Information System for Recommendation List Formation of Clothes Style Image Selection According to User's Needs Based on NLP and Chatbots

Vitaliy Husak[1], Olga Lozynska[0000-0002-5079-0544][2], Ihor Karpov[0000-0003-4885-5078][3], Ivan Peleshchak[0000-0002-7481-8628][4], Sofia Chyrun[0000-0002-2829-0164][5], Anatolii Vysotskyi[6]

[1-4]Lviv Polytechnic National University, Lviv, Ukraine
[5]IT Step University, Lviv, Ukraine
[6]Anat Company, Lviv, Ukraine

ygtwhy@gmail.com, olha.v.lozynska@lpnu.ua, shad1ksen@gmail.com, peleshchakivan@gmail.com, chyrunsofia@gmail.com[5], anat1957@gmail.com

Abstract. The work is devoted to the development of the information system for creating a list of recommendations for fashionable style of clothing meeting the users` needs using NLP and chat bots. It provides studying and practical use of chat bots as virtual assistants involving natural language processing. The purpose of this work is to develop software so that chat bot will be function on Telegram messenger base. During the implementation there were analyzed another chat bots that have similar functionality, their advantages and disadvantages. The server part of the system that process the oncoming data using Java programming language and Spring Framework is build. For development, IntelliJ IDEA as IDE is used. NLP also to the system using DialogFlow service so that helped to archive a high level of natural user language recognition that increased interactivity and communication is connected. Connection to Telegram messenger performs via Bot API. For processing data, there are used Hibernate as a database ORM and Jsoup as html parser.

Keywords: NLP, chatbots, content recommendation, content analysis, financial products, processes, services, Telegram, API, Data Mining, Machine Learning

1 Introduction

In the modern world, the use of chat bots in everyday life is becoming increasingly popular [1]. They allow the user to find useful information for him: weather forecast, exchange rate, book a plane or train ticket, order a taxi, etc. [2]. Nowadays, no one can say that chat bots can replace people, but can automate a huge amount of routine operations, answering frequently asked questions or providing background information [3]. In the course of the dialogue with the user; they offer the most relevant information and consistently narrow the list option [4]. Applying natural language processing, chat bots can interpret the data received in the user's language and using dialogue systems, the data into meaningful information and show the response to the request are decrypted [5]. Also in the application of machine learning, there is the possibility of "training" the chat bot, in consequence of which, over time it will recognize the greater amount of information as well as give a wider range of results [6]. Chat bots are empowered instantly provide the user with the information they need [7]. This is a big plus for business, since there is no need for 24-hour support with the people [8]. Two main functions that perform chat bots of online stores in buyers terms [9]: looking for goods on request; showing products without targeted search is in the form of catalogues, collections and other forms. The topicality of the work derives from the wide popularity of instant messaging systems and the development of alternative ways of interaction with Internet-shops [10]. It is also worth noting that, based on general trends concerning the development of artificial intelligence technologies, machine learning modern bots, including chat bots can take on more and more functions. The purpose of the work is to demonstrate and analyze the functionality of the chat-bot, built on the Telegram-messenger platform and with the additional use of non-recognized natural language recognition services. This will allow you to explore the business scope of chat bots and integrate them into online stores and specific functionality. To achieve this goal, the following tasks are identified:

- Familiarization and analysis of functions and means of communication presenting existing solutions [11];
- Researching and comparing existing chat bots on different messengers with similar functionality and using natural language processing for user interaction [12];
- Construction and demonstration of a chat bot that is ready for use and interacts with the necessary customized services [13].

The object of research is the chat bot and the integration of natural language processing into its functionality. The subject of research is the development, integration, interaction of information systems with chat bots. Research methods are a way of using chat bots in the field of clothing sales, construction of the active chat bot, integrated with certain third-party services and platforms. The scientific novelty of this work is in the construction of an information system for the development of chat-bots, with the possibility of integrating natural language packs. The practical value of the results of the work is in the popularity of the use of chat bots and their integration into online stores in our time and the obvious lack of use of modern capabilities of artificial intelligence.

2 Analysis of literary sources and subject field

At first sight, it seems that the problems that are solved by using NLP are rather trivial, although there is quite a large number of complexities in this area, linked both to technical implementation and to the human factor [14]. For example, a billion people speak English and each carrier uses it uniquely: there are different dialects, individual communication features [15]. One of the main problems is the ambiguity of words, expressions or phrases, for the correct interpretation of which a context is needed [16-21]. Often people in messages ignore the rules of grammar, or answer so short that it is almost impossible to restore the structure of the message [22-24].

Due to the large scale of contact centers used by hundreds of thousands of companies around the world, a large number of routine operations increasingly being are handed over to artificial intelligence: chat bots are used to answer frequently asked questions (according to the FAQ, but with the understanding of language and user requests) [25]. All this allows you to significantly reduce the cost of staff, as well as increase the bandwidth of operators without increasing state. The usual world has become even more comfortable when an assistant Alexa came up, who can turn on your favorite music, find and tell news, order products home or taxi. This is the first mass-market device in the US market that recognizes language at a high-quality level, even if there is a strong external noise. Voice assistants, such as Google Assistant from Google or Siri from Apple, interpret the voice message in a text and execute commands upon request [26]. Topical chat bots do a number of things [27-34]:

- People spend a lot of time typing in messengers that quickly develops this segment on the app market, they use them not only for friendly correspondence, but also for the benefit of business development;
- Rapid development of technologies of artificial intelligence, machine learning allows modern chat bots to maximally approximate their behavior to human;
- Modern chat bots rely not only on text but also on displaying media, links, and forms, creating an atmosphere of application use.

All of this allows you to use them for very different purposes, such as shopping assistance, back services, news, etc. A good bot should not have everything; the best solution when it is created to give it the ability to perform a certain action very well. You do not have to try to please everyone, it is best to focus on specific functionality, topics and audiences. Typically, chat bots are hybrid systems that combine machine learning models, classic algorithms, and a set of rules. The structure of the chat bot in Fig. 1 is depicted.

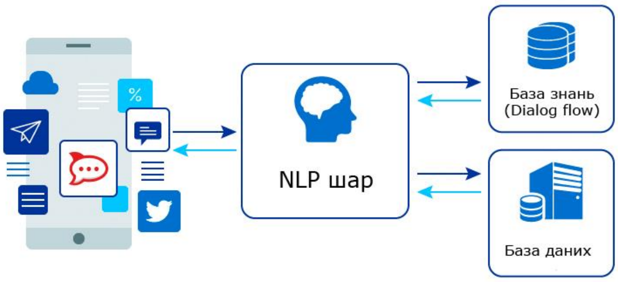


Fig. 1. Chart of data processing chat bot

3 Analysis of known problem solving tools

Consider the following chat bots in the clothing e-commerce segment Nike StyleBot, eBay ShopBot, H & M bot for Kik, chatShopper, Masha.ai. **Nike StyleBot** in Facebook Messenger gives users the ability to create outfits; they can create their own style or sneakers. It is possible to get a great set of styles for sports, work and much more. If the user likes the outfit, he can find out more on the official site Nike. The disadvantage is that the bot focuses only on women, since all the pictures are only female models for those looking for men's things; they can only use the official site (Fig. 2).

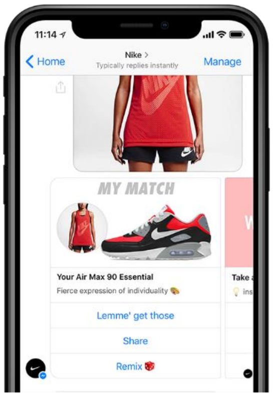

11:14
Home
Nike >
Typically replies instantly
Manage
MY MATCH
Your Air Max 90 Essential
Fierce expression of individuality
Lemme' get those
Share
Remix
Take

Fig. 2. StyleBot View

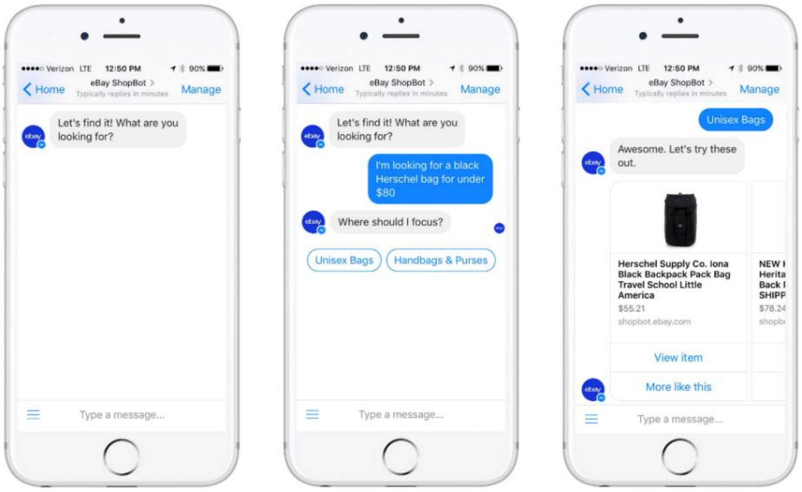
Advantages:

- A large range of new styles;
- Design and navigation;
- Creation of your own sneakers.

Disadvantages:

- The user will only associate with one brand;
- Clothes sets can only be picked up by women;
- The bot is only available on Facebook Messenger.

eBay ShopBot is a personal sales assistant whose purpose is to help the user find the best deals regardless of whether he is looking for something specific or just looking at new ideas (Fig. 3). Its feature is delivering the best eBay inventory to the user with a focus on products, fast and free shipping and special offers. The advantages of the bot include the fact that the user can use text, language or photos to tell what he is looking for. Also, from the best sides you can select a user-friendly design. The disadvantage is the connection to one service, as well as the presence of a chat bot on only one platform.



Let's find it! What are you looking for?

Let's find it! What are you looking for?

I'm looking for a black Herschel bag for under $80

Where should I focus?

Unisex Bags

Handbags & Purses

Unisex Bags

Awesome. Let's try these out.

Herschel Supply Co. Iona Black Backpack Pack Bag Travel School Little America
$55.21
shopbot.ebay.com

NEW Herita Back SHIPP
$78.24
shopb

View item

More like this

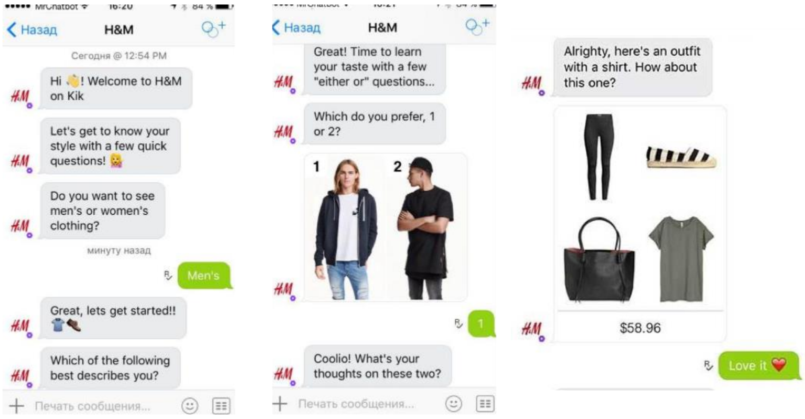
Fig. 3. Examples of communication with ShopBot

H & M Kik Bot is chat bot for Kik messenger (Fig. 4). It provides an opportunity to find a set of clothing for a user with respect to his own preferences. Its features include the availability of good communicative skills sufficient for the user to have the impression that he communicates with the real person. Advantages:

- Convenient navigation;
- The ability to choose a style;

Disadvantages:

- Available in Kik, a narrowly distributed messenger;
- Low level of speech recognition.



