

В. О. Гнатюк¹, І. О. Бондаренко², І. С. Каплун²

¹Національний авіаційний університет
Проспект Любомира Гузара, 1, 03058 Київ, Україна

²Вінницький національний технічний університет
Хмельницьке шосе, 95, 21021 Вінниця, Україна

Використання систем обміну миттєвими повідомленнями для автоматизації надання консультативних послуг

Сучасні системи обміну миттєвими повідомленнями дозволяють швидко обмінюватися повідомленнями та інформаційними ресурсами, зокрема такі системи все частіше використовуються при роботі консультативних центрів, забезпечуючи автоматизацію надання консультативних послуг. Метою роботи є автоматизація надання консультативних послуг із використанням сучасних систем обміну миттєвими повідомленнями, зокрема Telegram. Для цього проаналізовано сучасні системи обміну миттєвими повідомленнями, досліджено роботу ботів, визначено проблемні питання, з якими найчастіше звертаються до консультативних центрів, розроблено та експериментально досліджено Telegram-бот.

Ключові слова: API, система обміну миттєвими повідомленнями, СОМП, Telegram-бот, консультативний центр, КЦ.

Вступ

На сучасному етапі розвитку суспільство не уявляє своє життя без використання інформаційних технологій, що застосовуються в усіх сферах діяльності людини. Важливою частиною життя сучасної людини є використання систем обміну миттєвими повідомленнями (СОМП), які дозволяють швидко отримувати необхідну інформацію, обмінюватися повідомленнями, файлами, зображеннями, звуковими сигналами, створювати та використовувати групові чати, здійснювати дзвінки, опитування тощо. Використання систем обміну миттєвими повідомленнями набуває неабиякого використання і при роботі консультативних центрів, забезпечуючи автоматизацію надання консультативних послуг.

Огляд літератури

Серед основних служб для обміну повідомленнями виділяють: Skype, Telegram, Viber, Facebook Messenger, WhatsApp тощо. Найбільш популярними СОМП в Україні у 2020–2021 роках були Viber, яким користувалися 99 % користувачів смартфонів у віці від 13 до 55 років і Facebook Messenger, Telegram; WhatsApp та Skype поступово втрачають користувачів, ними користуються все менше, в той же час популярність Telegram зростає [1], особливо серед молоді. Відповідно до [2] Telegram являє собою багатоплатформний клауд-месенджер з функціями VoIP для смартфонів, планшетів і ПК, який дозволяє обмінюватися текстовими, голосовими та відеоповідомленнями, наліпками та фотографіями, файлами багатьох форматів. Також має функції відео- і аудіодзвінків, організації відеоконференцій у групах і каналах. Клієнтські програми Telegram доступні для Android, iOS, Windows Phone, Windows, macOS і GNU / Linux. Кількість щомісячних активних користувачів сервісу станом на січень 2021 року становить близько 500 млн. осіб. Окрім обміну повідомленнями в діалогах і групах, у месенджері можна зберігати необмежену кількість файлів, вести канали (мікроблоги), створювати та використовувати ботів. Боти являють собою сторонні програми, які працюють всередині Telegram. Користувачі можуть взаємодіяти з ботами, надсилаючи їм повідомлення, команди та вбудовані запити. Управління ботами здійснюється за допомогою HTTP-запитів до API-ботів Telegram. Також, Telegram дедалі частіше інтегрується з різноманітними електронними системами, IoT тощо [3–10].

Сьогодні заклад вищої освіти (ЗВО), відповідно до Закону України «Про вищу освіту» являє собою окремий вид установи, яка є юридичною особою приватного або публічного права, діє згідно з виданою ліцензією на провадження освітньої діяльності на певних рівнях вищої освіти, проводить наукову, науково-технічну, інноваційну та/або методичну діяльність, забезпечує організацію освітнього процесу і здобуття особами вищої освіти, післядипломної освіти з урахуванням їхніх покликань, інтересів і здібностей. Під час вступу до ЗВО варто керуватися умовами та правилами вступу, проте, зважаючи на велику кількість інформації, з якою необхідно ознайомитися вступникам та її складністю, у вступників виникає значна кількість запитань. Зважаючи на це, у структурах ЗВО функціонують консультаційні центри (КЦ), але в пікові періоди, за активної фази вступної кампанії, кількість звернень вступників зростає у сотні, тисячі разів. Для прикладу на рис. 1. відображено статистику відвідування веб-сайту приймальної комісії ЗВО України за 2021 рік, де можна побачити у пікові періоди до 60 000 відвідувань за добу.

