DOI: https://doi.org/10.53052/9788367652001.01

Mykhailo GOLOVAN[1]

Supervisor: Viktor GNATYK[2]

SYSTEM NATYCHMIASTOWYCH WIADOMOŚCI DLA CENTRUM KONTAKTOWEGO

Streszczenie: Artykuł analizuje współczesne systemy komunikatorów, bada działanie botów, identyfikuje problematyczne kwestie, którymi najczęściej zajmują się centra konsultacyjne. Bot Telegram został opracowany i eksperymentalnie zbadany. Bot Telegram pozwala odciążyć centrum konsultacyjne i w wygodny sposób przekazać użytkownikom niezbędne informacje.

Słowa kluczowe: centrum kontaktowe, komunikator, bot Telegram

INSTANT MESSAGING SYSTEM FOR THE CONTACT CENTER

Summary: The article analyzes modern instant messaging systems, examines the operation of bots, identifies problematic issues with which consultation centers are most often addressed. The Telegram bot was developed and experimentally researched. The Telegram bot allows you to reduce the burden on the consultation center and provide users with the necessary information in a convenient way.

Keywords: contact center, instant messaging system, Telegram bot

1. Introduction

At the current stage of development, society cannot imagine its life without the use of information technologies, which are used in all areas of human activity. An important part of the life of a modern person is the use of instant messaging systems (IMS), which allow you to quickly receive the necessary information, exchange messages, files, images, sound signals, create and use group chats, make calls, polls, etc. The use of instant messaging systems is gaining considerable use in the work of consulting centers, providing automation of the provision of consulting services.

[1] National Aviation University, Educational scientific institute of continuous education, Telecommunications and radio engineering, 2199038@stud.nau.edu.ua
[2] National Aviation University, Faculty of Aeronautics, Electronics and Telecommunications, Telecommunications and radio engineering, viktor.hnatiuk@npp.nau.edu.ua

2. Literature review

Among the main services for exchanging messages are: Skype, Telegram, Viber, Facebook Messenger, WhatsApp, etc. The most popular IMS in Ukraine in 2020-2021 were Viber, which was used by 99% of smartphone users aged 13 to 55, and Facebook Messenger, Telegram; WhatsApp and Skype are gradually losing users, they are used by less than half of users, at the same time Telegram is growing in popularity [1], especially among young people. According to [2], Telegram is a multi-platform cloud messenger with VoIP functions for smartphones, tablets and PCs, which allows the exchange of text, voice and video messages, stickers and photos, files of many formats. It also has the functions of video and audio calls, organization of video conferences in groups and channels. Telegram client programs are available for Android, iOS, Windows Phone, Windows, macOS and GNU / Linux. The number of monthly active users of the service as of January 2021 is about 500 million people. In addition to exchanging messages in dialogues and groups, you can store an unlimited number of files in the messenger, run channels (microblogs), create and use bots. Bots are third-party programs that run inside Telegram. Users can interact with the bots by sending them messages, commands, and built-in requests. Our bot is managed using HTTPS requests to the Telegram bot API. Also, Telegram is increasingly integrated with various electronic systems, IoT, etc. [3-10].

Today, the university is a separate type of institution, which is a legal entity under private or public law, operates in accordance with the issued license to conduct educational activities at certain levels of higher education, conducts scientific, scientific and technical, innovative and/or methodical activities, ensures the organization of the educational process and acquisition of higher education and post-graduate education by individuals, taking into account their vocations, interests and abilities. During admission to the university, one should be guided by the conditions and rules of admission, however, taking into account the large amount of information that applicants need to familiarize themselves with and its complexity, applicants have a significant number of questions. Considering this, counseling centers (CC) operate in university structures, but in peak periods, during the active phase of the admissions campaign, the number of applications from applicants increases hundreds, thousands of times. For example in Fig. 1.

