УПРАВЛІННЯ В ОРГАНІЗАЦІЙНИХ СИСТЕМАХ

MANAGEMENT IN ORGANIZATIONAL SYSTEMS

DOI: 10.20998/2079-0023.2025.01.03
UDC 004.8:378:004.91

O. V. IVASHCHENKO, Doctor of Philosophy (PhD), National Technical University "Kharkiv Polytechnic Institute", Associate Professor at the Department of Software Engineering and Management Intelligent Technologies, Kharkiv, Ukraine; e mail: oksana.ivashchenko@khpi.edu.ua; ORCID: https://orcid.org/0000-0003-3636-3914
S. FILIP, Eng., Ph.D., Associate Professor, Bratislava University of Economics and Management, Bratislava, Slovakia; e mail: stanislav.filip@vsemba.sk; ORCID: https://orcid.org/0000-0003-3000-9383
B. V. RATUSHNYI, Student, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; e-mail: bohdan.ratushnyi@cs.khpi.edu.ua

DEVELOPMENT OF AN EDUCATIONAL CHATBOT WITH A CONTEXTUAL INTENT SYSTEM ON THE DIALOGFLOW PLATFORM

In the context of digital transformation in higher education, the development of intelligent agents capable of maintaining continuous and effective interaction with students is becoming increasingly relevant. This article presents a complete life cycle of the creation of the contextual chatbot “Pytayko z PIITU” for the Department of Software Engineering and Intelligent Control Technologies of NTU “KhPI”. The chatbot is designed to provide quick and intuitive access to information about academic procedures, communication channels, scholarships, documents, and other common questions related to students' interaction with the department and its website. The system was developed using the Dialogflow platform with Telegram integration and Google Cloud Functions as the fulfillment handler. The core of the system is a structured multi-level intent architecture, where each intent group corresponds to a thematic category such as admissions, documents, or course schedules. This allows the bot to maintain conversation context, ensure precise routing of requests, and reduce ambiguity in user interaction. The prototyping model was selected as the life cycle methodology due to the need for active user feedback and iterative improvement. Based on the analysis of the departmental website and survey data from students, an intent system was created that organizes user queries by categories, each with its own fallback intent and context-based clarification mechanisms. Special attention was paid to the dynamic distribution of queries using webhook logic and centralized reusable intent blocks. The article presents the development algorithm, intent architecture, testing process, and analysis of interaction history. The testing phase included multiple validation cycles, real-time sessions via Telegram, and the assessment of fallback effectiveness. The final implementation achieved a high accuracy rate (~91%) and low error percentage (~3 %), demonstrating the feasibility of using Dialogflow for educational automation scenarios. The chatbot architecture supports future scalability and provides 24/7 support for student inquiries without additional administrative workload

Keywords: chatbot development, Dialogflow, higher education, student services, prototyping model, intent system, fallback logic, natural language understanding, educational automation, Telegram integration.

Introduction. In the context of digitalization of higher education and increasing competition among universities, institutions are facing numerous new challenges – one of the most significant being the need to ensure prompt, accessible, and personalized communication with students and prospective applicants. As noted in studies [1–2], the digital learning environment has become a key component of university transformation, while the development of “smart universities” through the implementation of AI technologies is emerging as a strategic priority for many higher education institutions.

One of the most effective tools supporting this transformation is the implementation of intelligent chatbots. These systems significantly reduce administrative workload, enable 24/7 communication, and improve the speed and quality of responses by automating typical student inquiries.

Given this, the aim of the present study was to develop a contextual chatbot for the Department of Software Engineering and Management Intelligent Technologies at NTU “KhPI” – named “Pytayko z PIITU”. The solution is designed with the potential for further adaptation and transfer of the developed design, development, testing, and implementation approaches to a similar project for the Bratislava University of Economics and Management. The chatbot’s functionality is based on the Dialogflow platform, which supports natural language processing and is integrated with the Telegram messenger.

The analysis of the department’s website revealed that, despite its structured architecture and comprehensive content covering various departmental activities, users (particularly first-year students and applicants) face difficulties in locating up-to-date information – a finding confirmed by survey results. Therefore, using a chatbot as an interface to the department’s knowledge base is a justified and effective decision.

The core logic of the chatbot is implemented through a structured system of intents, developed based on an analysis of the department's website, patterns of user interaction with digital services, and stakeholder survey results. Through the integration of both general and category-specific fallback intents, as well as dynamic query routing using webhook functions (Google Cloud Functions), the system demonstrated improved response relevance even when user input was incomplete or imprecise.

Thus, the development of the chatbot presented in this study not only aligns with current trends in the digital transformation of higher education [3–4], but also contributes to improving the quality of departmental services by providing users with convenient, intuitive, and accessible access to essential information.

Literature review. According to the analysis of literature and practical implementations, there are two main approaches to chatbot development. The first approach is based on the use of programming languages such as Java, Python, PHP, C++, Ruby, Lisp, etc., which requires a high level of technical expertise and significant development time, but provides maximum flexibility, logic customization, and integration capabilities [5–6].

The second approach involves the use of specialized platforms and frameworks that provide pre-built components for intent recognition, natural language processing (NLP/NLU), fulfillment handling, analytics, and integration with popular messaging platforms. These platforms allow developers to focus on the content and dialogue design rather than technical infrastructure. Given the available functionality, this method ensures high-quality implementation and is particularly suitable for developing educational and administrative chatbots that address frequently asked questions and assist users in navigating university websites [5, 7, 8].

The integration of NLP for clarification and fallback logic corresponds to methods for textual analysis and comprehensibility evaluation of structured business models, as demonstrated by [9].

Among the most widely used platforms are:

1. Google Dialogflow [10] – a powerful platform supporting multi-channel integration (Telegram, Facebook Messenger, websites, etc.), featuring a visual interface and webhook support.