Подібний приріст звернень спостерігається і при роботі КЦ, як наслідок, працівники КЦ фізично не можуть здійснити обслуговування всіх звернень. З огляду на це, виникає актуальна науково-практична задача автоматизації надання консультативних послуг щодо вступу до університету.

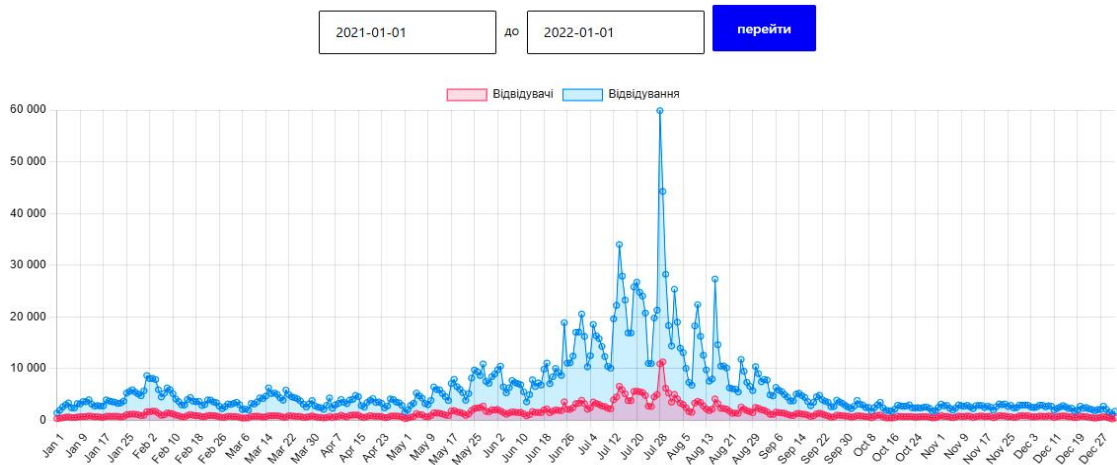


Рис. 1. Статистика відвідування веб-сайту приймальної комісії провідного ЗВО України за 2021 рік

Мета роботи

Зважаючи на зростаючу високу популярність Telegram серед молоді, простий та інтуїтивний інтерфейс, забезпечення конфіденційності даних, забезпечення доступу з кількох пристроїв одночасно, відсутності обмежень у розмірі мультимедіа та чатів, наявність відкритого вихідного коду та API для розробників, захищеність від перехоплення та зламу, а також наявність широкого функціоналу метою роботи є автоматизація надання консультативних послуг щодо вступу до університету з використанням сучасних систем обміну миттєвими повідомленнями, використовуючи Telegram.

Розробка Telegram-бота

При розробці Telegram-бота насамперед інстальємо інтегроване середовище розробки (Integrated development environment — IDE). Середовищем розробки обрано IntelliJ IDEA — комерційне інтегроване середовище розробки для різних мов програмування (Java, Kotlin, Python, Scala, PHP тощо) від компанії JetBrains. Система поставляється у вигляді урізаної за функціональністю безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну ліцензію. Сирцеві тексти Community-версії поширюються в рамках ліцензії Apache 2.0. Бінарні збірки підготовлені для Linux, Mac OS X і Windows.

Як інструмент автоматизації збірки використовуємо Maven. Проєкти Maven налаштовуються за допомогою об'єктної моделі проєкту (POM), яка зберігається у pom.xml-файлі (рис. 2).

