

DOI: <https://doi.org/10.15276/hait.05.2022.18>

UDC 004.67:004.891.2:330.322

Application of chat bots for personalized financial advice

Vitaliy M. Kobets¹⁾ORCID: <https://orcid.org/0000-0002-4386-4103>; vkobets@kse.org.ua. Scopus Author Id: 56006224700Kyrylo H. Kozlovskiy¹⁾ORCID: <https://orcid.org/0000-0003-3505-9902>; b4559667@gmail.com¹⁾ Kherson State University. 27, Universitetska St. Kherson, 73003, Ukraine

ABSTRACT

Chat bots are web-based systems that have recently received a lot of attention due to their ability to provide professional financial advice at a lower cost than a human advisor. We considered anthropomorphic design of chat bot as a positive impact on human perception, which is an important factor in human-computer interaction, because increasing the level of anthropomorphism of the system can potentially compensate for the lack of contact with a person. The influence of the anthropomorphism effect and anthropomorphic design of the bot on the feeling of social presence, trust, and the desire to follow recommendations are investigated. High cost of financial advice, the volatility of markets and problems with finding professional financial advisors have created favourable conditions for the use of robo-advisors. Due to the growing popularity of robo-advisors and chat bots, new opportunities appeared for banks for attracting new customers. There are a large number of chat bots that can provide financial advice, but usually, these are complex solutions and they require quite a lot of time to work with them, so if the user's goal is to get a quick and specific answer, such solutions are not the best option. The goal of our research is to develop a Telegram chat bot for personalized financial advice. To meet this goal we need consider examples of chat bots advisors for potential investors to analyse available software solutions for financial advisors and create our own software product for personalized financial advice. Research methods include analytical mathematical methods and technologies, such as Python, Telegram Bot API, Aioogram, Docker, Amazon Web Services, EC2, and Ubuntu Server. We created our own software product, a telegram bot for automated financial consultations to achieve the goals of private investors by calculating constant level of consumption during determined period under expected inflation rate to decide whether invest or not. Our telegram bot provides option for receiving financial advice and helping potential investors with capital distribution.

Keywords: Chat bots; robo-advisor; anthropomorphic design; investors

For citation: Kobets V. M., Kozlovskiy K. H. Application of chat bots for personalized financial advice. *Herald of Advanced Information Technology*. 2022; Vol. 5 No. 3: 229–242. DOI: <https://doi.org/10.15276/hait.05.2022.18>

INTRODUCTION. PROBLEM STATEMENT

Nowadays, when computer technologies are developing rapidly and IT is an integral part of our lives, chat bots have become an everyday thing. Now it is difficult to find a person who would never use chat bots, because they are able to take their place in many areas and provide a wide range of services. Almost without any effort, the user can get financial advice, get information about his/her bank account, and even find help for using the services.

There are a large number of chat bots that can provide financial advice, but usually, these are complex solutions and they require quite a lot of time to work with them, so if the user's goal is to get a quick and specific answer, such solutions are not the best option.

Investors who believe that AI is more capable than human intelligence are more likely to adopt algorithmic advice of robo-advisor (RA) [1]. ‘The majority of consumers still express a preference for human financial advisors, because of RAs’ lack of a “human touch” and a human ability to understand and personalize investment advice to the consumers unique financial situation’ [2]. Decision support tool depends greatly on its usability and unwillingness to engage manage investment questions with RA [3].

Financial mistakes of inexperienced investors, ‘who are the largest part of the population, vary for different individuals (e.g., low statistical skills) and in different situations (e.g. stress, cognitive overload etc.)’, thus, inexperienced investors need good decision support [4]. Discretion is investors’ ability to override robo-advisors’ recommendation [5]. Clusters of investors can be determined using investors’ characteristics [6].

© Kobets V., Kozlovskiy K., 2022

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/deed.uk>)

The architecture of a Robo-adviser based on open data about cryptocurrency is developed in papers [7, 8], [9, 10].

Each person's income changes over time. Circumstances of a change of job, place of residence, or health condition arise that require reserve funds to equalize the level of an individual's income. The lack of permanent income or the presence of unstable income for an individual, the need for significant expenses for a human consultant to get investment advice and the lack of access to RA in Ukraine, generate the need for an affordable and quick tool that could provide an opportunity for a person to stabilize the level of his/her expenses over time. To meet this purpose, we have developed a model of constant consumption of a person with unstable incomes, which allows dividing his/her income between consumption and savings in order to determine a stable level of consumption throughout life, depending on the level of inflation, which affects the feasibility or impracticality of savings program. Inflation, which reduces the solvency of an individual, requires a review of one's own decisions: whether to save or invest immediately in tangible assets. If the return on savings corresponds to the minimum level of an individual's income, it is advisable to invest in current tangible assets. On the other hand, low inflation contributes to the fact that savings increase the future level of individual consumption.

The **goal** of our research is to develop a chat bot for personalized financial advice to distribute income of person between consumptions and savings to achieve constant level of consumption during long life period under expected level of inflation rate.

To our best knowledge, there are currently no analogues of such a model that would allow making decisions based on the consumer price index and the income of individuals that will help an individual make a rational decision depending on his/her income level, investment horizon and inflation rate. The spread of such a model of a constant level of consumption throughout life is realized by the development of a chat bot.

To implement this task we need consider examples of chat bots advisors for potential investors analyse available software solutions for financial advisors and create our own software product for personalized financial advice.

Research methods includes mathematical model of constant consumption throughout life using technologies: Python, Telegram Bot API, Aiogram, Docker, Amazon Web Services, EC2, Ubuntu Server.

The structure of our paper as follows: in section 2, we consider related works, in section 3, we analyze software products for financial advice; in section 4, we build mathematical model and the architecture of our chat bot; in section 5, we justify the application of our chat bot; finally, we present our conclusions.

RELATED WORKS

Advisor bots are web-based systems that have recently received a lot of attention due to their ability to provide professional financial advice at a lower cost than a human advisor. Each bot uses different implementation options for financial information analysis algorithms. However, process automation of financial advice as full digital services often goes hand in hand with the loss of human contact and a decrease in trust.

In a recent study, users were asked to talk with a human advisor before making an investment using a bot advisor [11]. It is interesting that the user's goal was not to get more information, but to make sure of the reliability of the chat bot, using advice from a human consultant. That is why personal contact takes an important place in decision-making and contributes to the creation of a trust relationship between the user and the chat bot in the subsequent adoption of investment advice. However, it is not clear whether a human advisor is actually necessary, or whether a more human-like design of a bot advisor can compensate for the lack of human contact, and have a positive impact on the trust and lead the user to accept the recommendations.

To accept chat bot's advice by a person, anthropomorphism is used, that is, the assimilation of anything that is not a person to a person or the transfer of its physical and intellectual properties to creatures, things and phenomena of the surrounding world. Anthropomorphism refers to the external assimilation to a person – in proportions, form, appearance features and attributes, as well as by motivations, intentions or emotions. Thanks to anthropomorphism, a person understands and develops the external world with which he/she interacts [12]. Anthropomorphism of products is a popular marketing strategy. Recently, the effect of anthropomorphism in a commercial context has attracted considerable interest from scientists. Previous research has linked anthropomorphism to preferences for product design, consequences of product failures, and trust in advertising messages [13]. Usually, when interacting with user interfaces, the effect of anthropomorphism appears due to anthropomorphic signals. Because it provides an

opportunity to influence the IT user, software and hardware developers try to apply anthropomorphic features and design to give people a “familiar feeling” and promote the effect of social presence when working with an information system. This approach to design makes it easier to familiarize yourself with the features and further use of the software. One of the main aims of anthropomorphic design is a positive impact on human perception, which is an important factor in human-computer interaction [14]. That is, increasing the level of anthropomorphism of the system can potentially compensate for the lack of contact with a person, increasing the effect of the social presence of the system.