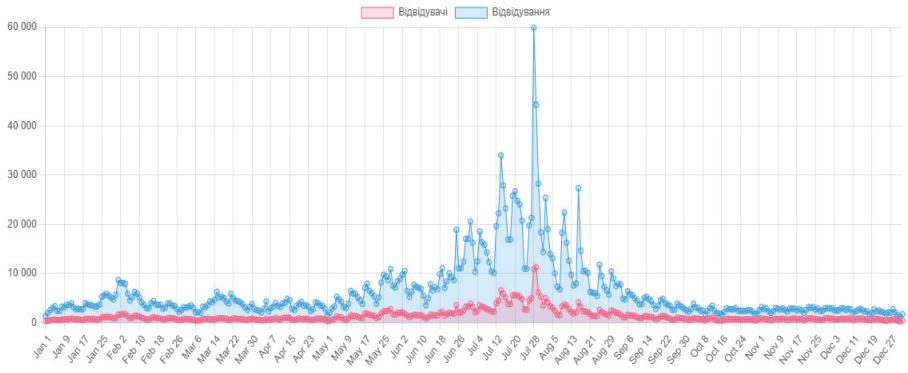

Figure 1. Statistics of visits to the website of the admissions committee of a leading university of Ukraine for 2021

Displayed statistics of visits to the website of the admission committee of the University of Ukraine for 2021, where you can see up to 60,000 visits per day in peak periods.
A similar increase in appeals is also observed during the work of contact centers, as a result, CC employees are physically unable to service all appeals. In view of this, there is an urgent scientific and practical task of automating the provision of advisory services regarding admission to the university.
Due to the growing high popularity of Telegram among young people, a simple and intuitive interface, ensuring data privacy, ensuring access from several devices at the same time, the absence of limits on the size of multimedia and chats, the availability of open source code and API for developers, protection against interception and hacking, and the presence of a wide functionality, the purpose of the work is to automate the provision of advisory services regarding admission to the university using modern instant messaging systems.

3. Telegram bot development

When developing a Telegram bot, first of all, we install an integrated development environment. IntelliJ IDEA, a commercial integrated development environment for various programming languages (Java, Kotlin, Python, Scala, PHP, etc.) from the JetBrains company, was chosen as the development environment. The system is delivered in the form of a reduced-functionality free version "Community Edition" and a full-featured commercial version "Ultimate Edition", for which active developers of open projects have the opportunity to obtain a free license. The source texts of the Community version are distributed under the Apache 2.0 license. Binary builds are prepared for Linux, Mac OS X and Windows.
We use Maven as a build automation tool. Maven projects are configured using the project object model (POM), which is stored in the pom.xml file (Fig. 2).

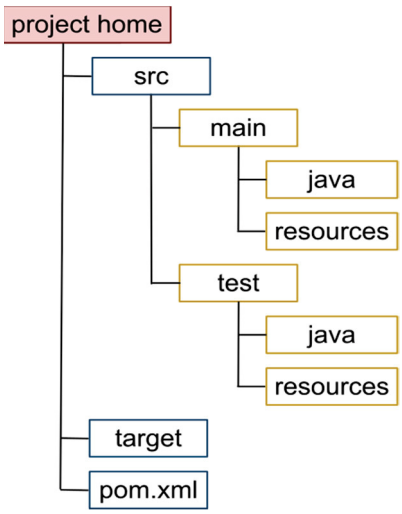

Figure 2. A directory structure for a Java project automatically generated by Maven

Maven considers two aspects of building software: how the software is built and its dependencies. Unlike previous tools like Apache Ant, it uses conditionals for the build procedure. Only exceptions need to be specified. The XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plugins. It comes with pre-defined goals to perform certain well-defined tasks, such as compiling code and packaging it. Maven dynamically loads Java libraries and Maven plugins from one or more repositories, such as the central Maven 2 repository, and stores them in a local cache. This local cache of downloaded artifacts can also be updated with artifacts created by local projects.

Public repositories can also be updated. Maven is built using a plugin-based architecture, allowing you to use any program that can be controlled using standard input. A native C / C++ plugin is supported for Maven 2.

In the pom.xml file, we connect the dependency:

```
<dependency>
  <groupId>org.telegram</groupId>
  <artifactId>telegrambots</artifactId>
  <version>4.9</version>
</dependency>
```

The object-oriented programming language Kotlin, which runs on top of the Java Virtual Machine and is developed by the JetBrains company, will be used for development. In the pom.xml file, connect Kotlin:

```
<dependency>
  <groupId>org.jetbrains.kotlin</groupId>
  <artifactId>kotlin-stdlib-jdk8</artifactId>
  <version>${kotlin.version}</version>
</dependency>
<dependency>
  <groupId>org.jetbrains.kotlin</groupId>
  <artifactId>kotlin-test</artifactId>
  <version>${kotlin.version}</version>
  <scope>test</scope>
</dependency>
```

The next step is to create a Bot class that will inherit the TelegramLongPollingBot() class. Next, we create methods:

```
override fun getBotToken() = "BotToken";
override fun getBotUsername() = "NameBot";
override fun onUpdateReceived(update: Update) {…}.
```

To obtain the values of "BotToken" and "NameBot", we will use Telegram BotFather, register the Telegram bot and get the necessary "BotToken" and "NameBot". Then we can specify them in the getBotToken() and getBotUsername() methods.