Maven розглядає два аспекти створення програмного забезпечення — спосіб його створення та залежності. На відміну від попередніх інструментів, таких як Apache Ant, він використовує умовні правила для процедури збірки. Необхідно вказати лише винятки. Файл XML описує програмний проєкт, який створюється, його залежності від інших зовнішніх модулів і компонентів, порядок збірки, каталоги та необхідні плагіни. Він поставляється з попередньо визначеними цілями для виконання певних чітко визначених завдань, таких як компіляція коду та його

пакування. Maven динамічно завантажує бібліотеки Java та плагіни Maven з одного або кількох сховищ, таких як центральний репозиторій Maven 2, і зберігає їх у локальному кеші. Цей локальний кеш завантажених артефактів також можна оновлювати за допомогою артефактів, створених локальними проектами.

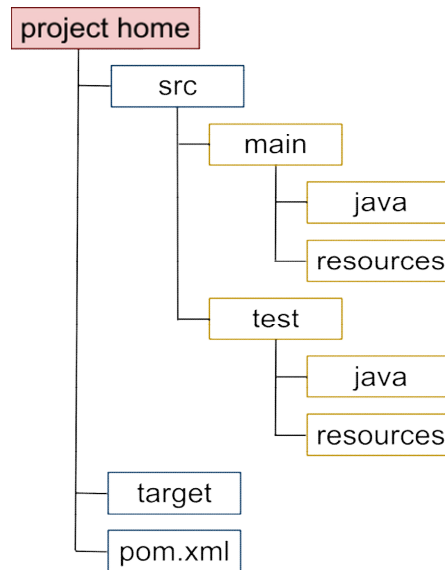


Рис. 2. Структура каталогів для проекту Java, яка автоматично створена Maven

Загальнодоступні репозиторії також можна оновлювати. Maven створено з використанням архітектури на основі плагінів, що дозволяє використовувати будь-яку програму, якою можна керувати за допомогою стандартного введення. Для Maven 2 підтримується рідний плагін C / C++.

У pom.xml-файлі підключаємо dependency (рис. 3).

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

Рис. 3. Використання центрального репозиторію Maven

Для розробки використаємо об'єктно-орієнтовану мову програмування Kotlin, що працює поверх Java Virtual Machine і розробляється компанією JetBrains. У pom.xml-файлі підключаємо Kotlin (рис. 4).

Наступним кроком створюємо клас Bot, що буде успадковувати клас TelegramLongPollingBot(). Далі створюємо методи:

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

```
<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>
```

Рис. 4. Залежність Kotlin у pom.xml-файлі

Задля отримання значень «BotToken» та «NameBot» використовуємо Telegram BotFather, реєструємо Telegram-бота і отримуємо необхідні «BotToken» та «NameBot». Після чого можемо їх вказати у методах `getBotToken()` та `getBotUsername()`.

Здійснюємо ініціалізацію та реєстрацію бота (рис. 5).

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

Рис. 5. Ініціалізація та реєстрація бота

Далі реалізуємо метод `onUpdateReceived(update: Update)`. Для цього використаємо глобальну команду `/start`, яка починає «спілкування» з користувачем (наприклад, надсилає вітальне повідомлення). Для обробки `/start` використаємо умовний вираз із кількома гілками `when`, схожий на `switch`-твердження в С-подібних мовах програмування. `When` перевіряє співпадіння зі своїм аргументом за всіма гілками послідовно, доки не буде виконано деяку умову розгалуження. `When` можна використовувати як вираз або як твердження. Якщо він використовується як вираз, значення першої відповідної гілки стає значенням загального виразу. Якщо він використовується як оператор, значення окремих гілок ігноруються. Так само, як і в `if` випадку, кожна гілка може бути блоком, а її значення є значенням останнього виразу в блоці. `Else`-відгалуження виконується, якщо жодна з інших умов розгалуження не задовольняється (наприклад, рис. 6).

```
when (x) {
    1 -> print("x == 1")
    2 -> print("x == 2")
    else -> {
        print("x is neither 1 nor 2")
    }
}
```

Рис. 6. Приклад використання умовного виразу when

Реалізація методу `onUpdateReceived(update: Update)` з використанням `when` виглядає таким чином:

```
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")
        )
    }
}
```

Рис. 7. Реалізація методу `onUpdateReceived(update: Update)`, `/start`

```
" Button1" -> {
    val chatId1 = update.message.chatId
    execute(SendMessage()
        .setText("□" + " Крок 1" +
            "\n" +
            "□" + " Крок 2. " +
            "\n" +
            "□" + " Крок ... " +
            "\n" +
            "□" + " Крок n. ")
        .setChatId(chatId1)
        .setReplyMarkup(
            InlineKeyboardMarkup()
        )
        apply {
            keyboard = listOf(listOf(
                InlineKeyboardButton("Text")
            ))
            apply {
                url = "https://..."
            }
        }
    )
}
```

Рис. 8. Реалізація методу `onUpdateReceived(update: Update)`, `Button1`

Аналогічним чином (рис. 8) здійснюється реалізація методу `onUpdateReceived(update: Update)` для інших кнопок згідно таблиці.