As indicated in the study [15], three categories of anthropomorphic features in the design of information systems can be distinguished: visual, auditory, and mental. Visual features aim to improve social engagement with the technology by using anthropomorphic visual characteristics. It is important to consider that visual features can be transmitted both through software and through physical hardware. Auditory features as a method of anthropomorphic interaction can be represented as speech synthesizers, thanks to which it is possible to transmit additional information, which can lead to different reactions of the end user. Mental traits, such as emotional intelligence or personality, can also have a significant impact on the user and their decisions and at the same time have multiple implementation options.

As an example, we can consider anthropomorphic bots of bank assistants, the purpose of which is to replace the traditional role of a human consultant, for example, bank assistant, investment or insurance advisor [16]. Performing these obligations, chat bots can be responsible for managing savings or helping to make financial decisions. For digital economy, the use of chat bots as financial and insurance agents is becoming increasingly popular, which may eventually provide an opportunity to target new customer segments of the financial industry.

Referring to research [17], we will consider a laboratory experiment that was set up to examine the consequences of the anthropomorphization of robo-advisors and their impact on the perception of social presence, trust, and the likelihood of following recommendations.

Four hypotheses were also proposed:

- The increased level of anthropomorphism of the robo-advisor is positively associated with the user's sense of social presence.

- The sense of social presence of the robo-advisor is positively associated with the user's trust in the robo-advisor.

- Users who show a high level of perceived social presence are more likely to follow the recommendations of the chat bot.

- Users who demonstrate a higher level of trust in chat bot are more likely to follow the chat bot's recommendations.

To evaluate the proposed hypotheses and the effect of anthropomorphization of robo advisors in the form of chat bots, three different levels of anthropomorphism were implemented in the experiment: low, medium, and high [17]. Their goal was to make the robo-advisor more or less human-like when interacting with it. The bot with a low level of anthropomorphism was not given a name, the bot with a medium level of anthropomorphism was given the name “Robo-Advisor”, and the bot with a high level of anthropomorphism was given the name “Charles”. The app used three different avatars for bots that corresponded to levels of anthropomorphism, i.e. at a low level the avatar did not include any anthropomorphic features, at a medium level the avatar had anthropomorphic features, at a high level the avatar was as human-like as possible. To simulate a human-like messaging process, the highly anthropomorphized bot was set to have a response delay, while the low and medium bots were set to have an instantaneous response time. In addition, the highly anthropomorphic bot had a graphical notification indicator when it was ready to send a message, while the medium and low anthropomorphic bots did not. To simulate human communication, bots from the medium and high level sent greetings to the user and said goodbye after the interaction was over, and after receiving a message from the user, the bots sent a thank you message, and the bot with a low level of anthropomorphism did not have this functionality. In the application, chat bots from the middle and high level groups addressed themselves, using personal pronouns and could remember the user's name when interacting with him/her and used it for addresses. Unlike them, the bot with a low level of anthropomorphism avoided personal appeals and did not remember the user's name. In summary, the experiment presented three different chat bots with different design levels of anthropomorphism.

Participants were asked to interact with a chat bot to receive investment advice and decide whether or not to invest in three proposed fictitious scenarios, where the robo-advisor recommended one of them. After interacting with the bot, participants were

asked to fill out a questionnaire about their interaction with the chat bot. Because of the experiment, support for the first and second hypotheses was found, which confirms the expected effect of an increase in the level of anthropomorphism on the user's sense of social presence and trust. According to the results, there was no significant difference in the probability of following recommendations between bots with different levels of anthropomorphism. The feeling of social presence did not have a direct effect on the probability of following the chat bot's recommendations, but a positive indirect effect of user trust in the advisor bot on the likelihood of following recommendations was found. In addition, the results indicate that user trust has a significant effect on the likelihood of following recommendations, thus confirming the fourth hypothesis.

According to the results of the study, several recommendations were highlighted for the development of anthropomorphic design of financial robo-advisors in the form of chat bots. First, it is necessary to ensure a high level of user trust to the robo-advisor. Secondly, it is necessary to contribute the creation and provision of a sufficient level of the effect of social presence for the user to increase the level of trust.

SOFTWARE PRODUCTS FOR FINANCIAL ADVISING

Incentives to use Chat Bots

Innovations in the field of customer service have always been aimed at ensuring lower costs, quick turnover and solving the problem at the first contact [18]. The first chat bots were created to work with a limited number of requests. Quite a few simple solutions check the input for keywords and compare it to existing patterns in the database, and then provide a response based on that information. It allows providing a low price and quick turnover, but this approach of data processing makes the bot incapable of solving complex problems.

Recent developments in areas such as Natural language processing (NLP) and Machine learning (ML) provide more technological opportunities for bot developers, which allows creating software products capable of performing tasks of increased complexity. Large companies, for example Google, Apple, Amazon and Microsoft, make the biggest contribution to the development of modern and ultra-sophisticated chat bots. Solutions from market leaders provide the ability to design complex user-bot interactions while remaining commercially

viable. Such chat bots are able to process spoken language, understand and store the context of the conversation, and handle complex queries, making such solutions as an option for using in financial advice and helping with the distribution of savings.

The usually high cost of financial advice, the volatility of markets and problems with finding professional financial advisors have created favorable conditions for the use of robo-advisors, where a previously created algorithm is responsible for the consultation process itself and can replace a person.

History of the Robo-Advisors and Chat Bots Development

One of the first-generation of robo-advisors is Betterment, which was created during the 2008 crisis by the firm of the same name. The firm's personalized financial advice is based on the principles of robo-advisor technology that uses computer algorithms. In addition, licensed financial robo-advisors can provide advice to users over the phone [19].

The first robo-advisors were created to work directly with the client and provided the opportunity to trade a limited set of exchange traded funds (ETF), which allowed the firm to offer low-cost advisory services. Over time, a large number of banks adopted this experience and created their own robo-advisors. Such solutions were usually created using surveys to obtain information about the user, for example, his/her financial status and wishes, then, using specially developed algorithms, an optimal investment portfolio was built. Such software products performed their task, but did not have a sufficient level of interactivity when editing a portfolio and taking into account wishes, so customer support services were often used to solve such issues.

With the growing popularity of chat bots, new opportunities appeared for banks for attracting new customers. Due to the interest of firms, and the merging of robo-advisor and chat bot technologies, chat bot advisors were created, which, compared to previous generations of technology, could perform various tasks, for example: opening an investment account, familiarizing clients with the operation of the service, creating an investment portfolio, provide answers to questions, process bank transactions. As a result, this functionality made them an integral part of the modern financial industry.

Examples of chat bots and Robo-Advisors

There are now a large number of software solutions for automated consultations. Chat bot

technology is actively used and already implemented in many industries, in particular for the purpose of customer support [20].

As an example of such a bot, consider the solution from Bank of America – Erica (Fig. 1) [21]. The bot was introduced in late 2018 – early 2019, it had 6.3 million users and 16.5 million applications [22]. This solution is built into the bank's mobile application, its functionality includes sending notifications about banking activity, identifying available opportunities to reduce costs, i.e. counseling customers and giving advice on saving money, notifying about changes in credit rating and simplifying bill payments.