Сегодня @ 12:54 PM

Hi 👋! Welcome to H&M on Kik

Let's get to know your style with a few quick questions! 💁

Do you want to see men's or women's clothing?

минуту назад

Men's

Great, lets get started!! 👕👞

Which of the following best describes you?

Great! Time to learn your taste with a few "either or" questions...

Which do you prefer, 1 or 2?

1

Coolio! What's your thoughts on these two?

Alrighty, here's an outfit with a shirt. How about this one?

$58.96

Love it ❤️

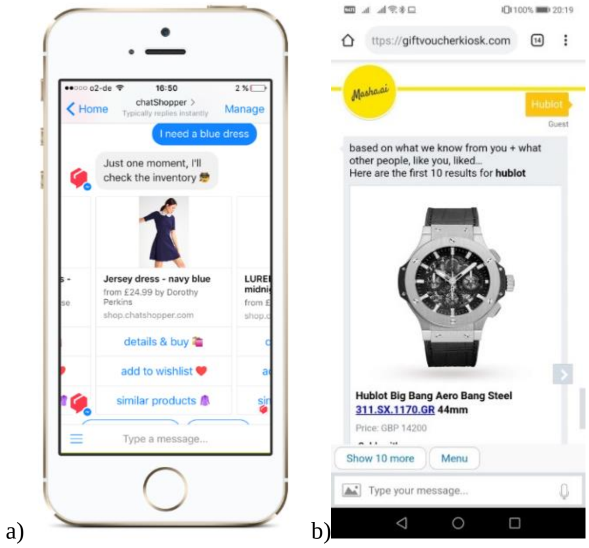
Fig. 4. An example of communication in H & M Kik bot

chatShopper is Emma's personal assistant for finding fashionable fashion clothes in Facebook Messenger (Fig. 5a). It was nominated for the position of one of the best chat bots in 2016. The peculiarity of this bot is the high level of recognition of the user's language, which increases the chance of conversion. Emma's personal assistant is distinguished by the interactivity and the ability to conduct a dialogue with the user.

Among the advantages is the possibility to search from various retailers (no binding to a particular store), as well as the availability of convenient navigation.

This chat bot has some minor disadvantages, including:

- Average level of speech recognition;
- Support only German and English;
- Only a few requests filtering functions.



I need a blue dress

Just one moment, I'll check the inventory

Jersey dress - navy blue
from £24.99 by Dorothy Perkins
shop.chatshopper.com

based on what we know from you + what other people, like you, liked...
Here are the first 10 results for **hublot**

Hublot Big Bang Aero Bang Steel 311.SX.1170.GR 44mm
Price: GBP 14200

Fig. 5. An example of a) communication with Emma and b) Masha.ai result on request

Masha.ai is a personal text assistant for making purchases in more than 400 brands (Fig. 5b). The feature of this chat bot is the use of AI. In addition to the fact that the chat bot is present in Facebook Messenger, there is an individual chat site application, as well as the ability to integrate into Alexa and Google Assistant. Advantages:

- Large number of choices;
- Convenient navigation;
- Interactivity;
- The possibility of narrow-directed search.

Disadvantages:

- User is limited in communication;
- Low level of speech recognition.

4 Functionality of the software system

Functionality of system that you need to design:

1. Filling the user's speech recognition system with basic possible keywords (brand name, color, size) that will be used for recognition;
2. Interactivity - helping the user to use the provided chat bot functionality;
3. The ability to recognize the user's language using natural language processing;
4. Providing a wide range of products to find the right amount of things, regardless of the complexity of the search filter;
5. Interacting of the chat bot with the user with the help of small-talk;
6. Authentication authorization - the user may authorize the system and be recognized in subsequent sessions;
7. Separated architecture for possible further integration of other messengers;
8. Availability of at least two online stores in the database as resources for finding products that meet the requirements;
9. Creation of an admin-panel for managing the chat bot and configuration changes;
10. Providing an intuitive user interface;
11. Informing the user about relevant products through the notification system;
12. Application Program Interface (API) for communication with systems;
13. The use of artificial intelligence - the use of methods and algorithms that allow the computer to make conclusions and proposals based on computing by special methods and programs;
14. Processing of existing information in databases;
15. Datamining (Duplication of data) is ensuring the detection of data in previously unknown, non-trivial, practically useful and accessible for interpretation of knowledge in large arrays.
16. Machine Learning of the system during communication with the user;
17. Ensuring the possibility of processing requests with incomplete information and, if necessary, making certain clarifications;
18. The possibility of a chat bot to process user requests that contain language or typographical errors.

Table 1. Attributes of functions

Function	Status	Ptior.	Diffic.	Risk	Stability	Vers.
1	Included	Critical	Average	Low	Average	0.2
2	Included	Critical	High	Average	High	0.2
3	Included	Critical	High	Average	Average	0.4
4	Included	Critical	High	Average	Average	0.6
5	Approved	Important	Average	Average	Average	0.6
6	Included	Important	High	Low	Low	1.0
7	Approved	Critical	Low	Low	Low	0.4
8	Approved	Important	Average	Average	Low	0.6
9	Approved	Important	High	Low	Average	1.0
10	Included	Useful	Average	Average	Average	1.2
11	Included	Useful	Average	Low	Low	1.2

Function	Status	Ptior.	Diffic.	Risk	Stability	Vers.
12	Approved	Useful	Low	Low	Low	1.2
13	Included	Useful	High	High	Average	1.4
14	Included	Useful	Average	Average	Average	1.8
15	Approved	Useful	Average	Average	Average	2.0
16	Approved	Useful	High	Average	Average	2.2
17	Approved	Useful	Average	Average	Average	1.8
18	Approved	Useful	Average	Average	Average	1.8

The most obvious area of natural language processing can be called machine interpreters, chat bots and voice assistants. The chat bot will calmly handle the user's requests, even with possible errors in the text, depending on the degree of learning with the help of NLP. The difficulty of creating a chat bot correlates with the atypical interaction of the user with it. Even considering the high-level NLP chat bots architecture, one can conclude that the development of its own specialized chat bot may require research and solving quite complex tasks. It depends largely on how the bot perform complex and specialized tasks and how the user formulates them.

5 System analysis and reasoning of the problem

System analysis (SA) is aimed at solving complex problems. The problem arises when there is a distinction between the desired and the actual that is, an abstract category that reflects the understanding of people's motives in their activities. Problems are generated and solved by people, and therefore the concept of "problem" has human features of perception, which gives rise to the following difficulties:

- Uncertainty about understanding the problem;
- The complexity of problem solving in the long-term perspective;
- The difficulty of classifying problems and, as a consequence, the choice of inadequate means of solving them;
- Distorted evaluation of problems (close, but small problems obscure large but remote);
- Incorrect assessment of the significance of problems due to the narrow professional aspect;
- Mixing the goals are achieved with the means to achieve them.

The purpose of applying the SA to a particular problem is to increase the degree of validity of the decision that is taken and to overcome the above difficulties.

For SA, the following methodological principles are important:

- Organic unity of subjective and objective;
- The structure of the system, which determines the integrity and stability of the characteristics of the system;
- Dynamism of the system;
- Interdisciplinary nature of systemic research;
- Organic unity of formal and informal.

Information system is a collection of information resources, technical means, and methods of information processing. The result of the work of the information system are:

- Storage of information;
- Data processing;
- Issuance of information.

The information model of the software error tracking system, which should become the basis for the functioning of the system, must record the value of the characteristics of the system and objects of observation. Because of this, the model should have the following characteristics:

- Perform the decomposition of entities;
- Keep the states of the object relative to time;
- Remove the object from the priority state.

At the first stage of the system creation, there should be a definition of the problem.

It is necessary to correctly and accurately formulate problems. To this end, a large problem is subjected to decomposition (breaking one problem into several smaller ones). The objects of information include:

- Users;
- Developed systems;
- Control of defect states.

The purpose of the system development is to create a chat bot and interactive elements for the information system of the formation of the list of clothes according to the user's request. All system tasks can be divided into the following groups:

- Filling in information about the user;
- Conversation with the user;
- Obtaining the necessary information from the user's language for the issuance of a relevant result.

When creating a new information system, attention should be paid to all aspects that affect the realization of this task. In general, the following aspects can be distinguished:

- Hardware support;
- Software;
- Custom aspects;
- Economic aspects.

Each aspect must be thoroughly checked. This is necessary for the adequate and proper functioning of the system. The work of each system should be aimed at achieving its goal. Understanding the purpose helps to understand the essence of the systems being studied. Formulating a goal is a complex logical operation that is implemented within the SA. The arbitrariness in the process of allocating subsystems implemented in the system can lead to the failure of a systematic study. First, for the system, you need to create a configuration file that allows you to manage flexibly the entire service, for example, to change the data to access the database. The next step should be to select and connect the system to handle the natural language of the user and select the messenger that serves as the ecosystem for the chat bot.

In addition, we must not forget that any system should be convenient for our users, because each system created should be convenient first for it. An important factor is the efficiency and speed of the system, which is determined in the rapid development of the search for the most complex and critical errors.

5.1 Tree of Goals

Tree of Goals represents a definite visual representation of the relationship and sequence of goals representing the division of the mission and the main purpose of the task by subdivision or individual actions (Fig. 6).

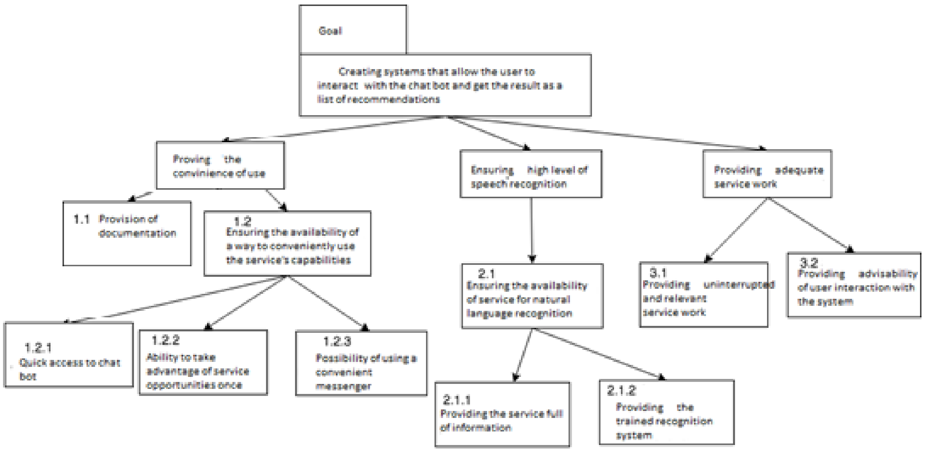

Goal
Creating systems that allow the user to interact with the chat bot and get the result as a list of recommendations
Proving the convinience of use
Ensuring high level of speech recognition
Providing adequate service work
1.1 Provision of documentation
1.2 Ensuring the availability of a way to conveniently use the service's capabilities
2.1 Ensuring the availability of service for natural language recognition
3.1 Providing uninterrupted and relevant service work
3.2 Providing advisability of user interaction with the system
1.2.1 Quick access to chat bot
1.2.2 Ability to take advantage of service opportunities once
1.2.3 Possibility of using a convenient messenger
2.1.1 Providing the service full of information
2.1.2 Providing the trained recognition system

Fig. 6. Tree system goals

Figure 6 depicts the purpose tree of the system considered in this work, the service intended for the work of the chat bot, which generates a list of clothing recommendations using NLP. According to the target tree, the service has four main goals:

1. Ensuring ease of use;
2. Ensuring high level of speech recognition;
3. Ensure the adequacy of the service.

Consider each of these goals. Ease of use:

- Ensuring the availability of documentation will allow you to easily and conveniently receive a prompt to the user in a particular situation related to the work or configuration of the service.