2. Rasa [11–12] – an open-source framework that combines flexible natural language understanding (NLU) and dialogue management (Core). Intent classification is based on the fastText mechanism using pre-trained word vectors (e.g., GloVe), ensuring robustness even with a limited number of training examples. Rasa offers high customizability but requires more effort to implement.

3. Microsoft Bot Framework [13], Amazon Lex [14], and IBM Watson Assistant [15] – commercial cloud-based solutions with strong integration into their respective ecosystems (Azure, AWS, IBM Cloud).

Based on a comparison of five chatbot development platforms (Dialogflow, Microsoft Bot Framework, Watson Assistant, Rasa, and Amazon Lex) using eight criteria – visual dialogue management, availability of prebuilt agents, number of integration channels, knowledge engine support, language coverage, development flexibility, support for modular architecture, and testing capabilities – the following conclusions can be drawn regarding functionnality, developer usability, and integration capabilities [8]:

1. Dialogflow shows a high level of support for visual dialogue management, the widest multilingual support (over 120 languages), and a rich library of prebuilt agents. In addition, it offers convenient tools for agent testing and validation, such as a built-in simulator.

2. Rasa, in contrast to other platforms, lacks a visual interface and built-in templates, but excels in flexibility due to its open-source nature. It is an ideal choice for developers who require on-premises deployment and full control over chatbot logic.

3. Microsoft and Watson Assistant offer robust visual design tools and integration features. Microsoft, in particular, supports the greatest number of communication channels. However, compared to Dialogflow, both platforms may require higher technical effort and development costs.

4. Amazon Lex, despite its more limited language support and basic testing features, remains competitive due to its seamless integration with AWS services, simple pricing model, and support for voice channels.

In addition to commercial and open-source chatbot development platforms, several studies have explored

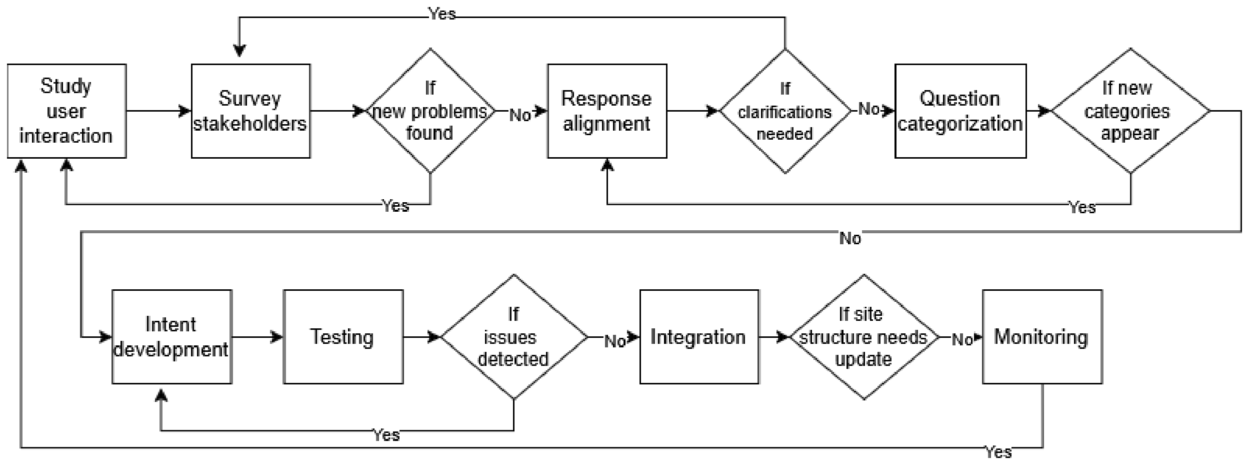


Fig. 1. The proposed algorithm for developing a chatbot

hybrid and expert-system-based approaches for chatbot integration with mobile platforms and external knowledge bases. For example, [16] proposed the use of Telegram chatbots integrated with rule-based expert systems via the Karkas platform, which enables layered knowledge representation and logical inference in asynchronous environments [16]

As a result, the choice of Google Dialogflow in this project is based on the following factors:

- simplicity and speed of prototyping, which align with the chosen development life cycle model;
- availability of built-in NLU components;
- support for context-aware dialogues (input/output contexts);
- direct integration with Telegram via webhook;
- successful use in similar educational chatbot projects [6].

The developed chatbot is intended to handle frequently asked questions related to student interaction with the department and its website. Therefore, a platform that allows rapid intent updates and supports fallback logic is the optimal solution.

Chatbot development: prototyping approach and creation algorithm. To develop the chatbot "Pytayko from PIITU", which is designed to answer the most common and contextually relevant questions frequently encountered by applicants, students, and other users – many of which relate to information available on the PIITU department website [17] – the prototyping life cycle model was chosen. This approach allows you to quickly create a chatbot prototype, receive feedback from users in real time and, if necessary, improve the functionality of the solution taking into account the needs and feedback of users. As noted in [18–19], such a model is effective in conditions of uncertain or dynamic requirements for the developed system, in conditions where the system interface is oriented towards the end user. Also, the prototype model is appropriate in cases where it is important not only to implement basic functionality, but also to test user interaction scenarios with the system, adaptability of responses and ease of use, which is key for interfaces based on natural language processing, such as chatbots.

The proposed algorithm for developing a chatbot is consistent with the typical stages of the information system life cycle:

1. User needs analysis – a study of interaction with the site was conducted, information was collected through surveys and interviews with stakeholders.
2. Requirements formulation – problems were classified, typical requests were agreed upon and grouped by main areas (introduction, training, practice, administrative issues).
3. Bot interface design – an intent structure was created in Dialogflow taking into account contexts and fallback scenarios.
4. Implementation and testing – intents were tested using different query options, for example, queries with errors, the use of synonyms, short and extended questions.
5. Integration – the chatbot was connected to the Telegram platform.
6. Monitoring and improvement – regular updating of the knowledge base is provided based on new user requests and monitoring by department employees.