We will also consider the Ally Assist bot (Fig. 2) [23], which was launched in 2015 by Ally Bank. Ally Assist is a chat bot built-in into the Ally Mobile app that you can interact with using voice commands and text messages. The functionality of the bot includes making payments, money transfers, P2P transactions and deposits. In addition, the user can get information about the bank account or transaction history. Using Machine Learning (ML), Ally Assist can predict a user's needs through account and transaction analysis to provide relevant messages with recommendations. In addition, the bot uses Natural Language Processing technologies to answer frequently asked questions of the client.

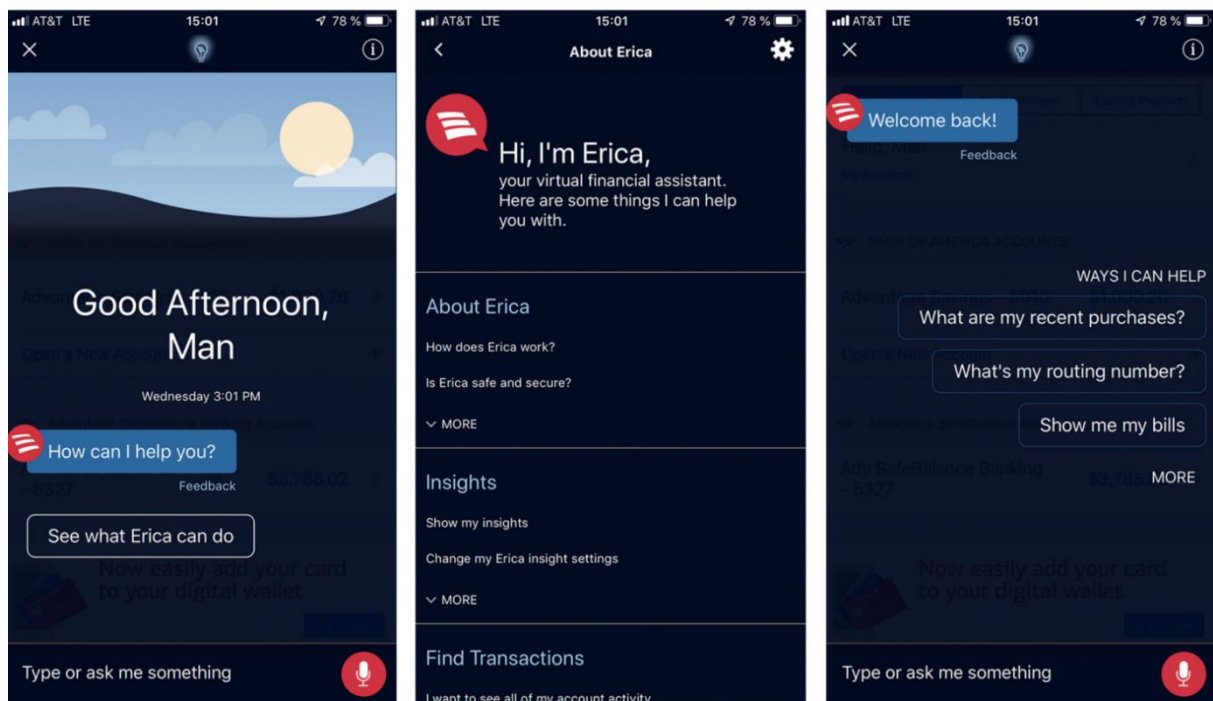


Fig. 1. Chat bot Erica

Source: compiled by the authors

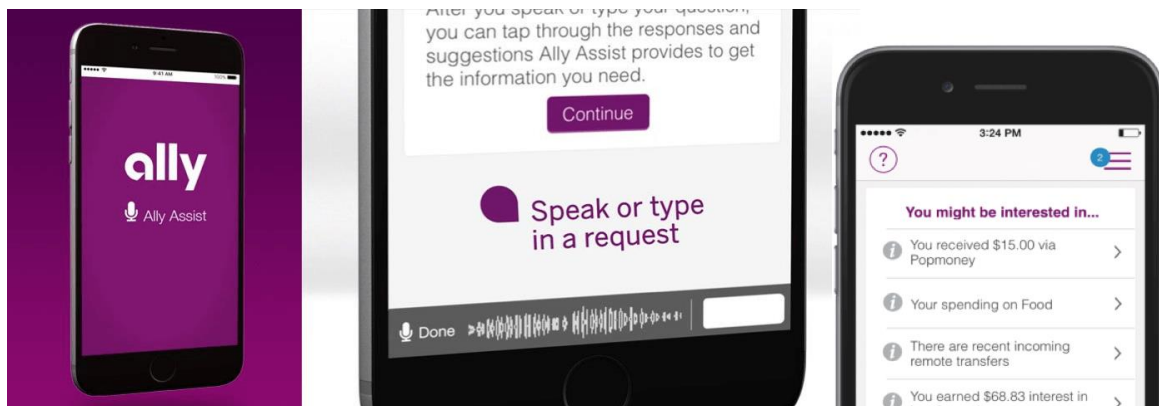


Fig. 2. Chat bot Ally Assist

Source: compiled by the authors

As an application, you can consider robo-advisors. They provide affordable opportunities to invest and manage money at a percent of the cost historically available in traditional models, allowing more people to save, invest and grow their capital to achieve their purposes [24]. There are many examples of robo-advisors, but let's consider only one of the best Wealthfront (Fig. 3) [25].

Launched in 2008, Wealthfront offers an exceptional portfolio management experience and remains a fully digital, automated financial investment solution with an extensive suite of portfolio management tools and a wide variety of financial products.

One of the most important features of the platform is the account aggregation tool. Using this feature, one can get an overview of the complete financial picture and conveniently use powerful financial planning, goal setting and progress tracking tools that help users support in achieving their financial goals. This aspect of the platform can be adjusted to the user's personal preferences, i.e. take into account current changes in the financial component, and adapt the investment model based on this information. That's why Wealthfront is the leader in the robo-advisor space.

MODEL OF TELEGRAM BOT

Mathematical model

The first stage of calculations is the determination of the future value of the annuity according to the formula (1):

$$FVA = S \cdot \frac{(1 + rr)^{N_1} - 1}{rr}, \quad (1)$$

where S – savings of individual; N_1 – is the period of the savings program; rr is the real interest rate, which is calculated according to the Fisher formula:

$$rr = \frac{1+r}{1+i} - 1, \quad (2)$$

where i is the inflation rate, r is the nominal rate.

The second stage is the determination of the present value of the annuity according to the formula (3), where C a constant level of consumption is, N_2 is the period of using the saved funds:

$$PVA = C \cdot \frac{1 - (1 + rr)^{-N_2}}{rr}. \quad (3)$$

To ensure a constant level of consumption C , the following condition has to be met:

$$FVA = PVA. \quad (4)$$

Personal income Y is divided into consumption and savings: $Y = C + S$, hence $S = Y - C$.

Technology for Telegram bot development

The main task of this research is to develop a telegram bot that helps the investor to get automated financial advice and answer to the questions about the amount of annual savings contributions at a real interest rate to ensure a stable level of consumption during defined period.