- Ensuring the availability of a way to conveniently manage service capabilities in this task involves creating an appropriate system that is easy to manage. The convenient way provides for the possibility of quick access to the chat-bot, the opportunity to take advantage of the features of the service at a time and the possibility of using a user-friendly messenger.

High-level language recognition target: Ensuring the availability of the service for recognizing the user's natural language. The service should is filled with information and "trained" in order for the system to give the user a relevant result. The purpose of the adequacy of the service is one goal, which directly relates to the actual mechanism of service. It is ensuring the uninterrupted operation of the service and expediency of user interaction, with the system and will make this system competitive and useful in use.

5.2 Specifying the functioning of the system

Data Flow Diagrams (DFDs) are the main means of modeling the functional requirements of future software. With their help, these requirements are divided into functional components (processes) and are presented as a network linked by data flows. The main purpose of such tools is to demonstrate how each process converts its input data on the output, as well as to identify the relationship between these processes.

When using this model, the system is represented in the form of a hierarchy of data flow diagrams that describe the asynchronous process of transforming information from the moment it is introduced into the system before being issued to the user. At each subsequent level of the hierarchy, the refinement of processes takes place until the next process is recognized as elementary. Models of data flows were independently proposed initially by E. Jordan (1975), then by C. Hayn and T. Sarson (1979). Based on these models, the classical methodologies of structural analysis and software design are based on the Jordan-de Mark and Guyana-Sarson software respectively.

The basis of the model are the notion of external nature, process, storage (data storage) and data flow. External entity is a material object or an individual acting as sources or receivers of information, such as customers, staff, etc.

The process is the transformation of input data streams into outputs according to a certain algorithm. Each process in the system has its own number and is associated with the executor that makes the transformation. As in the case of functional diagrams, physically computers can do conversion, by hand or by special devices. On the upper levels of the hierarchy, when processes are not yet defined, instead of the notion of "process" use the notion of "system" and "subsystem", which denote respectively the system as a whole or its functionally completed part. Data warehouse is an abstract device for storing information. The type of device and methods of placement, deletion, and storage for such a device do not detail. Physically it can be a database, a file, a table in RAM, a card file on paper, and the like. Data stream is the process of transmitting some information from source to receiver. Physically, the process of transmitting information can occur on cables running an application or program system or manually involving devices or people outside the projected system.

Figure 7 depicts a context diagram of data streams that describes Sub target 4.2 from the target tree, namely, "Ensuring the user's interaction with the system." From this diagram, one can determine that a user communicates with the system through a chat bot, fills in information about himself and sends search requests to the system (Fig. 8).

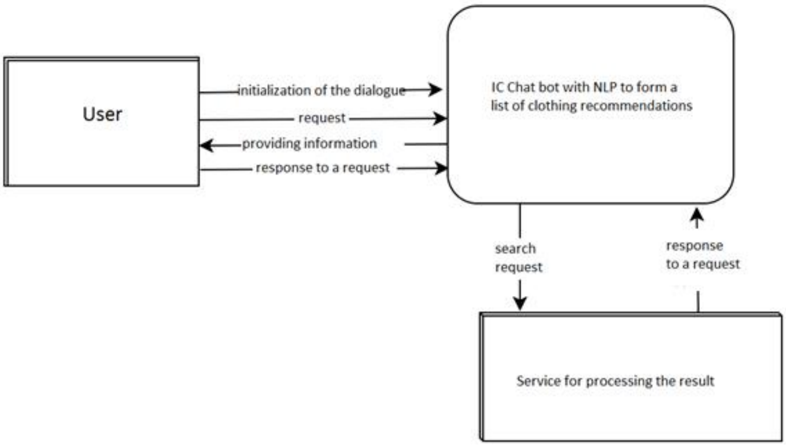

User
initialization of the dialogue
request
providing information
response to a request
IC Chat bot with NLP to form a list of clothing recommendations
search request
response to a request
Service for processing the result

Fig. 7. Context Data Flow Chart

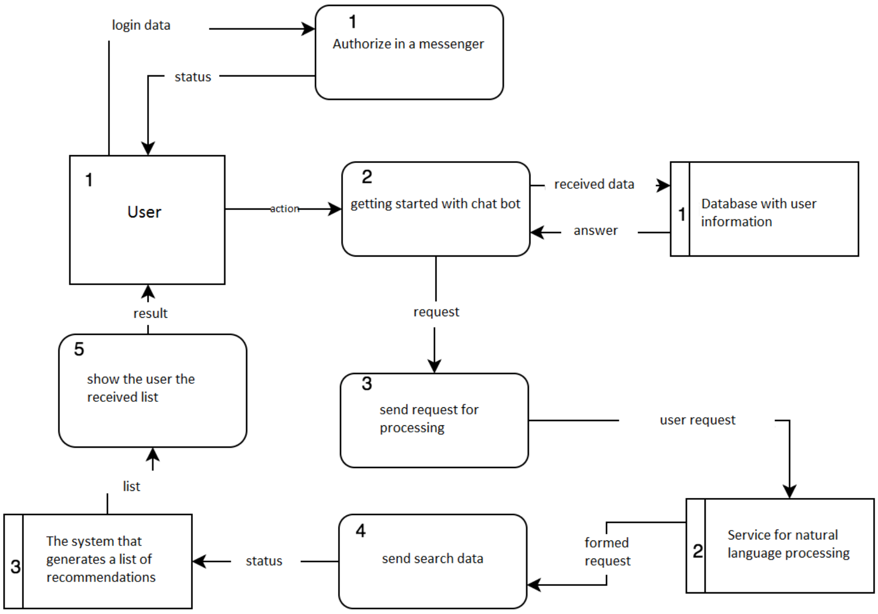

login data
1
Authorize in a messenger
status
1
User
action
2
getting started with chat bot
received data
answer
1
Database with user information
request
result
5
show the user the received list
3
send request for processing
user request
list
3
The system that generates a list of recommendations
status
4
send search data
formed request
2
Service for natural language processing

Fig. 8. Diagram of data flows in the Guine-Sarson notation of the first detail.

External entity 1 is "User" (initial data: "login data" (login and password for authorization in the messenger), "action" (necessary steps to get started with the chat bot); input: "status" (system notifications about the result of the authorization / registration of the user), "result" (result with a molded list of recommendations of the list of clothes)).

Process 1 is "Log in to the Messenger" (input: "user data", "response" (reply to the authorization status logger); output: "status", "requested data" (login data)).

Process 2 is "Getting Started with a Chat Bot" (input: "action", "response" (response from a database about the presence of data about a particular user); source data: "received data", "user information" (data about the user who received from the database or from the messenger)).

Process 3 is "Submitting a Processing Request" (input: "request"; output: "user request").

Process 4 is "Sending data to search" (input: "generated query" (query translated into system-understandable format); output: "status" (generated query result)).

Process 5 is "Obtained list to display to the user" (input: "status"; output: "result" (result in the form of the established list of clothing recommendations)).

Recipient 1 is "Database with information about users" (input: "received data"; output: "response").

Drive 2 is "Service for the processing of natural language" (input: "user request"; output: "formed request").

5.3 Building a hierarchy of processes (functions, tasks)

Figures 9 - 11 detail the processes of the system for evaluating practical knowledge and skills by manipulating: send a processing request, send data to the search, and display the user the list received. It should be understood that the figures show the main characteristics of the system, its purpose and purpose. Other features and additional features will be considered as an example in section 4. Process 3.1 is "Complete the request with the necessary information" (input: "user request"; output: "request"). Process 3.2 is "Form the essence of the request" (input: "completed query"; output: "test request").

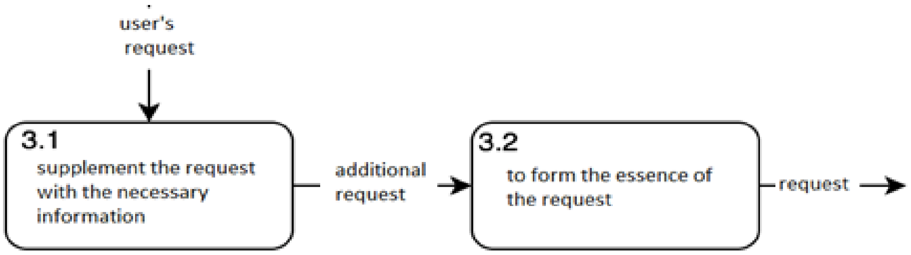

user's request
3.1
supplement the request with the necessary information
additional request
3.2
to form the essence of the request
request

Fig. 9. Detail process "Send Processing Request".

Process 4.1 is "Generate data for search" (input: "data from the query"; output: "data for the search"). Process 4.2 is "Prepare Search Query" (input: "data to search"; output: "query"). Process 4.3 is "Send request and determine transaction status" (input: "request"; output: "status" (generated response from the server)). Drive 2 is "Service for the processing of natural language" (input: "user request"; output: "data from the query" (generated data)). Drive 3 is "The system that generates a list of recommendations" (input: "status"; initial data: "list" (formed list of recommendations)).