Next, we initialize and register the bot:

```
fun main(){
    ApiContextInitializer.init();
    TelegramBotsApi().registerBot(Bot())
}
```

Next, we implement the `onUpdateReceived(update: Update)` method. To do this, we will use the global /start command, which starts "communication" with the user (for example, sends a welcome message). To process /start, we will use a conditional expression with several branches when, similar to the switch statement in C-like programming languages. When checks against its argument on all branches in sequence until some branching condition is met.

The implementation of the `onUpdateReceived(update: Update)` method using when looks like this:

```
when (update.message.text) {
"/start" -> {
val keyboard = ReplyKeyboardMarkup().
setResizeKeyboard(true)keyboard.
keyboard = listOf(
KeyboardRow().
apply {
add(KeyboardButton("Button1"))
add(KeyboardButton("Button2"))}, KeyboardRow().apply {
add(KeyboardButton("…"))
add(KeyboardButton("Button_n"))})
execute(SendMessage()
 .setReplyMarkup(keyboard)
 .setChatId(update.message.chatId)
 .setText("Text"))
}" Button1" -> {
val chatId1 = update.message.chatId
execute(SendMessage() .setText("✓" + "  Step 1" +
"\n" +"✓" + "   Step 2. " +
"\n" +"✓" + "   Step ... " +
"\n" +"✓" + "   Step n. ").setChatId(chatId1)
 .setReplyMarkup (InlineKeyboardMarkup().
 apply {keyboard = listOf(listOf
 (InlineKeyboardButton("Text").
 apply { url = "https://..." }))
 }))}
```

Similarly, the `onUpdateReceived(update: Update)` method is implemented for other buttons.

After conducting an analysis of the most frequently asked questions by entrants to the counseling center, a Telegram bot was developed that provides information on: the

sequence of steps (algorithm of actions) when entering a university for the "Bachelor's" and "Master's" programs; specialties (educational programs) of the university, according to which the specialists of OS "Bachelor" and OS "Master" are trained; cost and duration of training; university admission rules; educational centers "Donbas-Ukraine" and "Crimea-Ukraine"; common questions-answers about admission; useful information for entrants and contacts of the admissions committee (address, telephone numbers, schedule of the consultation center, etc.).

Using the data and implementation of the `onUpdateReceived(update: Update)` method, a Telegram bot keyboard was developed (Fig. 3), which is activated by the global /start command.

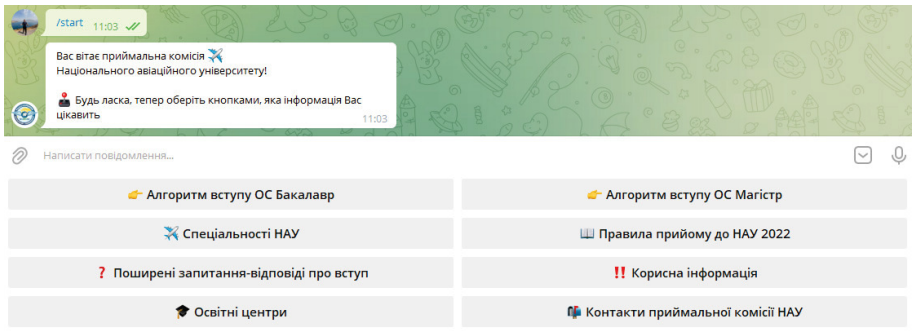

Figure 3. Telegram bot keyboard

In addition to the Telegram bot keyboard, a menu of university specialties was developed to display information on each individual specialty, including: educational degrees, names of educational programs, forms of education, tuition fees by educational degree, and a link to the website of the specialty.

To implement the menu of specialties, let's set their set:

$$\mathbf{S} = \{\bigcup_{j=1}^{m} \mathbf{S}_j\} = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m\} \tag{1}$$

where $\mathbf{S}_j \subseteq \mathbf{S}$, $(j = \overline{1,m})$, m – the number of university specialties.

So, using expression (1) and data [11], at $m = 49$ we get:

$$\mathbf{S}_{NAU} = \{\bigcup_{j=1}^{49} \mathbf{S}_j\} = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{49}\} = \{011, 022, ..., 293\} \tag{2}$$

where $\mathbf{S}_1 = \mathbf{S}_{011} = 011$, $\mathbf{S}_2 = \mathbf{S}_{022} = 022$ – specialties (codes) of the National Aviation University.

