

**Р. А. Наконечний, Т. Т. Іваницький**  
Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

## **СИСТЕМА МОНІТОРИНГУ НАВЧАЛЬНОГО ПРОЦЕСУ НА ОСНОВІ БРОКЕРА ПОВІДОМЛЕНЬ МІЖ КОРИСТУВАЧЕМ ТА СЕРВІСОМ**

© *Наконечний Р. А., Іваницький Т. Т., 2018*

**Розглянуто проблему моніторингу навчального процесу та причини пошуку інших шляхів для відображення розкладу занять у навчальних закладах. Наведено вибір платформи для вирішення проблеми, вибір типу інтерфейсу користувача для відображення навчального процесу, а також наведено переваги брокера повідомлень між користувачем та сервісом для вирішення проблеми моніторингу навчального процесу.**

**Ключові слова:** брокер повідомлень, месенджер, черга повідомлень, чат-бот.

**R. Nakonechnyi, T. Ivanytkyi**  
Lviv Polytechnic National University,  
Computer Engineering Department

## **THE MONITORING SYSTEM OF THE EDUCATIONAL PROCESS BASED ON BROKER MESSAGES BETWEEN THE USER AND THE SERVICE**

© *Nakonechnyi R., Ivanytkyi T., 2018*

**The problem of monitoring of the educational process and the reasons for finding other ways to display schedule in educational institutions. Are the choice of platform for the solution to the problem, the choice of the type of user interface to display the educational process, and also provides the benefits of a Broker messages between the user and service to solve the problem of monitoring the training process.**

**Key words:** Broker messages, Instant Messenge, Message Queuing, chat bot.

### **Вступ**

У кожному навчальному закладі працівники та учні чи студенти повинні дотримуватись навчального плану. В різних навчальних закладах розклад занять взнають з різних джерел: розклад занять на сайті навчального закладу; стенд в учительській/коридорі навчального закладу; файли з розкладом занять на сайтах навчальних закладів. Але, враховуючи, що комп'ютерні технології стають невід'ємною частиною нашого життя, то план навчального процесу намагаються зробити доступним для всіх і все частіше відображають його на сайтах навчальних закладів, а не вивіщують у коридорі закладу, щоб учням не потрібно було йти кудись, переписувати розклад на листок та перевіряти час-від-часу чи він не був зміненим.

### **Стан проблеми**

У сучасних умовах життя, коли більшість людей користується смартфонами всі шукають потрібну їм інформацію у мережі Інтернет. Це твердження справедливе також для учнів та студентів навчальних закладів, вони шукають інформацію про розклад занять у інтернеті. Для цього їм потрібно відкрити браузер на смартфоні, завантажити сайт навчального закладу, знайти там розклад, потрібний клас чи групу і тільки тоді вони зможуть побачити розклад занять на сьогодні чи на будь-

який інший день. Для сучасного світу, де все відбувається дуже швидко, а також коли учні чи студенти мають недостатньо часу це може тривати надто довго і вони запізняться на важливий урок або лекцію. Звісно є програми з розкладом занять навчальних закладів завдяки яким час на пошук свого розкладу зменшується у рази, але дані програми можуть займати багато простору у пам'яті пристрою і це може привести до того, що користувач не зможе встановити програму, якою він користувався би частіше.

### Постановка задачі

Знайти спосіб швидкого моніторингу навчального процесу без встановлення на пристрій зайвих додатків, щоб не займати на смартфоні простір пам'яті необхідний для роботи та встановлення інших додатків.

### Розв'язання задачі

**1. Вибір технології для моніторингу навчального процесу.** В даний момент в зв'язку з тим, що в Україні вже давно заблокували доступ до соціальної мережі «ВКонтакте» набули популярності різні месенджери, такі як: Viber (900 млн. користувачів), Telegram (100 млн. користувачів), Facebook Messenger (1 млрд. користувачів), WhatsApp (1 млрд. користувачів). Найбільший ріст користувачів спостерігається у месенджера Telegram. Це пов'язано з тим, що у ньому немає нічого зайвого, але, водночас, він є багатофункціональним. У нього гарний і простий інтерфейс, він споживає мінімальну кількість трафіку, а найголовніше – малу кількість системних ресурсів і бонусом є найкраща система захисту даних користувача та повідомлень від злону лиходіїв [1].

Для вирішення проблеми використано месенджер Telegram, оскільки він вимагає мало ресурсів смартфона, а також розробники цього месенджера одні з перших відкрили можливість створювати штучний інтелект за допомогою відкритих API. З їхньою допомогою можна задати список команд, налаштувати клавіатуру і зв'язати бота з сайтом. Можливості ботів в телеграмі: відправлення повідомлень; показ прогнозу погоди; переклад текстів; прийом платежів; створення ігор; опитування; стікери та багато іншого. Користувачі можуть спілкуватись з ботами за допомогою клавіатур, команд (наприклад, /help або /menu), звичайних повідомлень і картинок [2].

