Bot Framework* [online]. Microsoft, 2016 [visited on 2017-05-06]. Available from: `https://dev.botframework.com/`.

30. *Meya Bot Platform* [online]. Locl Interactive Inc., 2017 [visited on 2017-05-06]. Available from: `https://meya.ai/`.

31. *Botkit* [online]. XOXCO, INC., 2017 [visited on 2017-05-06]. Available from: `https://www.botkit.ai/`.

32. *WooCommerce – The Best eCommerce Platform for WordPress* [online]. WOOCOMMERCE, 2017 [visited on 2017-05-06]. Available from: `https://woocommerce.com/`.

33. *Stripe* [online]. Stripe, 2017 [visited on 2017-05-06]. Available from: `https://stripe.com/`.

34. *Zendesk | Customer Service Software & Support Ticket System* [online]. Zendesk, 2017 [visited on 2017-05-06]. Available from: `https://www.zendesk.com/`.

35. *AXA ASSISTANCE CHATBOT* [online]. Pragonauts, 2017 [visited on 2017-05-17]. Available from: https://pragonauts.com/#axabot.

36. TÓTHOVÁ, Lucia. *Jak se zrodil, čím byl inspirován a kolik stál první český pojišťovací chatbot od společnosti AXA Assistance — #fintechcowboys.cz* [online]. Fintechcowboys, 2017 [visited on 2017-05-17]. Available from: https://fintechcowboys.cz/se-zrodil-cim-inspirovan-kolik-stal-prvni-cesky-pojistovaci-chatbot-od-spolecnosti-axa-assistance/.

37. *Lemonade. Forget Everything You Know About Insurance.* [online]. Lemonade Insurance Agency, LLC, 2016 [visited on 2017-05-17]. Available from: https://www.lemonade.com/.

38. SAWERS, Paul. *P2P insurance firm Lemonade launches out of stealth, powered by chatbots, morals, and big bucks | VentureBeat | Apps | by Paul Sawers* [online]. VentureBeat, 2017 [visited on 2017-05-17]. Available from: https://venturebeat.com/2016/09/21/lemonade-insurance-social-good-p2p/.