The following software tools were used to develop this bot:

Python is a high-level, interpreted general-purpose programming language. Currently, the most relevant release is Python 3, which fixes the problems

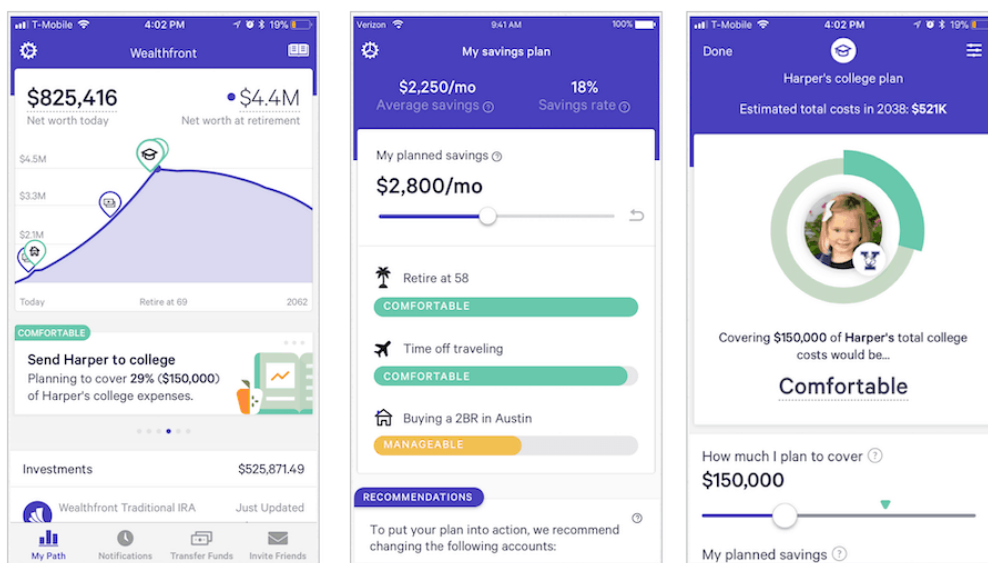


Fig. 3. Wealthfront Robo-Advisor
 Source: compiled by the authors

of versions 1.0 and 2.0 and includes additional features [26].

Telegram Bot API is an HTTP-based developer interface that is an extension of the Telegram API. This API allows bots to be connected to Telegram systems [27], that is, special accounts that do not require an additional phone number to create and simplifies their development. Bots act as an interface to code running on the server.

Aiogram is a simple and completely asynchronous framework for the Telegram Bot API developed using the Python programming language version 3.7 using the *asyncio* and *aiohttp* asynchronous libraries [28]. The exclusive features of the framework are the presence of a built-in Finite-state machine (FSM) and better performance indicators when executing the code compared to similar software solutions.

Docker is a set of platform as a service (PaaS) products that use virtualization at the operating system level to deliver software in packages called containers [29]. Docker packages software into containers i.e. standardized blocks. Each container has everything necessary for the application to work: libraries, system tools, code and a virtual environment. Docker works like a virtual machine, namely it creates a virtual representation of the server hardware and eliminates the need directly manage the server. With Docker, it is possible to quickly deploy and scale applications in any environment and be sure that this does not affect the performance of the code.

Amazon Web Services is a platform for cloud computing. The technology allows using a full-fledged virtual cluster of computers accessible via the Internet. Virtual machines provided to subscribers have most of the attributes of a computer, including hardware devices, operating system, network and pre-installed applications such as: web server, database, CRM, etc. Every Amazon Web Services system can be accessed via a web browser thanks to console Input/Output virtualization.

Elastic Compute Cloud (EC2) is a service based on Amazon Web Services, which allows users to have at their disposal a virtual cluster of computers that are available at any time [30], that is, a service for providing virtual server platforms. The user can choose between self-assembly of the server and a pre-configured machine with an installed server operating system.

Ubuntu Server is a Linux distribution based on Debian, consisting mostly of open source software [31]. It is used as a platform for Web servers, database servers, DNS servers, file servers and other types of servers. The main feature of this distribution is solving the issues of reliability, security and productivity.

The architecture of the developed Telegram bot

Our online application is developed based on an open-to-use complex template for the development of Telegram bots (Fig. 4), which helps to structure the program code and solves the problems of development and scaling of the application.

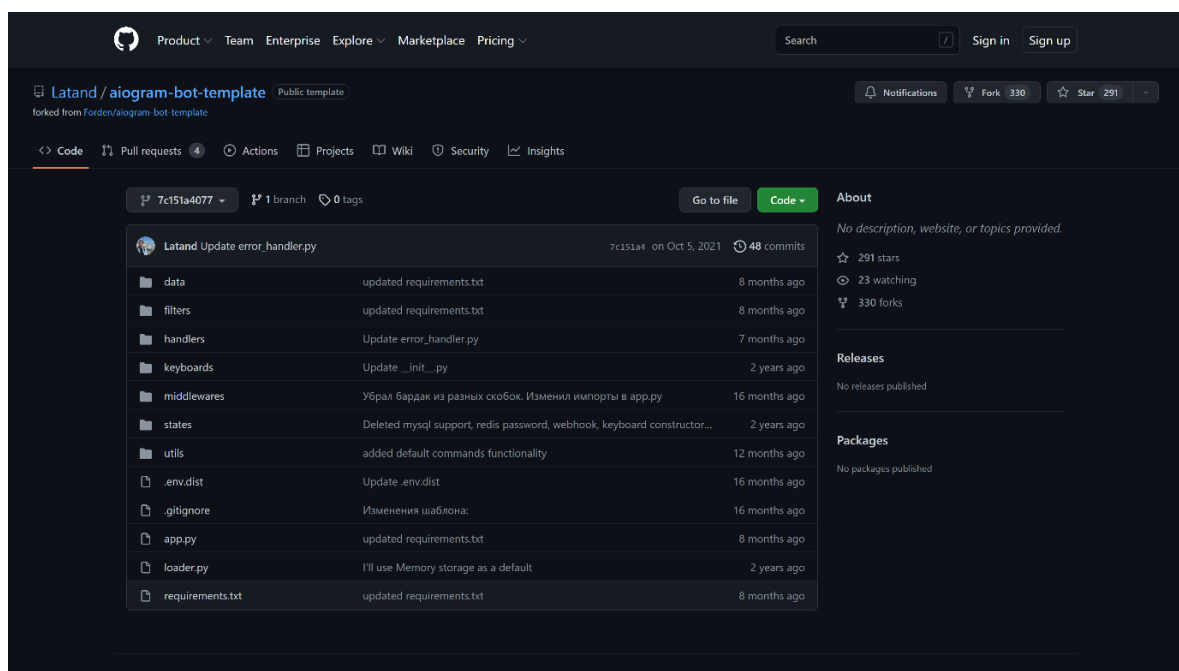


Fig. 4. Telegram bot template on GitHub

Source: compiled by the authors

The main packages of the program are handlers and utils. The handler’s package includes modules for processing incoming messages, and the utils package includes a module for calculations (Fig. 5).

The bot is hosted on the basis of Amazon Web Services – EC2. The server runs on the Ubuntu Server operating system. The bot is launched using a Docker container, polling the server is performed by the polling method (Fig. 6) [32], i.e. by constantly polling Telegram services for updates.

The user interacts with the bot thanks to a text interface based on personal messages (Fig. 7), which

is both informative, easy to use and easy to understand.