Провівши аналіз питань, з якими найчастіше звертаються вступники до консультативного центру було розроблено Telegram-бот, який надає інформацію про: послідовність кроків (алгоритм дій) при вступі до університету за ОС «Бакалавр» та ОС «Магістр»; спеціальності (освітні програми) університету, за якими здійснюється підготовка фахівців ОС «Бакалавр» та ОС «Магістр»; вартість і термін навчання; правила прийому до університету; освітні центри «Донбас-Україна» та «Крим-Україна»; поширені запитання-відповіді про вступ; корисна інформація вступникам і контакти приймальної комісії (адреса, номери телефонів, графік роботи консультативного центру тощо) (див. таблицю).

Назви кнопок і результат їхнього спрацювання

№	Назва кнопки	Результат спрацювання
1	Алгоритм вступу ОС «Бакалавр»	Відображення покрокового алгоритму вступу за ОС «Бакалавр» Детальніше ознайомитися з алгоритмом вступу до бакалаврату університету можна за посиланням
2	Алгоритм вступу ОС «Магістр»	Відображення покрокового алгоритму вступу за ОС «Магістр» Детальніше ознайомитися з алгоритмом вступу до магістратури університету можна за посиланням
3	Спеціальності університету	Перелік спеціальностей університету Введіть шифр спеціальності (наприклад: 172) та натисніть кнопку відправити, для отримання детальної інформації про освітні програми та вартість навчання Детальна інформація про спеціальності університету (за посиланням)
4	Правила прийому до університету	Правила прийому до університету за посиланням
5	Освітні центри	Відображення пунктів меню 5.1–5.4
5.1	Крим-Україна	Відповідальна особа за роботу освітнього центру Адреса освітнього центру «Крим-Україна» Контакти Графік роботи центру Подати заяву на реєстрацію для участі у вступних випробуваннях для вступників ОЦ «Крим-Україна»
5.2	Донбас-Україна	Відповідальна особа за роботу освітнього центру Адреса освітнього центру «Донбас-Україна» Контакти Графік роботи центру Подати заяву на реєстрацію для участі у вступних випробуваннях для вступників ОЦ «Крим-Україна»
5.3	Загальноосвітній навчальний заклад «Школа Екстернів»	Контакти Загальноосвітнього навчального закладу «Школа Екстернів» Детальна інформація для вступників з ТОТ (за посиланням)
5.4	Повернутися до головного меню	Відображення головного меню
6	Поширені запитання-відповіді	Відображення пунктів меню 6.1-6.3
6.1	ОС Бакалавр	Відображення переліку поширених питань з відповідями при вступі на ОС «Бакалавр»
6.2	ОС Магістр	Відображення переліку поширених питань з відповідями при вступі на ОС «Магістр»
6.3	Повернутися до головного меню	Відображення головного меню
7	Корисна інформація	Відображення корисної інформації вступникам
8	Контакти приймальної комісії університету	Графік роботи консультаційного центру Електронна адреса Номери телефонів Детальна інформація про контакти приймальної комісії університету (за посиланням)

Використовуючи дані таблиці та реалізацію методу onUpdateReceived (update: Update) (рис. 3–8) було розроблено клавіатуру Telegram-бота (рис. 9), що активується глобальною командою /start.