Using expression (2), we implement the `onUpdateReceived(update: Update)` method for each specialty.

As a result of the implementation of the `onUpdateReceived(update: Update)` method for the code of the specialty, for example 172, we will receive a display of information about the specialty by its code (Fig. 4).

Using expression (2) when implementing the `onUpdateReceived(update: Update)` method, we can, as an example (Fig. 4), for each specialty, display information about educational degrees, names of educational programs, forms of education, cost of education by educational degree, and a link to the website of the specialty.

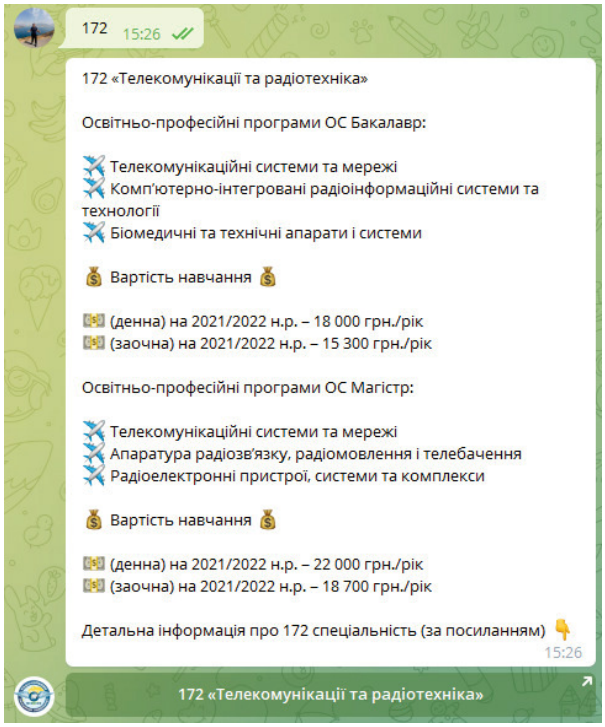

Figure 4. Display information about the specialty by its code

With the help of graphic notation UML, we visualize the developed software (Fig. 5-8).

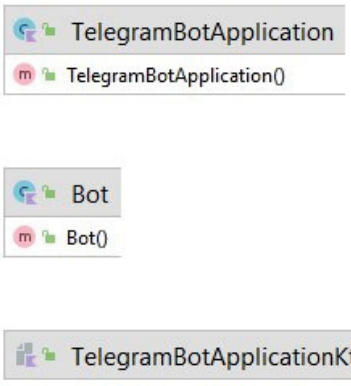

Figure. 5. Classes with constructors

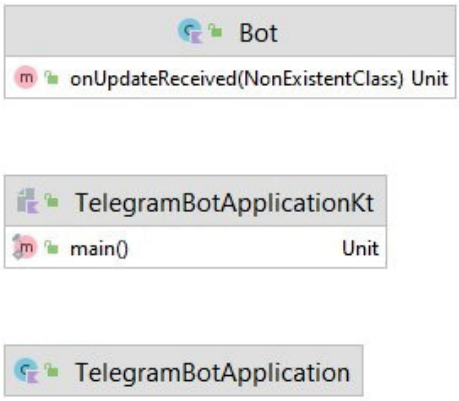

Figure. 6. Classes with methods

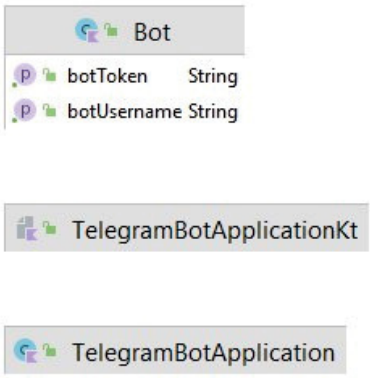

Figure. 7. Classes with properties

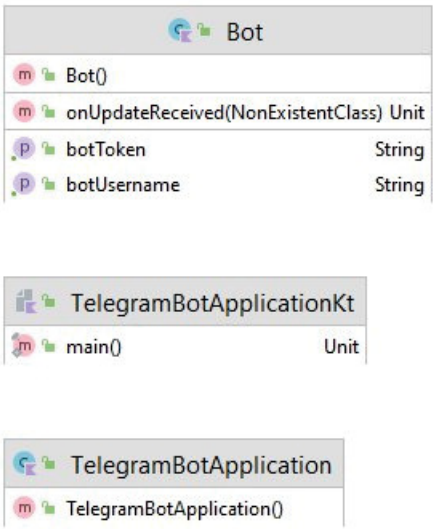

Figure. 8. Classes with fields, constructors, methods and properties

So, using the unified modeling language (Fig. 5-8), the developed software was visualized.