The developed algorithm is visualized in the form of a flowchart (Fig. 1), which reflects the logic of interaction between stages, providing the opportunity to return to previous stages in the event of uncertainty, new needs, or areas of concern.

As part of the design of a context-aware chatbot system for the department's website, visual modeling was employed – specifically, use case diagrams, which help describe the system's functional requirements, and sequence diagrams, which reflect the dynamic interactions between system components.

The use case diagram captures the main user roles within the system and their interaction scenarios with the chatbot. Three key roles have been identified:

- User – initiates information requests to obtain information or resolve an issue;
- Administrator – is responsible for the development, validation and implementation of intents;
- Department Staff – provides the content of responses to user requests and checks the correctness of the responses.

Several key interaction scenarios are provided:

- forming and sending a request to the system;
- receiving a response;
- developing and maintaining the component.

The development and maintenance of the component involves a number of further actions: filling the knowledge base with the most common queries, setting logical constraints, developing new intents and response templates, validating and updating them, as well as analyzing and processing unrecognized queries. This approach allows you to reflect not only the external behavior of the system, but also the internal processes of maintaining and updating knowledge within the chatbot.

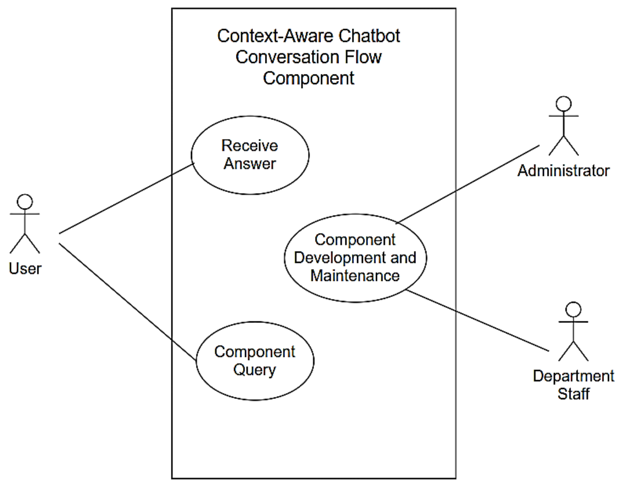

Context-Aware Chatbot Conversation Flow Component
Receive Answer
Component Development and Maintenance
Component Query
User
Administrator
Department Staff

Fig. 2. Use Case Diagram for the Context-Aware Chatbot System

To describe the dynamics of message processing, two sequence diagrams were created, each covering a specific aspect of the functionality of the context-aware chatbot.

Figure 3 presents a sequence diagram illustrating the real-time processing of a user request. This process is essential for the chatbot's operation and is implemented using a modern serverless architecture that ensures scalability, reliability, and minimal infrastructure costs. The user request processing flow includes sending a message via Telegram, intent recognition in Dialogflow, optional invocation of a Webhook, execution of logic in Google Cloud Functions, and finally – response generation and delivery to the user.

The sequence diagram shown below (Fig. 4) describes the internal process of knowledge base formation and maintenance, which underpins the chatbot's functionality. It covers both reactions to new (unrecognized) user requests and the scheduled development of content. Key actors involved in the process include the administrator and department staff. Conceptually, the process is divided into three stages:

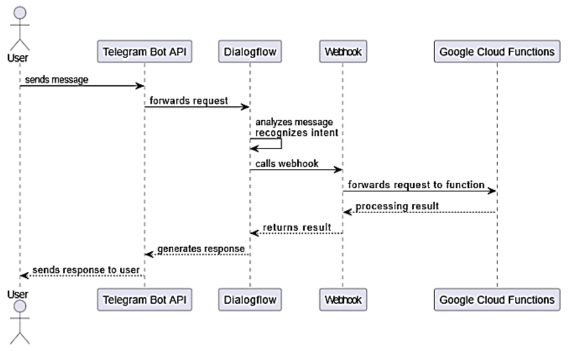


Fig. 3. User Request Processing Sequence Diagram

1. Standard intent development. The department staff provides information related to frequently asked questions, mandatory topics (such as department management contacts), and other content that, in the staff's opinion, must be delivered to chatbot users. Based on this information, the administrator creates intents. These intents then undergo structural validation and content review. If needed, they are revised and improved.

2. Reaction to new user requests. When the chatbot receives a query for which no matching intent is found, the system logs it as an unrecognized request. These requests are subsequently analyzed to improve the system.

3. Periodic analysis of new requests. Selected requests are further reviewed and refined, serving as the basis for developing new intents. These new intents undergo all necessary validation and verification stages before being deployed into the system.

The chatbot is implemented on the Dialogflow platform using a Webhook connection to Google Cloud Functions. The Telegram Bot API is used for message exchange. The architecture supports:

- scalability (via Google Cloud Functions);
- high availability (through cloud infrastructure);
- secure data transmission (HTTPS, IAM);
- seamless updates without downtime (via the Dialogflow Console).

Intent System Development: Architecture, Fallback Mechanisms, and Integration. The development of intents followed methodological principles recommended in Dialogflow documentation and general best practices for building natural language interactions:

- intent core: 5–10 typical user phrases, collected from surveys;
- response format: concise and clear, with a link to the relevant department webpage;
- follow-up intents: clarification logic implemented (e.g., awaiting-course, extracurricular-followup);
- contexts: input/output contexts are used to maintain topic continuity in a conversation;
- fallbacks: implemented on two levels – general and category-specific (e.g., Default Fallback, Contacts.Fallback).

There is the intent example format:

Intent name: Admissions.ProgramDetails.

Trigger phrases: "What programs are available in the department?", "What do students study in the program?".

Response: "The Department of Information and Communication Technology offers the following academic programs: ... [link]".

Context: admissions-followup.

Fallback: A_Admissions.Fallback.