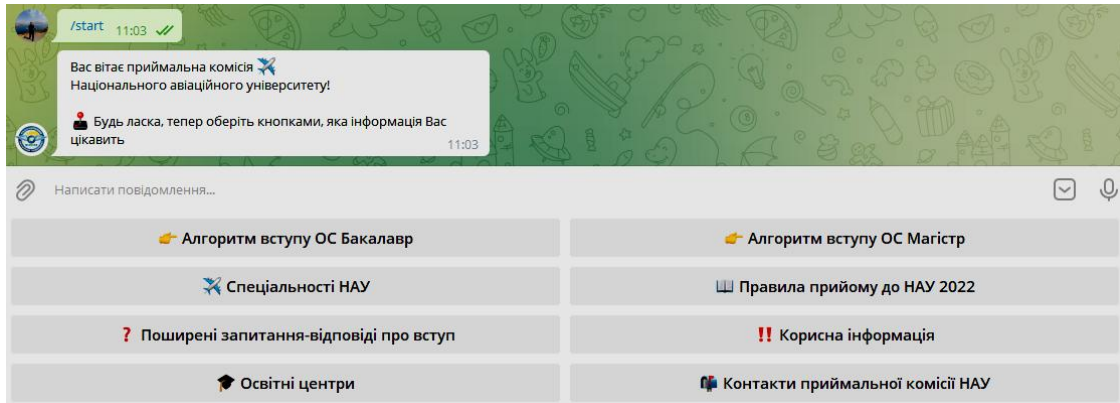


Рис. 9. Клавіатура Telegram-бота

Окрім клавіатури Telegram-бота було розроблено меню спеціальностей університету для відображення інформації за кожною окремою спеціальністю, зокрема: освітні ступені, назви освітніх програм, форми навчання, вартість навчання за освітнім ступенем і посилання на веб-сторінку спеціальності.

Для реалізації меню спеціальностей задамо їхню множину S :

$$S = \left\{ \bigcup_{j=1}^m S_j \right\} = \{S_1, S_2, \dots, S_m\}, \quad (1)$$

де $S_j \subseteq S$, $(j = \overline{1, m})$, m — кількість спеціальностей університету.

Отже, використовуючи вираз (1) та дані [11], при $m = 49$ отримаємо

$$S_{NAU} = \left\{ \bigcup_{j=1}^{49} S_j \right\} = \{S_1, S_2, \dots, S_{49}\} = \{011, 022, \dots, 293\}, \quad (2)$$

де $S_1 = S_{011} = 011$, $S_2 = S_{022} = 022$ — спеціальності (шифри) Національного авіаційного університету.

Використовуючи вираз (2), реалізуємо метод onUpdateReceived(update: Update) для кожної спеціальності за прикладом рис. 8.

У результаті реалізації методу onUpdateReceived(update: Update) для шифру спеціальності, наприклад 172, отримаємо відображення інформації про спеціальність за її шифром (рис. 10).

Використовуючи вираз (2) при реалізації методу onUpdateReceived(update: Update), можемо за прикладом (рис. 10) для кожної спеціальності відображувати інформацію про освітні ступені, назви освітніх програм, форми навчання, вартість навчання за освітнім ступенем і посилання на веб-сторінку спеціальності.

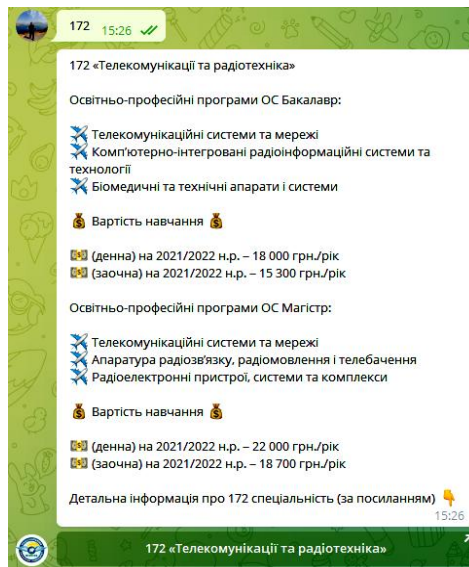


Рис. 10. Відображення інформації про спеціальність за її шифром

За допомогою графічної нотації UML візуалізуємо розроблене програмне забезпечення (рис. 11–14).