APPLICATION OF THE DEVELOPED TELEGRAM BOT

The given telegram bot is easy to use, but complex in structure. After finding the bot with the username “@AutoFinAdvisorBot” and pressing the “Start” button, the bot greets the user and sends a list of commands (Fig. 7). The anthropomorphism of the bot consists in communication that begins with a greeting, after which the communication is kept in the form of "bot question-user answer".

```

1 from aiogram.dispatcher import FSMContext
2 from aiogram.dispatcher.filters import Command
3
4 from filters import IsNumber, IsGreater2, IsToGreater2, IsDevInBound, IsWholeNumber
5 from filters.age_filter import IsLess125, IsLess250
6 from filters.number_filter import IsLess
7 from loader import dp
8 from aiogram import types
9 from aiogram import types
10
11 from states import PollStates
12 from utils.misc import Calculations
13
14 # Main message handler
15 @dp.message_handler(Command("cancel_poll"),
16                     state=[PollStates.q1_income_state, PollStates.q2_start_invest_state, PollStates.q3_end_invest_state,
17                          PollStates.q4_use_invest_state, PollStates.q5_rate_state, PollStates.q6_rate_dy_state,
18                          PollStates.generate_result_state])
19 async def cancel_poll(message: types.Message, state: FSMContext):
20     await state.finish()
21     await message.answer("Введите команду /start")
22
23
24 @dp.message_handler(Command("start_poll"), state=None)
25 async def start_poll(message: types.Message):
26     await message.answer("Введите сумму депозита (в рублях) (например: 100000 руб/евр)")
27     await PollStates.q1_income_state.set()
28
29
30 @dp.message_handler(IsNumber(), IsGreater2(), IsLess(), state=[PollStates.q1_income_state])
31 async def answer_start_age(message: types.Message, state: FSMContext):
32     avg_income_data = message.text
33     await state.update_data(avg_income_data=avg_income_data)
34     await message.answer("Введите срок в месяцы (например: 36 месяцев) (например: 36 мес/евр)")
35     await PollStates.q2_start_invest_state.set()
36
37
38 @dp.message_handler(IsNumber(), IsWholeNumber(), IsGreater2(), IsLess125(), state=[PollStates.q2_start_invest_state])
    
```

Fig. 5. Basic modules of Telegram bot
 Source: compiled by the authors

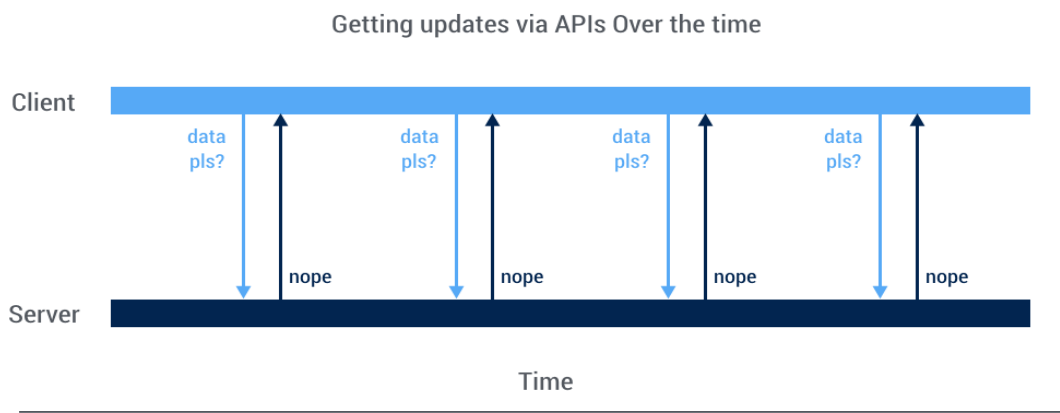


Fig. 6. Poling
 Source: compiled by the authors

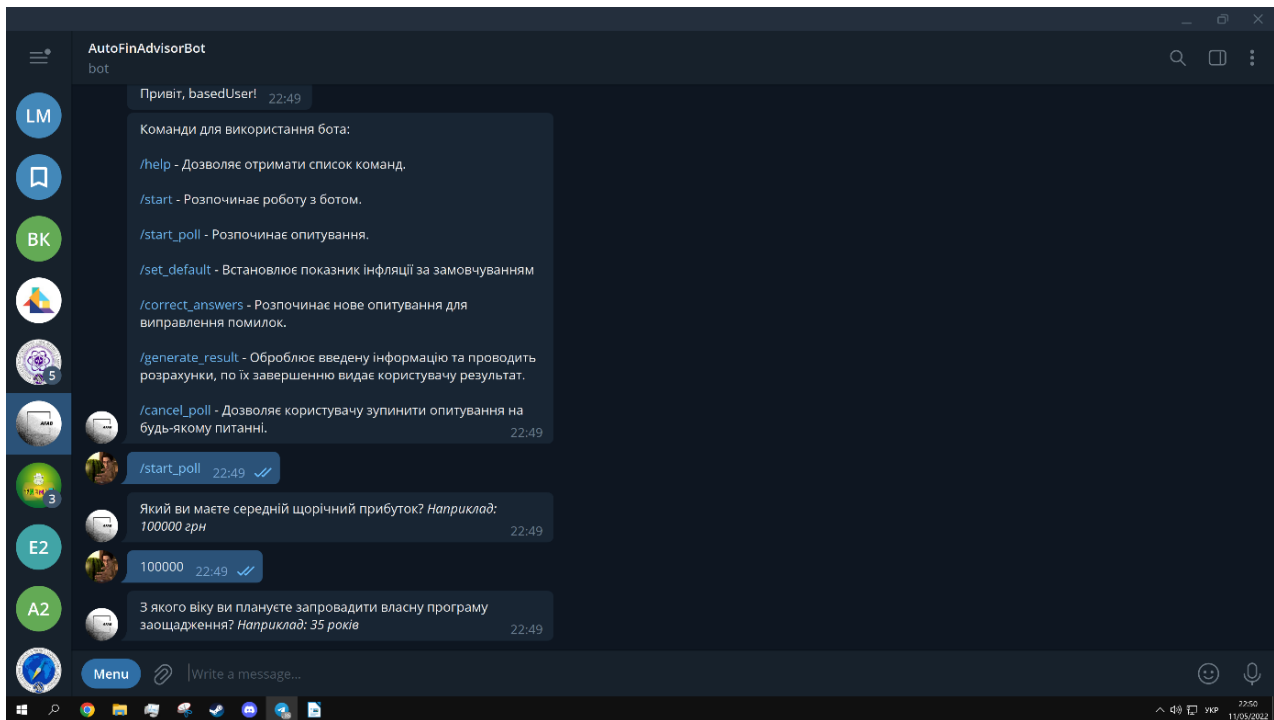


Fig. 7. Telegram bot interface

Source: compiled by the authors

The following commands are provided to work with the bot:

- /help – allows the user to get a list of all the commands necessary to interact with the bot;
- /start – when entered, starts working with the bot;
- /start_poll – after entering this command, the user will start a poll, during which all the information necessary for the operation of the bot will be collected;
- /set_default – sets the default inflation rate;
- /correct_answers – provides an opportunity to correct survey results;
- /generate_result – gives the result of calculations with explanations;
- /cancel_poll – stops a previously started poll at any stage.

The main tool for interaction between the user and the bot is polling, this process works on the basis of Finite State Machine (FSM), that is, when the polling process is started, the bot enters a specially created state, during which the answer to the question posed to the user has to be received. If the answer is satisfactory, it will be stored in the memory and transferred to the appropriate variables, then the bot will move to the next state and the process will be repeated, otherwise the question will be about getting the correct answer or stopping the survey. Sent responses pass through special filters that ignore incorrect options and notify about it (Fig. 8).

Let us consider an example of how this Telegram bot works. Suppose it is necessary to determine how much an investor should annually invest in savings S at a real interest rate r to ensure a constant level of consumption over a specified period.

1. We find the bot with the username “@AutoFinAdvisorBot”, press the “Start” button and receive a greeting with a list of commands, from the list we choose the “/start_poll” command, which will start the poll.