As a result of analyzing the department website structure, observing user behavior, and conducting surveys among students and staff, a hierarchical system of intents was developed in Dialogflow. It classifies typical queries into thematic categories such as admissions, communication, documents, extracurricular activities, scholarships, schedules, contacts, platform support, mobility, and campus partnerships.

Each category includes a set of specific intents (e.g., Admissions.ProgramDetails, Contacts.Department, Statements.Info,) and a dedicated category-specific fallback (e.g., Admissions.Fallback). This structure enables: staying within the current topic scope during conversations; politely prompting users for clarification; maintaining context within the same category.

Unlike a single default fallback intent, each category features its own fallback, allowing the system to handle incorrectly phrased or incomplete queries while maintaining focus on the current theme. For instance, if a student asks "When do classes start?" in an ambiguous form, the Schedule.Fallback intent helps clarify the question without switching to a general fallback, eventually guiding the user to the correct response.

This approach aligns with best practices in contextual conversation design and ensures: dialogue flexibility; fewer routing failures; professional tone even in uncertain scenarios.

One of the key functional features of the system is the implementation of centralized intents, such as CourseSelection and Communication.CourseLink.PhD, which rely on a custom Google Cloud Function triggered via Webhook. This function accepts the student's or PhD candidate's course number as input, processes the data or generates a dynamic response accordingly, and is reused across multiple categories including Communication, Admissions, and Campus.

This supports the reuse across intents principle, which aligns with serverless architecture by separating business logic from static responses.

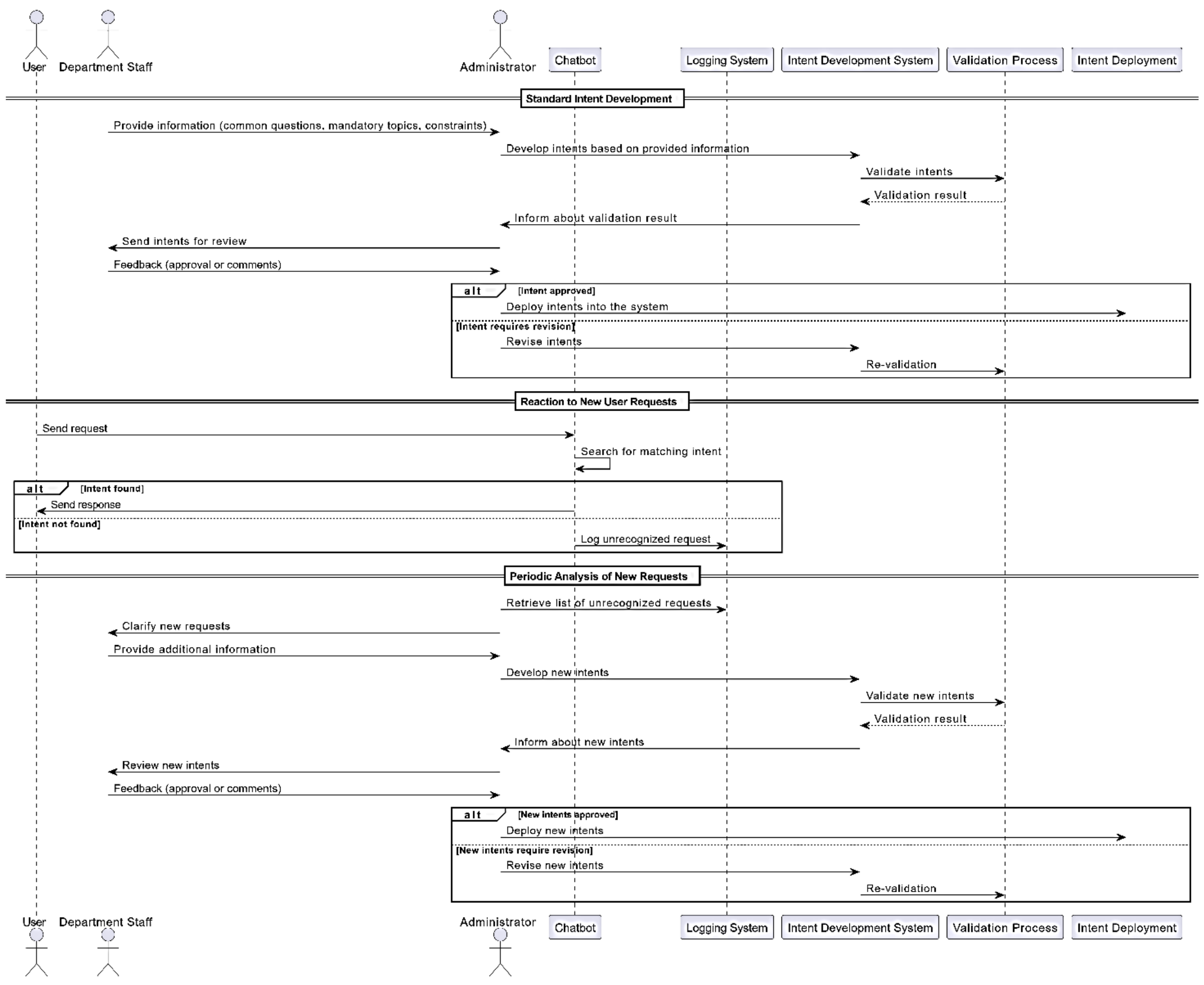


Fig. 4. Sequence Diagram of Knowledge Base Development and Maintenance

The intent architecture implemented in the system features the following key elements:

1. Thematic grouping with multi-level fallback support, ensuring improved dialogue accuracy.

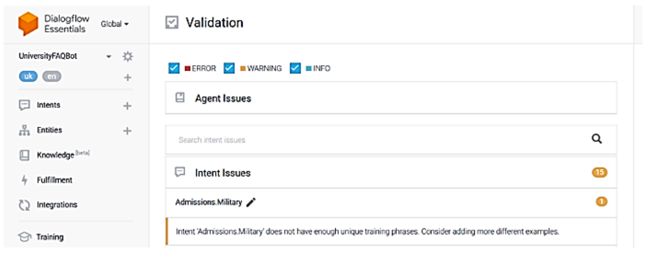
2. Centralized Webhook components that allow scalable bot expansion without logic duplication.