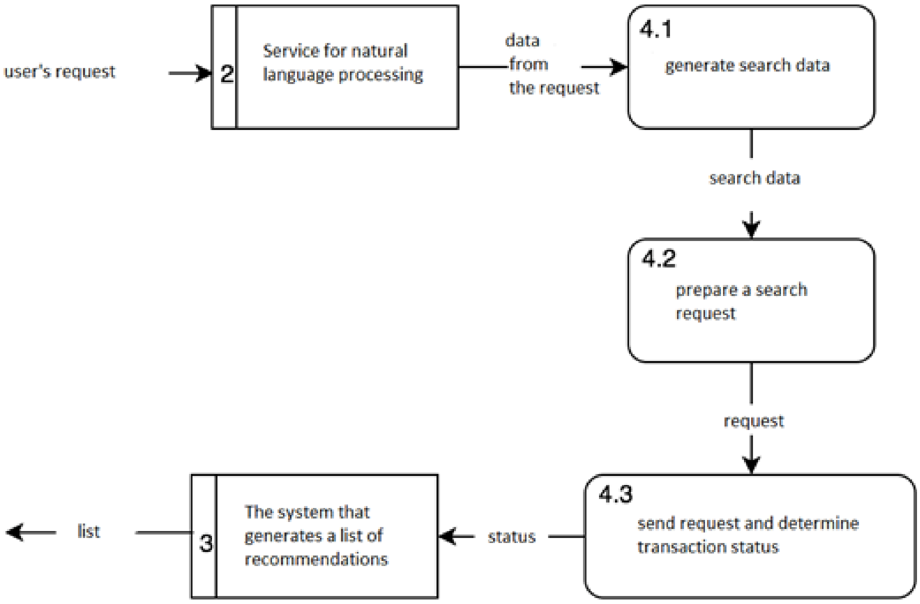

user's request
2
Service for natural language processing
data from the request
4.1
generate search data
search data
4.2
prepare a search request
request
4.3
send request and determine transaction status
status
3
The system that generates a list of recommendations
list

Fig. 10. Detail process "Send data to search".

Process 5.1 is "Process List" (input: "list of recommendations"; output: "list"). Process 5.2 is "Send the generated list to the user" (input: "list"; output: "result").

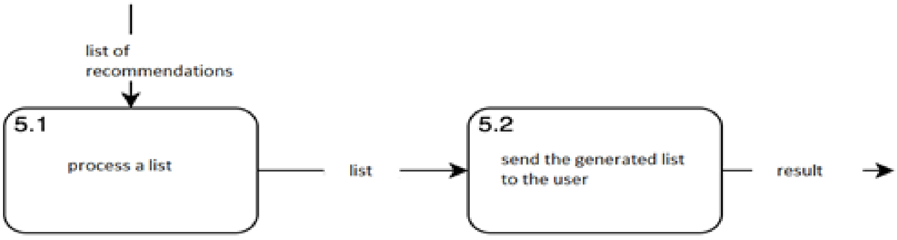


Fig. 11. Detail process "Display user list received".

During the implementation of this section, a comparison is made between the design of the systems, the objectives tree was constructed, and a detailed description of the system and its processes is made.

6 Choice and justification of the means to solve the problem

The most important goal of the system under development was to provide the user with the desired information in a convenient way, using methods of processing natural language. Therefore, the main issues for solving problems were:

1. Method of natural language processing;
2. How to communicate with the chat bot;
3. How to get information to display the results to the user.

Method of natural language processing:

1. The user enters a message;
2. The system verifies whether the message satisfies the conditions for internal processing;
3. If the internal processing failed, the program sends the request to the DialogFlow API;
4. The system receives the DialogFlow API response in the JSON format for further output of the result.

How to communicate with a chat engine:

1. Since the implementation of the bot is carried out in the telegram messenger, respectively, the communication system will be conducted using the Telegram API;
2. Using the API, the system receives the message and sends the necessary data to the next step;
3. The system receives the processing result and sends a message to the user.

Method of delivering information to display results to the user:

1. The system has a list of stores and clothing suppliers and other fashionable products;
2. For each item in the list there is an algorithm implemented for obtaining results with the support of the filter;
3. System services communicate with each other by sending a request, filtering and receiving the result.

After describing the main issues to solve the problems of the system implementation, you can start choosing the technologies for development selected software development tools.

7 Specifications of selected software development tools

7.1 Java programming language and Spring Boot framework

Before starting to work on the project, the choice of programming language fell on Java, considering that Telegram has already been created for this language API, which has accelerated the development process. Java is an object-oriented programming language with its own specification, as well as application development technology. One of the biggest advantages of Java is the independence of the platform (processor). The basis of Java technology is a byte code, that is, the program is not stored in machine codes, but in a special code independent of the platforms. In order for this code to be executed on a specific processor, there is a special virtual java-machine JVM. It's just a special program that translates the byte code into the machine code for a particular processor.

Java Advantages:

- Multiplatform is byte code generated by the Java machine is interpreted by the Java Virtual Machine (JVM), which allows you to use the program regardless of the platform of a particular machine;
- Portability is architectural neutrality and lack of dependencies on the implementation of aspects of specifications;
- Simplicity is in understanding the basic concepts of object-oriented programming, the language will be easy to learn;
- Multithreading is the ability to create programs that will perform a large number of tasks simultaneously.
- Dynamism is Java programming has greater dynamism, compared with others, because it is designed to adapt to changing conditions.

Spring is one of the most popular frameworks for creating applications in the Java language. It is versatile and provides a solution to a large number of tasks encountered by Java developers that create a specific information system. Spring can be viewed as a collection of small frameworks in one framework that can work independently, but they provide greater functionality when combined.

Spring is based on the Inversion of Control (IoC) design template, the main idea of which is to eliminate the dependencies of the components or classes of applications from the specific implementations of auxiliary interfaces and delegate the authority to manage the creation of the necessary implementations of the IoC container.

Main advantages of IoC containers:

- dependency management;
- facilitating the reuse of classes and components;
- simplification of unit testing;
- writing more "clean" code.

The most common type of Inversion of Control is Dependency injection. It helps to provide the object with other objects that it needs, instead of creating them independently. This approach is very useful for testing, since it allows you to model dependencies or make stubs. Dependencies can be introduced in many ways, for example using a constructor or setter-methods.

Spring container is the foundation of the Spring Framework. It creates, connects, configures and manages objects, during their entire life cycle from creation to destruction. The Spring container uses an add-on injection to control the components that make up the application. Objects created in the container are called Spring Beans.

The container receives instructions on which objects to create, read, and configure by reading the configuration metadata. Metadata can be presented as XML, file properties, Java annotations or Java code. The Spring container uses Java classes and configuration metadata to create a fully configured and executable system or application.

7.2 Development environment of IntelliJ IDEA

IntelliJ IDEA is a commercial integrated development environment for various programming languages (Java, Python, Scala, PHP, etc.) from JetBrains. The system comes in the form of a truncated version of the "Community Edition" and the full-featured commercial version of "Ultimate Edition", for which active open source project developers have the opportunity to get a free license. IntelliJ IDEA is one of the most powerful and popular Integrated Development Environment (IDE) for Java.

IntelliJ IDEA has several high-performance features to complete the Java code. Its prediction algorithm can precisely assume that a programmer tries to enter, and completes it for him, even if he does not know the exact name of a particular class, member, or any other resource. Advantages of this development environment:

- IntelliJ IDEA really understands and understands deeply the written code, as well as the context of the program;
- smart code completion - supports the completion of the context code, the ability to list the most appropriate characters used in the current context;
- detection of duplicates - finds duplicate code snippets on the fly and gives notifications / suggestions to the user;
- Inspections and quick fixes - every time the environment detects an error, it reports the user.

IntelliJ IDEA is developed around the principle that developers should be given the opportunity to write code with as little distraction, so a large number of auxiliary tools are integrated into the environment. This toolkit includes version control (Git, SVN, etc.), project development tools (Maven, Gradle, Ant, etc.), testing tools, decompiler, terminal, database communication tools, application servers (Tomcat, WebLogic, GlassFish).

7.3 Tool for natural language processing DialogFlow

Dialogflow (formerly Api.ai, Speaktoit) is a technology of human-computer interaction based on spoken languages. The company is known for creating a virtual assistant for smartphones Android, iOS and Windows Phone, which performs tasks and answers the questions of users in the natural language. Voice and spoken interfaces created with Dialogflow work with a wide range of devices, including phones, cars and other smart devices. It supports 14 languages, including Brazilian Portuguese, Chinese, English, Dutch, French, German, Italian, Japanese, Korean, Portuguese, Russian, Spanish and Ukrainian. Dialogflow supports a range of services related to the entertainment and hospitality industry. Dialogflow also includes an analysis tool that can measure interaction or session metrics, such as usage patterns, delay issues, etc.

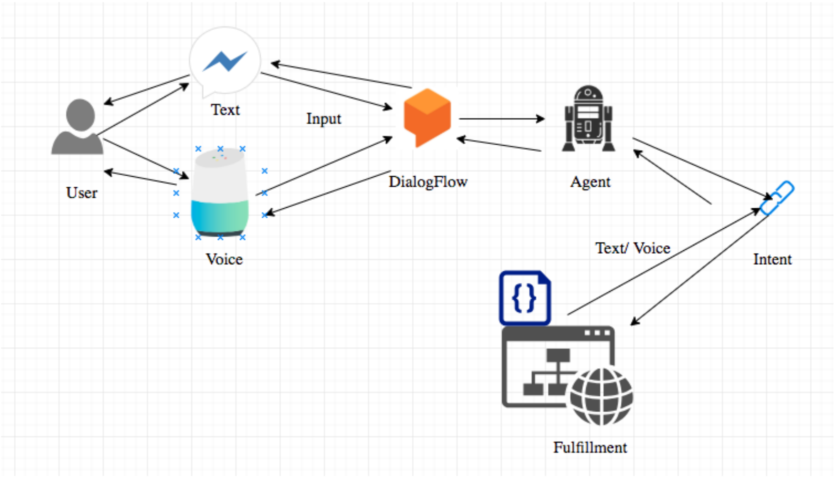


Fig. 12. DialogFlow talk stream

7.4 Hibernate

The conditions for the organization of one-to-many and many-to-many relationship between classes are supported. Hibernate supports displaying the types of eigenvalues. This makes possible the following scenarios:

- redefining the type in the SQL table by default when comparing a column with a property;
- mapping a separate property into several columns.

7.5 Telegram API

Telegram has fully opened the client code. The Telegram architecture implies that both Bot and API have practically the same methods - https://core.telegram.org/methods. In fact, Telegram represents not just a chat messenger, but also a social platform, access to which is open to various applications. They can provide additional privileges to users, instead using a ready-made network of users and a server for delivering messages. Also in the open space, there is a project to facilitate the use of the Telegram API using the Java language - https://github.com/rubenlagus/TelegramBots. This library allows the developer to easily use all available chat management methods in Telegram Messenger.

7.6 Jsoup

Used tools for developing a web-based information system for chat rooms using NLP:

- programming language - Java using the Spring framework;
- development environment - IntelliJ IDEA;
- system for natural language recognition - DialogFlow;
- database management system - MySQL using rendering using MySQL Workbench;
- connection of the system with a chat bot - Telegram API.

8 Software to solve the problem

8.1 Description of the created software

The purpose of the created software product is the implementation of a web application that will receive messages from the chat bot and depending on the content of the request to give the user the desired result. The developed program for the formation of the list of recommendations fashion image of clothing is called CloBot. The bot itself is not part of the system created; it works on the Telegram side and only sends queries to the web application. The system is designed to diversify the purchase of clothing on the Internet. As already mentioned, the chat bot itself can not replace the purpose of an online store, but can diversify the way of communication with the client, which can lead to increased sales and increase interactivity. The program consists of three parts:

- communication with the web-based chat-bot;
- communication of the web application with the system of recognition of natural language;
- the system of formation of the result.