2. We answer the questions asked by the bot. They are necessary for planning a constant consumption level C during a certain period (from t_1 to t_3):

2.1. What is your average annual income? Answer: $Y = 100000$ UAH (Checking the positivity condition of Y).

2.2. At what age do you plan to start your own savings program? Answer: $t_1 = 35$ years (Checking the positivity condition t_1).

2.3. At what age do you plan to complete your own savings program? Answer: $t_2 = 60$ years (Checking the condition $t_2 > t_1$).

2.4. Until what age do you plan to use your own savings fund? Answer: $t_3 = 80$ years (Checking the condition $t_3 > t_2$).

2.5. What a nominal annual interest rate on savings do you expect to receive? Answer: $r = 10\%$ (Checking the positive condition of r).

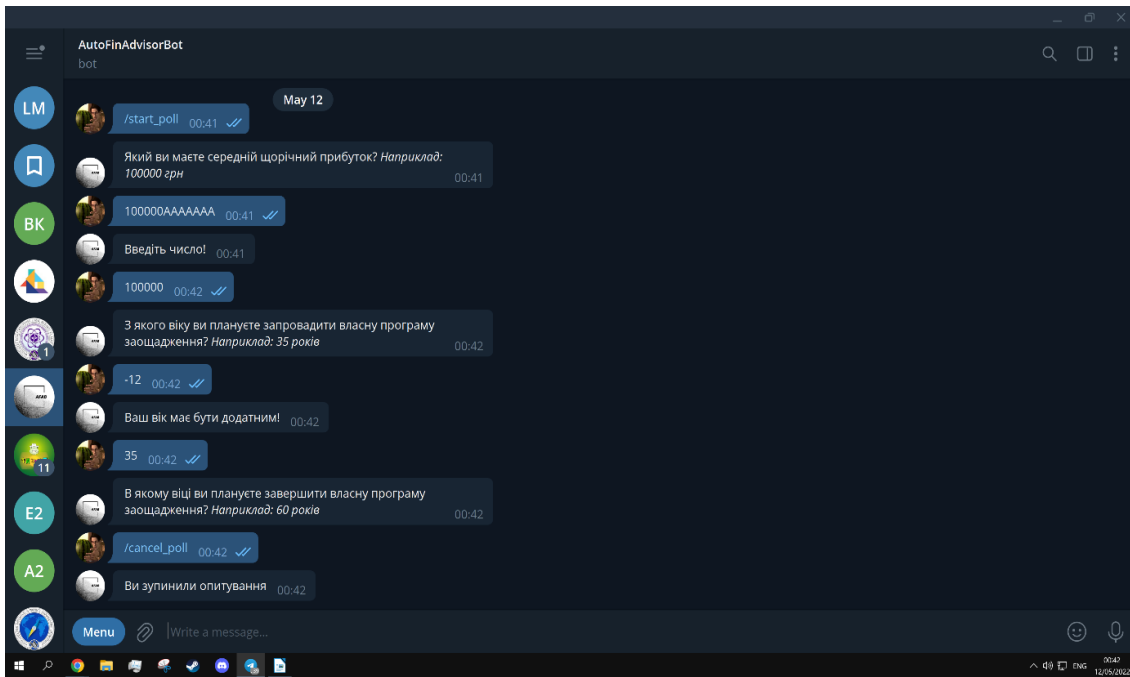


Fig. 8. Examples of filter operation of Telegram bot

Source: compiled by the authors

2.6. What percentage reduction from your determined average return would be critical for you to decide to reconsider (discontinue) the savings program? Answer: $h = 30\%$ (Checking the condition $0\% < h \leq 100\%$).

2.7. What is the value of the inflation indicator to use for calculations? Answer: $i = 6.3\%$ (Checking the positivity condition i).

3. After answering all the questions, we receive a message in which all the given answers are indicated (Fig. 9).

4. To obtain the result of calculations based on the provided answers, we use the command `/generate_result`, after entering which we receive a message stating: “To ensure a constant level of consumption from 35 to 80 years in the amount of UAH 73,267. You need to save UAH 26,733 every year per year at a nominal interest rate of 10%”. (Fig. 10). The calculation model is presented below.