3. Use of contexts and follow-up intents to support natural multi-turn interactions.

4. A modular, reusable, and scalable design – essential for systems dealing with diverse informational topics.

Validation, Testing, and Training of Intents. At the initial stage, following the creation of the base intent structure, the first agent validation was conducted using Dialogflow Essentials. The system automatically flagged several warnings such as "Intent does not have enough unique training phrases," particularly for the Admissions.Military intent (fig. 5). This indicated that certain intents were undertrained and might not account for sufficient variation in user phrasing.

Additionally, during internal testing, thematic overlaps were discovered between intents from different categories – for example, faculty-related questions were sometimes incorrectly classified under "Communication Channels" instead of "Contacts." This posed a risk of ambiguity in response routing.



Intent 'Admissions.Military' does not have enough unique training phrases. Consider adding more different examples.

Fig.5. Dialogflow validation showing a warning for the Admissions.Military intent due to an insufficient number of unique training phrases

To address these issues:

- the number of training phrases per intent was expanded;
- existing phrases were refined for clarity and variety;
- output contexts were revised to reduce topic overlap;
- clarification scenarios were implemented via follow-up intents for semantically similar inputs.

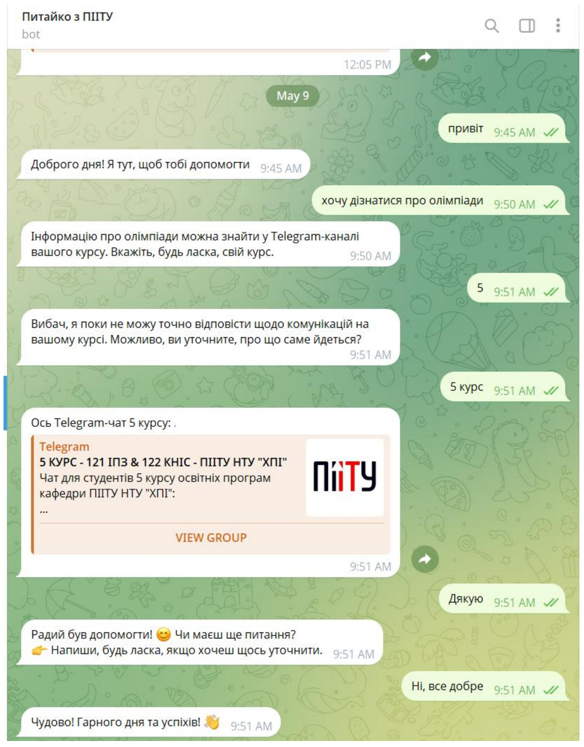
Following the initial validation, several phases of testing were performed, including manual evaluation of intent assignment in Dialogflow, real-world user interaction via Telegram, and training based on actual queries.

Manual testing involved using the Dialogflow Training section to evaluate classification accuracy. The use of contexts was crucial in disambiguating similar phrases and avoiding misclassification. A specific context, awaiting-course, was introduced to support clarification through Webhook logic.

After integration with Telegram, user interactions in a real environment were tested. This revealed instances of inappropriate fallback triggering and helped refine clarifycation flows. As a result, responses for conversation-ending phrases such as Thank.You and No.More.Questions were implemented, allowing for graceful session closure.

Real-user training played a key role as well. The system processed highly varied natural language inputs and gradually learned to correctly classify even short or incomplete queries thanks to an updated training set and refined category-specific fallback intents. This allowed the system to preserve dialogue context and guide users through clarification without breaking conversation logic.

After deploying the chatbot in Telegram, examples of complete multi-turn conversations were documented, demonstrating the effectiveness of the intent architecture, fallback mechanisms, and contextual design. One such conversation (fig. 6) involved a general inquiry: “I want to learn about olympiads.”



Питайко з ПІІТУ
bot

12:05 PM

May 9

привіт 9:45 AM

Доброго дня! Я тут, щоб тобі допомогти 9:45 AM

хочу дізнатися про олімпіади 9:50 AM

Інформацію про олімпіади можна знайти у Telegram-каналі вашого курсу. Вкажіть, будь ласка, свій курс. 9:50 AM

5 9:51 AM

Вибач, я поки не можу точно відповісти щодо комунікацій на вашому курсі. Можливо, ви уточните, про що саме йдеться? 9:51 AM

5 курс 9:51 AM

Ось Telegram-чат 5 курсу: .
Telegram
5 КУРС - 121 ІПЗ & 122 КНІС - ПІІТУ НТУ "ХПІ"
Чат для студентів 5 курсу освітніх програм кафедри ПІІТУ НТУ "ХПІ":
...
VIEW GROUP
9:51 AM

Дякую 9:51 AM

Радий був допомогти! 😊 Чи маєш ще питання?
👉 Напиши, будь ласка, якщо хочеш щось уточнити. 9:51 AM

Ні, все добре 9:51 AM

Чудово! Гарного дня та успіхів! 👋 9:51 AM

Fig. 6. Test Dialogue with the Chatbot via Telegram Interface

The system followed this flow:

1. The bot identified the topic but required additional clarification – the awaiting-course context was triggered.

2. The user entered only “5”, which was too brief, prompting an internal fallback that asked for clarification.

3. After the user responded with “5th year,” the bot correctly identified the course level and dynamically generated a link to the relevant Telegram group.

4. The bot then entered a follow-up state (awaiting-followup) and processed the user’s thank-you message appropriately.

5. The session concluded smoothly, staying within the category and avoiding fallback to a general flow.

The testing process resulted in the following key findings related to intent testing and training:

1. The tests confirmed that a sufficient variety of training phrases, along with analysis of real user input, is critical for accurate classification and response relevance.