This program does its work only with one type of user - the user of the chat-bot, the person who started the chat. User functionality:

- starts a conversation with the bot;
- sends a message;
- when successful processing of messages, receives a meaningful result in response;
- establishes information about himself;
- view items that are issued by the bot in response to requests.

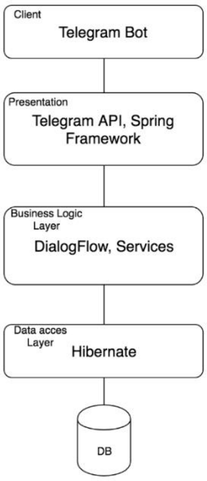
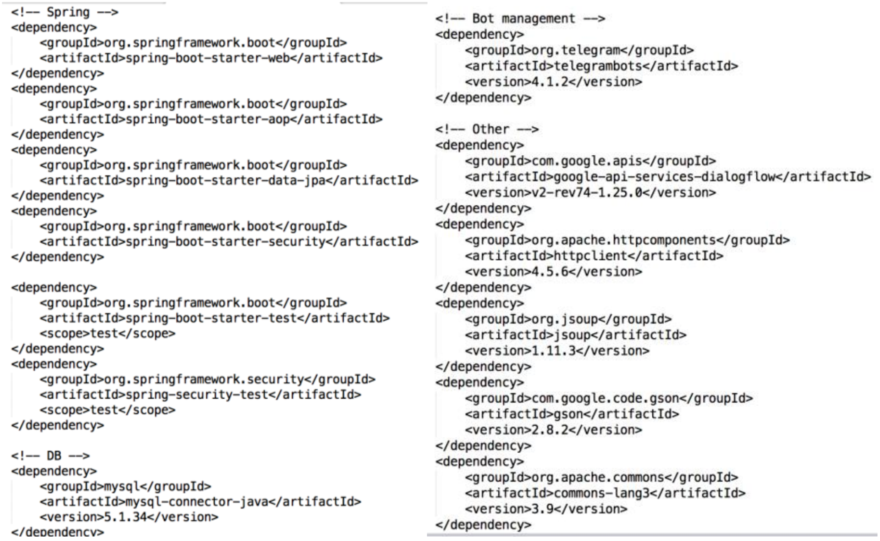


Fig. 13. System architecture

When creating a Web application, the interface interfaces and their implementation were carried out in order to further cover the Unit-tests. Since the project created is Maven-specific, the following libraries were Spring: Web, AOP, Data JPA. Additionally, auxiliary libraries and utilities were added, a complete list of which can be seen in Fig. 14.



```
<!-- Spring -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-aop</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-test</artifactId>
    <scope>test</scope>
</dependency>

<!-- DB -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.34</version>
</dependency>
```

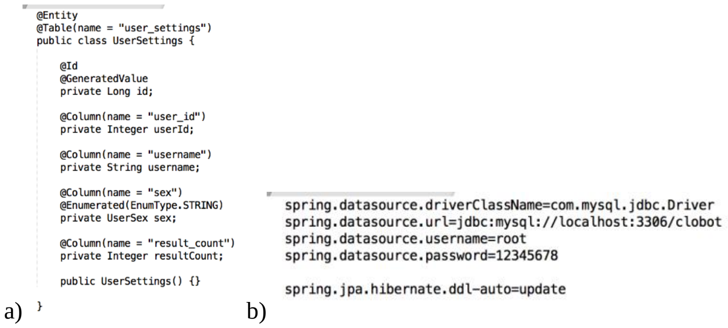
```
<!-- Bot management -->
<dependency>
    <groupId>org.telegram</groupId>
    <artifactId>telegrambots</artifactId>
    <version>4.1.2</version>
</dependency>

<!-- Other -->
<dependency>
    <groupId>com.google.apis</groupId>
    <artifactId>google-api-services-dialogflow</artifactId>
    <version>v2-rev74-1.25.0</version>
</dependency>
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.6</version>
</dependency>
<dependency>
    <groupId>org.jsoup</groupId>
    <artifactId>jsoup</artifactId>
    <version>1.11.3</version>
</dependency>
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.2</version>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.9</version>
</dependency>
```

Fig. 14. The list of dependencies in the pom.xml file

Maven is a compilation automation tool that is used mainly for Java projects.

Maven addresses two aspects of software development: first, he describes how software is created, and, secondly, describes his dependencies. Unlike previous tools, such as Apache Ant, it uses the conditions for the build procedure, and only exceptions should be recorded. The XML file (pom.xml) describes the project being built; its dependence on other external modules and components, the order of compilation, directories and necessary plug-ins. Maven dynamically loads Java libraries and Maven modules from one or more repositories, such as Maven's central repository, and saves them in the local cache. In this project, there is no need for a large database, since most of the information about the user is kept directly in the messengers. User account data is stored in DialogFlow, to avoid more confusion, only one table was created - user preferences using Hibernate, the model class can be seen in Fig. 15a.
Due to the connection of the database, it was necessary to further describe the configuration of the Spring Boot framework as on Fig. 15b.

Thanks to this configuration, the web application receives full access to the MySQL database, and the spring.jpa.hibernate.ddl-auto = update property emphasizes that when the program is launched, a database will be created from existing models. Here is how the UserSettings table looks like by posting an object-relational map:



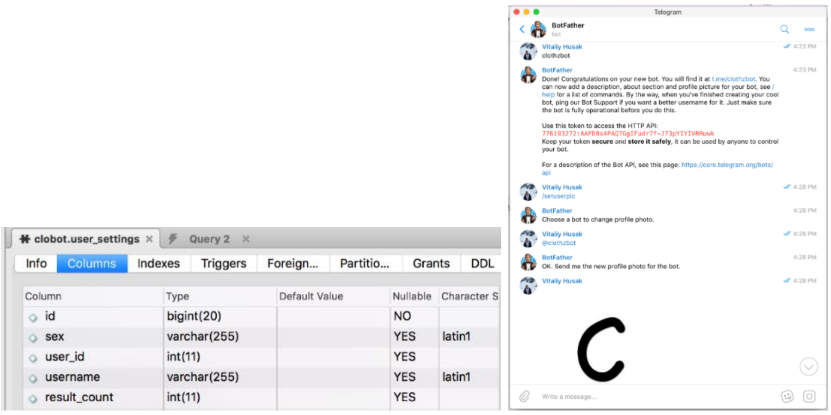
a)

```
@Entity
@Table(name = "user_settings")
public class UserSettings {

    @Id
    @GeneratedValue
    private Long id;

    @Column(name = "user_id")
    private Integer userId;

    @Column(name = "username")
    private String username;

    @Column(name = "sex")
    @Enumerated(EnumType.STRING)
    private UserSex sex;

    @Column(name = "result_count")
    private Integer resultCount;

    public UserSettings() {}

}
```

b)

```
spring.datasource.driverClassName=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/clobot
spring.datasource.username=root
spring.datasource.password=12345678

spring.jpa.hibernate.ddl-auto=update
```

Fig. 15. a) UserSettings class and b) the configuration file is application.properties



Column	Type	Default Value	Nullable	Character S
id	bigint(20)		NO	
sex	varchar(255)		YES	latin1
user_id	int(11)		YES	
username	varchar(255)		YES	latin1
result_count	int(11)		YES	

Vitaliy Husak 4:23 PM
clothzbot

BotFather 4:23 PM
Done! Congratulations on your new bot. You will find it at t.me/clothzbot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
776193272:AAFD8s4PAQ7GgIFudr7f-J73pYIYIVRRuwk
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page: https://core.telegram.org/bots/api

Vitaliy Husak 4:28 PM
/setuserpic

BotFather 4:28 PM
Choose a bot to change profile photo.

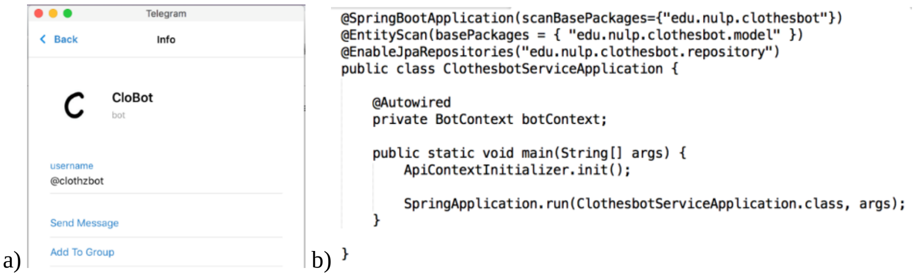
Vitaliy Husak 4:28 PM
@clothzbot

BotFather 4:28 PM
OK. Send me the new profile photo for the bot.

Vitaliy Husak 4:28 PM

Fig. 16. a) Table structure of user_settings and b) the process of creating a chat-bot in Telegram

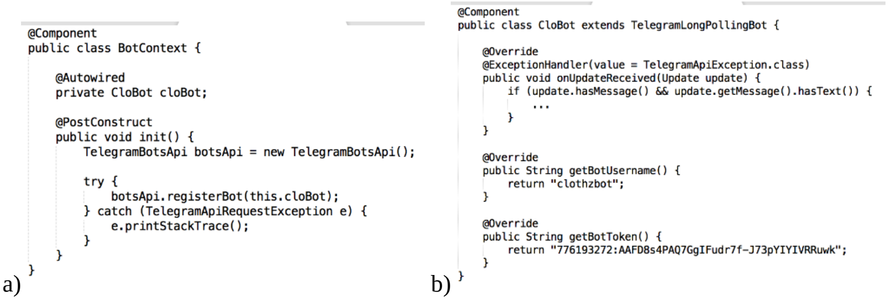
Telegram's bots look like ordinary users with the only difference that their names have the ending "Bot," and the user can start a dialogue with them with the / start command. Since the web application uses Spring Boot, a ClothesbotServiceApplication class is created that will run the program. Launching the program is comprehensive: the context is raised, filled with Java Beans, the database is created and all the services and necessary resources are debugged on embedded tomcat server, which is embedded in the framework.



```
@SpringBootApplication(scanBasePackages={"edu.nulp.clothesbot"})
@EntityScan(basePackages = { "edu.nulp.clothesbot.model" })
@EnableJpaRepositories("edu.nulp.clothesbot.repository")
public class ClothesbotServiceApplication {

    @Autowired
    private BotContext botContext;

    public static void main(String[] args) {
        ApiContextInitializer.init();

        SpringApplication.run(ClothesbotServiceApplication.class, args);
    }
}
```

Fig. 17. a) A created chat bot and b) class ClothesbotServiceApplication with the main method

The ClothesbotServiceApplication initializes the BotContext bin with the @Autowired annotation, the contents of this class are shown in Fig. 18, it provides the creation of a chat bot in the context. Now if you have a chat bot, you need to initialize it in the code of the program. Spring supports the ability to create an entire component - this means that the object responsible for the bot operation will be initialized only once during the lifetime of the web application. This will allow you to receive messages without interruption from all users of the system.



