

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпровський національний університет імені Олеся Гончара
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпровський національний університет імені Олеся Гончара

Кваліфікаційна наукова
праця на правах рукопису

АНТОНЮК ВЛАДИСЛАВ АНДРІЙОВИЧ

УДК 004.42

ДИСЕРТАЦІЯ
РОЗРОБЛЕННЯ КРОСПЛАТФОРМНОЇ СИСТЕМИ
АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ТА ВЗАЄМОДІЇ З
КОМП'ЮТЕРИЗОВАНИМИ ПРИСТРОЯМИ

12 Інформаційні технології
121 Інженерія програмного забезпечення

Подається на здобуття ступеня доктора філософії. Дисертація містить результати власних досліджень. Використання ідей, результатів та текстів інших авторів мають посилання на відповідне джерело.

_____В. А. Антонюк

Науковий керівник:
Сидорова Марина Геннадіївна
кандидат технічних наук, доцент

Дніпро – 2024

АНОТАЦІЯ

Антонюк В.А. Розроблення кросплатформної системи автоматизації процесів та взаємодії з комп'ютеризованими пристроями. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення – Дніпровський національний університет імені Олеся Гончара, Дніпро, 2024.

Сучасні технологічні умови характеризуються стрімким зростанням складності та обсягу рутинних задач. Це, в свою чергу, призводить до збільшення потреби в ефективних інструментах для управління процесами та задачами. Традиційні методи, що ґрунтуються на ручному введенні даних та виконанні повторюваних дій, стають все менш продуктивними та потребують значних витрат часу та ресурсів.

Одним із ключових факторів, що зумовлюють цю потребу, є зростаюча взаємодія між людьми та комп'ютеризованими системами. В сучасному світі люди постійно взаємодіють з комп'ютерами, смартфонами, планшетами та іншими пристроями для виконання різноманітних завдань. Ця взаємодія має бути простою, інтуїтивно-зрозумілою та ефективною, щоб мінімізувати час та зусилля, необхідні для досягнення цілей.

На сьогодні, багато існуючих систем не відповідають цим вимогам. Вони часто є складними, заплутаними та потребують спеціальних знань або навичок для використання. Це призводить до помилок, неефективності та незадоволеності користувачів.

Збільшення складності та об'єму рутинних задач, зростаюча взаємодія між людьми та комп'ютеризованими системами в сучасних технологічних умовах потребує створення вдосконалених інтерфейсів, здатних підвищити інтуїтивність та ефективність управління процесами та задачами з мінімальним втручанням людини.

Ця дисертаційна робота присвячена розробленню передової кросплатформної системи автоматизації процесів, що забезпечує інтуїтивно зрозумілий асоціативний графічний інтерфейс між користувачами та комп'ютеризованими пристроями, ефективне налаштування взаємодії процесів, інтеграцію з будь-якими існуючими сторонніми службами, такими як Slack, Amazon S3 Files, SendGrid тощо. Розроблення таких передових кросплатформних систем автоматизації процесів є актуальною та перспективною задачею, яка має значний потенціал для покращення продуктивності та якості життя людей у сучасному цифровому світі.

Дослідження спрямоване на всебічний аналіз поточного стану технологій автоматизованої інтерактивної комунікації, виявлення основних проблемних точок і можливостей для подальшого прогресу. В концептуальній частині підкреслюється значення розвитку інтерактивних систем для поліпшення якості взаємодії між людиною та машиною, а також важливу роль штучного інтелекту в цьому процесі. Виділяється необхідність поглибленого дослідження адаптації сучасних технологій до конкретних вимог користувачів і сценаріїв використання.

У першому розділі проведено всебічний аналіз існуючих технологій та систем автоматизації процесів, взаємодії між людиною та машинами, включаючи огляд актуальних трендів, таких як інтерфейси програм автоматизації та програм з використанням графічних ключів. Обговорюється поточний стан ринку і виявлені прогалини у функціональності та зручності, що мають бути вирішені. Визначення вимог до кінцевого продукту базується на дослідженні потреб користувачів і технічних можливостей.

Другий розділ присвячено проектуванню кросплатформної системи, розглянуто основні алгоритмічні концепції, які були запропоновані у роботі, описаний загальний алгоритм організації обчислювального процесу у розробленій системі, який включає в себе взаємодію між частинами програми, створення та відтворення макросів, а також ефективну реалізацію

перехоплення подій; представлена концепція асоціативного графічного інтерфейсу, яка передбачає використання графічних жестів для керування макросами; розглянуто алгоритми розпізнавання графічних жестів; запропонована модифікація алгоритму для можливості розпізнавання декількох ліній одночасно для побудови складніших графічних жестів; описана технологія автоматизованого створення макросів, яка використовує мовну модель штучного інтелекту для генерації макросів та розрахунку їх параметрів.

У третьому розділі описано технічні аспекти розроблення програмного забезпечення та його взаємодію з комп'ютеризованим обладнанням, включаючи структуру та функціональні можливості, архітектуру, алгоритм створення розширень, реалізацію технологій та інших інтерфейсів для забезпечення найбільш ефективної реакції системи на введення користувача, наводяться результати запропонованих інженерних особливостей, які демонструють ефективність пропонованого рішення. Описано розроблений план впровадження системи, що включає стратегію комерціалізації, механізми технічної підтримки та оновлення, а також методи забезпечення безпеки та конфіденційності даних користувачів, з урахуванням потреб усіх зацікавлених сторін. Також надано аналітичні графіки, що демонструють поточний стан системи.

Четвертий розділ присвячено можливостям практичного застосування розробленої системи, розкрито перспективи розширення роботи системи, а також описано приклади застосування системи у найрізноманітніших задачах і предметних галузях. Також надано детальну інструкцію користувача, яка описує покрокові дії для роботи з системою для кожного типу користувачів, надано інструкцію для звичайних користувачів, розробників та адміністраторів.

Висновки підводять підсумки проведеної роботи, виокремлюють основні наукові та практичні здобутки автора у розробленні кросплатформної

системи автоматизованої взаємодії. Розроблення передової кросплатформної системи автоматизації процесів є актуальною та перспективною задачею. Робота має значний потенціал для подальшого дослідження та розширення у сферах, де важлива автоматизація обслуговування, керування пристроями, а також у ситуаціях, що вимагають індивідуальної взаємодії користувача з технологічними системами.

Наукова новизна одержаних результатів полягає в такому:

1. Уперше запропоновано концепцію асоціативного графічного інтерфейсу у системах автоматизації процесів на основі застосування графічних ключів для визначення команд, що дозволяє підвищити інтуїтивність та ефективність управління процесами та задачами, що є особливо актуальним у контексті збільшення складності та об'єму задач.

2. Уперше розроблено архітектуру програмного забезпечення для автоматизації процесів, яка базується на поєднанні мікроядра плагінів та чистої архітектури, що дозволяє оптимізувати структуру програмного коду, зменшуючи залежності та дублювання, що сприяє підвищенню читабельності, обслуговуваності та тестованості коду, а також забезпечує покращення метрик якості, включаючи зниження цикломатичної складності та підвищення індексу підтримуваності.

3. Удосконалено методологію інтеграції програмного забезпечення з іншими бізнес-системами та сервісами за допомогою розроблення модульних кросплатформних з'єднувачів, які значно розширюють можливості автоматизації та інтегрованості корпоративного програмного забезпечення.

4. Уперше запропоновано технологію застосування мовної моделі ІІІ для автоматичної побудови макросів у системах автоматизації, що дозволяє користувачам будувати робочі процеси, описуючи лише вимоги до них, без необхідності поглиблюватись у знання програмування або навіть самої системи.

5. Удосконалено метод розпізнавання графічних жестів, що дозволило розпізнавати жести, які складаються з кількох ліній, тим самим розширюючи кількість можливих асоціацій для більш складних команд.

Практичне значення роботи полягає у створенні ефективної та гнучкої кросплатформної системи автоматизації процесів та взаємодії з комп'ютеризованими пристроями, яка може бути використана у промисловій сфері для управління виробничими процесами, інформаційних технологіях для моніторингу інфраструктури, системах «Розумний будинок» або «Smart city» для управління міськими послугами тощо. Розроблена система автоматизації процесів має значну практичну значимість для підприємств та організацій різного розміру. Її можна використовувати для автоматизації широкого кола завдань, таких як обробка замовлень, управління проектами, обслуговування клієнтів, фінансовий облік та багато іншого.

Ключові слова: багаторівнева архітектура, інформаційна технологія, ранжування ролей, комп'ютерні інформаційні системи, розпізнавання образів, проектування, система обробки інформації, кросплатформна система, автоматизована людино-машинна взаємодія, асоціативний графічний інтерфейс, штучний інтелект, автоматизація процесів, макрос, модульні кросплатформні з'єднувачі, смарт-пристрої.

ANNOTATION

Antonyuk V.A. Development of a cross-platform system for automating processes and interacting with computerized devices. – Qualifying scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 121 Software Engineering – Oles Honchar Dnipro National University, Dnipro, 2024.

Modern technological conditions are characterized by a rapid increase in the complexity and volume of routine tasks. This, in turn, leads to an increased need for effective tools for managing processes and tasks. Traditional methods based on manual data entry and repetitive actions are becoming less and less productive and require a significant investment of time and resources.

One of the key factors driving this need is the growing interaction between humans and computerized systems. In today's world, people are constantly interacting with computers, smartphones, tablets, and other devices to perform a variety of tasks. This interaction should be simple, intuitive, and effective to minimize the time and effort required to achieve goals.

Today, many existing systems do not meet these requirements. They are often complex, intricate, and require specialized knowledge or skills to use. This leads to errors, inefficiencies, and user dissatisfaction.

The increase in the complexity and volume of routine tasks, the growing interaction between people and computerized systems in modern technological conditions requires the creation of improved interfaces that can increase the intuitiveness and efficiency of process and task management with minimal human intervention.

This dissertation focuses on the development of an advanced cross-platform process automation system that provides an intuitive associative GUI between users and computerized devices, efficient customization of process interactions, integration with any existing third-party services such as Slack, Amazon S3 Files,

SendGrid, etc. The development of such advanced cross-platform process automation systems is an urgent and promising task that has significant potential to improve the productivity and quality of life of people in today's digital world.

The study is aimed at a comprehensive analysis of the current state of automated interactive communication technologies, identifying the main problem points and opportunities for further progress. The conceptual part emphasizes the importance of the development of interactive systems to improve the quality of human-machine interaction, as well as the important role of artificial intelligence in this process. The need for an in-depth study of the adaptation of modern technologies to specific user requirements and use cases stands out.

The first section provides a comprehensive analysis of existing technologies and systems for automating processes, human-machine interactions, including an overview of current trends, such as interfaces of automation programs and programs using graphic keys. The current state of the market is discussed and gaps in functionality and convenience that need to be addressed are identified. The definition of requirements for the final product is based on the study of user needs and technical capabilities.

The second section is devoted to the design of a cross-platform system, the main algorithmic concepts that were proposed in the work are considered, the general algorithm for organizing the computing process in the developed system is described, which includes the interaction between parts of the program, the creation and reproduction of macros, as well as the effective implementation of event interception; the concept of an associative graphical interface is presented, which involves the use of graphic gestures for control Macros; algorithms for recognizing graphic gestures are considered; a modification of the algorithm for the ability to recognize several lines at the same time for the construction of more complex graphic gestures is proposed; The technology of automated macro creation is described, which uses a language model of artificial intelligence to generate macros and calculate their parameters.

The third section describes the technical aspects of software development and its interaction with computerized equipment, including the structure and functionality, architecture, algorithm for creating extensions, implementation of technologies and other interfaces to ensure the most effective response of the system to user input, the results of the proposed engineering features that demonstrate the effectiveness of the proposed solution. A developed plan for the implementation of the system is described, including a commercialization strategy, technical support and update mechanisms, as well as methods for ensuring the security and confidentiality of user data, taking into account the needs of all stakeholders. Analytical graphs demonstrating the current state of the system are also provided.

The fourth section is devoted to the possibilities of practical application of the developed system, the prospects for expanding the work of the system are disclosed, and examples of the application of the system in a variety of tasks and subject areas are described. A detailed user manual is also provided, which describes step-by-step actions for working with the system for each type of user, instructions for ordinary users, developers and administrators are provided.

The conclusions summarize the results of the work carried out, highlight the main scientific and practical achievements of the author in the development of a cross-platform system of automated interaction. The development of an advanced cross-platform process automation system is an urgent and promising task. The work has significant potential for further research and expansion in areas where service automation, device management is important, as well as in situations that require individual interaction of the user with technological systems.

The scientific novelty of the obtained results is as follows:

1. For the first time, the concept of an associative graphical interface in process automation systems based on the use of graphic keys for determining commands is proposed, which allows to increase the intuitiveness and efficiency of

process and task management, which is especially relevant in the context of increasing the complexity and volume of tasks.

2. For the first time, a software architecture for process automation has been developed, which is based on a combination of a plug-in microkernel and a pure architecture, which allows you to optimize the structure of the program code, reducing dependencies and duplication, which helps to increase the readability, maintainability and testability of the code, as well as provides an improvement in quality metrics, including a decrease in cyclomatic complexity and an increase in the support index.

3. The methodology of software integration with other business systems and services has been improved through the development of modular cross-platform connectors, which significantly expand the possibilities of automation and integration of enterprise software.

4. For the first time, the technology of using the AI language model for the automatic construction of macros in automation systems is proposed, which allows users to build workflows, describing only the requirements for them, without the need to delve into knowledge of programming or even the system itself.

5. The method of recognition of graphic gestures has been improved, which made it possible to recognize gestures that consist of several lines, thereby expanding the number of possible associations for more complex commands.

The practical significance of the work lies in the creation of an effective and flexible cross-platform system for automating processes and interacting with computerized devices, which can be used in the industrial sphere to control production processes, information technology for infrastructure monitoring, "Smart Home" or "Smart City" systems for managing urban services, etc. The developed process automation system has significant practical significance for enterprises and organizations of different sizes. It can be used to automate a wide range of tasks,

such as order processing, project management, customer service, financial accounting, and more.

Keywords: multi-level architecture, information technology, role ranking, computer information systems, pattern recognition, design, information processing system, cross-platform system, automated human-machine interaction, associative graphical interface, artificial intelligence, process automation, macro, modular cross-platform connectors, smart devices.

Список опублікованих праць за темою дисертації

Статті у наукових фахових виданнях України:

1. Antonyuk V., Sydorova M. Synthesis of software architectures for cross-platform application development. *Actual problems of automation and information technology*. 2021. Vol.25. P. 3-12. DOI: [10.15421/432101](https://doi.org/10.15421/432101) URL: <https://actualproblems.dp.ua/index.php/APAIT/article/download/192/136> **(фахове видання категорії Б).**
2. Antonyuk V., Sydorova M. A Cross-Platform Mobile Development for accelerating software development lifecycle. *Actual problems of automation and information technology*. 2022. Vol.26. P. 3-8. DOI: [10.15421/432201](https://doi.org/10.15421/432201) URL: <https://actualproblems.dp.ua/index.php/APAIT/article/view/209> **(фахове видання категорії Б).**
3. Antonyuk V., Sydorova M. The concept of associative graphical interface in the workflow automation system. *System technologies*. 2023. Vol. 5 No. 148. P. 133-140. DOI: [10.34185/1562-9945-5-148-2023-12](https://doi.org/10.34185/1562-9945-5-148-2023-12) URL: <https://journals.nmetau.edu.ua/index.php/st/article/view/1552> **(фахове видання категорії Б).**
4. Antonyuk V., Sydorova M. Integration and use of artificial intelligence for automated macros creation. *System technologies*. 2024. Vol. 5 No. 154. P. 16-23. DOI: [10.34185/1562-9945-5-154-2024-02](https://doi.org/10.34185/1562-9945-5-154-2024-02) URL: <https://journals.nmetau.edu.ua/index.php/st/article/view/1863> **(фахове видання категорії Б).**

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Антонюк В.А., Сидорова М.Г. Microkernel architecture у розробці сучасних мобільних додатків. *Сучасні інформаційні системи та технології: тези доповідей III Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених*. Херсон, 30 листопада 2020 р., С. 96-

97, URL: <http://www.tsatu.edu.ua/vmf/wp-content/uploads/sites/17/cict2020-1.pdf>.

2. Антонюк В.А., Сидорова М.Г. Жестова взаємодія як засіб прискорення роботи користувача з комп'ютером. *Проблеми прикладної математики та інформаційних технологій*: тези доповідей наукової конференції за підсумками науково-дослідної роботи ДНУ за 2020 рік. Дніпро, 2021, С. 7, URL: <http://repository.dnu.dp.ua:1100/upload/1c39b5f616fadcd4da0a7142df2742cTezi-2021.pdf>.

3. Антонюк В.А., Сидорова М.Г. Використання ідеї асоціативного мислення людини у програмному забезпеченні. *Інформаційні технології – 2021*: тези доповідей VIII Всеукраїнської науково-практичної онлайн конференції молодих науковців, Київ, 20 травня 2021 р., С. 112-114, URL: <https://zcit.kubg.edu.ua/index.php/journal/issue/download/9/20>.

4. Антонюк В.А., Сидорова М.Г. Порівняльна характеристика систем автоматизації програмного забезпечення у розробці власного додатку. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2021)*: тези доповідей XIX міжнародної науково-практичної конференції, Дніпро, 17-19 листопада 2021, С. 5-6, URL: http://mpzis.dnu.dp.ua/wp-content/uploads/2021/11/mpzis_2021.pdf.

5. Антонюк В.А., Сидорова М.Г. Нова архітектура у .NET MAUI як еволюція XAMARIN.FORMS. *Сучасні інформаційні системи та технології*: матеріали IV Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених, Херсон, 30 листопада 2021 р., С. 11-12, URL: <https://kntu.net.ua/index.php/ukr/content/download/93825/537598/file/CICT2021.pdf>.

6. Антонюк В.А., Сидорова М.Г. Автоматизація аналізу та форматування коду. *Інформаційні технології – 2022*: тези доповідей IX

Всеукраїнської науково-практичної конференції молодих науковців, Київ, 19 травня 2022 р., С. 7-9, URL: <https://zcit.kubg.edu.ua/index.php/journal/issue/download/10/19> .

7. Антонюк В.А., Сидорова М.Г. Використання візуальних елементів керування для покращення взаємодії користувача. *Проблеми математичного моделювання: тези доповідей всеукраїнської науково-методичної конференції*, Кам'янське, 25-27 травня 2022 р., С.69-70, URL: https://www.dstu.dp.ua/uni/downloads/zbirka_konf_pm.pdf.