**2. Взаємодія «людина-комп'ютер».** Вибір типу інтерфейсу користувача для реалізації системи моніторингу навчального процесу. Для взаємодії зі штучним інтелектом потрібно вводити команди або ж тиснути на доступні кнопки. Такі типи інтерфейсу називаються інтерфейс командного рядка та графічним інтерфейсом користувача [3].

Інтерфейс командного рядка – різновид інтерфейсу людини та комп'ютера, в якому інструкції комп'ютеру задаються тільки шляхом вводу з клавіатури текстових рядків команд [4].

На рис. 1 показано послідовність роботи інтерфейсу командного рядка.



Рис. 1. Послідовність роботи інтерфейсу командного рядка

Графічний інтерфейс користувача – це система засобів для взаємодії користувача з комп'ютером, яка ґрунтується на представленні усіх доступних користувачу системних об'єктів і функцій у вигляді графічних компонент екрану, а саме: вікна, іконки, піктограми, меню, списків, кнопок тощо [5].

На рис. 2 показано послідовність роботи графічного інтерфейсу користувача.

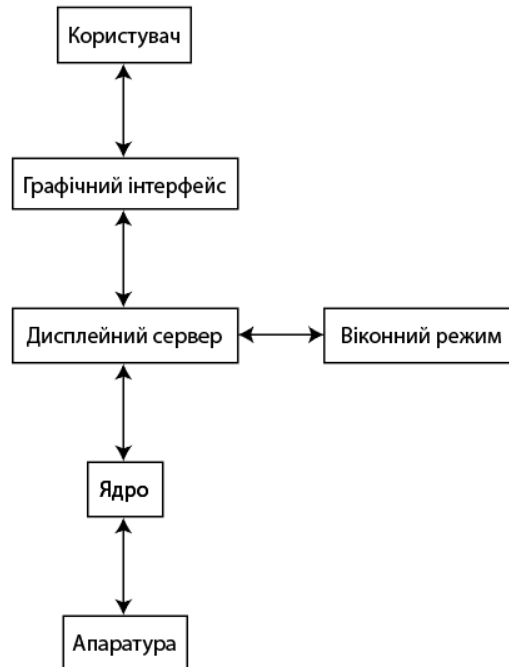


Рис. 2. Послідовність роботи графічного інтерфейсу користувача

**3. Вибір технології обробки запитів користувача.** Для обробки запитів вибрано чергу повідомлень, або як її ще називають – брокер повідомлень [6].

Повідомлення, в одному ряду з блоками обчислення і зберігання, складають три основні блоки майже в кожній блок-схемі системи. Черги повідомлень, по суті, є сполучною ланкою між різними процесами в додатках та забезпечують надійний і масштабований інтерфейс взаємодії з іншими підключеними системами і пристроями.

Черга – структура даних з дисципліною доступу до елементів «перший прийшов – першим вийшов». Додавання елемента можливе лише в кінець черги, вибірка тільки з початку черги, при цьому обраний елемент з черги видаляється [7].

Далі наведено причини використання черги повідомлень для вирішення проблеми моніторингу навчального процесу.

Слабке зв'язування – черги повідомлень створюють неявні інтерфейси обміну даними, які дозволяють процесам бути незалежними один від одного, тобто потрібно просто визначити формат повідомлень, що відправляються від одного процесу іншому.

Надмірність – черги повідомлень дозволяють уникнути випадків неекономного використання ресурсів процесу (наприклад пам'яті) в результаті зберігання необробленої (зайвої) інформації.

Масштабованість – черги повідомлень дозволяють розподілити процеси обробки інформації. Таким чином, вони дозволяють легко нарощувати швидкість, з якою повідомлення додаються в чергу і обробляються.

Еластичність і можливість витримувати пікові навантаження – черги повідомлень можуть виконувати роль свого роду буфера для накопичення даних в разі пікового навантаження, пом'якшуючи тим самим навантаження на систему обробки інформації і не допускаючи її відмови.

Відмовостійкість - черги повідомлень дозволяють відокремити процеси один від одного, так що якщо процес, який обробляє повідомлення з черги припиняє свою роботу, то повідомлення

можуть бути додані в чергу на обробку пізніше, коли система відновиться.

Гарантована доставка - використання черги повідомлень гарантує, що повідомлення буде доставлено і оброблено в будь-якому випадку (поки є хоча б один обробник).

Гарантований порядок доставки - велика частина систем черг повідомлень здатні забезпечити гарантії того, що дані будуть оброблятися в певному порядку (найчастіше в тому порядку в якому вони надійшли).