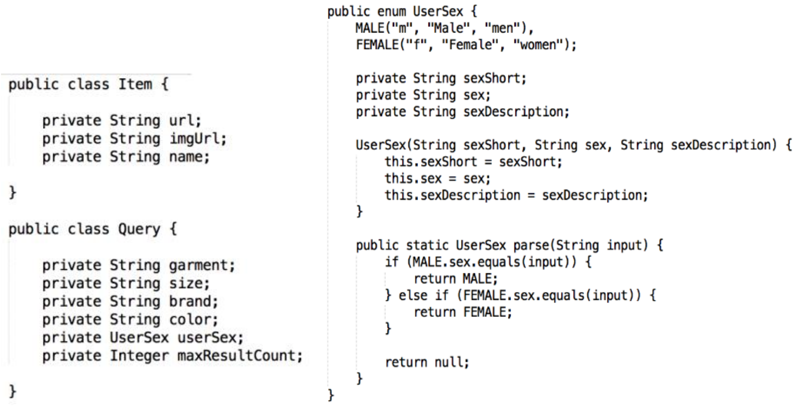
```
@Component
public class BotContext {

    @Autowired
    private CloBot cloBot;

    @PostConstruct
    public void init() {
        TelegramBotsApi botsApi = new TelegramBotsApi();

        try {
            botsApi.registerBot(this.cloBot);
        } catch (TelegramApiRequestException e) {
            e.printStackTrace();
        }
    }
}
```

```
@Component
public class CloBot extends TelegramLongPollingBot {

    @Override
    @ExceptionHandler(value = TelegramApiException.class)
    public void onUpdateReceived(Update update) {
        if (update.hasMessage() && update.getMessage().hasText()) {
            ...
        }
    }

    @Override
    public String getBotUsername() {
        return "clothzbot";
    }

    @Override
    public String getBotToken() {
        return "776193272:AAFD8s4PAQ7GgIFudr7f-J73pYIYIVRRuwk";
    }
}
```

Fig. 18. a) BotContext class and b) basic initialization of the chat-bot class

To transfer data between services, it is necessary to announce several POJO (Plain Old Java Object). Including:

- • Item is information about the resultant item;
- • Query is information about the user's request;
- • UserSex is the listed data type, used to determine user statistics, in UserSettings.



```
public class Item {

    private String url;
    private String imgUrl;
    private String name;

}

public class Query {

    private String garment;
    private String size;
    private String brand;
    private String color;
    private UserSex userSex;
    private Integer maxResultCount;

}
```

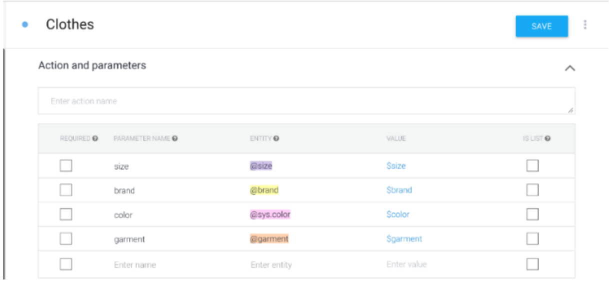
```
public enum UserSex {
    MALE("m", "Male", "men"),
    FEMALE("f", "Female", "women");

    private String sexShort;
    private String sex;
    private String sexDescription;

    UserSex(String sexShort, String sex, String sexDescription) {
        this.sexShort = sexShort;
        this.sex = sex;
        this.sexDescription = sexDescription;
    }

    public static UserSex parse(String input) {
        if (MALE.sex.equals(input)) {
            return MALE;
        } else if (FEMALE.sex.equals(input)) {
            return FEMALE;
        }

        return null;
    }
}
```

Fig. 19. Declared POJO classes

Before you start publishing a particular result to a user, you need to do two things: run the system to process queries and write a service to find the result.

The DialogFlow system was chosen for query processing, it allows you to parse the request and find out if it has the necessary information.

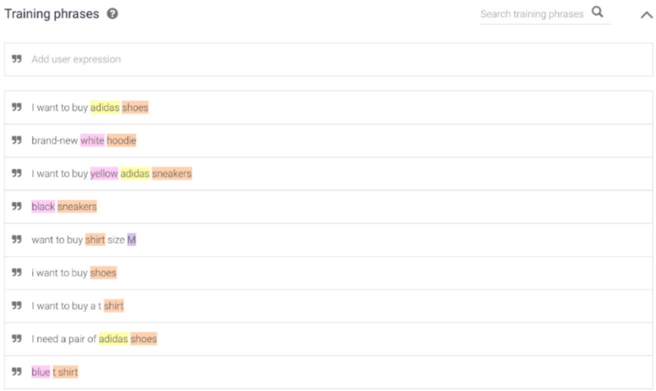
Dialogflow uses intentions to categorize user intentions. Intents have Training phrases, which are examples of what the user can say to the agent. For example, a user might say "Adidas shoes", "I need a pair of Adidas shoes", or just say "Adidas shoes". All these inquiries are unique, but have the same intention: to find the Adidas sneakers.



REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	size	@size	$size	□
□	brand	@brand	$brand	□
□	color	@sys.color	$color	□
□	garment	@garment	$garment	□
□	Enter name	Enter entity	Enter value	□

Fig. 20. Designed intention Clothes with options

Parameters, such as size, which describe the size of the sought-after clothing or shoes, are added also to the intention. Parameters are important and relevant words or phrases in the user's request that are extracted from it so that the agent can provide the appropriate response. Now you need to add some training phrases. Dialogflow uses training phrases as examples for a Machine Learning model to extract information from inquiries using intents. The Machine Training model checks the request for each agent's intention, gives each rating a rating, and selects the int for the highest score. If the highest scoring score has a very low score, then the appropriate backup is selected. To ensure better accuracy of phrases recognition, it is necessary to increase the list of training phrases. The initial state of the system with a certain set of training phrases is shown in Fig. 21.

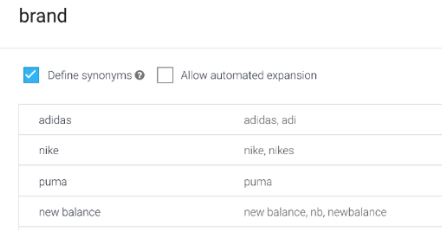


Add user expression

- I want to buy adidas shoes
- brand-new white hoodie
- I want to buy yellow adidas sneakers
- black sneakers
- want to buy shirt size M
- i want to buy shoes
- I want to buy a t shirt
- I need a pair of adidas shoes
- blue t shirt

Fig. 21. List of training phrases

DialogFlow also describes each parameter, which is called entity. These objects allow you to display synonyms to the main value. For example, to name the brand 'new balance' the user can write 'new balance' or 'nb', this is what is described at this stage.



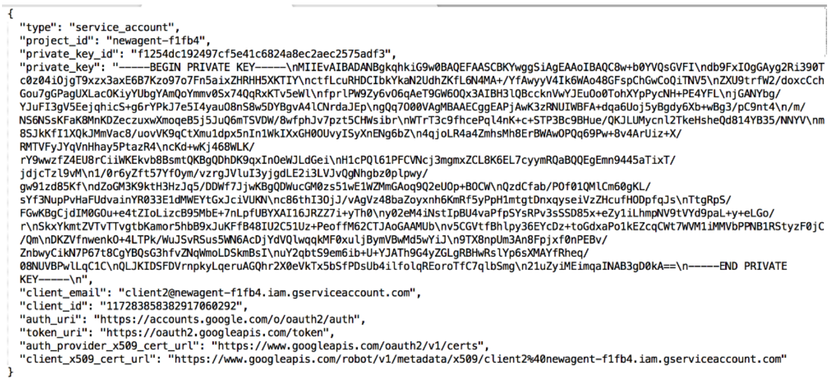
brand

☑ Define synonyms □ Allow automated expansion

adidas	adidas, adi
nike	nike, nikes
puma	puma
new balance	new balance, nb, newbalance

Fig. 22. Description of the synonyms for the 'brand' essence

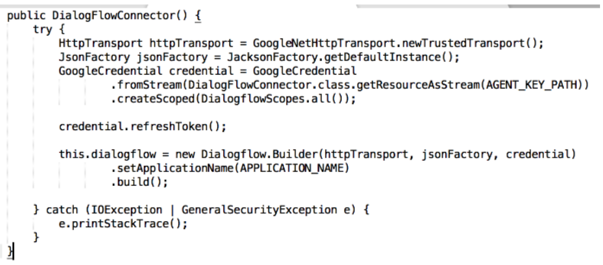
The system will contact DialogFlow using the API for which the client library is written in Java. You must take the service access key and configure the system to send queries to the DialogFlow server.



```
{
  "type": "service_account",
  "project_id": "newagent-f1fb4",
  "private_key_id": "f1254dc192497cf5e41c6824a8ec2aec2575adf3",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQC8w+b0YVQsGVFI\ndb9FxIOgGAyg2Ri390T
  c0z04iOjgT9xzx3axE6B7Kzo97o7Fn5aixZHRHH5XKTIY\nctfLcuRHDCIbkYkaN2UdhZKfL6N4MA+/YfAwyyV4Ik6WAo48GFspChGwCoQiTNV5\nZXU9trfW2/doxcCch
  Gou7gGPagUXLacOKiyYUbgYAmQoYmmv0Sx74QqRxKTv5eWl\nfprlPW9Zy6vO6qAeT9GW6OQx3AIBH3lQBccknVwYJEuOo0TohXYpPycNH+PE4YFL\njGANYbg/
  YJuFI3gV5EejqhicS+g6rYPkJ7e5I4yauO8nS8w5DYBgvA4lCNrdaJEp\ngQq7O00VAgMBAAECggEAPjAwK3zRNUIWBFA+dqa6Uoj5yBgdy6Xb+wBg3/pC9nt4\n/m/
  NS6NSsKFaK8MnKDZeczuxwXmoqeB5j5JuQ6mTSVDW/8wfphJv7pzt5CHWsibr\nWTrT3c9fhcePql4nK+c+STP3Bc9BHue/QKJLUMycnl2TkeHsheQd814YB35/NNYV\nm
  8SJkKfI1XQkJMmVac8/uovVK9qCtXmu1dpx5nIn1WkIXxGH0OUvyISyXnENg6bZ\n4qjoLR4a4ZmhsMh8ErBWAwOPQq69Pw+8v4ArUiz+X/
  RMTVFyJYqVnHhay5PtazR4\ncKd+wKj468WLK/
  rY9wwzfZ4EU8rCiiWKEkvb8BsmtQKBgQDhDK9qxInOeWJLdGei\nH1cPQl61PFCVNcj3mgmxZCL8K6EL7cyymRQaBQQEgEmn9445aTixT/
  jdjcTzl9vM\n1/0r6yZft57YfOym/vzrgJVluI3yjgdLE2i3LVJvQgNhgbz0plpwy/
  gw91zd85Kf\ndZoGM3K9ktH3HzJq5/DDWf7JjwKBgQDWucGM0zs51wE1WZMmGAoq9Q2eUOp+BOCW\nQzdCfab/POf01QMlCm60gKL/
  sYf3NupPvHaFUdvainYR033E1dMWEYtGxJciVUKN\nc86thI3OjJ/vAgVz48baZoyxnh6KmRf5yPpH1mtgtDnxqyseiVzZHcufHODpfqJs\nTtgRpS/
  FGwKBgCjdIM0GOu+e4tZIoLizcB95MbE+7nLpfUBYXAI16JRZZ7i+yTh0\ny02eM4iNstIpBU4vaPfpSYsRPv3sSSD85x+eZy1iLhmpNV9tVYd9paL+y+eLGo/
  r\nSkxYkmtZVTvTTvgtbKamor5hbB9xJuKFfB48IU2C51Uz+PeoffM62CTJAoGAAMUb\nv5CGVtfBhlpy36EYcDz+toGdxaPo1kEZcqCWt7WVM1iMMVbPPNB1RStyzF0jC
  /Qm\nDKZVfnwenkO+4LTPk/WuJSvRSus5WN6AcDjYdVQlwqqkMF0xuljBymVBwMd5wYiJ\n9TX8npUm3An8Fpjxf0nPEBv/
  ZnbwyCikN7P67t8CgYBQsG3hfvZNqWmoLDSkmBsI\nuY2qbtS9em6ib+U+YJATh9G4yZGLgRBHwRslYp6sXMAYfRheq/
  08NUVBPwlLqC1C\nQLJKIDSFDVrnpkyLqeruAGQhr2X0eVkTx5bSfPDsUb4ilfolqREoroTfC7qlbSmg\n21uZyiMEimqaINAB3gD0kA==\n-----END PRIVATE
  KEY-----\n",
  "client_email": "client2@newagent-f1fb4.iam.gserviceaccount.com",
  "client_id": "117283858382917060292",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/client2%40newagent-f1fb4.iam.gserviceaccount.com"
}
```

