

Система обміну миттєвими повідомленнями для автоматизації надання консультативних послуг авіаційного підприємства

У роботі було проаналізовано сучасні системи обміну миттєвими повідомленнями, досліджено роботу ботів, визначено проблемні питання, з якими найчастіше звертаються до консультаційних центрів, розроблено та експериментально досліджено Telegram-бота.

Вступ.

На сучасному етапі розвитку суспільство не уявляє своє життя без використання інформаційних технологій, що застосовуються у всіх 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 і використовувати групові чати, здійснювати дзвінки, опитування тощо. Серед основних служб для обміну повідомленнями виділяють: Skype, Telegram, Viber, Facebook Messenger, WhatsApp тощо. Найпопулярнішими 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 менше половини користувачів, в той же час популярність Telegram зростає [1], особливо серед молоді. Відповідно до [2] Telegram являє собою багатоплатформовий клауд-месенджер з функціями VoIP для смартфонів, планшетів та ПК, який дозволяє обмінюватися текстовими, голосовими та відеоповідомленнями, наліпками та фотографіями, файлами багатьох форматів. Управління ботами здійснюється за допомогою HTTPS-запитів до API ботів Telegram. Також, Telegram дедалі частіше інтегрується з різноманітними електронними системами, IoT тощо [3-10].

Сучасне авіаційне підприємство здійснює велику кількість завдань, серед яких, надання консультативних послуг. Для автоматизації цього процесу у роботі розроблено та експериментально досліджено Telegram-бота.

Розробка Telegram-бота.

При розробці Telegram-бота насамперед інсталиємо 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.

У якості інструменту автоматизації збірки використовуємо Maven. Проекти Maven налаштовуються за допомогою об'єктної моделі проекту (POM), яка зберігається у pom.xml-файлі (рис. 1).

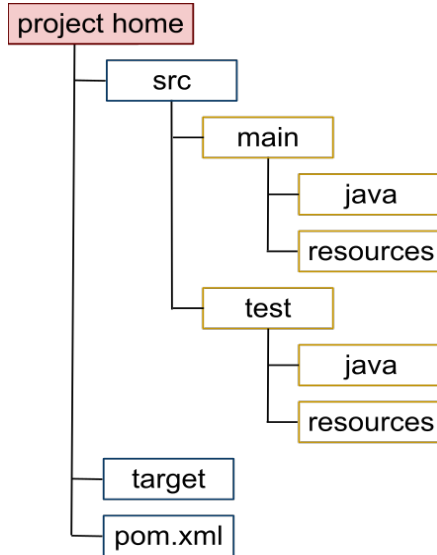


Рис. 1. Структура каталогів для проекту Java, автоматично створена Maven

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

У pom.xml-файлі підключаємо dependency (рис. 2).

```

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

```

Рис. 2. Використання центрального репозиторію Maven

Для розробки буде використано об'єктно-орієнтована мова програмування Kotlin, що працює поверх Java Virtual Machine і розробляється компанією JetBrains. У pom.xml-файлі підключаємо Kotlin (рис. 3).

```

<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. 3. Залежність Kotlin у pom.xml-файлі

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) {...}.

```

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

Далі здійснюємо ініціалізацію та реєстрацію бота (рис. 4).

```

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

```

Рис. 4. 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.

Завершальним етапом створюємо меню сервісів із заздалегідь запрограмованими відповідями на питання користувачів.

Для реалізації меню задамо множину сервісів 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 сервісів.

У роботі вирішена актуальна науково-практична задача автоматизації надання консультативних послуг з використанням сучасних СОМП. Для цього було проаналізовано сучасні СОМП, досліджено роботу ботів, визначено проблемні питання, з якими найчастіше звертаються до консультаційних центрів авіаційних підприємств, розроблено та експериментально досліджено Telegram-бота.

Список літератури

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 ІТС, 24.07.2020. Режим досту-пу в Інтернет: <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; <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, volume 456. Pp. 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), volume 1. Pp. 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). Pp. 106-111.
8. Integration of Student Academic Information Systems with Telegram Bot as an Automatic Answering Machine. Harminto Mulyo. Jurnal Disprotek, Vol 13, No. 1, Januari 2022, Pp. 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. 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.