4. Conclusions

The article solves the actual scientific and practical task of automating the provision of consulting services using modern IMSs. A Telegram bot was developed for this purpose, there are no similar Telegram bots in the leading universities of Ukraine. As a result of an experimental study of the Telegram bot, it was found that the number of users, depending on the day of the week, reaches an average of 80 per day, thus reducing the load on the university's counseling center and providing users with the

necessary information in a convenient way. The statistics of switching between Telegram bot menu items are kept, the response time is less than 1 second.

REFERENCES

1. InMind: The most popular messenger in Ukraine is Viber (used by 99% of respondents aged 25-34), followed by Facebook Messenger, Telegram, WhatsApp and Skype ITC, 24.07.2020. Internet access mode: *https://itc.ua/news/inmind-samyj-populyarnyj-messendzher-v-ukraine-viber-im-polzuyutsya-99-oproshennyh-v-vozraste-25-34-let-dalee-idut-facebook-messenger-telegram-whatsapp-i-skype/.*
2. Official website of Telegram. Internet access mode: *https://telegram.org/.*
3. FRANCHINI M., PIERONI S., MARTINI N., RIPOLI A., CHIAPPINO D., DENOTH F., NORMAN LIEBMAN M., MOLINARO S., DELLA LATTA D.: Shifting the Paradigm: The Dress-COV Telegram Bot as a Tool for Participatory Medicine. International Journal of Environmental Research and Public Health. 2020, 17(23), 8786; https://doi.org/10.3390/ijerph17238786.
4. ROSID M.A., RACHMADANY A., MULTAZAM M.T., NANDIYANTO A.B.D, ABDULLAH A.G., WIDIATY I.: Integration Telegram Bot on E-Complaint Applications in College. The 2nd Annual Applied Science and Engineering Conference (AASEC 2017). IOP Conf. Series: Materials Science and Engineering 288 (2018) 012159 doi:10.1088/1757-899X/288/1/012159.
5. The Development of Telegram BOT Through Short Story. Proceedings of the Brawijaya International Conference on Multidisciplinary Sciences and Technology (BICMST 2020). Advances in Social Science, Education and Humanities Research, volume 456. Pp. 209-212.
6. IDHOM M., RONGGO ALIT, WAHANANI H.E.: Implementation System Telegram Bot for Monitoring Linux Server. International Conference on Science and Technology (ICST 2018). Atlantis Highlights in Engineering (AHE), volume 1. Pp. 1089-1093.
7. RASYIDAH NUR AISYAH, DEWI MASITHO ISTIQOMAH, MUHAMMAD MUCHLISIN: Developing E-learning Module by Using Telegram Bot on ICT for ELT Course. Advances in Social Science, Education and Humanities Research, Series Volume Number 534. Proceedings of the 5th International Conference on Arts Language and Culture (ICALC 2020). Pp. 106-111.
8. HARMINTO MULYO. JURNAL DISPROTEK: Integration of Student Academic Information Systems with Telegram Bot as an Automatic Answering Machine. 13(2022)1, 11-20.
9. PENYIRAM TANAMAN OTOMATIS BERBASIS ARDUINO DENGAN KONTROL TELEGRAM. DAFFA EKA NADINDRA, JOKO CHRISTIAN CHANDRA: Sistem IoT. SKANIKA: Sistem Komputer dan Teknik Informatika. Volume 5, Nomor 1, Januari 2022, Halaman 104-114.

10. Sistem Pendukung Keputusan Rekomendasi Penempatan Lokasi ATM Berbasis Bot-Chat Telegram Menggunakan Metode FUZZY-ELECTRE. Mahendra, Rizqi Yahya (2022) Sistem Pendukung Keputusan Rekomendasi Penempatan Lokasi ATM Berbasis Bot-Chat Telegram Menggunakan Metode FUZZY-ELECTRE. Undergraduate thesis, Universitas Pembangunan Nasional "Veteran" Jawa Timur.
11. List of specialties of the National Aviation University. Internet access mode: *https://pk.nau.edu.ua/spetsialnosti/*.