Fig. 23. DialogFlow access key

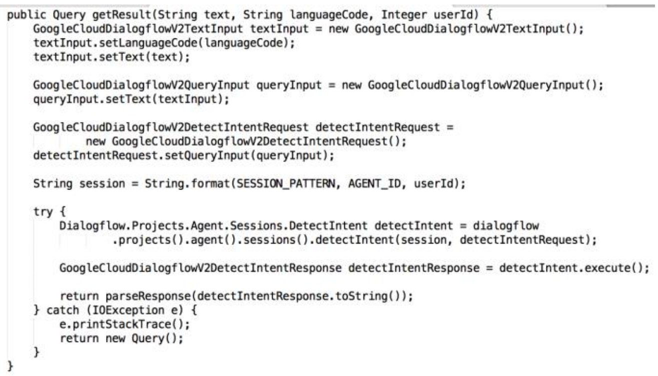
Connection to the DialogFlow service is described in the DialogFlowConnector class. The connection to the service opens to get started, as shown in Fig. 24.



```
public DialogFlowConnector() {
    try {
        HttpTransport httpTransport = GoogleNetHttpTransport.newTrustedTransport();
        JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();
        GoogleCredential credential = GoogleCredential
                .fromStream(DialogFlowConnector.class.getResourceAsStream(AGENT_KEY_PATH))
                .createScoped(DialogflowScopes.all());

        credential.refreshToken();

        this.dialogflow = new Dialogflow.Builder(httpTransport, jsonFactory, credential)
                .setApplicationName(APPLICATION_NAME)
                .build();

    } catch (IOException | GeneralSecurityException e) {
        e.printStackTrace();
    }
}
```

Fig. 24. Open a connection to the service

Already during work, the program uses a connection to get the result as a Query object described above.



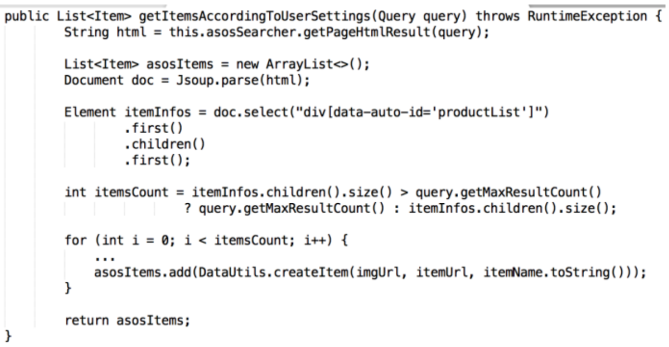
```
public Query getResult(String text, String languageCode, Integer userId) {
    GoogleCloudDialogflowV2TextInput textInput = new GoogleCloudDialogflowV2TextInput();
    textInput.setLanguageCode(languageCode);
    textInput.setText(text);

    GoogleCloudDialogflowV2QueryInput queryInput = new GoogleCloudDialogflowV2QueryInput();
    queryInput.setText(textInput);

    GoogleCloudDialogflowV2DetectIntentRequest detectIntentRequest =
            new GoogleCloudDialogflowV2DetectIntentRequest();
    detectIntentRequest.setQueryInput(queryInput);

    String session = String.format(SESSION_PATTERN, AGENT_ID, userId);

    try {
        Dialogflow.Projects.Agent.Sessions.DetectIntent detectIntent = dialogflow
                .projects().agent().sessions().detectIntent(session, detectIntentRequest);

        GoogleCloudDialogflowV2DetectIntentResponse detectIntentResponse = detectIntent.execute();

        return parseResponse(detectIntentResponse.toString());
    } catch (IOException e) {
        e.printStackTrace();
        return new Query();
    }
}
```

Fig. 25. Sending a request to the DialogFlow side and getting the result

The system has two resources connected to the clothing that might be of interest to users - Adidas and ASOS clothing stores. Due to the fact that ASOS has a large number of products of different brands, it is possible to provide the user with a wide choice.

Communication with these stores takes place in the AsosItemService and AdidasItemService classes, depending on the user's request, the getItemsAccordingToUserSettings method calls one of these classes to get a list of things.

Using jsoup, an html-page is processed from which the necessary data is extracted to be displayed in the messenger chat-bot.

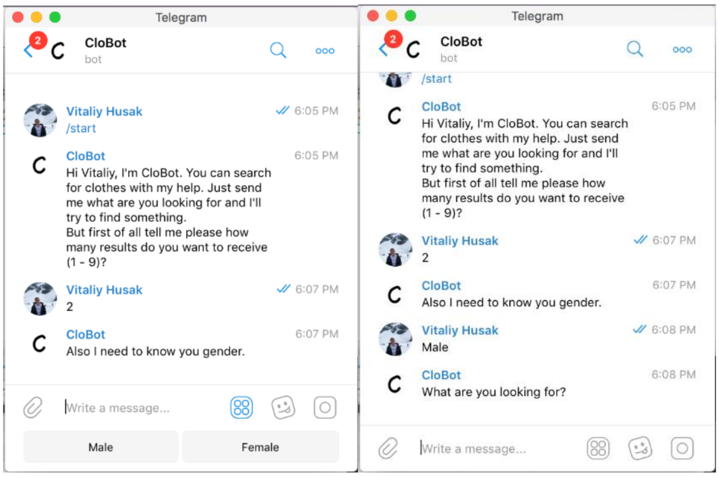


```java
public List<Item> getItemsAccordingToUserSettings(Query query) throws RuntimeException {
        String html = this.asosSearcher.getPageHtmlResult(query);

        List<Item> asosItems = new ArrayList<>();
        Document doc = Jsoup.parse(html);

        Element itemInfos = doc.select("div[data-auto-id='productList']")
                .first()
                .children()
                .first();

        int itemsCount = itemInfos.children().size() > query.getMaxResultCount()
                        ? query.getMaxResultCount() : itemInfos.children().size();

        for (int i = 0; i < itemsCount; i++) {
            ...
            asosItems.add(DataUtils.createItem(imgUrl, itemUrl, itemName.toString()));
        }

        return asosItems;
}
```

Fig. 26. An example of implementing the getItemsAccordingToUserSettings method

8.2 Analysis of the control sample

To start working with a bot in Telegram, you need to find it by the name @clothzbot. The correspondence begins with the / start command, on which the user receives bot information and the following instructions to get started.



Vitaliy Husak 6:05 PM
/start

CloBot 6:05 PM
Hi Vitaliy, I'm CloBot. You can search for clothes with my help. Just send me what are you looking for and I'll try to find something.
But first of all tell me please how many results do you want to receive (1 - 9)?

Vitaliy Husak 6:07 PM
2

CloBot 6:07 PM
Also I need to know you gender.

/start

CloBot 6:05 PM
Hi Vitaliy, I'm CloBot. You can search for clothes with my help. Just send me what are you looking for and I'll try to find something.
But first of all tell me please how many results do you want to receive (1 - 9)?

Vitaliy Husak 6:07 PM
2

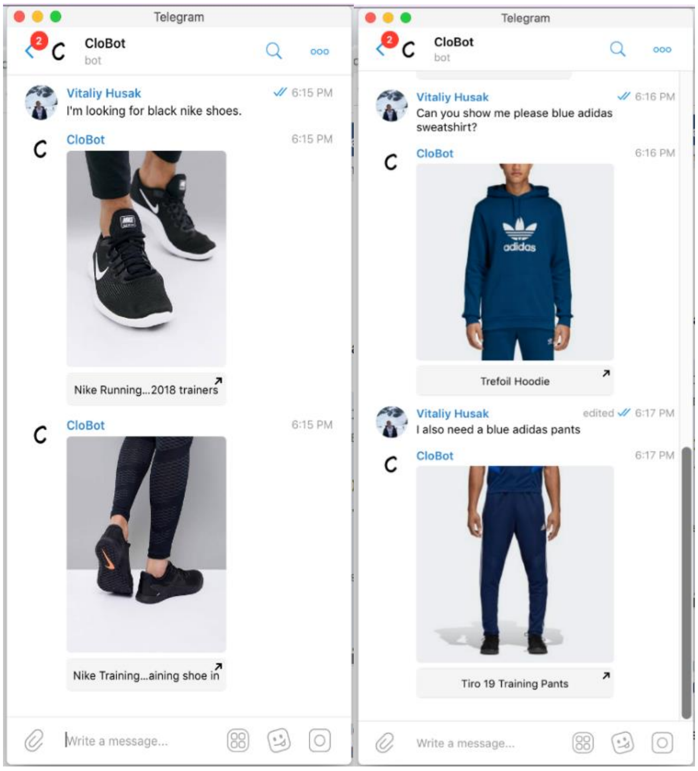
CloBot 6:07 PM
Also I need to know you gender.

Vitaliy Husak 6:08 PM
Male

CloBot 6:08 PM
What are you looking for?

Fig. 27. Beginning communication

After the bot has received the necessary information about the user, it is possible to make a specific request.



Vitaliy Husak 6:15 PM
I'm looking for black nike shoes.

CloBot 6:15 PM
Nike Running...2018 trainers

CloBot 6:15 PM
Nike Training...aining shoe in

Vitaliy Husak 6:16 PM
Can you show me please blue adidas sweatshirt?