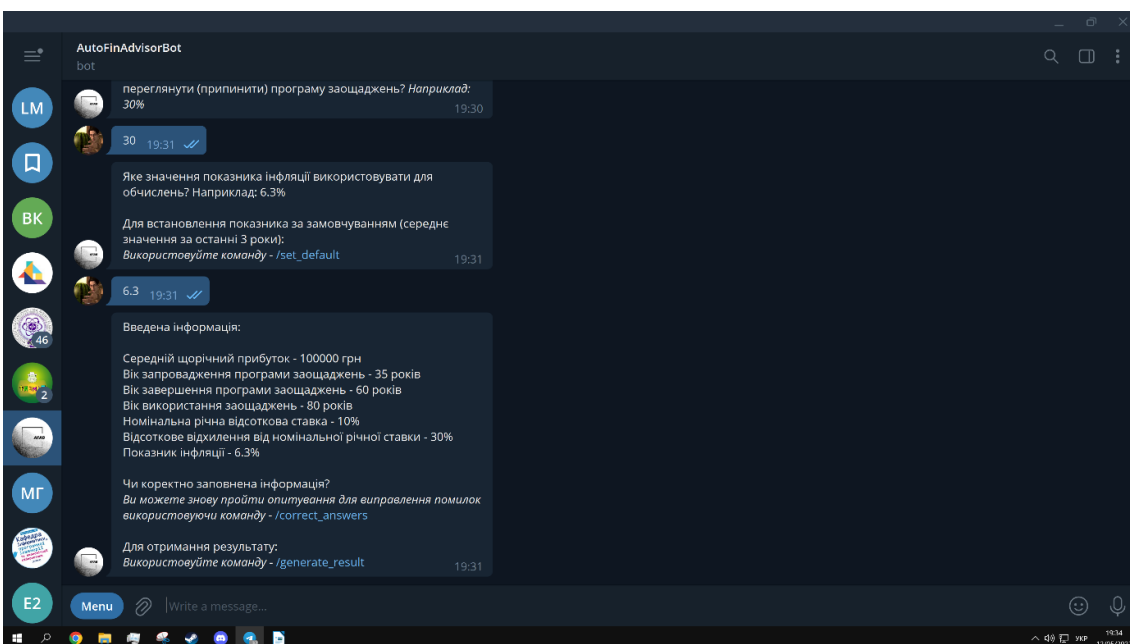


Fig. 9. Conversation with Telegram bot using specified answers

Source: compiled by the authors

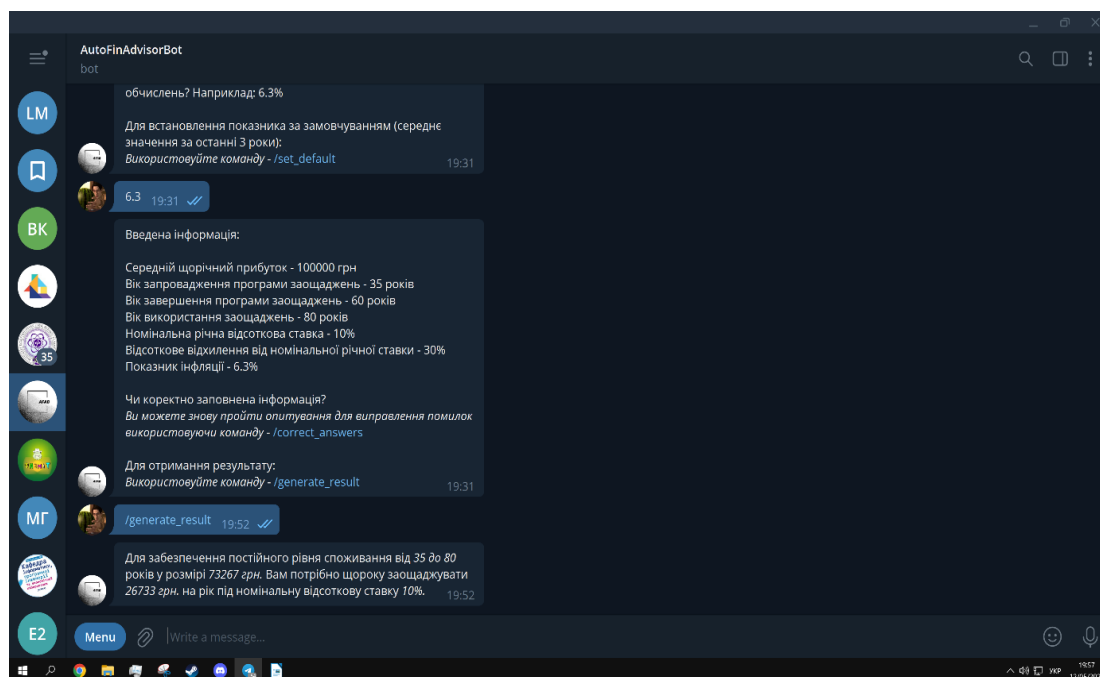


Fig. 10. Notification about the result of calculations

Source: compiled by the authors

If the minimum pension expected by the individual is lower than the income of UAH 73,267 per year, it means that this savings program is appropriate. In the opposite case, alternative use of funds is advisable.

Although the authors' Telegram bot provided in the example does not demonstrate all available Telegram bot API tools and does not outperform commercial solutions, it is an easy-to-use and accessible tool for personalize financial advice for all Telegram users.

CONCLUSIONS

The influence of the anthropomorphism effect and anthropomorphic design of the bot on the feeling of social presence, trust, and the desire to follow recommendations was investigated. We pointed out the recommendations to develop an anthropomorphic design of chat bots.

The incentives to use chat bots were analysed, that is the influence of various factors on the firm's decision to develop its own software product. One was considered the stages of chat bot development and robo-advisors technologies.

The existing software solutions of chat bots and robo-advisors were analysed, and information about their basic functionality was proposed.

We developed our own software product, which is based on a model of constant level of consumption during long-life period, this telegram bot for automated financial consultations to meet the goals of private investors in decision making about savings program taking into account the inflation rate. The Telegram bot is a fast and easy-to-use application provides viable option for receiving financial advice and helping potential investors with income distribution when expected benefits are more than expected expenses.

REFERENCES

1. Walter, B., Kremmel, D. & Jäger, B. "The impact of lay beliefs about AI on adoption of algorithmic advice". *Mark Lett.* 2021; 33: 143–155. DOI: <https://doi.org/10.1007/s11002-021-09589-1>.
2. Hildebrand, C., Bergner, A. "Conversational robo advisors as surrogates of trust: onboarding experience, firm perception, and consumer financial decision making". *J. of the Acad. Mark. Sci.* 2021; 49: 659–676. DOI: <https://doi.org/10.1007/s11747-020-00753-z>.
3. Jung, D., Dorner, V., Weinhardt, C. et al. "Designing a robo-advisor for risk-averse, low-budget consumers". *Electron Markets.* 2018; Vol. 28: 367–380. DOI: <https://doi.org/10.1007/s12525-017-0279-9>.
4. Glaser F., Iliewa Z., Jung D. & Weber M. "Towards Designing Robo-advisors for Unexperienced Investors with Experience Sampling of Time-Series Data". In: Davis F., Riedl R., vom Brocke J., Léger PM.,

Randolph A. (eds) Information Systems and Neuroscience. *Lecture Notes in Information Systems and Organisation* 2019; 29: 133–138. DOI: https://doi.org/10.1007/978-3-030-01087-4_16.

5. D’Acunto, F. & Rossi, A.G. “Robo-Advising”. Springer Books. In: Raghavendra Rau et al. (ed.). *The Palgrave Handbook of Technological Finance*. Springer. 2021. p. 725–749. DOI: https://doi.org/10.1007/978-3-030-65117-6_26.

6. Kilinich, D. & Kobets, V. “Support of investors’ decision making in economic experiments using software tools”. In: *Proceedings of the 15th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. CEUR-WS. Kherson: Ukraine. 2019; 2393: 277–288. – Available from: http://ceur-ws.org/Vol-2393/paper_273.pdf. – [Accessed: Sep. 2021].

7. Snihoviyi, O., Ivanov, O. & Kobets, V. “Implementation of Robo-Advisors Using Neural Networks for Different Risk Attitude Investment Decisions”. In: *Proceedings of the 9th International Conference on Intelligent Systems*. IEEE 2018; 8710559: 332–336. DOI: <https://doi.org/10.1109/IS.2018.8710559>.

8. Kobets, V., Yatsenko, V., Mazur, A. & Zubrii, M. “Data analysis of private investment decision making using tools of Robo-advisers in long-run period”. In: *Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. CEUR-WS. Kyiv: Ukraine 2018; 2104: 144–159. – Available from: http://ceur-ws.org/Vol-2104/paper_162.pdf. – [Accessed: Sep. 2021].

9. Ivanov, O., Snihoviyi, O. & Kobets, V. “Implementation of Robo-advisors tools for different risk attitude investment decisions”. In: *Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. CEUR-WS. Kyiv: Ukraine. 2018; 2104: 195–206. – Available from: http://ceur-ws.org/Vol-2104/paper_161.pdf. – [Accessed: Sep, 2021].

10. Snihoviyi, O., Kobets & V., Ivanov, O. “Implementation of Robo-Advisor Services for Different Risk Attitude Investment Decisions Using Machine Learning Techniques”. In: Ermolayev, V. et al. (eds.) ICTERI 2018, *Communications in Computer and Information Science*. Springer. Cham 2019; 1007: 298–321. DOI: https://doi.org/10.1007/978-3-030-13929-2_15.

11. Jung, D., Dorner, V., Glaser, F. & Morana, S. “Robo-Advisory - Digitalization and Automation of Financial Advisory”. *Business & Information Systems Engineering*. 2018; 60 (1): 81–86. DOI: <https://doi.org/10.1007/s12599-018-0521-9>.

12. Epley, N., Waytz, A. & Cacioppo, J. T. “On seeing human: A three-factor theory of anthropomorphism”. *Psychological Review*. 2007; 114 (4): 864–886. DOI: <https://doi.org/10.1037/0033-295x.114.4.864>.

13. Sheehan, B., Jin, H. S. & Gottlieb, U. “Customer service chat bots: Anthropomorphism and adoption”. *Journal of Business Research*. 2020; 115: 14–24. DOI: <https://doi.org/10.1016/j.jbusres.2020.04.030>.

14. Hudlicka, E. “To feel or not to feel: The role of affect in human–computer interaction”. *International Journal of Human-Computer Studies*. 2003; 59 (1-2): 1–32. DOI: [https://doi.org/10.1016/s1071-5819\(03\)00047-8](https://doi.org/10.1016/s1071-5819(03)00047-8).

15. Pfeuffer, N., Benlian, A., Gimpel, H. & Hinz, O. “Anthropomorphic Information Systems”. *Business & Information Systems Engineering*. 2019; 61 (4): 523–533. DOI: <https://doi.org/10.1007/s12599-019-00599-y>.