8. Антонюк В.А., Сидорова М.Г. Можливості використання програмного продукту «Draw & GO» у навчальному процесі. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022): тези доповідей XX міжнародної науково-практичної конференції*, Дніпро, 23-25 листопада 2022 р., С. 15-16, URL: <http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf>.

9. Антонюк В.А., Сидорова М.Г. Застосування стенографії для захисту програм від несанкціонованого доступу. *Інформаційна безпека та інформаційні технології (ІБІТ-2022): тези доповідей IV Міжнародної науково-практичної конференції*, Львів, 30 листопада 2022 р., С. 140-142, URL: <https://sci.ldubgd.edu.ua/jspui/bitstream/123456789/11434/1/IV%20International%20Scientific%20and%20Practical%20Conference.pdf> .

10. Антонюк В.А., Сидорова М.Г. Використання штучного інтелекту у розробці клієнт-орієнтованих додатків. *Інформаційні технології – 2023: тези доповідей X Всеукраїнської науково-практичної конференції молодих науковців*, Київ, 18 травня 2023 р., С. 5-6, URL: <https://www.eportfolio.kubg.edu.ua/data/conference/9325/document.pdf>.

ЗМІСТ

ВСТУП	17
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ СИСТЕМ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ТА ПІДХОДІВ ДО ЇХ РОЗРОБЛЕННЯ.....	23
1.1 Огляд програмних продуктів автоматизації процесів	24
1.2 Поняття графічного жесту (ключа) та його практичне застосування. 31	
1.3 Дослідження сучасних інструментів розроблення кросплатформних програмних додатків	37
1.4 Дослідження архітектурних рішень	41
1.5 Висновки до розділу 1	45
РОЗДІЛ 2. АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ «DRAW & GO»	47
2.1 Організація робочого процесу.....	47
2.2 Концепція асоціативного графічного інтерфейсу та розпізнавання графічних жестів	51
2.3 Технологія автоматизованого створення макросів на основі застосування мовної моделі штучного інтелекту	59
2.4 Висновки до розділу 2.....	65
РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ «DRAW & GO».....	67
3.1 Структура та функціональні можливості.....	67
3.2 Операції з макросами	83
3.3 Розроблення бібліотеки розширень для «Draw & GO»	89
3.4 Впровадження системи «Draw & GO»	91
3.5 Висновки до розділу 3.....	95
РОЗДІЛ 4. ПРАКТИЧНЕ ЗАСТОСУВАННЯ СИСТЕМИ «DRAW & GO».....	97
4.1 Посібник користувача	97
4.2 Інтеграція з комп'ютеризованими пристроями.....	102
4.3 Імітація трафіку на сайт	103

4.4	Використання у навчальному процесі.....	105
4.5	Автоматизація повсякденних дій.....	106
4.6	Інші приклади застосування системи.....	107
4.7	Висновки до розділу 4.....	109
	ВИСНОВКИ.....	110
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	113

ВСТУП

Обґрунтування вибору теми дослідження. У контексті стрімкого розвитку цифрових технологій, проблематика ефективної взаємодії людини з комп'ютеризованими пристроями набуває особливої актуальності. Сучасні інформаційні системи вимагають від користувачів здатності швидко адаптуватися до нових інтерфейсів та управлінських парадигм, що часто призводить до підвищення когнітивного навантаження та зниження продуктивності. З огляду на це, виникає необхідність створення більш інтуїтивних, природних способів взаємодії, зокрема через використання графічних жестів.

Паралельно, світ стикається з проблемою фрагментації цифрових пристроїв та платформ, що ускладнює процес їх інтеграції та уніфікованого управління. Відсутність універсальної кросплатформної системи, яка б забезпечувала безшовну взаємодію між різними пристроями та сервісами, є значною перешкодою для досягнення оптимальної ефективності робочих процесів.

Іншою критичною проблемою є високий рівень складності та витрат часу, пов'язаних з автоматизацією рутинних задач. Розроблення систем, здатних самостійно генерувати та оптимізувати макроси на основі штучного інтелекту та машинного навчання, може стати рішенням, що здатне радикально змінити ситуацію.

Крім того, багато існуючих рішень зосереджені на задоволенні потреб окремих ніш або специфічних секторів ринку, що призводить до обмеженої адаптивності та масштабованості. Така ситуація вимагає розроблення гнучких алгоритмів та архітектур, здатних до швидкої інтеграції з різноманітними системами та сервісами.

Враховуючи викладені проблеми, актуальною науково-технічною задачею є розроблення кросплатформної системи автоматизації процесів та

взаємодії з комп'ютеризованими пристроями, вирішенню якої і присвячена ця дисертаційна робота.

Мета та завдання дослідження. Метою дисертаційної роботи є створення ефективної та гнучкої кросплатформної системи автоматизації процесів та взаємодії з комп'ютеризованими пристроями, яка забезпечує управління на основі графічних жестів, застосування штучного інтелекту для автоматичної генерації макросів, інтеграцію з будь-якими існуючими сторонніми службами та має такі характеристики:

1. Ефективність: система має бути здатною до автоматизації складних процесів з максимальною точністю та мінімальними часовими витратами.
2. Гнучкість: система має бути легко налаштованою для адаптації до різних потреб та задач користувачів.
3. Управління на основі графічних жестів: система має дозволяти користувачам керувати процесами за допомогою простих та інтуїтивно зрозумілих графічних жестів.
4. Застосування штучного інтелекту: система має використовувати штучний інтелект для автоматичної генерації макросів, що дозволить користувачам економити час та зусилля.
5. Інтеграція з сторонніми службами: система має бути легко інтегрованою з будь-якими існуючими сторонніми службами, що розширить її функціональні можливості та зробить її більш універсальною.

Відповідно до поставленої мети основними задачами дослідження є:

1. Аналіз сучасних програмних засобів автоматизації процесів, технологій взаємодії з комп'ютеризованими пристроями;
2. Аналіз і удосконалення технологій застосування графічного ключа та алгоритмів розпізнавання графічних жестів;

3. Розроблення кросплатформного програмного забезпечення з удосконаленням архітектури та методології інтеграції з іншими системами та сервісами, яке буде відповідати вищезазначеним характеристикам;

4. Розроблення алгоритму створення розширень та бібліотеки макрокоманд для автоматизації типових завдань, які можуть бути використані користувачами без глибоких технічних знань в області програмування;

5. Розроблення технології автоматичної побудови макросів у системах автоматизації процесів із застосуванням мовних моделей штучного інтелекту;

6. Практична реалізація та тестування системи в реальних умовах, щоб забезпечити її надійність, масштабованість та готовність до впровадження у виробництво;

7. Розроблення плану впровадження системи, що включає стратегію комерціалізації, механізми технічної підтримки та оновлення, а також методи забезпечення безпеки та конфіденційності даних користувачів, з урахуванням потреб усіх зацікавлених сторін.

Реалізація цих завдань дозволить створити кросплатформну систему, яка не тільки відповідатиме академічним вимогам, але й буде готова до комерційного використання, вносячи значний вклад у розвиток галузі автоматизованих технологій взаємодії з комп'ютерним обладнанням.

Об'єктом дослідження є процеси розроблення програмних систем автоматизації процесів та взаємодії з комп'ютеризованими пристроями.

Предметом дослідження є методи розроблення алгоритмічного та програмного забезпечення систем автоматизації процесів та взаємодії з комп'ютеризованими пристроями.

Методи дослідження: методи та технології інженерії програмного забезпечення, людино-машинної взаємодії, теорії алгоритмів, баз даних, розпізнавання графічних жестів.

Наукова новизна одержаних результатів полягає у наступному:

1. Уперше запропоновано концепцію асоціативного графічного інтерфейсу у системах автоматизації процесів на основі застосування графічних ключів для визначення команд, що дозволяє підвищити інтуїтивність та ефективність управління процесами, що є особливо актуальним у контексті збільшення складності та об'єму задач.

2. Уперше розроблено архітектуру програмного забезпечення для автоматизації процесів, яка базується на поєднанні мікроядра плагінів та чистої архітектури, що дозволяє оптимізувати структуру програмного коду, зменшуючи залежності та дублювання, що сприяє підвищенню читабельності, обслуговуваності та тестованості коду, а також забезпечує покращення метрик якості, включаючи зниження цикломатичної складності та підвищення індексу підтримуваності.

3. Удосконалено методологію інтеграції програмного забезпечення з іншими системами та сервісами за допомогою розроблення модульних кросплатформних з'єднувачів, які значно розширюють можливості автоматизації та інтегрованості корпоративного програмного забезпечення.

4. Уперше запропоновано технологію застосування мовної моделі ІІІ для автоматичної побудови макросів у системах автоматизації процесів, що дозволяє користувачам будувати робочі процеси, описуючи лише вимоги до них, без необхідності поглиблюватись у знання програмування або навіть самої системи.

5. Удосконалено метод розпізнавання графічних жестів, що дозволило розпізнавати жести, які складаються з кількох ліній, тим самим розширюючи кількість можливих асоціацій для більш складних команд.

Практичне значення одержаних результатів полягає у наступних аспектах:

1. Застосування програми «Draw & GO» усуває необхідність повторного виконання монотонних операцій, забезпечуючи можливість

швидкого графічного створення макросів, що значно спрощує роботу користувачів.

2. «Draw & GO» надає гнучкий механізм робочого процесу з архітектурою плагінів, що дозволяє користувачам швидко адаптувати робочі процеси під власні потреби, від простих до складних.

3. Широка інтеграція зі сторонніми сервісами та гнучкість у налаштуванні дозволяють використовувати «Draw & GO» в контексті широкого діапазону ділових та особистих завдань.

4. Програма забезпечує високий рівень організації при роботі процесів завдяки вбудованій автоматизованій синхронізації даних в реальному часі.

5. «Draw & GO» гарантує широку сумісність, працюючи на всіх основних операційних системах та пристроях, забезпечуючи користувачам можливість застосування програми в різних середовищах і ситуаціях.

6. Загальнодоступність і легкість у використанні програми, поєднанні з сучасними технологіями автоматизації та API інтерфейсами, роблять «Draw & GO» важливим інструментом, здатним підвищити продуктивність та ефективність в широкому спектрі професійних діяльностей.

7. Система може зробити комп'ютеризовані пристрої більш доступними для людей з обмеженими можливостями.

Особистий внесок здобувача. Аналіз літературних даних, розроблення алгоритмів та програмного забезпечення, обробка отриманих результатів здійснені безпосередньо автором. Постановка мети і завдань дослідження, а також аналіз і узагальнення отриманих результатів проводились спільно з науковим керівником к.т.н., доц. М. Г. Сидоровою.

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідалися й обговорювалися на XIX та XX Міжнародних науково-практичних конференціях «Математичне та

програмне забезпечення інтелектуальних систем (MPZIS)» (м. Дніпро, 2021, 2022), IV Міжнародній науково-практичній конференції «Інформаційна безпека та інформаційні технології (ІБІТ)» (м. Львів, 2022), III-IV Всеукраїнських науково-практичних інтернет-конференціях студентів, аспірантів та молодих вчених «Сучасні інформаційні системи та технології» (м. Херсон, 2020, 2021), підсумкових наукових конференціях Дніпровського національного університету імені Олеся Гончара (м. Дніпро, 2020, 2021), IX-X Всеукраїнських науково-практичних конференціях молодих науковців «Інформаційні технології» (м. Київ 2021, 2022, 2023), Всеукраїнській науково-методичній конференції «Проблеми математичного моделювання» (м. Кам'янське, 2022).

Публікації. Основні положення й результати дисертаційної роботи опубліковано у 14 роботах: 4 статті у наукових фахових виданнях України категорії Б та 10 тез доповідей у збірниках матеріалів наукових конференцій.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась у відповідності з індивідуальним планом підготовки аспіранта кафедри математичного забезпечення ЕОМ Дніпровського національного університету імені Олеся Гончара. Дослідження за темою дисертації здійснювалися також в рамках науково-дослідної роботи № ФПМ-2-22 "Розроблення програмного забезпечення аналізу та кластеризації часових рядів" 2022-24 рр. № держреєстрації 0122U001465.

Структура та обсяг дисертації. Робота містить вступ, 4 розділи, висновки та список використаних джерел, що містить 103 найменування на 12 сторінках. Загальний обсяг дисертації – 124 сторінки, обсяг основного тексту – 98 сторінок. Робота містить 43 рисунки та 1 таблицю.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ СИСТЕМ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ТА ПІДХОДІВ ДО ЇХ РОЗРОБЛЕННЯ

Системи автоматизації процесів (Workflow Automation Systems) – це технології, засновані на використанні комп'ютерного програмного забезпечення для автоматизації повторюваних бізнес-процедур і завдань у рамках організації з метою покращення ефективності операцій. Вони забезпечують автоматичне управління потоками документації, задачами та даними, раціоналізують робочі процеси та знижують ймовірність людських помилок [31].

У сучасному динамічному світі посилюється потреба в оптимізації та автоматизації рутинної діяльності. Системи автоматизації процесів відіграють ключову роль у підвищенні продуктивності, зниженні витрат та покращенні точності виконання завдань.

Макроси в контексті автоматизації – це заздалегідь визначені послідовності команд або дій, які можуть бути викликані одним вводом, наприклад, натисканням на кнопку інтерфейсу користувача. Використання макросів дозволяє стандартизувати і прискорити виконання рутинних завдань, які раніше потребували множинних ручних операцій.

У системах автоматизації процесів макроси не є обов'язковим елементом, але їх включення зазвичай вважається вигідним для підвищення продуктивності. Вони надають користувачам можливість швидко змінювати робочі процедури без потреби у знаннях програмування.

У підрозділі 1.1 проаналізовано існуючі програмні продукти для автоматизації процесів. У підрозділі 1.2 розкрито поняття графічних ключів, охарактеризовано основні підходи до їх визначення та практичного застосування. Підрозділи 1.3 і 1.4 присвячені дослідженню інструментарію для розроблення кросплатформних програмних додатків і архітектурних

рішень, які лежать в основі ефективної роботи з цими програмами. Висновки за розділом подані у підрозділі 1.5.

1.1 Огляд програмних продуктів автоматизації процесів

У роботі проведено огляд основних характеристик та можливостей сучасного програмного забезпечення для автоматизації процесів, включаючи функціональність, масштабованість та інтеграцію макросів.

Механізм робочого процесу – це програма, яка запускає програмне забезпечення цифрового робочого процесу. Це дозволяє компаніям створювати та автоматизувати робочі процеси – у багатьох випадках за допомогою візуальних конструкторів із низьким кодом.

Механізм робочого процесу перетворює традиційні ручні робочі процеси із завдань, керованих ІТ, на процеси, керовані людиною та програмним забезпеченням, які направляють шляхи інформації, обов'язки та канали співпраці для найбільш ефективного використання ресурсів [87, 88, 92].

Ринок програм механізму робочого процесу є висококонкурентним. Найбільшими та найрозвиненішими гравцями на цьому ринку є Microsoft Power Automate [50] і Workflow Core [89]. Ці два продукти є фактичними галузевими стандартами автоматизації робочого процесу. Microsoft Power Automate – це продукт, який дозволяє користувачам автоматизувати робочі процеси в кількох програмах і службах [103].

Як видно на рисунку 1.1, який ілюструє інтерфейс системи Microsoft Power Automate, користувачу надається інтуїтивно зрозумілий робочий простір для створення автоматизованих процесів за допомогою візуального редактора потоків. Ця платформа дозволяє розробляти рішення за схемою причина-наслідок, де можливо з легкістю зв'язати тригери, такі як отримання

нового електронного листа, з відповідними діями, наприклад, записом інформації у таблицю. Функціонал даної системи не обмежується простими завданнями і підтримує складні багаторівневі бізнес-сценарії, забезпечуючи інтеграцію з широким спектром зовнішніх додатків та сервісів, що робить Microsoft Power Automate потужним інструментом для оптимізації процесів на підприємстві.

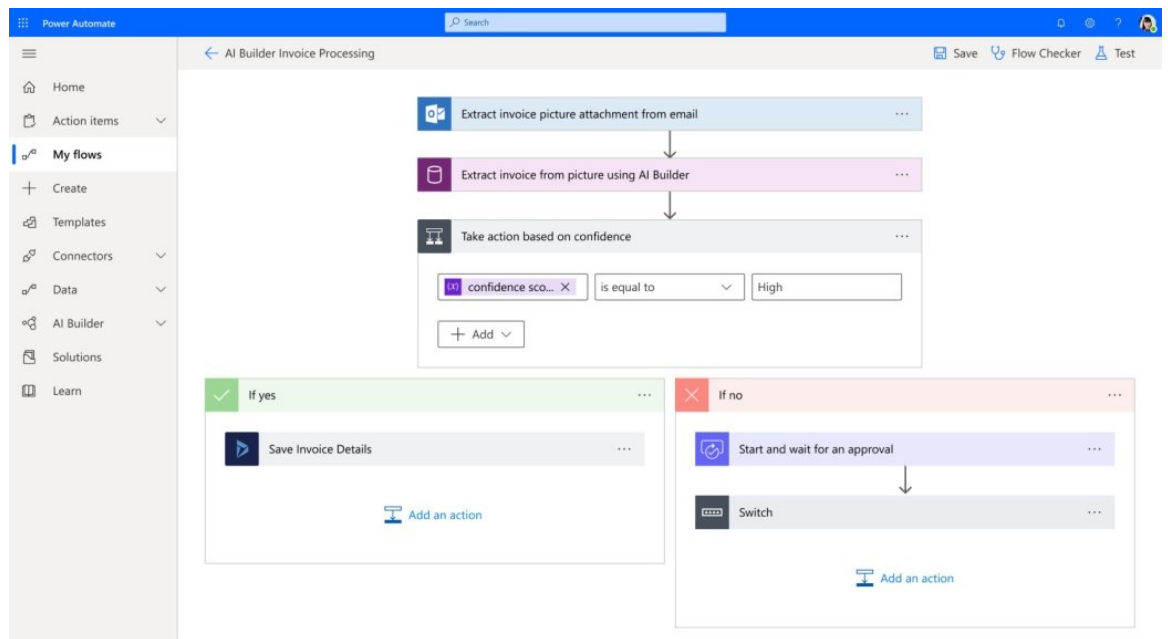
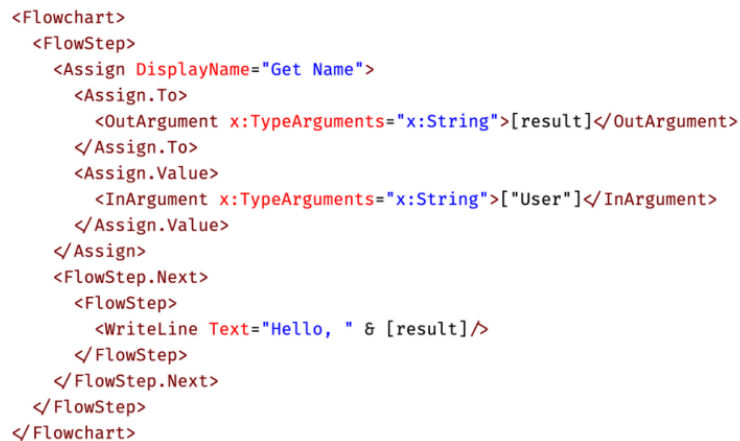


Рисунок 1.1 – Інтерфейс Microsoft Power Automate

Хоча послуга пропонує широкий набір функцій, вона може бути занадто коштовною та складною у використанні для багатьох користувачів. У результаті зростає кількість конкурентів, які пропонують подібні послуги за нижчою ціною. Workflow Core – це механізм робочого процесу з відкритим кодом, написаний на .NET Core, забезпечує простий спосіб реалізації складних процесів у будь-якій програмі .NET. На рисунку 1.2 представлено приклад опису flow для Workflow Core.



```

<Flowchart>
  <FlowStep>
    <Assign DisplayName="Get Name">
      <Assign.To>
        <OutArgument x:TypeArguments="x:String">[result]</OutArgument>
      </Assign.To>
      <Assign.Value>
        <InArgument x:TypeArguments="x:String">["User"]</InArgument>
      </Assign.Value>
    </Assign>
    <FlowStep.Next>
      <FlowStep>
        <WriteLine Text="Hello, " & [result]/>
      </FlowStep>
    </FlowStep.Next>
  </FlowStep>
</Flowchart>

```

Рисунок 1.2 – Приклад опису flow у Workflow Core

Хоча Workflow Core набирає популярності в галузі, існують інші варіанти, які пропонують різні переваги: AutoIt [14], Make [54], Zapier [20], Macro Maker [52], JetBit Macro Recorder [45] тощо. Ці програми дозволяють встановлювати макроси для полегшення виклику програм.