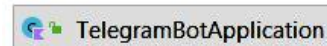
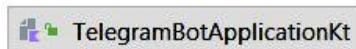
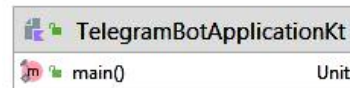
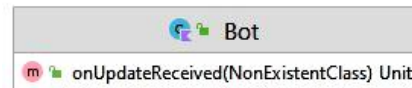


Рис. 11. Класи з конструкторами

Рис. 12. Класи з методами

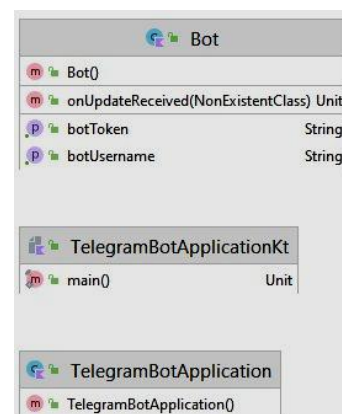
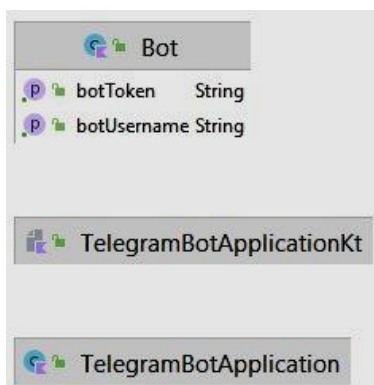


Рис. 13. Класи з властивостями

Рис. 14. Класи з полями, конструкторами, методами та властивостями

Висновки

Вирішено актуальну науково-практичну задачу автоматизації надання консультативних послуг з використанням сучасних СОМП. Для цього було розроблено Telegram-бота, подібних Telegram-ботів у провідних ЗВО немає. У результаті експериментального дослідження Telegram-бота виявлено, що кількість користувачів, залежно від дня тижня, у середньому сягає 80 на добу. Таким чином, це дозволяє зменшити навантаження на консультаційний центр університету та у зручній формі надавати користувачам необхідну інформацію. Ведеться статистика перемикання між пунктами меню Telegram-бота, час відповіді менше 1 секунди.

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. URL: <https://itc.ua/news/inmind-samyj-populyarnyj-messendzher-v-ukraine-viber-im-polzuyutsya-99-oproshennyh-v-vozraste-25-34-let-dalec-idut-facebook-messenger-telegram-whatsapp-i-skype/>

2. Official website of Telegram. Internet access mode: <https://telegram.org/>

3. Shifting the Paradigm: The Dress-COV Telegram Bot as a Tool for Participatory Medicine. Michela Franchini, Stefania Pieroni, Nicola Martini, Andrea Ripoli, Dante Chiappino, Francesca Denoth, Michael Norman Liebman, Sabrina Molinaro and Daniele Della Latta. *International Journal of Environmental Research and Public Health*. 2020. **17**(23), 8786. URL: <https://doi.org/10.3390/ijerph17238786>.

4. Integration Telegram Bot on E-Complaint Applications in College. M.A. Rosid, A.Rachmadany, M.T. Multazam, A.B.D Nandiyanto, A.G. Abdullah, and I. Widiaty. 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. Vol. 456. P. 209–212.

6. Implementation System Telegram Bot for Monitoring Linux Server. Mohammad Idhom, Ronggo Alit, Henni Endah Wahanani. International Conference on Science and Technology (ICST 2018). Atlantis Highlights in Engineering (AHE). Vol. 1. P. 1089–1093.

7. Developing E-learning Module by Using Telegram Bot on ICT for ELT Course. Rasyidah Nur Aisyah, Dewi Masitho Istiqomah, Muhammad Muchlisin. 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). P. 106–111.

8. Integration of Student Academic Information Systems with Telegram Bot as an Automatic Answering Machine. Harminto Mulyo. *Jurnal Disprotek*. Januari 2022. Vol 13, No. 1. P. 11–20.

9. Sistem IoT Penyiram Tanaman Otomatis Berbasis Arduino Dengan Kontrol Telegram. Daffa Eka Nadindra, Joko Christian Chandra. SKANIKA: Sistem Komputer dan Teknik Informatika. Januari 2022. Vol. 5, No. 1. 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/>

Надійшла до редакції 15.11.2021