2. Identified overlaps between intent categories led to a refined, more segmented context system.

3. Validation in both Dialogflow and Telegram showed that real-world bot behavior requires flexible fallback mechanisms and support for continuous retraining.

4. Each testing phase led to adjustments in intent design, resulting in an iterative and adaptive training process.

5. The combination of intents, contexts, and Webhook logic enabled natural, multi-turn interactions – even when user queries were imprecise or incomplete.

6. Fallback intents function not only as error handlers but as effective tools for refining user intent without disrupting dialogue flow.

Chatbot Usage Analytics. After the chatbot was deployed in the Telegram environment and the testing period began, detailed interaction statistics were collected. These allowed the team to evaluate not only the effecttiveness of individual intents but also general user behavior patterns.

According to Dialogflow analytics collected during the testing period, the system recorded an average of 1 to 2 sessions per day. Each session typically included between 5 and 18 user interactions, with noticeable peaks observed on days of intensive testing.

This confirms that users engaged in full conversations with follow-up clarifications, rather than submitting just a single query.

Table 1 – Intent Activity Distribution during the period 05.05.25–11.05.25

Intent	Sessions	Interactions	Exit, %
CourseSelection	10	15	18.75
Admissions.EntryConditions	4	4	0.00
B_Communication.Channels	4	6	0.00
Extracurricular.Projects	3	4	0.00
Admissions.OnlineSubmission	3	6	6.25
Default Fallback Intent	4	9	0.00

The analysis of the table indicates that the CourseSelection intent was the most frequently triggered and had a relatively high exit rate – a result consistent with its function of redirecting users to course-specific Telegram chats for further information. In contrast, the Default Fallback Intent showed a 0 % exit rate, suggesting that the system successfully managed even unrecognized queries. Notably, no fallback intent led to the termination of a session, which confirms the effectiveness of the category-specific fallback mechanisms in maintaining user engagement.

The Dialogflow History section revealed that the most frequent user queries were related to admissions – including questions such as "admission", "deadlines", and "how to submit documents" – as well as inquiries about department contact details, communication channels with lecturers, project-based learning, digital signatures, and scholarships. These findings confirm that the chatbot's intent structure is well aligned with real user needs, as identified through stakeholder surveys and website analytics.

Conclusions. The choice of a prototyping model was justified, as it enabled a flexible and iterative development process with user feedback incorporated at each stage:

1. The website structure was analyzed, stakeholder surveys were conducted, and common user queries were identified – these became the foundation for the intent architecture.

2. A system of over 30 active intents was developed, covering key informational categories: admissions, documents, schedules, contacts, scholarships, mobility, communication channels, support platforms, and extracurricular activities.

3. More than 100 unique queries were processed during testing, allowing the intent structure to be adapted to real user needs.

4. A multi-level fallback logic was implemented, including both a general fallback for unrecognized queries and category-specific fallbacks. This allows the system to request clarifications within the context of the conversation, maintaining a coherent dialogue flow.

5. The chatbot was integrated with Telegram, providing students with an intuitive mobile interface for interaction.

6. Several rounds of validation, manual testing, and automated testing were carried out, which helped identify and resolve conflicts between intents.

7. After deployment, analytical monitoring revealed a high accuracy rate (91 %), a low misclassification rate (~3 %), and effective fallback handling. The system was designed such that the chatbot's entire knowledge base is implemented as intents in Dialogflow, allowing it to be updated or extended flexibly without modifying the infrastructure or processing logic.

While still in the pre-deployment phase, the chatbot "Pytayko from PIITU" has been developed as an advanced and scalable solution intended to serve as part of the department's intelligent service ecosystem – capable of providing 24/7 support, adapting to user needs, and reducing the administrative burden through automated responses to frequently asked questions. Notably, this joint development also establishes a solid foundation for future implementation at the Bratislava University of Economics and Management, demonstrating the transferability and relevance of the proposed approach in international academic environments