Make – це хмарна платформа автоматизації, схожа на Microsoft Power Automate, але з деякими ключовими відмінностями. Make допомагає користувачам створювати, керувати та оптимізувати робочі процеси в різних програмах і середовищах. Порівняно з Microsoft Power Automate, Make забезпечує більшу гнучкість щодо даних і процесів із широкою різноманітністю варіантів перетворення даних. Це також дозволяє користувачам виконувати дії в середині робочого процесу та планувати завдання. За допомогою Make користувачі можуть автоматизувати такі завдання, як обробка даних, вебзбирання, маркетинг електронною поштою, публікація в соціальних мережах, керування файлами тощо. Make також підтримує інтеграцію з такими популярними сервісами, як Gmail, Google Drive, Dropbox, Slack, Twitter тощо. Як показано на рисунку 1.3, Make також надає візуальний інтерфейс, який дозволяє користувачам проєктувати та виконувати робочі процеси автоматизації без програмування.

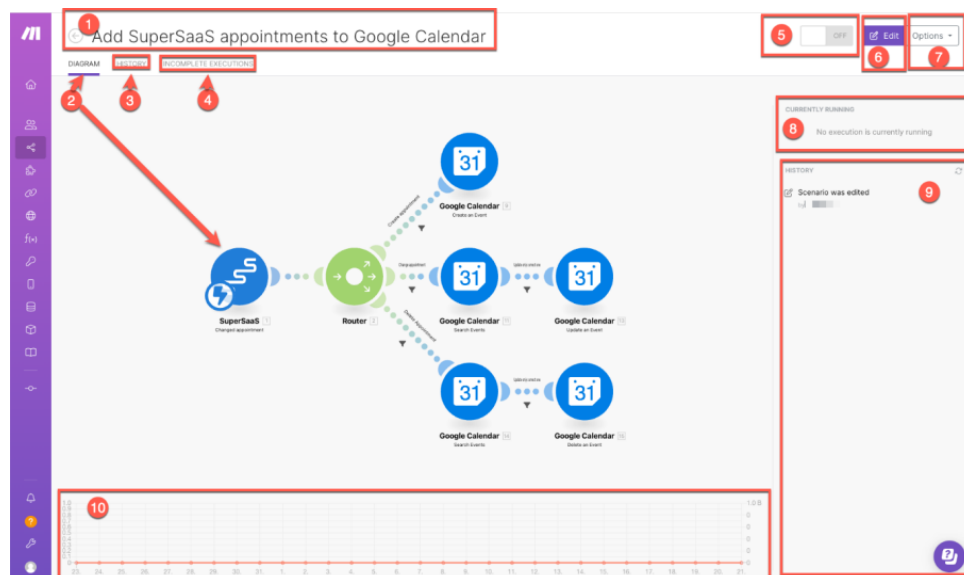


Рисунок 1.3 – Приклад сценарію автоматизації у Make

Сильними сторонами Make є простий у використанні інтерфейс, інтуїтивно зрозумілий конструктор із функцією перетягування, гнучкі можливості перетворення даних, планування завдань та виконання дії в середині робочого процесу, доступні тарифні плани. До недоліків можна віднести обмежену підтримку API та відносну обмеженість документації.

Zapier – це потужна платформа автоматизації, яка дозволяє користувачам підключати вебдодатки, що робить їх можливими для автоматизації. Використовує простий тригер і дію для створення команд, таких як «якщо це трапиться, то зробіть це». Наприклад, користувач може використовувати Zapier, щоб додавати нових потенційних клієнтів до списку розсилки або зберігати вкладення електронної пошти на Google Drive.

Zapier може допомогти у автоматизації маркетингу, керуванні даними, обслуговуванні клієнтів, управлінні проєктами тощо. Як показано на рисунку 1.4, користувач може вибрати з понад 5000 програм, які підтримує Zapier, або створити власні інтеграції [40].

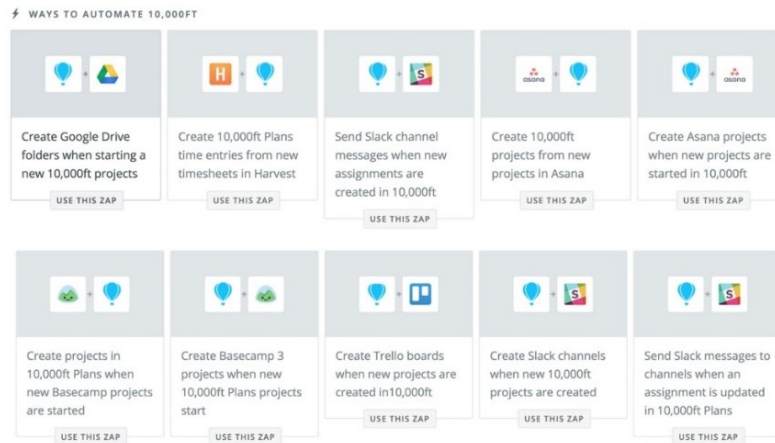


Рисунок 1.4 – Приклади інтеграцій із Zapier

Zapier вирізняється простим у використанні інтерфейсом та інтуїтивно зрозумілим конструктором із функцією перетягування, містить широкий спектр можливостей інтеграції вебдодатків, пропонує доступні тарифні плани, водночас маючи слабкі сторони, такі як обмежену підтримку API, відсутність вбудованого планувальника та неможливість реалізації складної автоматизації робочого процесу.

Macro Maker – це програма, яка записує дії користувача та може використовуватися для автоматизації повторюваних завдань і виконання макросів на комп'ютерах Windows. Як видно на рисунку 1.5, Macro Maker дозволяє користувачам записувати рухи миші та введення з клавіатури, а також використовувати вбудовані функції та змінні.

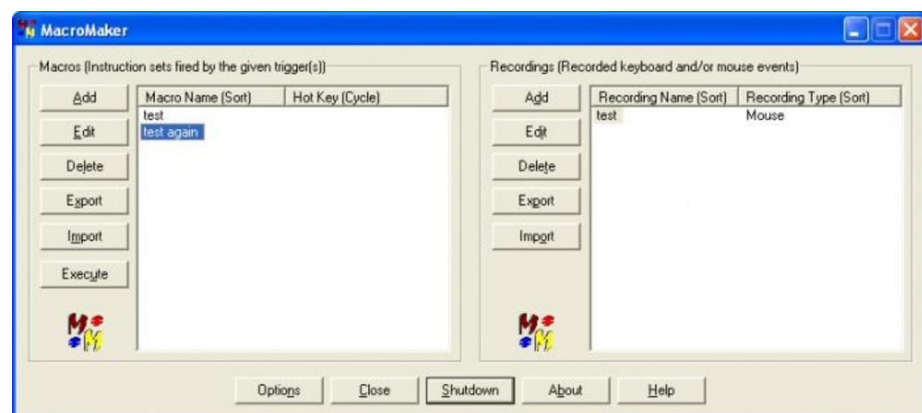


Рисунок 1.5 – Менеджер макросів у Macro Maker

Програма має ряд сильних сторін, серед яких ключовими є можливість безкоштовного завантаження та використання, невеликий розмір, що сприяє легкій інтеграції в різні системи, а також здатність автоматизувати повсякденні процеси з високою продуктивністю. Однак слабкими сторонами є непростий інтерфейс для початківців та обмежені можливості для налаштування складних процесів, що може ускладнити застосування програми для певних категорій користувачів і специфічних завдань.

JetBit Macro Recorder – програма з широким набором вбудованих функцій для автоматизації завдань. Вона дозволяє записувати та автоматизувати дії миші та клавіатури (рисунк 1.6), а також редагувати та налаштовувати макроси за допомогою редактора сценаріїв.

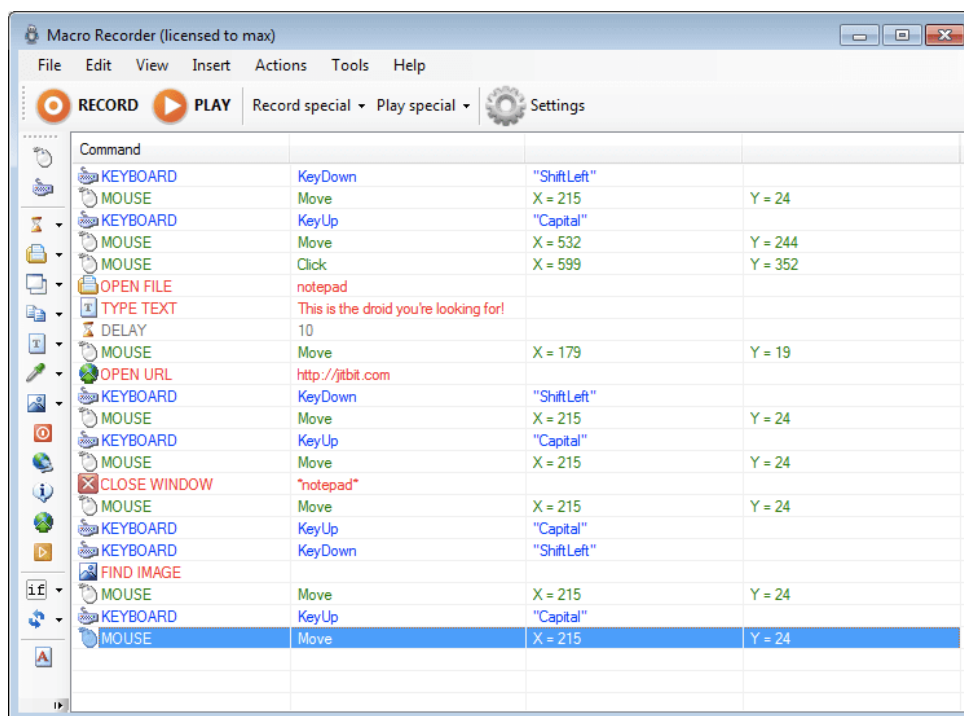


Рисунок 1.6 – Запис макросу у JetBit Macro Recorder

Користувач також може конвертувати свої макроси у виконувані файли, які можна запускати на будь-якому Windows-сумісному комп'ютері. JetBit Macro Recorder можна використовувати для різних цілей, наприклад

для автоматизації рутинних завдань, створення навчальних посібників, тестування програм тощо.

Програма має ряд переваг, таких як комплексний набір інструментів і функцій, доступних для використання в автоматизованих рішеннях, гнучкість у налаштуванні графіка відтворення для оптимізації робочого процесу, а також інтелектуальні засоби для аналізу середовища відтворення, що забезпечують повну сумісність з потребами користувачів. Однак існують обмеження, насамперед, програма сумісна тільки з операційними системами Windows, що ускладнює використання для користувачів з різноманітними пристроями та іншими операційними системами.

Macro Recorder – це кросплатформна програма, яка дозволяє користувачам записувати та відтворювати дії миші та клавіатури. Користувач може редагувати, відтворювати, планувати та експортувати свої макроси за допомогою цього інструменту (рисунок 1.7) [53].

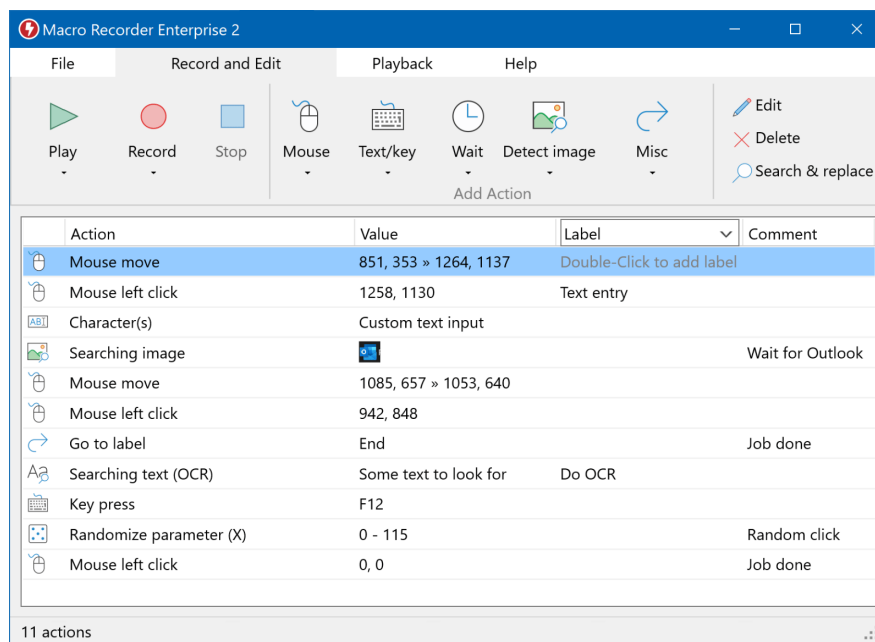


Рисунок 1.7 – Редагування макросу у Macro Recorder

Сильні сторони даної програми включають кросплатформну сумісність з операційними системами Windows і macOS, можливість точної реєстрації координат миші та визначення кольору пікселів, функціональність імітації

натискань клавіш, а також розпізнавання зображень на екрані та встановлення локації вікон. Інтеграція з оболонкою Windows доповнює планування задач, а вбудований редактор підвищує ефективність розроблення автоматизованих сценаріїв. Програма також підтримує необмежений запис та відтворення макросів із зрозумілим користувацьким інтерфейсом. Проте слабкості програми полягають у відсутності розширеної підтримки для складних процесів та API, що може заважати ефективному впровадженню програми в різноманітні робочі середовища.

Ключовим обмеженням сучасних програм для автоматизації процесів є недостатність механізмів, що дозволяють впровадження жестово-графічного управління. Цей тип сприяв би більш інтуїтивному та ефективному управлінню, що є особливо актуальним у контексті збільшення складності та об'єму задач. Інтеграція жестово-графічного управління може виступати як значний прорив у сфері автоматизації, відкриваючи додаткові можливості для оптимізації процесів, зниження ризиків помилок та створення більш надійної та доступної робочої оболонки [32, 36, 59, 74, 100].

1.2 Поняття графічного жесту (ключа) та його практичне застосування

Інформаційна технологія розвивається в геометричній прогресії. До цього моменту основними лишились текстові паролі, але зараз все більшою популярністю користуються графічні ключі [67].

Графічні ключі – це вид ключів автентифікації, які використовують зображення або графічний шаблон, наприклад, перетворюючи графічний об'єкт на точки з певною локацією, замість традиційного тексту або числового коду [3].

Ідея графічної системи автентифікації була представлена в 1996 році. З того часу було запропоновано багато різних графічних систем автентифікації, і кількість нових систем графічних паролів поступово зростала. Цей вид автентифікації базується на дослідженнях, які показують, що людина краще пам'ятає об'єкти та образи, ніж послідовності символів. Це робить графічні ключі привабливими для застосування у таких сферах, як безпека даних і автентифікація користувачів.

Щоб зрозуміти тенденцію розвитку графічних жестів, у роботі [4] було проаналізовано роки, в які були введені різні схеми систем графічних паролів (рис 1.8).

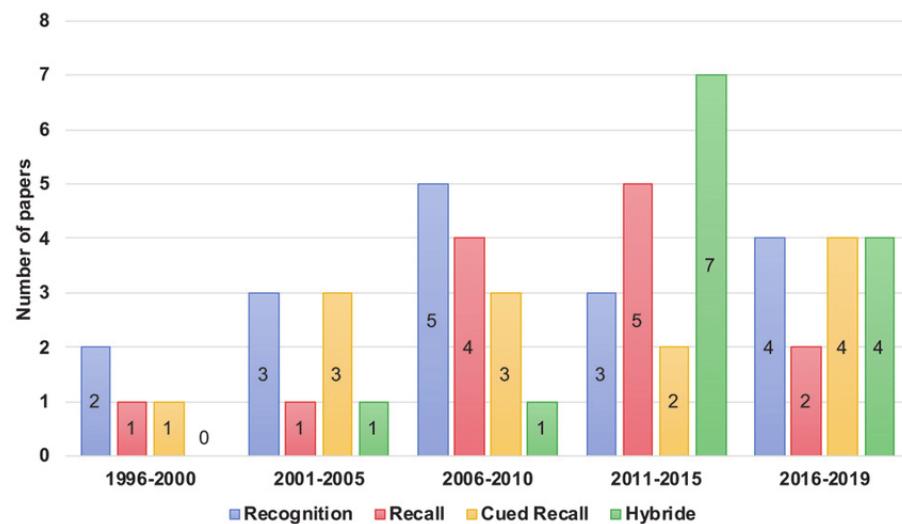


Рисунок 1.8 – Графічні схеми паролів за роками видання.