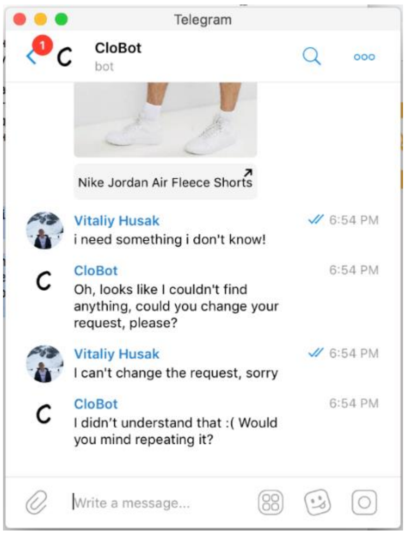
CloBot 6:16 PM
Trefoil Hoodie

Vitaliy Husak edited 6:17 PM
I also need a blue adidas pants

CloBot 6:17 PM
Tiro 19 Training Pants

Fig. 28. Example chat bots answers

If the user fails to request or it is not possible to remove any information from his message, he will receive the message as pictured in Fig. 29.



Nike Jordan Air Fleece Shorts

Vitaliy Husak 6:54 PM
i need something i don't know!

CloBot 6:54 PM
Oh, looks like I couldn't find anything, could you change your request, please?

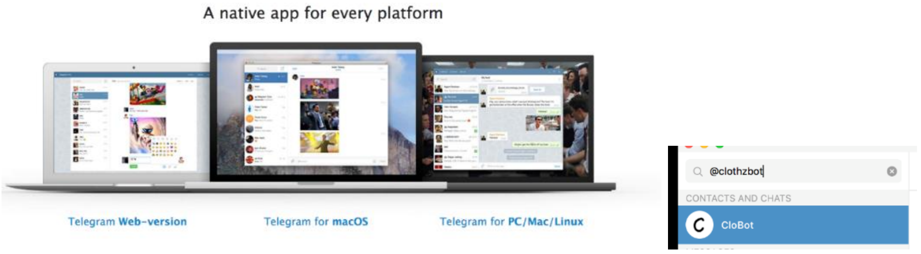
Vitaliy Husak 6:54 PM
I can't change the request, sorry

CloBot 6:54 PM
I didn't understand that :(Would you mind repeating it?

Fig. 29. Answers to incorrect user queries

8.3 Instruction for the user

First of all, the user needs to install Telegram on any available device: mobile phone or personal computer. Then in the search box you must enter the name of the chat bot - @clothzbot and select the bot named CloBot from the list as shown in Fig. 30.



@clothzbot

CloBot

Fig. 30. a) Download Telegram and b) search chat bot

In the chat that opens the chat starts with the / start command. The user provides the bot with information about himself and can continue to send requests for example:

- I'm looking for black nike shoes.
- I need Air Jordan t-short size M.

9 Conclusions

In this work the basic functions for the information system of forming the list of recommendations of the fashion style clothes according to the needs of the user using the NLP and chat bots are developed. The debuted software provides users with another way to search and select branded things in their own messenger, which in some cases can help increase sales to businesses as additional verticals.

Characteristics of the created system:

- An application that operates around the clock, if it is running on the server;
- Narrow selection of results;
- The ability to study the system for better recognition of phrases.

The system in the Java programming language is created. The use of natural language processing has increased the level of recognition of the user's language. The architecture of the system is constructed in such a way that the developer in the future can expand the functionality and create test cases. Expansion options include the addition of support for a new messenger such as Facebook Messenger or Viber - it is enough to describe the functionality using the APIs available for this. It is also possible to expand the range by adding support for new online stores and brands. This system is functionally complete - it can perform the tasks and can have practical application. In the future, the system can be improved and expanded to realize a larger range of tasks.

However, the chat bot is smarter and their capabilities grow, while online chat shops are usually an add-on that performs certain functions. There are no full-fledged online chats, which represent American or European companies. Although in the Chinese market, most entrepreneurs are engaged in Internet sales through the application for messaging WeChat exactly where the common chat bots are. The reason that chat bots can not replace sites or add-ons for online stores is the ability to design - some important functions are difficult to adequately migrate to the chat.

References

1. Iturrioz, J., Azpeitia, I., Díaz, O.: Generalizing the like button: empowering websites with monitoring capabilities. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing, 743-750. (2014).
2. Maggi, F., Robertson, W., Kruegel, C., Vigna, G.: Protecting a moving target: Addressing web application concept drift. In: International Workshop on Recent Advances in Intrusion Detection, Springer, Berlin, Heidelberg, 21-40. (2009).
3. Spring Boot Reference Guide / Spring – 2017. https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/
4. Craig Walls - Spring in Action, Fourth Edition, 2017 624 p.
5. Pawlak, Z.: Rough sets: Theoretical aspects of reasoning about data. In: Springer Science & Business Media, 9. (2012)
6. Eckel, Bruce. Thinking in Java / Bruce Eckel.—4th ed., 2006 1079 p.
7. Taming Text: How to Find, Organise, and Manipulate it / Andrew L. Farris, Grant S. Ingersoll, Thomas S. Morton; 2013, 320 p.

8. Christensen, J. H.: Using RESTful web-services and cloud computing to create next generation mobile applications. In: the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, 627-634. (2009).
9. Neugschwandtner, M., Neugschwandtner, G., Kastner, W.: Web services in building automation: Mapping knx to obix. In: Int. Conf. on Industrial Informatics, 87-92. (2007).
10. Tkachenko, R., Izonin, I., Kryvinska, N., Chopyak, V., Lotoshynska, N., Danylyuk, D.: Piecewise-linear Approach for Medical Insurance Costs Prediction using SGTM Neural-Like Structure. In: CEUR Workshop Proceedings, 170-179. (2018)
11. Tepla, T., Izonin, I., Duriagina, Z., Tkachenko, R., Trostianchyn, A., Lemishka, I., Kulyk V., Kovbasyuk, T.: Alloys selection based on the supervised learning technique for design of biocompatible medical materials. In: Archives of Materials Science and Engineering, 93/1, 32-40. (2018)
12. Vysotska, V., Lytvyn, V., Burov, Y., Gozhyj, A., Makara, S.: The consolidated information web-resource about pharmacy networks in city. In: CEUR Workshop Proceedings, 239-255 (2018)
13. Zdebskyi, P., Vysotska, V., Peleshchak, R., Peleshchak, I., Demchuk, A., Krylyshyn, M.: An Application Development for Recognizing of View in Order to Control the Mouse Pointer. In: CEUR Workshop Proceedings, Vol-2386, 55-74. (2019)
14. Rusyn, B., Lytvyn, V., Vysotska, V., Emmerich, M., Pohreliuk, L.: The Virtual Library System Design and Development, Advances in Intelligent Systems and Computing, 871, 328-349 (2019)
15. Rusyn, B., Lutsyk, O., Lysak, O., Lukeniuk, A., Pohreliuk, L.: Lossless Image Compression in the Remote Sensing Applications. In: Int. Conf. on Data Stream Mining & Processing (DSMP), 195-198 (2016)
16. Emmerich, M., Lytvyn, V., Yevseyeva, I., Fernandes, V. B., Dosyn, D., Vysotska, V.: Preface: Modern Machine Learning Technologies and Data Science (MoMLeT&DS-2019). In: CEUR Workshop Proceedings, Vol-2386. (2019)
17. Rusyn, B., Vysotska, V., Pohreliuk, L.: Model and architecture for virtual library information system. In: Computer Sciences and Information Technologies, CSIT, 37-41 (2018)
18. Vysotska, V., Burov, Y., Lytvyn, V., Demchuk, A.: Defining Author's Style for Plagiarism Detection in Academic Environment, Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, 128-133 (2018)
19. Lytvyn, V., Peleshchak, I., Vysotska, V., Peleshchak, R.: Satellite spectral information recognition based on the synthesis of modified dynamic neural networks and holographic data processing techniques. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 330-334 (2018)
20. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, 113-116 (2018)
21. Lytvyn, V., Sharonova, N., Hamon, T., Cherednichenko, O., Grabar, N., Kowalska-Styczen, A., Vysotska, V.: Preface: Computational Linguistics and Intelligent Systems (COLINS-2019). In: CEUR Workshop Proceedings, Vol-2362. (2019)
22. Burov, Y., Vysotska, V., Kravets, P.: Ontological Approach to Plot Analysis and Modeling. In: CEUR Workshop Proceedings, Vol- 2362, 22-31. (2019)
23. Vysotska, V., Lytvyn, V., Burov, Y., Berezin, P., Emmerich, M., Basto Fernandes V.: Development of Information System for Textual Content Categorizing Based on Ontology. In: CEUR Workshop Proceedings, Vol- 2362, 53-70. (2019)
24. Lytvyn, V., Vysotska, V., Kuchkovskiy, V., Bobyk, I., Malanchuk, O., Ryshkovets, Y., Pelekh, I., Brodyak, O., Bobrivetc, V., Panasyuk, V.: Development of the system to inte-

grate and generate content considering the cryptocurrent needs of users. In: Eastern-European Journal of Enterprise Technologies 1(2-97), 18-39 (2019)

25. Lytvyn, V., Kuchkovskiy, V., Vysotska, V., Markiv, O., Pabyrivskyy, V.: Architecture of system for content integration and formation based on cryptographic consumer needs. In: Computer Sciences and Information Technologies, CSIT, 391-395 (2018)
26. Lytvyn, V., Vysotska, V., Demchuk, A., Demkiv, I., Ukhanska, O., Hladun, V., Kovalchuk, R., Petruchenko, O., Dzyubyk, L., Sokulska, N.: Design of the architecture of an intelligent system for distributing commercial content in the internet space based on SEO-technologies, neural networks, and Machine Learning. In: Eastern-European Journal of Enterprise Technologies, 2(2-98), 15-34. (2019)
27. Veres, O., Rishnyak, I., Rishniak, H.: Application of Methods of Machine Learning for the Recognition of Mathematical Expressions. In: CEUR Workshop Proceedings, Vol- 2362, 378-389. (2019)
28. Lytvyn, V., Vysotska, V., Rusyn, B., Pohreliuk, L., Berezin, P., Naum O.: Textual Content Categorizing Technology Development Based on Ontology. In: CEUR Workshop Proceedings, Vol-2386, 234-254. (2019)
29. Lytvyn, V., Vysotska, V., Dosyn, D., Lozynska, O., Oborska, O.: Methods of Building Intelligent Decision Support Systems Based on Adaptive Ontology. In: International Conference on Data Stream Mining and Processing, DSMP 2018, 145-150 (2018)
30. Basyuk, T.: The main reasons of attendance falling of internet resource. In: Proc. of the X-th Int. Conf. Computer Science and Information Technologies, CSIT'2015, 91-93. (2015).
31. Lytvyn, V., Vysotska, V., Rzheuskyi, A.: Technology for the Psychological Portraits Formation of Social Networks Users for the IT Specialists Recruitment Based on Big Five, NLP and Big Data Analysis. In: CEUR Workshop Proceedings, 2392, 147-171. (2019)
32. Vysotska, V., Burov, Y., Lytvyn, V., Oleshek, O.: Automated Monitoring of Changes in Web Resources. In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 348–363. (2020)
33. Demchuk, A., Lytvyn, V., Vysotska, V., Dilai, M.: Methods and Means of Web Content Personalization for Commercial Information Products Distribution. In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 332–347. (2020)
34. Vysotska, V., Mykhailyshyn, V., Rzheuskyi, A., Semianchuk, S.: System Development for Video Stream Data Analyzing. In: In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 315-331. (2020)