Буферизація - черги повідомлень дозволяє відправляти і отримувати повідомлення при цьому працюючи з максимальною ефективністю, пропонуючи буферний шар - процес запису в чергу може відбуватися настільки швидко, наскільки швидко це в змозі виконати черга повідомлень, а не обробник повідомлення.

Розуміння потоків даних - черги повідомлень дозволяють виявляти вузькі місця в потоках даних програми, легко можна визначити яка з черг забивається, яка простоє і визначити що необхідно робити - додавати нових обробників повідомлень або оптимізувати поточну архітектуру.

Асинхронний зв'язок - черги повідомлень надають можливість асинхронної обробки даних, яка дозволяє помістити повідомлення в чергу без обробки, дозволяючи системі обробити повідомлення пізніше, коли з'явиться можливість.

На рис. 3 зображено приклад роботи черги повідомлень.

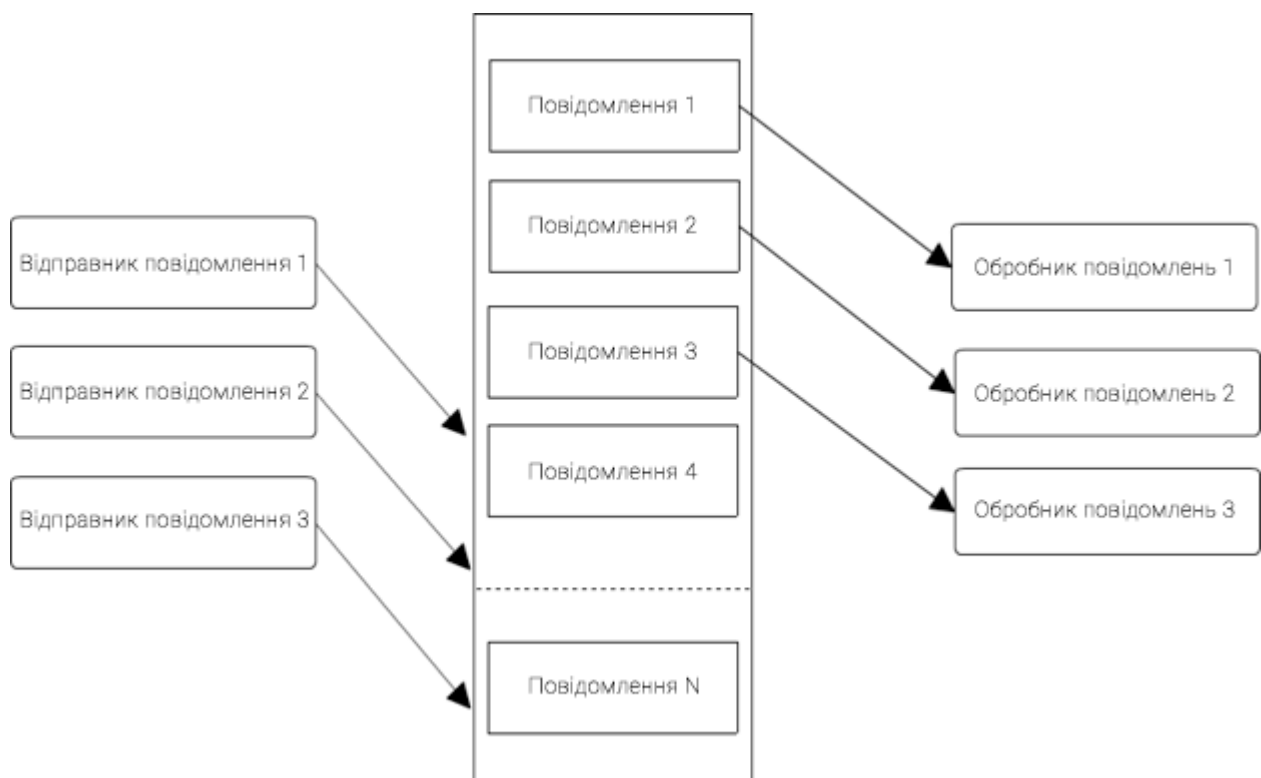


Рис. 3. Приклад черги повідомлень

Брокер складається з: сервісів – іменних клієнтів, які працюють в режимі відповідь на запит; клієнтів – анонімних клієнтів, які виконують запит до сервісу через брокер і очікують на відповідь.

Точкою підключення до брокера є сокет виду ROUTER. При прийомі даних від REQ сокету (від клієнтів), генерується повідомлення, яке містить в собі унікальний код клієнта. Далі створюється пакет для запиту в сервіс, який складається з назви сервісу, номера клієнту і решти даних [8]. Схематично це зображається так, як показано на рис. 4.