Як показано на рисунку 1.8, з 1996 по 2000 рр. було введено відносно небагато нових графічних систем автентифікації, і жодної гібридної системи автентифікації не було розроблено в цей період. У 2001–2005 рр. кількість систем графічних паролів у всіх категоріях зросла і були представлені перші гібридні графічні системи паролів. У 2006–2010 роках кількість графічних систем паролів продовжувала зростати, а кількість нових систем графічних паролів на основі розпізнавання досягла максимуму. У 2011–2015 роках кількість гібридних схем автентифікації різко зросла, тоді як кількість систем

графічних паролів на основі розпізнавання впала. За останні 10 років гібридні схеми стали більш популярними, ніж інші методи.

Це пов'язано з тим, що сучасні пристрої здатні сприймати перо, дотик та рухи миші, і стає актуальним отримання корисного значення з більш ніж просто натисків. Одним із таких методів є розпізнавання графічного жесту або форми [18].

Розпізнавання графічних жестів має ряд переваг перед іншими способами взаємодії з комп'ютером, такими як використання миші або клавіатури. До переваг розпізнавання графічних жестів відносяться:

1. Асоціативність. Жести є більш природними для людини, ніж використання комбінацій букв та цифр [57, 98].
2. Інтерактивність. Жести можуть зробити інтерфейс користувача більш інтерактивним і захоплюючим [97].

Розпізнавання графічних жестів має також ряд недоліків [2, 3, 51, 80]. До недоліків розпізнавання графічних жестів відносяться:

1. Точність. Алгоритми розпізнавання графічних жестів можуть бути не дуже точними, особливо якщо жест виконується нечітко.
2. Відмова. Жести можуть бути відхилені алгоритмом розпізнавання, якщо вони не відповідають встановленим критеріям.
3. Навчання. Алгоритми розпізнавання графічних жестів, які навчаються на основі прикладів, вимагають попереднього навчання на наборі даних прикладів жестів.

Існує багато різних алгоритмів розпізнавання графічних жестів [19, 22, 27, 28, 90]. Деякі з найпоширеніших типів алгоритмів включають:

1. Алгоритми на основі ознак. Ці алгоритми використовують статичні або динамічні ознаки жесту для його ідентифікації. Статичні ознаки – це характеристики жесту, такі як його форма, розмір і розташування. Динамічні ознаки – це характеристики жесту, які змінюються з часом, такі як швидкість, прискорення і напрямки.

2. Алгоритми навчання без нагляду. Ці алгоритми навчаються розпізнавати жести без попереднього навчання на наборі даних прикладів жестів. Вони роблять це, використовуючи статистичні методи для виявлення закономірностей у наборі даних.

3. Алгоритми навчання на основі прикладів. Ці алгоритми навчаються розпізнавати жести на наборі даних прикладів жестів. Вони роблять це, використовуючи методи машинного навчання, такі як класифікація або регресія.

Вибір конкретного алгоритму розпізнавання графічних жестів залежить від багатьох факторів, таких як тип жестів, які потрібно розпізнавати, точність, яка необхідна, і складність алгоритму.

Розпізнавання графічних жестів є активною областю досліджень. Дослідники працюють над підвищенням точності результатів, зменшенням кількості відмов і спрощенням навчання алгоритмів [12].

Головною перевагою застосування графічних ключів є їхня здатність забезпечити зручність для користувачів, що базується на еволюційному аспекті людського мозку, який дає перевагу для запам'ятовування образів [11].

Однак цей вид ключів також має обмеження. Деякі з цих обмежень пов'язані з можливими підходами атак на системи графічних паролів або з можливими проблемами, пов'язаними з тим, що користувачі можуть вибирати прогнозовані або прості графічні шаблони. Також дані системи, засновані на розпізнаванні, слабо стійкі до атак через плече, brute-force, dictionary, соціальної інженерії, шпигунського програмного забезпечення та вгадування (рисунок 1.9) [4, 68].

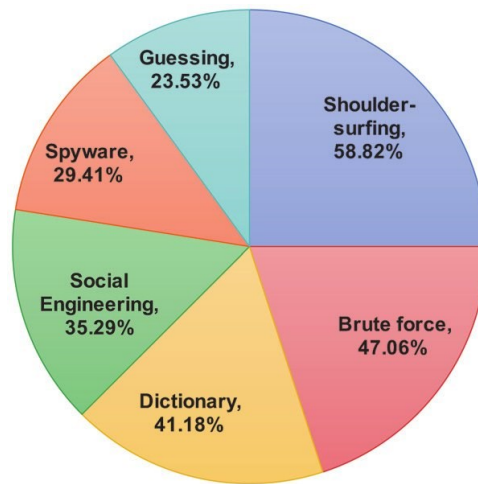


Рисунок 1.9 – Вразливість систем графічного пароля

Передовими розробниками графічного коду на сьогодні є такі програми, як MazeLock та графічний пароль Windows.

MazeLock є системою безпеки, яка базується на концепції розблокування пристроїв за допомогою унікальних графічних жестів, створюючи додатковий рівень захисту від несанкціонованого доступу (рисунок 1.10 а). Ця технологія дозволяє користувачам встановити власний шаблон розблокування, спрощуючи автентифікацію в порівнянні з традиційними PIN-кодами або паролями. Завдяки своїй інтуїтивності та ефективності, MazeLock корисний для застосування в особистих мобільних пристроях, а також надає додаткові опції безпеки для корпоративних користувачів, яким необхідно забезпечити конфіденційність даних на своїх пристроях [56].

MazeLock підтримує функцію створення персональних асоціативних блокувальних шляхів, що сприяє легкості запам'ятовування, але водночас може знижувати рівень безпеки, через потенціал передбачуваності таких шляхів.

Графічний пароль Windows дозволяє користувачам створювати власне графічне зображення, яке слугує паролем для доступу до комп'ютера з операційною системою Windows [66].

На рисунку 1.10 б демонструється використання графічного пароля в операційній системі Windows, який представляє собою запатентований механізм автентифікації користувача. Функція Picture Password дозволяє здійснювати вхід в систему за допомогою специфічних рухів, що виконуються на фоні обраного користувачем зображення. Цей метод вважається більш захищеним у порівнянні з традиційними текстовими паролями, оскільки він комбінує ряд атрибутів: локалізацію точок, напрямки жестів та послідовність дій. Графічний пароль є простим у використанні з точки зору користувацького досвіду, одночасно забезпечуючи надійний захист від несанкціонованого доступу, що робить його популярним вибором для забезпечення безпеки в особистих та робочих пристроях [37].



а)



б)

Рисунок 1.10 – Створення графічного пароля:

а) MazeLock; б) Windows

Інтуїтивність запровадження графічних паролів у Windows виражається у простоті запам'ятовування зображень у порівнянні з комплексними символічними паролями. Графічний пароль крім того може виступати додатковим рівнем захисту, підсилюючи багаторівневі системи автентифікації. Втім, користувачі можуть стикатися з проблемами безпеки при виборі передбачуваних та легких для вгадування зображень.

Налаштовування графічного паролю вимагає додаткового часу, а його функціональність обмежена лише платформою Windows, що робить неможливим застосування на інших операційних системах. Залежність від здатності користувача запам'ятовувати графічний пароль і ризик його перехоплення через спостереження за діями користувача також є суттєвими недоліками даної системи.

У контексті забезпечення інформаційної безпеки застосування системи графічних паролів створює додаткові ризики для захисту даних через потенційну вразливість до атак спостереження та аналізу поверхонь вводу. Проте, значущістю є рішення інтегрувати такі системи як засіб асоціації жестів і швидкого пошуку, опираючись на асоціативне мислення користувачів. У результаті, коли фокус переноситься з передовсім забезпечення безпеки на зручність та швидкість взаємодії із системою, графічні паролі можуть стати органічним розширенням натуральної поведінки людини, спираючись на здатність людського мозку більш ефективно запам'ятовувати образи та шаблони, а не абстрактні символічні послідовності, значно підвищуючи швидкість і точність виконання завдань користувачами [79].

1.3 Дослідження сучасних інструментів розроблення кросплатформних програмних додатків

Кросплатформне розроблення – це процес створення програмного забезпечення, яке здатне працювати на багатьох різних операційних системах та апаратних платформах з мінімальними змінами або без них взагалі. Головне завдання при кросплатформній розробці – створити застосунок, який буде однаково працювати на iOS, Android, Windows, macOS, Linux та інших платформах, використовуючи одне й те ж джерело коду.

Цей підхід дозволяє розробникам значно економити час і ресурси, оскільки не потрібно писати окремий код для кожного пристрою або системи. Також він максимізує охоплення цільової аудиторії, адже додаток стає доступним на багатьох платформах одразу після запуску. З іншого боку, кросплатформні рішення можуть вимагати компромісів щодо продуктивності та індивідуалізації інтерфейсу користувача під конкретні платформи.

Кросплатформне розроблення можливе завдяки застосуванню спеціальних фреймворків, таких як Flutter, React Native та .NET MAUI, які дозволяють створювати додатки з впізнаваним користувацьким інтерфейсом та функціональністю незалежно від того, на якій операційній системі вони працюють. Однак, кожен інструмент приносить не лише можливості, але й виклики. Розглянемо їх на прикладі трьох популярних фреймворків [1, 8].

Flutter пропонує потужну систему для створення гладких та красивих інтерфейсів. Перевагою є його висока продуктивність і відгук користувача, а також гнучкий дизайн, який можна регулювати під будь-яку платформу. Через це Flutter швидко набуває популярності серед розробників. Однак, мова програмування Dart, на якій Flutter базується, не так широко вживана, як JavaScript чи C#, що може бути перешкодою для тих, хто не знайомий з цією мовою. Також, хоча спільнота Flutter швидко розвивається, вона все ще молодша за інші і може мати менше ресурсів для вирішення специфічних проблем.

React Native використовує JavaScript і поєднує елементи веб та мобільного програмування, що робить його ідеальним вибором для веброзробників, які переходять до створення мобільних застосунків. Його здатність інтегруватись із нативним кодом дає можливість створювати дійсно потужні додатки. Проте, іноді досягнення узгодженості на iOS та Android може вимагати додаткових зусиль, і в цілому продуктивність може бути трохи нижчою порівняно з нативними додатками.

.NET MAUI від Microsoft – це сучасний інструмент, що об'єднує розроблення мобільних та настільних застосунків. Він дозволяє розробникам користуватись силами C# і .NET для створення кросплатформних додатків. Завдяки тісній інтеграції з Visual Studio, .NET MAUI пропонує високу продуктивність розроблення та зручні інструменти для дизайну інтерфейсів. Серед інших переваг .NET MAUI можна виділити наступні:

1. Єдиний проєкт для різних платформ. .NET MAUI дозволяє розробникам керувати всіма платформами в одному проєкті. Це означає, що для кожної цільової платформи не буде кількох проєктів. У .NET MAUI спільні файли ресурсів знаходяться в межах одного проєкту, єдиний маніфест програми, який визначає назву, ідентифікатор і версію програми, єдина кросплатформна точка входу в програму, яка допомагає швидше розробляти проєкти та набагато легша в обслуговуванні [35].

2. Єдине місце для управління всіма ресурсами. .NET MAUI керує шрифтами, зображеннями, піктограмами програм, заставкою, необробленими ресурсами та стилями з єдиного місця – папки «Ресурси». .NET MAUI оптимізує їх відповідно до унікальних вимог кожної платформи, що значно економить час і ресурси на підтримку [44].

3. Повна підтримка Hot Reload. Hot Reload дозволяє розробникам вносити зміни в код і застосовувати їх до запущеної програми. Перекомпіляція не потрібна, і, коли це можливо, стан програми зберігається без змін. Це значно прискорює розробку програми [49].

4. Підтримка Blazor. .NET MAUI також чудово підходить для веб-розробників, які хочуть запускати вебкод у рідних клієнтських програмах. Blazor ідеально підходить, коли потрібно створити вебсайт і мобільний додаток, які виглядатимуть однаково [82].

5. Асоціація бібліотек. .NET MAUI спрощує використання таких можливостей пристрою, як датчики, фотографії, контакти та інші служби,

якими клієнт користується регулярно, наприклад автентифікація, сповіщення, Bluetooth, геолокація, файлова система тощо [91].

6. .NET MAUI поставляється з більш ніж 40 компонентами інтерфейсу користувача з коробки. Однією з основних переваг використання рідного інтерфейсу користувача є успадкована підтримка доступності [102].

Проблемою .NET MAUI може бути відносна молодість проєкту, що призводить до меншої кількості навчальних матеріалів та менш зрілої спільноти порівняно з іншими фреймворками.

Для створення повноцінного кросплатформного продукту, крім додатку, необхідно розробити сайт. Існує кілька популярних фреймворків для розроблення сучасних вебпродуктів, кожен з яких має свої особливі переваги та обмеження.

Angular є одним із передових фреймворків, створених Google, який надає набір інструментів для створення динамічних і адаптивних односторінкових додатків (SPA). Завдяки двосторонньому зв'язуванню даних і розширеній вбудованій функціональності Angular дозволяє створювати високоефективні програми. Його недоліки включають високий бар'єр для входу для нових розробників і відносно нижчу продуктивність при роботі зі складними та великими програмами порівняно з деякими новішими фреймворками [42].

React, створений і підтримуваний Facebook, являє собою бібліотеку з відкритим вихідним кодом, що дозволяє розробникам будувати інтерфейси користувача за допомогою декларативного підходу. Головною особливістю React є використання віртуального DOM, який оптимізує повторну візуалізацію інтерфейсу завдяки селективному оновленню лише тих компонентів, які зазнали змін. Такий підхід сприяє підвищенню продуктивності вебдодатків [41].

Blazor, фреймворк розроблений Microsoft, відрізняється тим, що він дозволяє використовувати C# замість JavaScript для побудови вебдодатків.

Blazor використовує стандарт WebAssembly для виконання C# коду безпосередньо у браузері, дозволяючи розробникам втілювати клієнтську частину вебдодатків, використовуючи знайомі .NET парадигми та інструменти [7].

З точки зору інструментів створення програмного забезпечення перевагу у цій роботі було віддано .NET MAUI. Використовуючи .NET MAUI, можна розробляти програми, які можуть працювати на Android, iOS, macOS, Windows та Tizen, з однієї спільної бази кодів. Одна з ключових цілей .NET MAUI – дозволити реалізувати якомога більше логіки програми та макета інтерфейсу користувача в єдиній кодовій базі [61, 64].

Для створення вебдодатку було обрано Blazor. Blazor, реалізований на основі інноваційних технологічних рішень, що спираються на стандарт WebAssembly, що дозволяє використати повноцінний стек .NET для створення додатку. За допомогою Blazor можна реалізувати зручну та швидку взаємодію з користувачем, написавши логіку додатку на мові C# замість JavaScript. Використання Blazor сприяє раціоналізації розроблення вебдодатку, гарантуючи високу продуктивність і забезпечуючи кросплатформну інтерактивність користувацького інтерфейсу.

Кросплатформна розробка істотно економить час і ресурси, в свою чергу зменшуючи обсяг роботи і покращуючи якість кінцевого продукту.

1.4 Дослідження архітектурних рішень

Аналіз архітектури програмного забезпечення є фундаментальним для розуміння його функціонування, розширюваності та масштабованості. Важливість вибору відповідної архітектури не можна недооцінювати, адже вона закладає основу для розвитку, підтримки та продуктивності системи [13, 33].

В контексті інженерії програмного забезпечення, архітектурні рішення сприяють визначенню структурних рамок проєкту, зумовлюють його розвиток та адаптивність до змін. Розгляд архітектур має ґрунтуватися на глибокому аналізі потенційних переваг та обмежень, що є невід'ємною частиною формулювання комплексного інженерного вирішення [29].

Монолітна архітектура забезпечує згуртованість і єдність розроблення при реалізації всіх компонентів програми в одному процесі або пакеті. Дана архітектура підходить для імплементації невеликих чи середніх за обсягом систем і дозволяє застосувати прості та лінійні процеси розгортання. Однак масштабування таких систем обмежене, адже стрункість кодової бази може погіршуватися з часом. Це зумовлює зростання складності внесення змін і ускладнює процес оновлення [72, 83].

Архітектура мікросервісів може враховуватися як альтернатива монолітній під час необхідності гнучкого масштабування та ускладненої функціональності. Вона передбачає декомпозицію функціоналу на невеликі, відокремлені сервіси, кожен з яких є автономним та взаємодіє з іншими через легкі комунікаційні механізми, такі як REST чи gRPC. Це підвищує гнучкість, сприяє паралельній розробці та можливим поступовим оновленням системи. Однак керування кількома мікросервісами збільшує складність інфраструктури та потребує значного споживання ресурсів для координації та моніторингу системи [16, 25].

Керовані подіями архітектури розглядають системи як збірні компоненти, які реагують на події або стани, що змінилися. Такий підхід сприяє високому рівню розв'язки компонентів, збільшуючи масштабованість і реактивність системи. Водночас це вносить додаткову складність у визначення потоків даних, обробку помилок, а також збереження транзакційної цілісності у розподілених системах [76].

Архітектура мікроядра, що також відома як архітектура плагінів, передбачає розвиток системи навколо стабільного, мінімалістичного ядра,

котре реалізує фундаментальну функціональність, тоді як додаткові можливості реалізуються застосуванням модулів або компонентів. Цей підхід підвищує адаптивність і надійність системи за рахунок легкої заміни або оновлення плагінів без впливу на ядро. З іншого боку, він вимагає високого рівня дисципліни в архітектурі й чіткої дефініції контрактів інтерфейсів між ядром та плагінами для уникнення зайвих затримок в опрацюванні запитів [5, 26, 58, 63, 78, 81, 95].

Чиста архітектура. Сучасні програми мають як вебсайти, так і мобільні/настільні програми для синхронізації даних користувачів на різних платформах. Більшість моделей і бізнес-логіки дублюються, що затягує і ускладнює розроблення, оскільки функціонал потрібно синхронізувати. У цьому випадку бізнес-логіка може відрізнитися для вебсайту та мобільного додатку (вебсайт зберігає дані в базі даних, мобільний додаток надсилає запит на сайт). Чиста архітектура вирішує цю проблему шляхом вилучення загального коду – моделей та інтерфейсів, які можна використовувати як у веб, так і в мобільних програмах. Він називається «Домен» і «Додаток» відповідно [39, 55, 62, 77].

Спеціальний для платформи код реалізовано в інфраструктурі та інтерфейсі користувача. Це 4 базові рівні чистої архітектури (рисунки 1.11) [23].

Кількість рівнів може бути різною, але центром завжди є модель предметної області, тобто ті класи моделі, які використовуються в додатку і об'єкти яких зберігаються в базі даних [17].