16. Shah, K.B., Shetty, M.S., Shah, D.P. & Pamnani, R. “Approaches towards building a banking assistant”. *International Journal of Computer Applications*. 2017; 166 (11): 1–6. DOI: <https://doi.org/10.5120/ijca2017914140>.

17. Morana, S., Gnewuch, U., Jung, D., Granig, C. “The Effect of Anthropomorphism on Investment Decision-Making with Robo-Advisor Chatbots”. *Proceedings of European Conference on Information Systems (ECIS)*. Marrakech: Morocco. 2020.

18. “Chat based Robo-Advisors – A game changer in the banking space”: website. – Available from: <https://www.credenceanalytics.com/press-releases/IBS-Intelligence-and-Credence-Analytics-presents-Robo-Advisory-Whitepaper.html> – [Accessed: June, 2022].

19. “Betterment Company”: website. – Available from: [https://en.wikipedia.org/wiki/Betterment_\(company\)](https://en.wikipedia.org/wiki/Betterment_(company)). – [Accessed: June, 2022].

20. “Cecilie Bertinussen Nordheim. Trust in chat bots for customer service – findings from a questionnaire study”: website. – Available from: https://www.duo.uio.no/bitstream/handle/10852/63498/1/CecilieBertinussenNordheim_masteroppgaveV18.pdf. – [Accessed: June, 2022].

21. “How effective is the Human-AI interaction of BOA's Erica?": website. – Available from: https://medium.com/@man_hei_miranda_mo/how-good-is-the-human-ai-interaction-of-boas-erica-ff626f539a1a. – [Accessed: June, 2022].
22. “A deeper look into what makes a successful chatbot with Bank of America's Erica”: website. – Available from: <https://tearsheet.co/artificial-intelligence/a-deeper-look-into-what-makes-a-successful-chatbot-with-bank-of-americas-erica/>. – [Accessed: June, 2022].
23. “How chat bots improve user experience in online banking”: website. – Available from: <https://ergomania.eu/how-chatbots-improve-user-experience-in-online-banking>. – [Accessed: June, 2022].
24. “Best Robo-Advisors”: website. – Available from: <https://www.investopedia.com/best-robo-advisors-4693125>. – [Accessed: June, 2022].
25. “What is Wealthfront Financial?": website. – Available from: <https://www.groovypost.com/reviews/what-is-wealthfront-financial>. – [Accessed: June, 2022].
26. Kuhlman, D. “A python book: Beginning Python, Advanced Python, and Python Exercises”: website. – Available from: https://www.davekuhlman.org/python_book_01.pdf. – [Accessed: June, 2022].
27. “Bot API”: website. – Available from: <https://core.telegram.org/api>. – [Accessed: June, 2022].
28. “Welcome to aiogram's documentation”: website. – Available from: <https://docs.aiogram.dev/en/latest>. – [Accessed: June, 2022].
29. O'Gara, Maureen. “Ben Golub, Who Sold Gluster to Red Hat, Now Running dotCloud”. *SYS-CON Media*. 2013.
30. “Amazon Web Services”: website. – Available from: https://en.wikipedia.org/wiki/Amazon_Web_Services. – [Accessed: June, 2022].
31. “Ubuntu”: website. – Available from: <https://en.wikipedia.org/wiki/Ubuntu>. – [Accessed: June, 2022].
32. “Webhooks vs Polling”: website. – Available from: <https://community.stuart.engineering/t/webhooks-vs-polling>. – [Accessed: June, 2022].

Conflicts of Interest: The authors declare no conflict of interest

Received 15.09.2022
Received after revision 12.10.2022
Accepted 18.10.2022

DOI: <https://doi.org/10.15276/hait.05.2022.18>

УДК 004.89:339.3:338

Застосування чат ботів для персоналізованих фінансових консультацій

Кобець Віталій Миколайович ¹⁾

ORCID: <https://orcid.org/0000-0002-4386-4103>; vkobets@kse.org.ua. Scopus Author Id: 56006224700

Козловський Кирило Геннадійович ¹⁾

ORCID: <https://orcid.org/0000-0003-3505-9902>; b4559667@gmail.com

¹⁾ Херсонський державний університет, вул. Університетська, 27. Херсон, 73003, Україна

АНОТАЦІЯ

Чат-боти є веб-системами, які останнім часом привернули багато уваги завдяки своїй здатності надавати професійні фінансові консультації за нижчою ціною, ніж люди-консультанти. Ми розглянули антропоморфний дизайн чат-бота, який позитивно сприятиме на людське сприйняття, що є важливим фактором взаємодії людини з комп'ютером, оскільки підвищення рівня антропоморфізму системи потенційно може компенсувати відсутність контакту з людиною. Досліджено вплив ефекту антропоморфізму та антропоморфного дизайну бота на відчуття соціальної присутності, довіри та бажання виконувати рекомендації. Висока вартість фінансових консультацій, нестабільність ринків і проблеми з пошуком професійних фінансових консультантів створили сприятливі умови для використання автоматизованих фінансових

консультантів. Завдяки зростанню популярності чат-ботів у банків з'явилися нові можливості для залучення нових клієнтів. Існує велика кількість чат-ботів, які можуть надавати фінансові консультації, але, як правило, це комплексні рішення, і для роботи з ними потрібно досить багато часу, тому, якщо мета користувача отримати швидку і конкретну відповідь, такі рішення не найкращий варіант. Метою нашого дослідження є розробка чат-бота в Telegram для персоналізованих фінансових консультацій. Для досягнення цієї мети розглянуті приклади чат-ботів консультантів для потенційних інвесторів, проаналізовані доступні програмні рішення для фінансових радників і створений власний програмний продукт для персоналізованих фінансових консультацій. Методи дослідження включають аналітичні математичні методи та технології, такі як Python, Telegram Bot API, Aiogram, Docker, Amazon Web Services, EC2, Ubuntu Server. Ми створили власний програмний продукт, телеграм-бот для автоматизованих фінансових консультацій для досягнення цілей приватних інвесторів шляхом розрахунку постійного рівня споживання протягом визначеного періоду за очікуваного рівня інфляції, щоб вирішити, інвестувати чи ні. Наш телеграм-бот надає можливість отримати фінансову консультацію та допомогти потенційним інвесторам у розподілі капіталу.

Ключові слова: чат-боти; автоматизований фінансовий консультант; антропоморфний дизайн; інвестори

ABOUT THE AUTHORS



Vitaliy M. Kobets – Doctor of Economic Sciences, Professor, Professor of Computer Science and Software Engineering Department. Kherson State University. 27, Universitetska Str. Kherson, 73003, Ukraine
ORCID: <https://orcid.org/0000-0002-4386-4103>; vkobets@kse.org.ua. Scopus Author Id: 56006224700
Research field: Data Science in Economics; Evolutionary Microeconomics; Robo-Advisor

Кобець Віталій Миколайович – доктор економічних наук, професор, професор кафедри Комп'ютерних наук та програмної інженерії. Херсонський державний ун-т, вул. Університетська, 27. Херсон, 73003, Україна



Kyrylo H. Kozlovskiy – student of Computer Science and Software Engineering Department. Kherson State University. 27, Universitetska Str. Kherson, 73003, Ukraine
ORCID: <https://orcid.org/0000-0003-3505-9902>; b4559667@gmail.com
Research field: Information Systems; Software Engineering

Козловський Кирило Геннадійович – студент кафедри Комп'ютерних наук та програмної інженерії. Херсонський державний ун-т, вул. Університетська, 27. Херсон, 73003, Україна