References

1. Bygstad B., Øvrelid E., Ludvigsen S., Dæhlen M. From dual digitalization to digital learning space: Exploring the digital transformation of higher education. *Computers & Education.* 2022. Vol. 182. Art. 104463. URL: https://www.sciencedirect.com/science/article/pii/S0360131522000343 (access date: 07.05.2025).
2. Díaz-García V., Montero-Navarro A., Rodríguez-Sánchez J.-L., Gallego-Losada R. Digitalization and digital transformation in higher education: A bibliometric analysis. *Frontiers in Psychology.* 2022. Vol. 13. Art. 1081595. DOI: doi.org/10.3389/fpsyg.2022.1081595.
3. Rosak-Szyrocka J. The Era of Digitalization in Education: Where do Universities 4.0 Go? *Management Systems in Production Engineering.* 2024. Vol. 32, issue 1. P. 54–66. DOI: 10.2478/mspe-2024-0006. URL: https://www.researchgate.net/publication/378498067 (access date: 07.05.2025).
4. George B., Wooden O. Managing the Strategic Transformation of Higher Education through AI. *Administrative Sciences.* 2023. Vol. 13, no. 9. Art. 196. DOI: doi.org/10.3390/admsci13090196.
5. Kumar R., Mahmoud M. A. A Review on Chatbot Design and Implementation Techniques. *International Journal of Engineering and Technology.* 2020. Vol. 7, issue 2. P. 2791–2796. URL: https://www.researchgate.net/publication/340793645 (access date: 07.05.2025).
6. Barus S. P., Surijati E. Chatbot with Dialogflow for FAQ Services in Matana University Library. *International Journal of Informatics and Computation (IJICOM).* 2021. Vol. 3, no. 2. P. 68–78. DOI: 10.35842/ijicom.v3i2.75.
7. Nsaif W. S., Salih H. M., Saleh H. H., Al-Nuaimi B. T. Chatbot Development: Framework, Platform, and Assessment Metrics. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM).* 2024. Vol. 27. P. 50–62.
8. Dagkoulis I., Moussiades L. A. Comparative Evaluation of Chatbot Development Platforms. *26th Pan-Hellenic Conference on Informatics (PCI 2022).* ACM, 2022. P. 322–328. DOI: doi.org/10.1145/3575879.3576012.
9. Shevelev, V., & Kopp, A. (2024). Algorithm for comprehensibility evaluation of business process models using natural language processing. *Automation of Technological and Business Processes,* No. 15 (4). P. 76–83. DOI: doi.org/10.15673/atbp.v15i4.2721.
10 *Dialogflow ES documentation.* URL: https://cloud.google.com/dialogflow/es/docs (access date: 07.05.2025).
11. *Introduction to Rasa Open Source & Rasa Pro.* URL: https://legacy-docs-oss.rasa.com/docs/rasa/ (access date: 07.05.2025).
12. Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. Rasa: Open source language understanding and dialogue management. *arXiv preprint.* arXiv:1712.05181. 2017.
13. *Microsoft. My bots.* URL: https://dev.botframework.com/ (access date: 07.05.2025).
14. *Amazon Lex – AI Chat Builder.* URL: https://aws.amazon.com/lex/ (access date: 07.05.2025).
15. *AI Website Chatbot Comparison – 23 chatbots.* URL: https://www.chatlab.com/comparator/chatlab-ibm+watson/?gad_source=1&gad_campaignid=21863828536&gclid=CjwKCAjw_pDBBhBMEiwAmY02NjgZiJypYLZ6o4W76uWSx4mU_aVQRPd2WqoTEE26mpL_4dmO90ek4hoCVv0QAvD_BwE (access date: 07.05.2025).
16. Burdaev V. About the features of developing intelligent systems for the android platform. *Conference proceedings of the VII International Scientific-Practical Conference "Information Technologies in Education, Science and Technology" (ITEST-2024),* (Cherkasy, May 23-24, 2024). Cherkasy: ChSTU, 2024. P. 134–135. https://itest.chdtu.edu.ua/Conference-Proceedings-ITEST-2024_25_06.pdf (access date: 07.05.2025).
17. *Кафедра програмної інженерії та інтелектуальних технологій управління.* URL: https://web.kpi.kharkov.ua/asu/uk/ (access date: 07.05.2025).
18. Aminu H., Ogwueleka F. A. Comparative Study of System Development Life Cycle Models. *Journal of Emerging Technologies and Innovative Research (JETIR).* 2020. Vol. 7, issue 8. P. 200–210. URL: https://www.jetir.org/view?paper=JETIR2008025 (access date: 07.05.2025).
19. Diansyah A. F., Rahman M. R., Handayani R., Nur Cahyo D. D., Utami E. Comparative Analysis of Software Development Lifecycle Methods in Software Development: A Systematic Literature Review. *International Journal of Advances in Data and Information Systems.* 2023. Vol. 4, no. 2. P. 97–106. DOI: 10.25008/ijadis.v4i2.1295.

References (transliterated)

1. Bygstad B., Øvrelid E., Ludvigsen S., Dæhlen M. From dual digitalization to digital learning space: Exploring the digital transformation of higher education. *Computers & Education.* 2022. vol. 182, art. 104463. URL: https://www.sciencedirect.com/science/article/pii/S0360131522000343 (access date: 07.05.2025).

2. Díaz-García V., Montero-Navarro A., Rodríguez-Sánchez J.-L., Gallego-Losada R. Digitalization and digital transformation in higher education: A bibliometric analysis. *Frontiers in Psychology.* 2022, vol, 13. art. 1081595. DOI: doi.org/10.3389/fpsyg.2022.1081595.
3. Rosak-Szyrocka J. The Era of Digitalization in Education: Where do Universities 4.0 Go? *Management Systems in Production Engineering.* 2024, vol. 32, issue 1, pp. 54–66. DOI: 10.2478/mspe-2024-0006. URL: https://www.researchgate.net/publication/378498067 (access date: 07.05.2025).
4. George B., Wooden O. Managing the Strategic Transformation of Higher Education through AI. *Administrative Sciences.* 2023. Vol. 13, no. 9. Art. 196. DOI: doi.org/10.3390/admsci13090196.
5. Kumar R., Mahmoud M. A. A Review on Chatbot Design and Implementation Techniques. *International Journal of Engineering and Technology.* 2020. Vol. 7, issue 2. P. 2791–2796. URL: https://www.researchgate.net/publication/340793645 (access date: 07.05.2025).
6. Barus S. P., Surijati E. Chatbot with Dialogflow for FAQ Services in Matana University Library. *International Journal of Informatics and Computation (IJICOM).* 2021, vol. 3, no. 2, pp. 68–78. DOI: 10.35842/ijicom.v3i2.75.
7. Nsaif W. S., Salih H. M., Saleh H. H., Al-Nuaimi B. T. Chatbot Development: Framework, Platform, and Assessment Metrics. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM).* 2024, vol. 27, pp. 50–62.
8. Dagkoulis I., Moussiades L. A. Comparative Evaluation of Chatbot Development Platforms. *26th Pan-Hellenic Conference on Informatics (PCI 2022).* ACM, 2022, pp. 322–328. DOI: doi.org/10.1145/3575879.3576012.
9. Shevelev, V., & Kopp, A. (2024). Algorithm for comprehensibility evaluation of business process models using natural language processing. *Automation of Technological and Business Processes*. No.15(4), pp.76–83. DOI: doi.org/10.15673/atbp.v15i4.2721.
10 *Dialogflow ES documentation.* URL: https://cloud.google.com/dialogflow/es/docs (access date: 07.05.2025).
11. *Introduction to Rasa Open Source & Rasa Pro.* URL: https://legacy-docs-oss.rasa.com/docs/rasa/ (access date: 07.05.2025).
12. Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. Rasa: Open source language understanding and dialogue management. *arXiv preprint.* arXiv:1712.05181. 2017.
13. *Microsoft. My bots.* URL: https://dev.botframework.com/ (access date: 07.05.2025).
14. *Amazon Lex – AI Chat Builder.* URL: https://aws.amazon.com/lex/ (access date: 07.05.2025).
15. *AI Website Chatbot Comparison – 23 chatbots.* URL: https://www.chatlab.com/comparator/chatlab-ibm+watson/?gad_source=1&gad_campaignid=21863828536&gclid=CjwKCAjw_pDBBhBMEiwAmY02NjgZiJypYLZ6o4W76uWSx4mU_aVQRPd2WqoTEE26mpL_4dmO90ek4hoCVv0QAvD_BwE (access date: 07.05.2025).
16. Burdaev V. About the features of developing intelligent systems for the android platform. *Conference proceedings of the VII International Scientific-Practical Conference "Information Technologies in Education, Science and Technology" (ITEST-2024),* (Cherkasy, May 23-24, 2024). Cherkasy, ChSTU Publ., 2024, pp. 134–135. https://itest.chdtu.edu.ua/Conference-Proceedings-ITEST-2024_25_06.pdf (access date: 07.05.2025).
17. *Department of Software Engineering and Management Intelligent Technologies.* URL: https://web.kpi.kharkov.ua/asu/uk/ (access date: 07.05.2025).
18. Aminu H., Ogwueleka F. A. Comparative Study of System Development Life Cycle Models. *Journal of Emerging Technologies and Innovative Research (JETIR).* 2020, vol. 7, issue 8, pp. 200–210. URL: https://www.jetir.org/view?paper=JETIR2008025 (access date: 07.05.2025).
19. Diansyah A. F., Rahman M. R., Handayani R., Nur Cahyo D. D., Utami E. Comparative Analysis of Software Development Lifecycle Methods in Software Development: A Systematic Literature Review. *International Journal of Advances in Data and Information Systems.* 2023, vol. 4, no. 2, pp. 97–106. DOI: 10.25008/ijadis.v4i2.1295.