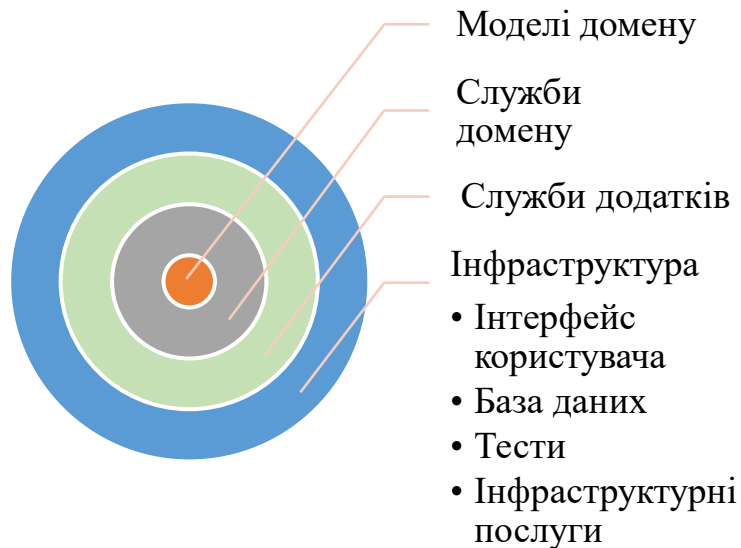


Рисунок 1.11 – Чиста архітектура

Чиста архітектура забезпечує кращу можливість тестування, оскільки модульний тест може бути створений для окремих шарів без впливу інших модулів програми [43].

У дослідженні [6] проведено порівняльний огляд відмінностей між монолітними та мікросервісними архітектурами, їх перевагами і недоліками. Зазначене дослідження покладається на теоретичне порівняння їх характеристик та практичне зіставлення різних сценаріїв використання.

Розгляд архітектур розкриває, що монолітні системи, навіть будучи відносно простішими у впровадженні та потенційно кращими в управлінні залежностями, стикаються із серйозними обмеженнями під час масштабування та внесення змін. З іншого боку, мікросервісна архітектура, хоча й забезпечує вищий ступінь гнучкості і незалежності компонентів, може привести до складнощів у координації сервісів та управлінні інфраструктурою [34, 65].

У зв'язку з цим, існує потреба у розробленні нового архітектурного рішення, здатного поєднати в собі переваги згаданих моделей та уникнути їх недоліків, актуальним є розроблення архітектури мікроядра з «центральною

ядром» (чиста архітектура), яка спрямована на забезпечення ефективності, масштабованості та високого рівня модульності [10].

1.5 Висновки до розділу 1

У першому розділі було проведено дослідження існуючих програмних продуктів для автоматизації процесів на основі запису макросів. Було виявлено, що дані програми є дуже актуальними і необхідними у наш час через те, що надають користувачам зручний спосіб автоматизувати повторювані завдання та підвищити продуктивність. Але хоча на даний момент і існують програмні продукти, які дозволяють встановлювати макроси для полегшення виклику програм, проте кожен з них має як переваги так і недоліки, які потребують вирішення.

Досліджено поняття графічного жесту, методи його розпізнавання та сфери застосування. Проаналізовано, що в контексті забезпечення інформаційної безпеки застосування системи графічних паролів має певні проблеми та створює додаткові ризики для захисту даних. Проте, з іншого боку, показано, що графічні ключі можуть бути важливим напрямом у пошуку альтернативи традиційним засобам людино-машинної взаємодії та запропоновано застосовувати їх як засіб асоціації і швидкого пошуку макросів у системах автоматизації процесів та керування комп'ютеризованими приладами, опираючись на асоціативне мислення користувачів. У результаті, коли фокус переноситься з забезпечення безпеки на зручність та швидкість взаємодії із системою, графічні паролі можуть стати органічним розширенням натуральної поведінки людини, спираючись на здатність людського мозку більш ефективно запам'ятовувати образи та шаблони, а не абстрактні символічні послідовності, значно підвищуючи швидкість і точність виконання завдань користувачами.

Розглянуто сучасні інструменти розроблення кросплатформних програмних додатків та обґрунтовано вибір технологій, що застосовуються у роботі.

Проаналізовано існуючі архітектурні рішення, їх особливості, переваги та недоліки. Обґрунтовано, що на даний час відсутнє ефективне рішення для систем автоматизації процесів та взаємодії з комп'ютеризованими пристроями та актуальною задачею є розроблення нової архітектури.

РОЗДІЛ 2. АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ «DRAW & GO»

При створенні системи автоматизації процесів «Draw & GO» було розроблено ряд рішень та концепцій, які складають алгоритмічне ядро системи. Загальний опис організації робочого процесу представлений у підрозділі 2.1. Запропонована концепція асоціативного графічного інтерфейсу та удосконалений алгоритм розпізнавання графічних жестів викладено у підрозділі 2.2. Підрозділ 2.3 присвячено технології автоматизованого створення макросів на основі застосування мовної моделі штучного інтелекту. Висновки за розділом подані у підрозділі 2.4.

2.1 Організація робочого процесу

«Draw & GO» – це інноваційний продукт, що революціонізує спосіб, яким людина взаємодіє з технологіями через мистецтво малювання та автоматизацію. Він дозволяє користувачам легко створювати, керувати та виконувати складні задачі, процеси або дії за допомогою інтуїтивно зрозумілого інтерфейсу, де основним засобом вводу є малювання та перетягування.

Завдяки «Draw & GO», користувачі можуть візуалізувати свої ідеї, концепції або алгоритми набагато швидше та ефективніше, ніж традиційними методами кодування або ручного введення. Це може бути особливо корисно в освітніх цілях, дизайні, програмуванні, а також для спрощення повсякденних завдань через автоматизацію.

На рисунку 2.1 представлено блок-схему, яка складає основу системи автоматизації «Draw & GO». Схема демонструє кроки від визначення робочого процесу до його виконання та моніторингу результатів. Візуалізація

яскраво відображає ланцюжок дій, що включає встановлення тригерів та умов, під якими викликаються певні задачі, визначення алгоритмів обробки даних та інтеграції з зовнішніми системами чи платформами.

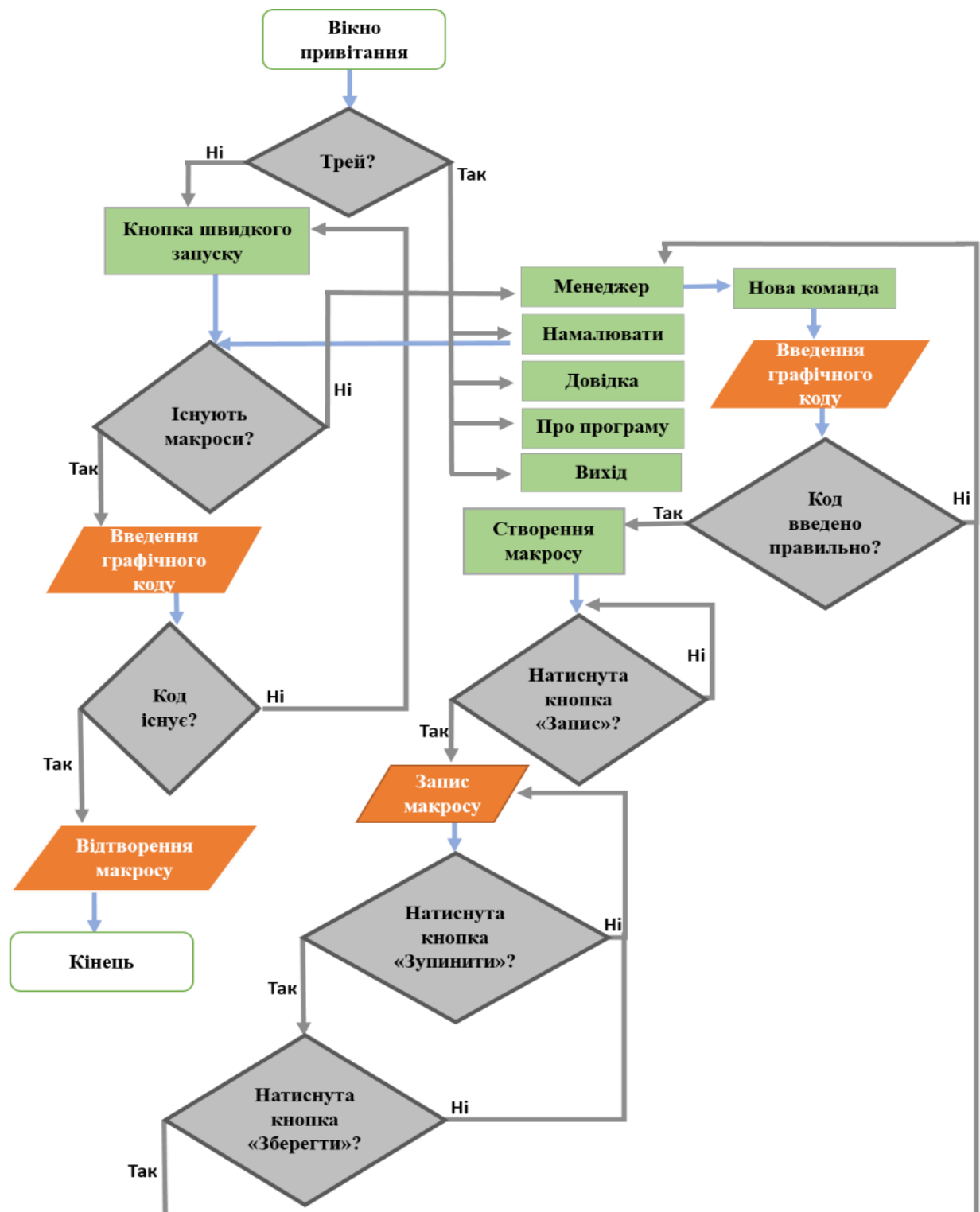


Рисунок 2.1 – Блок-схема організації робочого процесу

Програма запускається і відображає вікно привітання. У цьому вікні користувач може вибрати, чи хоче він відкрити вікно швидкого запуску.

Якщо користувач вибирає відкрити вікно швидкого запуску, то програма відкриває це вікно. У вікні швидкого запуску користувач може вибрати одну з наступних дій:

1. Створити новий макрос: користувач вводить графічний код макросу. Код макросу складається з послідовності команд, які будуть виконуватися при відтворенні макросу.
2. Відтворити існуючий макрос: користувач вибирає макрос із списку і натискає кнопку «Відтворити».
3. Вийти з програми: користувач закриває вікно швидкого запуску.

Якщо користувач вибирає «Створити новий макрос», то програма переходить до кроку створення макросу. На цьому кроці користувач повинен ввести графічний код для макросу.

Після введення коду макросу користувач повинен створити макрос. Макрос може бути введений у вигляді послідовності команд, які використовуються в програмі. Наприклад, це можуть бути команди для переміщення курсору, введення тексту, виконання дій з файлами тощо. Або натиснути кнопку «Запис». Це запустить процес запису макросу. Під час запису макросу програма записує кожну дію користувача, формуючи послідовність команд.

При роботі з макросами користувач може використовувати мишу та клавіатуру. Проте, для коректної роботи, необхідно відслідковувати дії миші та клавіатури, навіть якщо програма не активна.

Існує 2 типи перехватів – локальний та глобальний.

Перший контролює дії, що відбуваються для певної програми. Другий контролює всю систему [24].

У програмі «Draw & GO» використовується глобальний перехват миші та клавіатури. Для цього створені наступні обробники подій: MouseMove, MouseDown, MouseUp, MouseWheel, KeyDown, KeyUp.

Алгоритм запису макросу:

1. Користувач натискає кнопку запису. Одразу відбувається перехват подій миші та клавіатури. В залежності від події, яку виконує користувач, відповідний обробник додає подію до списку команд та збереже час останньої події.
2. Зберігаємо інформацію про розкладку клавіатури. Потрібно, аби вона збігалася з початковою, аби макрос відтворювався так само, як і під час запису.
3. Для завершення запису макросу користувач натискає кнопку зупинки. Зупиняється таймер, який рахує час запису макросу. Припиняється глобальний перехват подій.

Якщо користувач вибирає «Відтворити існуючий макрос», то програма переходить до кроку відтворення макросу. На цьому кроці програма виконує макрос, який був записаний раніше. Програма виконує код макросу послідовно, від початку до кінця. Користувач може зупинити відтворення макросу, натиснувши кнопку «Зупинити».

Алгоритм відтворення макросу:

1. Перед початком відтворення збереженого макросу програма встановлює мову, яка була на початку запису.
2. Користувач натискає кнопку «Відтворити». Відбувається відкриття макросу та додання кожної події до відповідного списку.
3. Для кожної записаної команди, програма повертає час між ними і, в залежності від події, яку виконує користувач, відповідний обробник відтворює команду.

Єдиним недоліком усіх перехватів є властивість уповільнювати систему, оскільки вони збільшують обсяг обробки і система повинна

оброблювати кожне повідомлення. Тому, бажано, встановлювати функцію тільки тоді, коли це необхідно, що і було використано у даній програмі.

Блок-схема чітко ілюструє потік виконання програми. Вона може бути використана для розуміння того, як працює програма, або для розроблення нових функцій програми.

2.2 Концепція асоціативного графічного інтерфейсу та розпізнавання графічних жестів

Використання графічних жестів як частини системи вводу відкриває нові можливості для взаємодії, створення та навігації. Користувачі можуть створювати унікальні жести, що співвідносяться з визначеними діями, створюючи особистий макрос для кожного жесту. Концепції асоціативного графічного інтерфейсу та розпізнавання графічних жестів включає не лише класичне введення за допомогою клавіатури та миші, а й нововведення такі як перо, сенсорні екрани та технології розпізнавання жестів.

Цей розділ фокусується на розгляді того, як графічні жести можуть бути використані для взаємодії з інтерфейсами, а також на розгляді технологій, що дозволяють це здійснювати. Розглянуто, як користувач може малювати жест (використовуючи мишу, перо або палець) для створення макросу і як система може розпізнавати цей жест для виконання відповідних дій.

Розпізнавання жестів за допомогою камери є одним із найбільш інноваційних напрямків у сфері інтерактивних технологій. Цей метод передбачає використання відеокамери або спеціалізованого сенсорного обладнання, яке здатне фіксувати рухи користувача, такі як махання рукою, і трансформувати їх у команди для управління комп'ютеризованими системами. Це дозволяє створити більш природній та інтуїтивний спосіб

взаємодії, який може бути особливо зручним у ситуаціях, коли використання традиційних пристроїв вводу неможливе або незручне [84].

Однак, існує кілька обмежень, пов'язаних з технологією розпізнавання жестів за допомогою камери. По-перше, для точного розпізнавання жестів необхідне спеціалізоване обладнання, таке як камери високої роздільної здатності та інфрачервоні сенсори, що здатні вловлювати рухи в тривимірному просторі. Таке обладнання може бути дорогим та вимагати додаткового простору для розміщення [86].

По-друге, системи розпізнавання жестів можуть бути обмежені у своїй здатності ідентифікувати широкий спектр жестів, особливо в умовах, де присутня зовнішня інтерференція або недостатнє освітлення. Також можуть виникати складнощі в інтерпретації схожих жестів, що вимагає від користувачів виконання дуже точних та відмінних один від одного рухів, що може бути не завжди зручно.

По-третє, використання камер для розпізнавання жестів може порушувати конфіденційність користувачів, оскільки вимагає постійного відеоспостереження, що може бути неприйнятно в деяких сценаріях використання.

На противагу цьому, системи вводу за допомогою миші або пера не мають багатьох з цих обмежень. Вони не вимагають спеціалізованого обладнання, крім самої миші або стилуса, та працюють у широкому діапазоні умов освітлення та без фізичних обмежень простору. Такі системи також дозволяють користувачам виконувати більш тонкі та складні дії, що можуть бути важко розпізнати за допомогою камери. Крім того, вони не створюють проблем із конфіденційністю, оскільки не здійснюють відеозапису дій користувача.

У роботі запропоновано використовувати графічні ключі, як спосіб зв'язування асоціативної команди з жестом.

Одним із таких методів є використання розпізнавання жесту або форми, а саме: користувач малює безперервну форму, а комп'ютер визначає, що було зроблено, зіставляючи зі заздалегідь визначеною бібліотекою форм. Після того, як комп'ютер визначить форму, для неї може бути задана будь-яка дія.

Для забезпечення більш гнучкої та інтуїтивно-зрозумілої взаємодії з цифровими системами, користувач отримує можливість контролювати систему шляхом малювання графічних жестів. Ці жести, що вводяться за допомогою таких пристроїв, як мишка, перо або навіть палець на сенсорному екрані, є ключами до заданого виконуваного макросу.

Створення макросу: Використовуючи будь-який з перерахованих пристроїв вводу, користувач створює унікальний графічний символ на екрані. Наприклад, користувач може намалювати букву “М” на сенсорному екрані пальцем. На основі цього жесту, користувач може визначити дію, яка має бути виконана при розпізнанні того ж жесту у майбутньому – ця дія може включати запуск окремої програми, виконання команди тощо. Так створюється макрос.

Розпізнавання: Після створення макросу, система розпізнає графічний символ або жест, який був асоційований з цим макросом, коли користувач використовує його в майбутньому. Наприклад, якщо користувач знову малює букву “М” пальцем на екрані, система розпізнає цей жест і виконує асоційовану з ним дію. Це полегшує ефективне використання системи і робить її більш інтуїтивно-зрозумілою для користувача.

У роботі було реалізовано 2 методи розпізнавання графічних жестів. Було визначено, що жест вважається розпізнаним, якщо точність перевищує 75%.

Метод розпізнавання жестів №1

За основу методу було використано Draw-A-Secret (DAS) [60]. У DAS користувачі повинні намалювати свій пароль на 2D (5×5) сітці за допомогою

стилуса або миші. Малюнок складається з одного або кількох штрихів, розділених «пером». Користувачі можуть малювати паролі скільки завгодно довго. Щоб пройти автентифікацію, користувачам потрібно перемалювати той самий пароль. Цей метод пропонує великий теоретичний простір, який можна порівняти з текстовими паролями [94].

Для створення графічного коду намальований жест перетворюється у набір символів:

- великі літери англійського алфавіту;
- числа.

Алгоритм створення графічного коду:

1. Визначаємо зону пошуку як прямокутник, який охоплює намальований код.
2. Накладаємо квадратну сітку над цим прямокутником. Нумерація полів сітки починається з 0 з верхнього лівого до правого нижнього кутів.
3. Здійснюємо відстеження полів, в яких переміщувався курсор миші, починаючи з події MouseDown і закінчуючи на події MouseUp;
4. Перетворюємо номери поля у відповідний текст;
5. Використаємо шифр Цезаря (зсув кожної літери на 1 позицію). За допомогою циклу виконуємо зсув, доки не повернемось до початкового коду [93];
6. Оскільки код малюється тричі (для кращого навчання), то отримаємо списки кодів. Виконуємо перевірку наявності однакових кодів, враховуючи їх довжину.