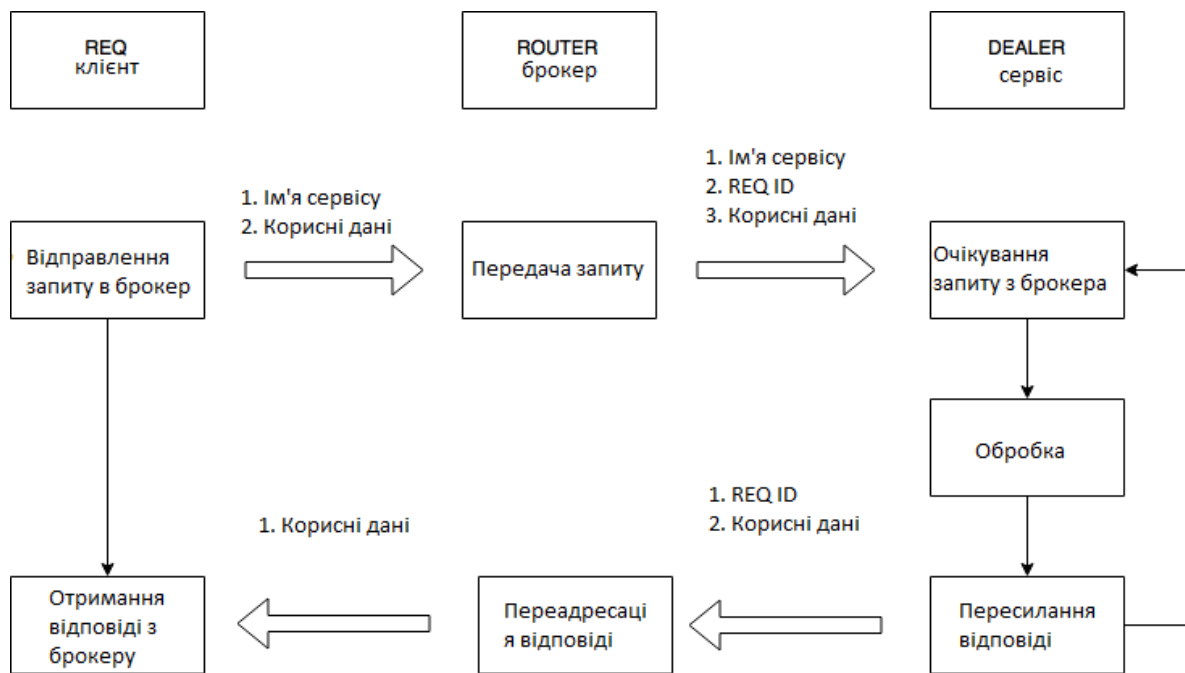


Рис. 4. Схематичне зображення роботи брокера повідомлень

### Висновки

У роботі розглянуто проблему моніторингу навчального процесу та причини пошуку інших шляхів для відображення розкладу занять у навчальних закладах. Знайдено спосіб моніторингу навчального процесу. Наведено вибір платформи для вирішення проблеми, вибрано тип інтерфейсу користувача для відображення навчального процесу, а також наведено переваги брокера повідомлень між користувачем та сервісом для вирішення проблеми моніторингу навчального процесу.

1. *Top 5 messenger in Ukraine [Elektronnyj resurs] : [Veb-sajt]. – Rezhym dostupu: <https://minfin.com.ua> (data zvernennja 17.09.2018). – Nazva z ekrana.* 2. *Mark Masse REST API Design Rulebook Designing Consistent RESTful Web Service Interfaces O'Reilly Media October 2011 p. 116.* 3. *Fisun A. P., Grashchenko L. A. i dr. Teoreticheskie i prakticheskie osnovy cheloveko-komp'yuternogo vzaimodejstviya: bazovye ponyatiya cheloveko-komp'yuternyh sistem v informatike i informacionnoj bezopasnosti / Orlovskij gosudarstvennyj universitet, 2004. — 169 p.* 4. *Piskoppel' A. A., SHCHedrovickij G. P. Inzhenernaya psihologiya i ehrgonomika. — Spravochnik-obzor. — M.: Put', 1996. — 207 p.* 5. *Berezko L. O. Konspekt lekcij po kursu «Doslidzhennya i proektuvannya interfejsiv korystuvacha»* 6. *Bach, M.J. The Design of the UNIX Operating System. ISBN 9780132017992.* 7. *Message Queuing (MSMQ) [Elektronnyj resurs] : [Veb-sajt]. – Rezhym dostupu: <https://msdn.microsoft.com> (data zvernennja 17.09.2018). – Nazva z ekrana.* 8. *Message Broker [Elektronnyj resurs] : [Veb-sajt]. – Rezhym dostupu: <https://habr.com> (data zvernennja 17.09.2018). – Nazva z ekrana.*