Received 19.05.2025

УДК 004.8:378:004.91

О. В. ІВАЩЕНКО, доктор філософії (PhD), Національний технічний університет «Харківський політехнічний інститут», доцент кафедри програмної інженерії та інтелектуальних технологій управління, м. Харків, Україна; e-mail: oksana.ivashchenko@khpi.edu.ua; ORCID: https://orcid.org/0000-0003-3636-3914
С. ФІЛІП, інженер, доктор філософії (PhD), доцент, Братиславський університет економіки та менеджменту, м. Братислава, Словаччина; e-mail: stanislav.filip@vsemba.sk; ORCID: https://orcid.org/0000-0003-3000-9383
Б. В. РАТУШНИЙ, Національний технічний університет «Харківський політехнічний інститут», студент, м. Харків, Україна; e-mail: bohdan.ratushnyi@cs.khpi.edu.ua

РОЗРОБКА ОСВІТНЬОГО ЧАТ-БОТА З КОНТЕКСТНОЮ СИСТЕМОЮ ІНТЕНТІВ НА ПЛАТФОРМІ DIALOGFLOW

У контексті цифрової трансформації вищої освіти дедалі більшої актуальності набуває розробка інтелектуальних агентів, здатних забезпечувати безперервну та ефективну взаємодію зі студентами. У статті представлено повний життєвий цикл створення контекстно-залежного чат-бота «Питайко з ПІТУ» для кафедри програмної інженерії та інтелектуальних технологій управління НТУ «ХПІ». Бот призначено для швидкого та інтуїтивного доступу до інформації про навчальні процеси, канали зв'язку, стипендії, документи та інші типові запитання, що виникають під час взаємодії студентів з кафедрою та її вебсайтом. Система розроблена на платформі Dialogflow з інтеграцією до месенджера Telegram та використанням Google Cloud Functions як засобу обробки запитів (fulfillment). Ядро системи становить структурована багаторівнева система інтентів, кожна з яких відповідає певній тематичній категорії (наприклад, вступ, документи, розклад занять). Така архітектура дозволяє підтримувати контекст діалогу, забезпечувати точне маршрутизування запитів та зменшувати неоднозначність взаємодії. У якості моделі життєвого циклу було обрано прототипування, що дозволило враховувати зворотний зв'язок користувачів і вдосконалювати систему ітеративно. На основі аналізу структури сайту кафедри та результатів опитування студентів сформовано тематичну систему інтентів з fallback-механізмами та контекстною логікою уточнення. Окрему увагу приділено динамічному розподілу запитів за допомогою Webhook і повторному використанню функціональних блоків. У статті представлено алгоритм розробки, архітектуру системи інтентів, процес тестування та аналіз історії взаємодій. Тестування включало декілька циклів валідації, реальні сесії у Telegram і оцінювання ефективності fallback-обробки. Підсумкова реалізація забезпечила високу точність (≈91 %) та низький відсоток помилок (~3 %), що підтверджує доцільність використання Dialogflow для автоматизації освітніх процесів. Архітектура чат-бота передбачає масштабування та забезпечує цілодобову підтримку студентських звернень без додаткового навантаження на адміністративний персонал.

Ключові слова: розробка чат-бота, Dialogflow, вища освіта, студентські сервіси, модель прототипування, система інтентів, fallback-логіка, обробкі природної мови, автоматизація освіти, інтеграція з Telegram.

Повні імена авторів / Author's full names

Автор 1 / Author 1: Іващенко Оксана Віталіївна / Ivashchenko Oksana Vitaliivna
Автор 2 / Author 2: Філіп Станіслав / Filip Stanislav
Автор 3 / Author 3: Ратушний Богдан Володимирович / Ratushnyi Bohdan Volodymyrovych