Алгоритм розпізнання графічного коду:

1. Завантажуємо збережені коди;
2. Перевіримо наявності щойно намальованого і обробленого за допомогою шифру Цезаря коду у списку раніше збережених кодів.

Як можна бачити на рисунку 2.2, дуже важливим моментом є кількість клітинок на які ділиться прямокутник.

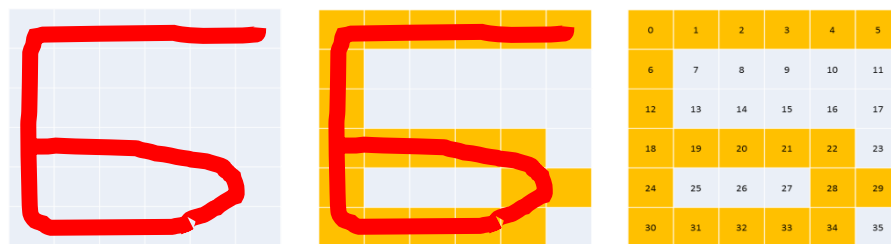


Рисунок 2.2 – Аналіз жесту

Якщо число занадто мале, то не буде достатньої кількості полів, аби розрізняти схожі жести. Щоб проілюструвати це, представимо, що найменший можливий дільник дорівнює 1, тобто не ділити прямокутника. Тому траєкторія кожного жесту дорівнюватиме 0, незалежно від того, буде це точка або мільйони рухів. З іншого боку, якщо ми розділимо прямокутник в занадто багатьох областях, є багато кодів, які є дуже схожими. Провівши декілька тестів, було виявлено, що найкраще використовувати число 6, як дільник. В результаті отримаємо 36 полів.

Метод розпізнавання жестів №2

На початку роботи алгоритму жест представляє собою набір нерівномірно розташованих точок (x_i, y_j) і може містити більше або менше потрібної точності.

1. Розіб'ємо графічний жест на сегменти і знайдемо бажану довжину кожного з них $L_{\text{бaj}} = \frac{\sum_{i=1}^{n-1} \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2}}{N}$, де n – кількість точок, N – бажана кількість точок.

У роботі N було обрано рівним 100, проте чим більше точок, які розподіляємо для жесту, тим точніше порівняння має бути (принаймні теоретично). Однак, чим більше точок звир'яєм для кожного жесту, тим повільніше буде загальне порівняння. Розподіл жесту передбачає можливість істотно збільшити чи зменшити кількість точок в жесті до певного фіксованого значення. Так, наприклад, простий жест, подібний до прямої лінії, може становити лише 2 точки, але складніший жест, подібний до М,

займе мінімум 5 точок. Насправді більшість жестів матимуть набагато більше точок, оскільки вони зроблені за допомогою миші, пальця або інших методів вводу.

2. Встановимо початкові значення: поточна довжина сегменту $L_{\text{пот}} = 0$, допоміжна точка дорівнює першій вхідній точці, вона ж є першою шуканою точкою.

3. Встановимо границі індексів: $1 \leq i < n$ та $1 \leq j \leq N$.

4. Встановимо поточну точку як P_i

5. Знайдемо приріст поточної довжини сегменту, як відстань між допоміжною точкою і поточною:

$$increment = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

6. Знайдемо тестову довжину сегменту:

$$l = L_{\text{пот}} + increment$$

7. Якщо $l < L_{\text{баж}}$, то $L_{\text{пот}}$ присвоюємо l , допоміжну точку присвоюємо поточній, переходимо на крок 4.

8. Підрахуємо коефіцієнт пропорційності для знаходження нової точки:

$$coef = \frac{L_{\text{баж}} - L_{\text{пот}}}{increment}$$

9. Знайдемо координати шуканої точки (рисунок 2.3)

$$X_j = (1 - coef_i) * x_{i-1} + coef_i * x_i$$

$$Y_j = (1 - coef_i) * y_{i-1} + coef_i * y_i$$

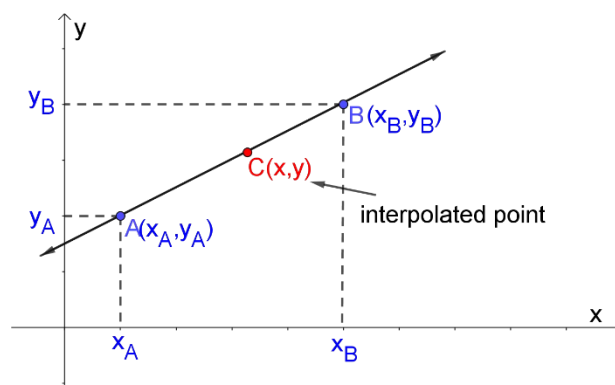


Рисунок 2.3 – Пошук точки інтерполяції

10. Якщо $j < N$, встановлюємо допоміжну точку за шукану, $L_{\text{пот}} = 0$, зменшуємо індекс «і» на 1 і повертаємося на крок 4.

11. Розподіл завершено. Після розподілу фігури рухів зберігаються, всі точки тепер рівномірно розподілені і містять потрібну кількість точок для порівняння.

12. Після завершення розподілу масив точок перетворюємо у масив кутів. Робимо це для порівняння кутової різниці між двома жестами. Отримуємо масив кутів для кожного жесту.

$$angle_j = \tan^{-1}(Y_{j+1} - Y_j, X_{j+1} - X_j)$$

13. Тепер, коли є кути для кожного жесту, можемо перебирати кожен кут у жесті та обчислити кутову різницю між двома кутами. Розмістимо їх в створеному масиві.

14. Якщо $angleDifferences_j = |angle_{j+1} - angle_j|$ та $angleDifferences_j > \pi$, то

$$angleDifferences_j = \pi - (angleDifferences_j - \pi)$$

15. За допомогою масиву обчислимо середню кутову різницю між двома жестами, і перетворимо її в точність збігу (від 0% до 100%). Це все, що потрібно для точного розпізнавання різниці між двома жестами. Як тільки є вірогідність, все помістимо в результуючий клас `PointPatternMatchResult`, що містить ідентифікатор жесту та ймовірності.

$$probability = 100 - \frac{\sum_{i=1}^n angleDifferences_i}{n} * \frac{100}{\pi}$$

Вищезначене – це порівняння масиву точок (для поточного жесту) з єдиним зразком для списку збережених жестів. Далі потрібно підрахувати ймовірність для кожного жесту в наборі зразків, що допоможе оцінити їх за кращою відповідністю, виходячи з найбільшої ймовірності.

Цей спосіб розпізнавання жестів та форм виявився значно швидшим і точнішим у порівнянні з першим методом розпізнавання.

Однак обидва методи працюють лише з однією лінією. Це обмежує кількість простих асоціацій. Тому було вирішено розширити і дозволити користувачам малювати кілька ліній. Для цього було розроблено розширення алгоритму розпізнавання жестів, щоб він міг обробляти кілька ліній.

Удосконалений алгоритм розпізнавання жестів складається з таких кроків:

1. Додавання ідентифікації ліній: кожна лінія у жесті повинна мати унікальний порядковий номер або ідентифікатор, який визначає її позицію в загальному порядку жесту.

2. Виконання алгоритму розпізнавання жестів для кожної лінії окремо: використовувати описаний вище алгоритм для кожної лінії з урахуванням її порядкового номера. Це означає, що потрібно розподілити точки, обчислити масив кутів, різницю кутів та ймовірність для кожної окремої лінії.

3. Збереження результатів для кожної лінії: після обробки кожної лінії зберігати результати в структурі, яка дозволяє асоціювати результати з відповідною лінією.

4. Порівняння жестів: коли всі лінії оброблено, порівняти результати для кожної лінії з відповідною лінією в зразку жесту. Це може включати порівняння порядкових номерів ліній і відповідних кутових різниць.

5. Обчислення загальної ймовірності: після порівняння всіх ліній, обчислити загальну ймовірність збігу, яка може бути середнім значенням ймовірностей для всіх ліній або ваговим середнім, якщо деякі лінії вважаються важливішими за інші.

6. Визначення найкращого збігу: вибрати жест з набору зразків, який має найвищу загальну ймовірність збігу з розпізнаним жестом.

7. Повернення результату: повернути результати розпізнавання, які можуть включати ідентифікатор найкращого збігу жесту, загальну ймовірність збігу та іншу відповідну інформацію.

Цей підхід дозволяє розширити алгоритм розпізнавання жестів для обробки багатолінійних жестів, незалежно від складності намальованого жесту.

2.3 Технологія автоматизованого створення макросів на основі застосування мовної моделі штучного інтелекту

У сучасному світі технологічного прогресу, використання штучного інтелекту в розробленні клієнт-орієнтованих додатків набуває все більшої популярності. Певні можливості та API, наприклад Azure OpenAI Service REST API [15], доступні для розробників, надають інструменти для підвищення ефективності та еластичності рішень, а також полегшують процес роботи із застосунками.

Актуальним напрямком досліджень є використання штучного інтелекту в системах автоматизації та документообігу, що базується на низці ключових факторів:

1. Зростання даних. Світ генерує безпрецедентні обсяги даних, які люди не можуть проаналізувати самотійно. Штучний інтелект дозволяє обробляти та інтерпретувати ці дані, виявляючи приховані закономірності та отримуючи цінну інформацію [46].

2. Персоналізований досвід. Штучний інтелект дозволяє створювати персоналізований досвід для користувачів на основі їхніх індивідуальних потреб і вподобань. Це призводить до кращого залучення та лояльності клієнтів.

3. Покращене прийняття рішень. Штучний інтелект може допомогти в кращому прийнятті рішень, надаючи аналітику та прогнози на основі даних. Це призводить до кращих результатів у бізнесі, науці та інших сферах.

Додатки штучного інтелекту стають все більш доступними та універсальними. Вони використовуються широким колом організацій, від малого бізнесу до великих корпорацій, а також державними та некомерційними організаціями [71].

Сьогодні важливими досягненнями в галузі штучного інтелекту є великі мовні моделі (LLM) [30, 85] та технології Prompt Engineering [70]. Деякі з найпопулярніших великих мовних моделей включають:

- OpenAI GPT (Generative Pre-trained Transformer) відомий своєю здатністю генерувати зв'язний, інформативний текст на основі інструкцій, які він отримує;
- Gemini, що є спробою створити більш універсальну та ефективну мовну модель, здатну адаптуватися до різноманітних завдань обробки мови;
- Facebook AI LLaMa, який також є високопродуктивною моделлю, призначеною для різноманітних додатків у сфері обробки природної мови.

З появою мовних моделей з'являється термін Prompt Engineering. Prompt Engineering є відносно новою дисципліною для розроблення та оптимізації підказок для ефективного використання мовних моделей для широкого кола додатків і тем досліджень [73].

Деякі з найпопулярніших методів оперативної інженерії:

1. Prompt chaining – це метод, при якому завдання розбивається на підзадачі з метою створення ланцюжка операційних операцій [69].
2. Few-shot learning – це метод, який дозволяє мовній моделі вчитися на невеликій кількості прикладів. Цей метод може бути використаний для навчання мовної моделі виконанню нових завдань, для яких не так багато даних [47, 75].

3. Zero-shot learning – це метод, який дозволяє мовній моделі виконувати завдання без будь-яких навчальних даних. Цей метод використовує знання, отримані мовною моделлю в ході попереднього навчання, з метою узагальнення на нові завдання [48].

У роботі запропоновано технологію застосування ШІ для розуміння людської мови і автоматичної побудови макросів [9]. Це дозволяє користувачам будувати робочі процеси, описуючи лише вимоги до них, без необхідності поглиблюватись у знання програмування або навіть самої системи та складається з таких кроків:

1. Обираємо мовну модель штучного інтелекту. Проаналізувавши можливості інтеграції LLMs, було виявлено, що OpenAI надає найпростіший спосіб використання, тому розглянемо саме цю мовну модель. Наразі штучний інтелект від OpenAI пропонує дві популярні моделі: GPT-3.5 та GPT-4. Хоча обидві моделі базуються на архітектурі перетворення, GPT-4, орієнтовно, демонструє більше покращень в точності, здатності до генерального навчання і розпізнавання шаблонів. Проте GPT-4 дорожча і повільніша модель, оскільки надає більш розгорнуту відповідь. Порівнявши обидві моделі і якість результату було обрано використовувати модель GPT-3.5.

2. Підключаємо бібліотеки для роботи з OpenAI. Для підключення моделі було використано OpenAI SDK NuGet пакет [21].

3. Здійснюємо prompt engineering з метою навчити модель ШІ розуміти систему. Необхідно провести декілька експериментів для визначення того, які саме техніки prompt engineering краще застосувати і якої інформації буде достатньо для повернення коректних даних. Наприклад, для інтеграції ШІ з системою «Draw & GO» було розроблено такі інструкції з користування кожною макрокомандою:

3.1. Надаємо базову роль, щоб система адаптувала відповіді, ніби експерт у області. Prompt: *«You know how to use «Draw & GO» plugins»*.

3.2. Надаємо список макрокоманд, що є у системі, для того, щоб мовна модель повертала результати, які зрозуміє «Draw & GO». Prompt: *«You know such plugins as: Start, Search, ConsoleLog, SendEmail, Text, Number, Boolean».*

3.3. Надаємо базову інформацію про макрокоманду. Prompt: *«Each plugin has its own set of parameters. The parameter has its name, type, and value. The parameter name is a string. The parameter value is an object. The parameter type is a string. The parameter type can be one of the following: string, number, boolean, or any.*

Each plugin has its own set of Connectors. There can be input and output connectors. Input connectors are used to connect plugins. Output connectors are used to connect plugins. Output connectors can only be set to input connectors».

3.4. Надаємо детальну інформацію про кожен параметр для кожної макрокоманди. Prompt: *«The start plugin has no input connectors. The start plugin has one output connector. The start plugin has no parameters.*

The start plugin has a single output parameter with the name Start.

The search plugin has one input connector with the name In. The search plugin has one output connector with the name Out. Search plugin has 2 parameters: Search (with type string) and SearchProvider (with type string).

ConsoleLog plugin has one input connector with the name In. The search plugin has one output connector with the name Out. Console plugin has 1 parameter: Message (with type string). »

3.5. За необхідності надаємо обмеження або правила для загальної системи, що допоможе звужити відповідь до більш точного результату. Prompt: *«Macros must contain one and only one Start plugin. It's the start of the flow execution. »*

3.6. Надаємо інформацію про те, які будуть запити та як система буде використовуватись. Prompt: *«The user will ask you to generate a flow diagram*

for «Draw & GO» plugins in general format. You will need to understand the user's request and generate a flow diagram for «Draw & GO» plugins. »

3.7. Задаємо вигляд у якому очікуємо отримати результат. Prompt: «In output, you must provide a valid JSON object with plugins and links (connectors). Example valid output:

```
{
  "name": "Weather in Dnipro",
  "macroCommands": [{
    "Type": "DrawGo.Plugin.Essentials.Extensions.Start",
    "Id": "68B34584-7649-4582-974F-FA8608570D58",
    "Position": "0:0",
    "FieldData": []
  }],
  "Type":
  "WebExtensions.SearchMacroCommand.SearchMacroCommand",
  "Id": "E7625C90-C643-4CA7-B093-7374BE9DBC71",
  "Position": "199:266",
  "FieldData": [
    { "DataType": 5, "Key": "Search", "Value": "Weather in
Dnipro" },
    { "DataType": 5, "Key": "SearchProvider", "Value":
"Google" } ] ],
  "links": [{
    "DestinationKey": "In",
    "DestinationNode": "E7625C90-C643-4CA7-B093-
7374BE9DBC71",
    "SourceKey": "StartFlow",
    "SourceNode": "68B34584-7649-4582-974F-FA8608570D58"
  } ] }»
```

3.8. Після загальних налаштувань система готова приймати команди від користувачів. Необхідно підкреслити, що для формування цих команд користувачам не потрібно мати глибокі технічні знання або вивчати складний синтаксис команд.

Система розрахована на використання звичайної людської мови для формулювання запитів, подібно до того, як люди вводять запити у пошукові системи, такі як Google. Наприклад, користувач може просто ввести “відкрити пошту” або “запустити музичний плеєр”, і система відповідно розпізнає і виконує ці команди. Прикладом такої команди може бути:

«Generate a flow diagram for «Draw & GO» plugins.

I want to search for a Smart Home in Bing. I want to write a log message with text containing the current date and time. »

В результаті отримаємо наступну відповідь (рисунок 2.4):

```
{
  "name": "Smart Home Search and Log",
  "macroCommands": [
    {
      "Type": "DrawGo.Plugin.Essentials.Extensions.Start",
      "Id": "1",
      "Position": "0:0",
      "FieldData": []
    },
    {
      "Type": "WebExtensions.SearchMacroCommand.SearchMacroCommand",
      "Id": "2",
      "Position": "200:100",
      "FieldData": [
        { "DataType": 5, "Key": "Search", "Value": "Smart Home" },
        { "DataType": 5, "Key": "SearchProvider", "Value": "Google" }
      ]
    },
    {
      "Type": "DrawGo.Base.Extensions.Logs.ConsoleLog",
      "Id": "3",
      "Position": "400:200",
      "FieldData": [ { "DataType": 5, "Key": "Message", "Value": "Current Date and Time" } ]
    }
  ],
  "links": [
    {
      "DestinationKey": "In",
      "DestinationNode": "2",
      "SourceKey": "StartFlow",
      "SourceNode": "1"
    },
    {
      "DestinationKey": "In",
      "DestinationNode": "3",
      "SourceKey": "Out",
      "SourceNode": "2"
    }
  ]
}
```

Рисунок 2.4 – Результат, згенерований за допомогою ІІІ

Отриманий результат у системі «Draw & GO» автоматично конвертується у макрос, яким надалі можна користуватися (рисунок 2.5):

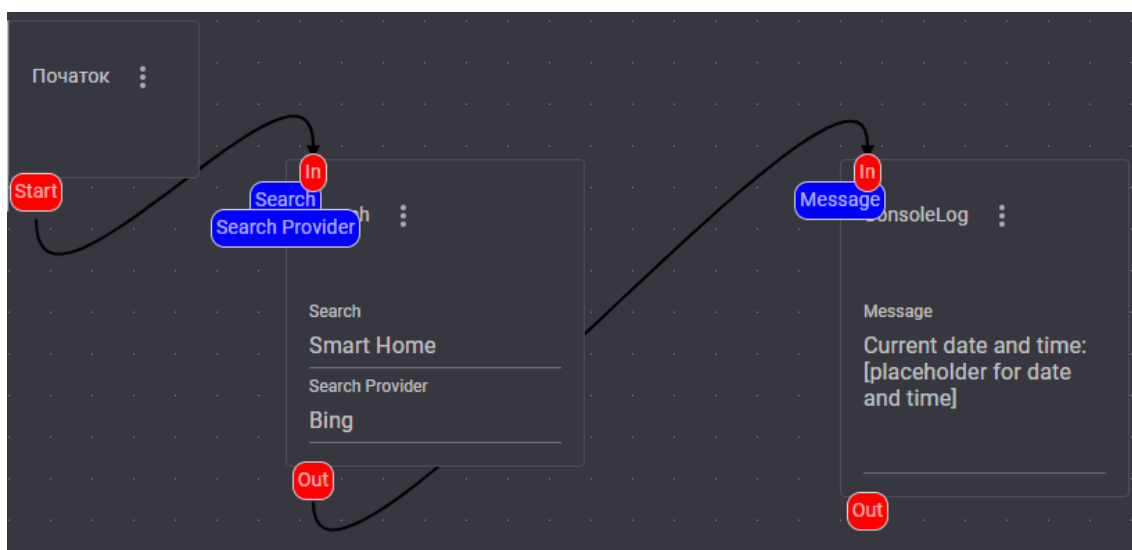


Рисунок 2.5 – Макрос, згенерований штучним інтелектом

2.4 Висновки до розділу 2

У цьому розділі було розглянуто основні алгоритмічні концепції, запропоновані у роботі:

1. Загальний алгоритм організації обчислювального процесу у розробленій системі, включаючи взаємодію між частинами програми, що дозволяє організувати робочі процеси, створення та відтворення макросів, ефективну реалізацію перехоплення подій.
2. Концепцію асоціативного графічного інтерфейсу, що полягає у застосуванні графічних жестів для керування макросами. В тому числі наведено два алгоритми розпізнавання графічних жестів. Порівнявши швидкість та якість розпізнавання було прийнято рішення використання алгоритму інтерполяції точок через його незалежність до намальованого жесту. Цей алгоритм є більш точним та ефективним з точки зору використання ресурсів пристрою. Також було запропоновано модифікацію

алгоритму 2, завдяки чому було досягнуто можливість розпізнавання декількох ліній одночасно для побудови більш складних графічних жестів.

3. Технологію автоматизованого створення макросів. В цьому контексті, застосування мовної моделі штучного інтелекту дає можливість генерувати макроси та розраховувати їх параметри. У роботі запропоновано технологію застосування ШІ для розуміння людської мови і автоматичної побудови макросів. Це дозволяє користувачам будувати робочі процеси, описуючи лише вимоги до них, без необхідності поглиблюватись у знання програмування або навіть самої системи. У додатку «Draw & GO», використання AI дозволяє створювати макроси ввівши лише запит за допомогою голосу або тексту.

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ «DRAW & GO»

У розділі розглянуто питання та особливості розроблення програмного забезпечення системи «Draw & GO».

Підрозділ 3.1 розкриває загальну структуру системи та її компонентів, запропоновану архітектуру та її переваги, надає опис бази даних та детальну інформацію про кожну макрокоманду, розкриває питання безпеки.

Підрозділ 3.2 описує методи організації роботи з макросами – програмним засобом, призначеним для автоматизації дій, які користувачі можуть легко визначати за допомогою візуального інтерфейсу, що є ключовим елементом розробленої системи.

Система «Draw & GO» підтримує плагіни та надає можливості створювати розширення, що додають нові функції до основної програми. Підрозділ 3.3 надає інформацію про розроблення ключової бібліотеки для створення розширень та описує інструкцію для їх розроблення.

Підрозділ 3.4 описує розроблений план впровадження системи, що включає стратегію комерціалізації, механізми технічної підтримки та оновлення, а також методи забезпечення безпеки та конфіденційності даних користувачів, з урахуванням потреб усіх зацікавлених сторін. Також надано аналітичні графіки, що демонструють поточний стан системи.

Висновки за розділом подані у підрозділі 3.5.

3.1 Структура та функціональні можливості

Система «Draw & GO» забезпечує інтеграцію з пристроями користувача для забезпечення зручності користування.

Доступні такі варіанти інтеграції:

1. Web app – вебдодаток, який можна використовувати на будь-якому пристрої з доступом до Інтернету. Цей варіант інтеграції є найзручнішим для користувачів, оскільки не вимагає установки додаткового програмного забезпечення. Він доступний за посиланням <https://drawgo.azurewebsites.net>.

2. Mobile app – мобільний додаток, який можна використовувати на смартфонах і планшетах. Доступний для пристроїв з операційними системами Android, iOS, Tizen. Додаток має такий самий функціонал, як і web app. Цей варіант інтеграції є зручним для користувачів, які хочуть мати доступ до системи «Draw & GO» на ходу.

3. Desktop app – настільний додаток, який можна використовувати на комп'ютерах з операційними системами Windows, macOS та Linux. Цей варіант інтеграції є зручним для користувачів, які хочуть мати доступ до системи «Draw & GO» в автономному режимі.

4. Browser extension – розширення для браузера, яке можна використовувати для швидкого запуску макросів прямо з браузера. Розширення для браузера системи «Draw & GO» доступне для браузерів Google Chrome та Microsoft Edge. Цей варіант інтеграції є зручним для користувачів, які хочуть мати швидкий доступ до системи «Draw & GO».

5. Інтеграція у Telegram – інтеграція з месенджером Telegram, яка дозволяє користувачам керувати макросами за допомогою команд Telegram. Цей варіант інтеграції є зручним для користувачів, які використовують Telegram.

Система «Draw & GO» забезпечує широкий спектр варіантів інтеграції з пристроями користувача. Це дозволяє користувачам вибрати той варіант, який найбільше відповідає їхнім потребам і зручності.

Як було показано у підрозділі 1.4 на даний момент не існує єдиного ефективного архітектурного рішення для систем автоматизації процесів,

кожен з існуючих має певні слабкі сторони. Тому у роботі було розроблено і застосовано нову архітектуру Clean Microkernel Architecture, яка поєднує в собі архітектуру мікроядра та чисту архітектуру.

Архітектура мікроядра дозволяє розробникам створювати та ділитися налаштованими плагінами для розширення можливостей механізму робочого процесу. Плагіни дозволяють користувачам підключати сервіси, автоматизувати завдання та інтегруватися з корпоративними системами. Розробники можуть легко додавати нові функції до системи та гарантувати бездоганну інтеграцію плагінів. Ця гнучкість дозволяє компаніям швидко створювати робочі процеси та вносити зміни, коли це необхідно. Це прискорює потік даних і покращує швидкість автоматизації процесу. Архітектура плагінів «Draw & GO» розроблена для забезпечення масштабованості механізму робочого процесу. Платформа мікроядра працює незалежно, тобто базові функції для всіх компонентів обробляються ядром, що дозволяє легко інтегруватись з іншими системами.

Рисунок 3.1 демонструє загальну архітектуру додатку, показує взаємодію між усіма рівнями системи. На рисунку чітко відстежується чиста архітектура, та можна бачити, що основна система залежить від DrawGo.Plugin.Base бібліотеки, що містить основу для роботи з макросами.

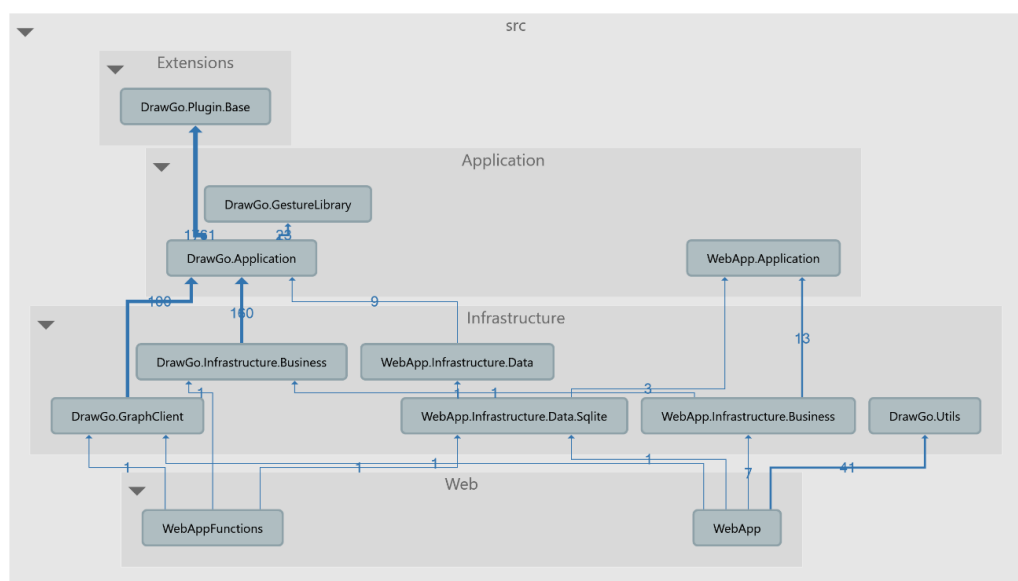


Рисунок 3.1 – «Draw & GO» архітектура рішення

Як можна бачити на діаграмах класів (рисунки 3.2, 3.3), рівні Domain і Services містять однакові проєкти та містять спільний код.

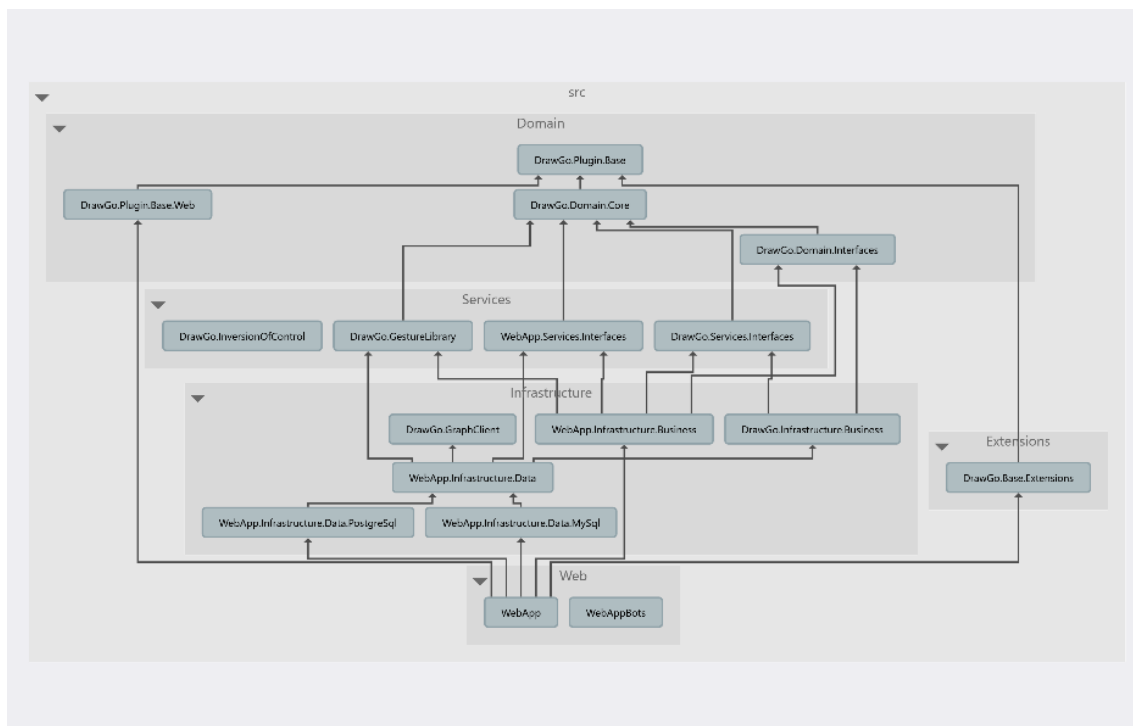


Рисунок 3.2 – Діаграма класів вебдодатку «Draw & GO»

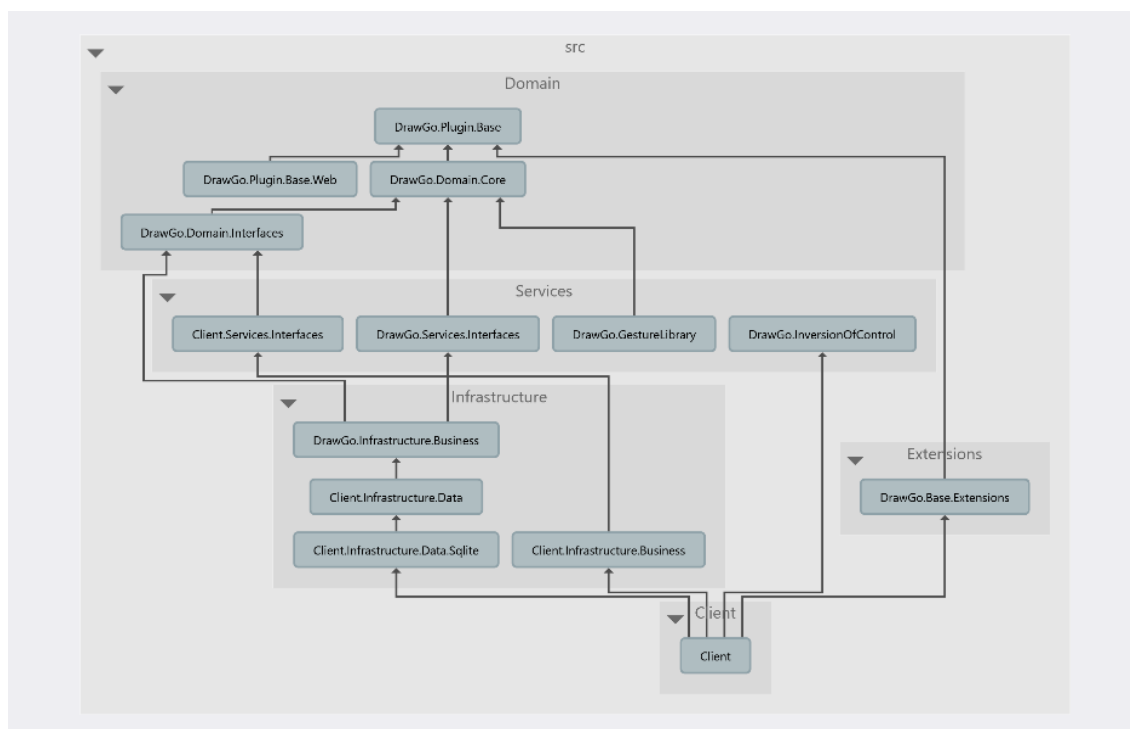


Рисунок 3.3 – Діаграма класів мобільного клієнта «Draw & GO»

Рівень домену містить код для сутностей, перерахувань і загальну логіку для плагінів.

Рівень програми містить інтерфейси, які будуть реалізовані для кожної програми. Він також містить загальну бібліотеку для підключення IoC та аналізу жестів.

У свою чергу, рівень інфраструктури має схожі конструкції, але вміст відрізняється бізнес-логікою та реалізацією бази даних для вебдодатку та клієнтського додатка. Клієнт також містить логіку для зв'язку зі службою шляхом надсилання http-запитів. Ця реалізація не потрібна у вебдодатку.

Нарешті, рівень інтерфейсу користувача, який абсолютно різний для WebApp і Client, оскільки програми працюють на різних платформах і мають різні засоби керування та обов'язки в цілому.

Зниження цикломатичної складності та підвищення індексу підтримуваності – це два поняття, які використовуються в контексті розроблення програмного забезпечення для покращення якості коду та його легкості в обслуговуванні.

Цикломатична складність (Cyclomatic Complexity) – це метрика, яка вимірює кількість лінійних шляхів через програму. Вона використовується для оцінки складності програми, заснована на кількості можливих шляхів виконання програми. Висока цикломатична складність означає більшу кількість шляхів виконання, що може ускладнити розуміння, тестування та підтримку коду. Тому зниження цикломатичної складності полягає у спрощенні структури коду, видаленні зайвих умовних операторів та циклів, рефакторингу та використанні шаблонів проєктування для зменшення кількості шляхів виконання.

Індекс підтримуваності (Maintainability Index) – це метрика, яка використовується для оцінки легкості підтримки програмного коду. Вона враховує такі фактори, як цикломатична складність, кількість рядків коду, згортання коментарів та інші атрибути, які можуть впливати на зусилля,

необхідні для зміни та підтримки програми. Підвищення індексу підтримуваності означає здійснення змін у коді, які роблять його більш читабельним, легшим для розуміння та модифікації, що включає в себе якісну документацію, зменшення залежностей між модулями та покращення структури коду.

Результати цих двох метрик відображені на рисунку 3.4.

Hierarchy	Maintainability Index	Cyclomatic Complexity
src\Extensions\DrawGo.Plugin.Base (Debug)	89	221
src\Application\DrawGo.GestureLibrary (Debug)	88	29
tests\Infrastructure\DrawGo.Infrastructure.Business.Tests (Debug)	87	56
tests\Extensions\DrawGo.Plugin.Base.Tests (Debug)	86	4
src\Application\DrawGo.Application (Debug)	85	1,129
tests\Application\DrawGo.Application.Tests (Debug)	82	10
tests\Application\DrawGo.GestureLibrary.Tests (Debug)	81	4
src\Infrastructure\DrawGo.Infrastructure.Business (Debug)	73	93

Рисунок 3.4 – Метрики спільної частини коду.

Clean Microkernel Architecture дозволила зменшити дублювання коду до 46% та пришвидшити роботу з кодом завдяки винесенню спільної частини у окремий модуль.

Архітектура мікроядра реалізована за допомогою MEF 2.0, що дозволяє розбирати бібліотеку та легко завантажувати необхідні класи (рисунки 3.5).

```
var configuration = new ContainerConfiguration();
var asm = Assembly.Load("Extension1.dll");
configuration.WithAssembly(asm);
var compositionContainer = configuration.CreateContainer();

var allMacroCommands = compositionContainer.GetExports<MacroCommand>();
```

Рисунок 3.5 – Приклад роботи з MEF 2.0. Завантаження всіх макрокоманд із Extension1.dll

Рисунок 3.6 демонструє загальну хмарну архітектуру та всі сервіси, що використовуються для побудови додатку.

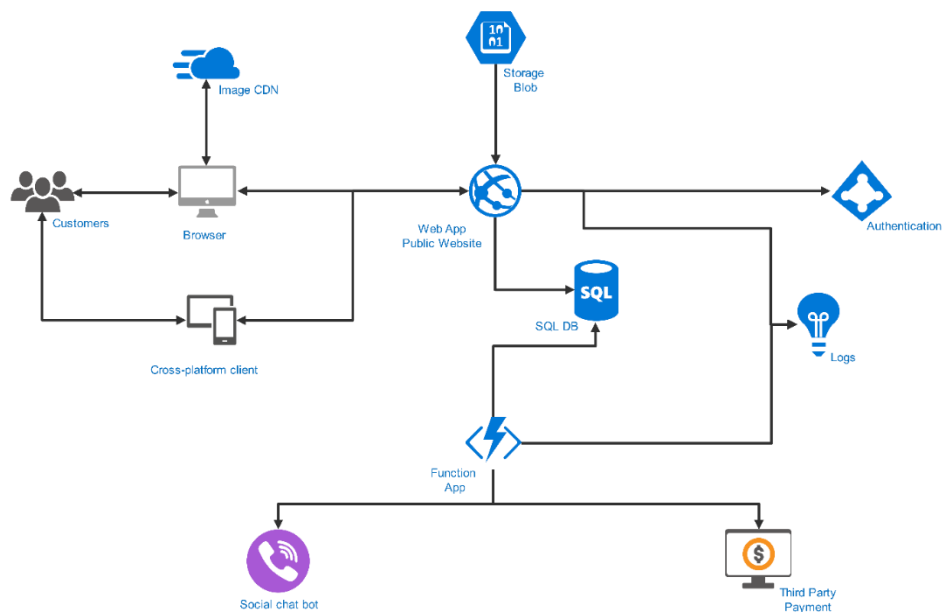


Рисунок 3.6 – «Draw & GO» хмарна архітектура

У «Draw & GO» значну увагу приділено безпеці даних. На сучасному етапі розвитку інформаційного суспільства, коли комп'ютери є чи не в кожній сім'ї, в умовах комерціалізації інтелектуальної власності актуальним є створення програмних продуктів, які дозволяють зручно та безпечно захистити особисті дані користувача при збереженні, передачі, і модифікації. Графічний пароль є зручною формою захисту даних, з іншого боку, різноманітні графічні зображення дозволяють створювати надійні паролі. Крім того, актуальність захисту даних підтверджується чинним законодавством: Концепцією національної системи захисту інтелектуальної власності № 1243-V від 27 червня 2007 року.

Його реалізація безпечно зберігає інформацію в базі даних і надає користувачам всеосяжний журнал аудиту, контроль конфіденційності даних і права доступу. Його комплексні заходи безпеки гарантують безпечне та захищене середовище для ефективного керування бізнес-процесами клієнтів.

Рисунок 3.7 демонструє загальну схему бази даних. Таблиця Users зберігає інформацію про користувачів системи. Таблиця LinkedAccount

дозволяє асоціювати користувача системи з соціальними мережами. Інформація про транзакції зберігається у таблиці Transactions.

Таблиця Components містить інформацію про Макроси та Папки. У таблиці використовується шаблон Composite для побудови ієрархічної структури. Налаштування макросів, макроси та жести зберігаються у таблицях MacroOptions, Graphs та GestureLines відповідно.

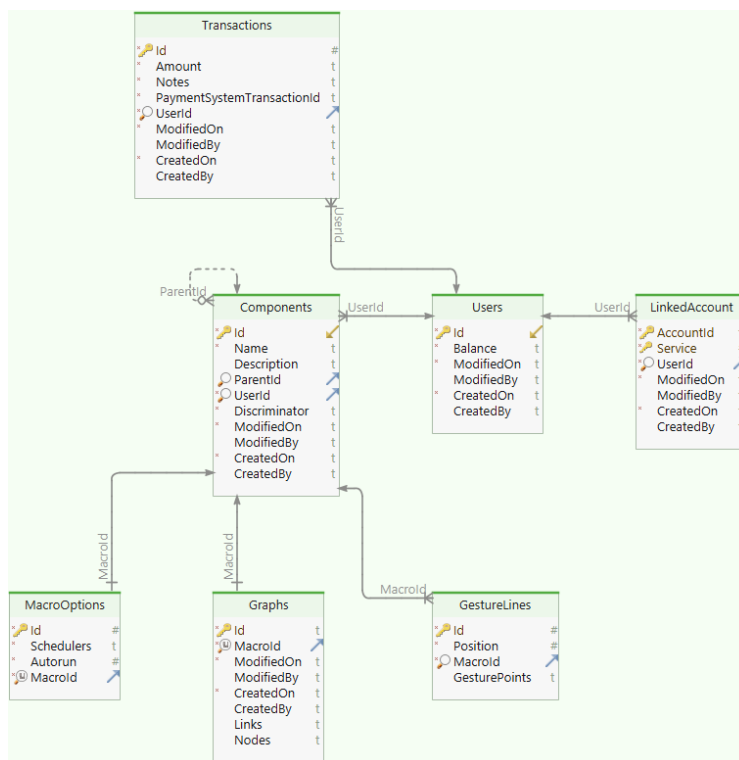


Рисунок 3.7 – Схема бази даних

«Draw & GO» має кілька функцій та переваг, які відрізняють його від інших механізмів робочого процесу, як-от:

1. Розпізнавання жестів: користувачі можуть малювати фігури на своїх екранах для створення процесів, не вводячи текст і не клацаючи.
2. Візуальний дизайнер: користувачі можуть перетягувати плагіни та з'єднувачі, щоб визначити умови, дії, тригери, цикли, розгалуження тощо.
3. Синхронізація в режимі реального часу: користувачі можуть бачити свої зміни миттєво на всіх пристроях.

4. Безпека: користувачі проходять автентифікацію за допомогою SSO від Google або Microsoft без необхідності зберігати паролі у додатку. Обидва провайдери підтримують новітні, найбезпечніші методи автентифікації, включаючи біометричну.

5. Налаштування та розширюваність: користувачі можуть створювати свої плагіни або використовувати наявні з бібліотеки.

6. Простота використання: «Draw & GO» не потребує програмування чи технічних навичок для створення автоматизації процесів.

7. Сумісність між платформами: «Draw & GO» працює на будь-якому пристрої або платформі, яка підтримує технології .NET MAUI або Blazor.

8. Інноваційний інтерфейс: «Draw & GO» є першим механізмом робочого процесу, який використовує розпізнавання жестів як метод введення.

Програма має багатомовний інтерфейс, що поширює коло осіб, які можуть її використовувати. На сьогодні «Draw & GO» доступна двома мовами: англійською та українською. Але в перспективі, за бажанням, можливо розширити список доступних перекладів.

Програма має функцію оновлення, що допоможе користувачеві отримати найостаннішу версію продукту лише за один клік.

Без уваги не залишено й створення інтернет-сторінки, завдяки якій можна продивитися приклади роботи програми, власноруч завантажити останню версію та документацію, дізнатися про рекомендовані параметри комп'ютера, що дозволять безперебійно працювати, та зв'язатися з розробниками у разі необхідності. Кожен користувач має зареєструватися на вебсайті для керування розширеннями та макросами.

«Draw & GO» містить широкий набір макрокоманд (таблиця 3.1), що допомагають автоматизувати та оптимізувати робочий процес, забезпечуючи швидкість та зручність використання.

Таблиця 3.1 – Макрокоманди, реалізовані у «Draw & GO»

Назва	Категорія	Опис
Generate Image	AI	Використовує штучний інтелект для генерації зображень на основі зазначених параметрів
Get Completion	AI	Отримує результат з AI-моделі на основі наданого текстового запиту
Plagiarism checker	AI	Перевіряє текст на наявність плагіату, використовуючи алгоритми штучного інтелекту
Break	Control	Перериває виконання циклу або інструкції
Delay	Control	Здійснює тайм-аут на задану кількість мілісекунд
For	Control	Виконує блок команд певну кількість разів
Sequence	Control	Виконує послідовно набір дій
TryCatchFinally	Control	Обробляє виключення та виконує код у блоках Try, Catch, і Finally
While	Control	Виконує блок команд, поки виконується задана умова

Продовження таблиці 3.1

Назва	Категорія	Опис
Query	Database.MSSQL	Виконує SQL-запит до бази даних Microsoft SQL Server
Select	Database.MSSQL	Виконує SQL-запит вибірки (SELECT) у базі даних Microsoft SQL Server
Query	Database.MySQL	Виконує SQL-запит до бази даних MySQL
Select	Database.MySQL	Виконує SQL-запит вибірки (SELECT) у базі даних MySQL
Query	Database.Postgres	Виконує SQL-запит до бази даних PostgreSQL
Select	Database.Postgres	Виконує SQL-запит вибірки (SELECT) у базі даних PostgreSQL
Show alert	Dialogs	Показує спливаюче вікно з попередженням або інформацією
Send email	ElasticEmail	Відправляє електронний лист за допомогою ElasticEmail сервісу
Create a Facebook post	Facebook	Створює публікацію у Facebook
Download Mega File	FileSystem	Завантажує файл з сервісу Mega

Продовження таблиці 3.1

Назва	Категорія	Опис
List Google Drive Files	FileSystem	Отримує список файлів з Google Drive
List Mega Files	FileSystem	Отримує список файлів з сервісу Mega
GetVariable	Graph	Отримує змінну з контексту виконання
SetVariable	Graph	Встановлює змінну у контексті виконання
Get	HTTP	Виконує HTTP GET-запит
JsonGet	HTTP	Виконує HTTP GET-запит та отримує JSON-відповідь
JsonPost	HTTP	Виконує HTTP POST-запит з JSON-даними
Post	HTTP	Виконує HTTP POST-запит
Application Insights Query	Integrations.Application Insights	Виконує запит до Microsoft Application Insights
SendEmail	Integrations.Mailgun	Відправляє електронний лист за допомогою Mailgun сервісу
SendEmail	Integrations.SendGrid	Відправляє електронний лист за допомогою SendGrid сервісу
SendText	Integrations.Twilio	Відправляє SMS за допомогою Twilio сервісу

Продовження таблиці 3.1

Назва	Категорія	Опис
GetIndex	Lists	Отримує індекс елементу в списку
GetValue	Lists	Отримує значення за індексом зі списку
LastOrDefault	Lists	Отримує останній або типовий елемент із списку
And	Logic	Логічне «І» між декількома умовами
Equals	Logic	Порівняння на еквівалентність
GreaterThan	Logic	Перевірка на більше
GreaterThanOrEqual To	Logic	Перевірка на більше або дорівнює
If	Logic	Виконує код на основі умови (якщо-то)
LessThan	Logic	Перевірка на менше
LessThanOrEqualTo	Logic	Перевірка на менше або дорівнює
Or	Logic	Логічне «АБО» між декількома умовами
ConsoleLog	Logs	Виводить повідомлення в консоль записів
FileLog	Logs	Заносить повідомлення у файл записів

Продовження таблиці 3.1

Назва	Категорія	Опис
Holidays	Lookup	Отримує інформацію про святкові дні
IP Geolocation	Lookup	Визначає географічні координати за IP
Add	Math	Виконує додавання чисел
Cos	Math	Обчислює косинус числа
Divide	Math	Виконує ділення чисел
Exponential	Math	Обчислює показник степені числа
Logarithmic	Math	Обчислює логарифм числа
Multiply	Math	Виконує множення чисел
Round	Math	Заокруглює число до найближчого цілого
Sin	Math	Обчислює синус числа
Subtract	Math	Виконує віднімання чисел
Tan	Math	Обчислює тангенс числа
Kill process	Processes	Завершує процес
List processes	Processes	Виводить список процесів
Start process	Processes	Запускає процес
Get random number	Random	Генерує випадкове число
Get random string	Random	Генерує випадковий рядок символів
DownloadFile	SCP	Завантажує файл за допомогою SCP

Продовження таблиці 3.1

Назва	Категорія	Опис
DownloadFile	SFTP	Завантажує файл за допомогою SFTP
UploadFile	SFTP	Вивантажує файл за допомогою SFTP
ExecuteCommand	SSH	Виконує команду через SSH
Login in Telegram	Telegram	Авторизується в Telegram
Send a message	Telegram	Відправляє повідомлення в Telegram
Delete tweet	Twitter	Видаляє твіт в Twitter
New tweet	Twitter	Створює новий твіт в Twitter
Concat	Utils	Об'єднує два або більше рядків
StringFormat	Utils	Форматує рядок за заданим шаблоном
ToJson	Utils	Конвертує дані у формат JSON
Email validation	Validation	Перевіряє правильність введення електронної адреси
IBAN validation	Validation	Перевіряє дійсність міжнародного банківського номеру рахунку (IBAN)

Продовження таблиці 3.1

Назва	Категорія	Опис
VAT validation	Validation	Перевіряє дійсність введення номеру платника податку на додану вартість (VAT)
Send a message	Viber	Відправляє повідомлення у Viber
Close all browser windows	WebBrowser	Закриває всі вікна браузера, відкриті «Draw & GO»
Close browser window	WebBrowser	Закриває вікно браузера за посиланням
Open Window	WebBrowser	Відкриває вікно браузера та виконує дію
Open Window and get reference	WebBrowser	Відкриває вікно браузера та отримує посилання на нього
Search	WebSearch	Виконує пошук інформації в пошуковій системі (Google, Bing тощо)
Дата	Базові типи	Представляє дату без часу
Дата і час	Базові типи	Представляє дату та час
Логічний тип	Базові типи	Представляє логічну змінну (true/false)
Номер телефону	Базові типи	Зберігає та обробляє номери телефонів

Назва	Категорія	Опис
Список	Базові типи	Дозволяє створювати та обробляти списки елементів
Текст	Базові типи	Представляє рядок тексту
Час	Базові типи	Представляє час без дати
Число	Базові типи	Представляє числові значення
Початок	Точка входу макросу	Позначає початок виконання макросу

3.2 Операції з макросами

Макрос у системі «Draw & GO» – програмний засіб, призначений для автоматизації дій. Вікно створення макросу представлено на рисунку 3.8, де користувачі можуть легко визначати автоматизовані дії за допомогою візуального інтерфейсу. Ця функція дозволяє зібрати серію команд у формі макросу, що може бути згодом викликаний одним натисканням кнопки або певною подією у системі. Інтерфейс включає інтуїтивне розміщення інструментів та полів, в які користувач вводить команди чи задає параметри, такі як клавіші доступу, тригери та умови запуску. Вікно створення макросу є ключовим для оптимізації рутинних завдань і спрощує автоматизацію процесів без потреби в написанні коду.

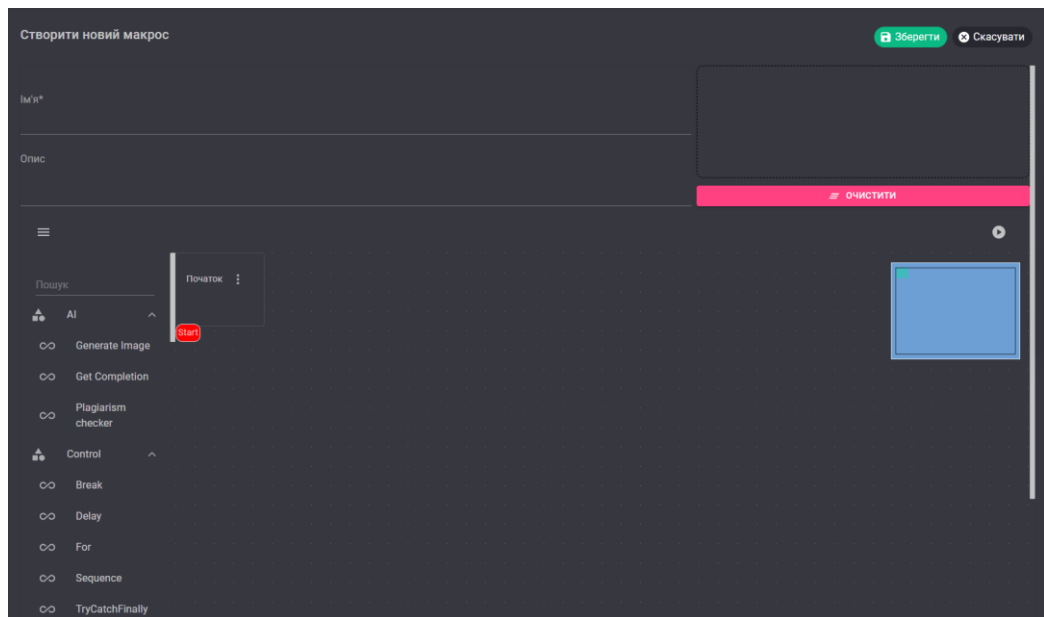


Рисунок 3.8 – Вікно створення макросу

Потрапивши у вікно створення макросу, необхідно ввести унікальне ім'я для макросу. Наступний крок полягає у створенні дій для макросу. У лівій частині можна обрати необхідну команду, натиснувши на неї. Команда буде додана до списку дій. Поєднавши порти обох команд, організується послідовність дій (рисунок 3.9). Початкова команда для запуску макросу має назву «Початок».

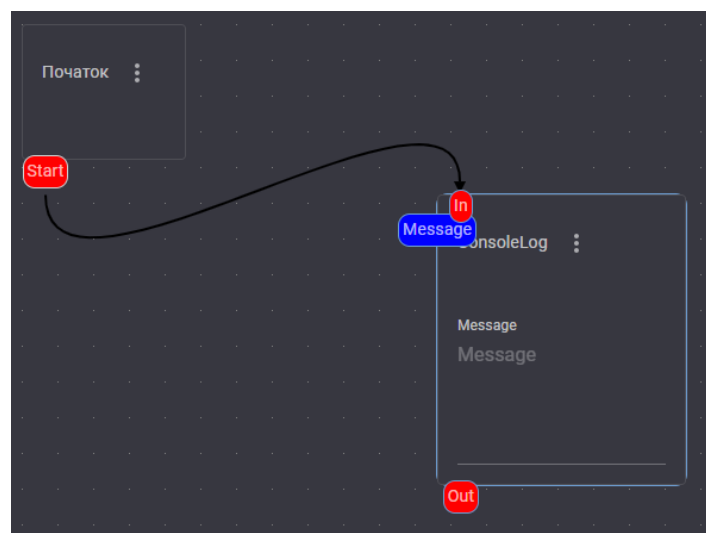


Рисунок 3.9 – Поєднання макросів

Після завершення побудови послідовності дій користувач може асоціювати макрос з відповідним графічним жестом. Користувач може намалювати будь-яку кількість ліній.

Так, для прикладу, може виглядати макрос для пошуку погоди (рисунок 3.10):

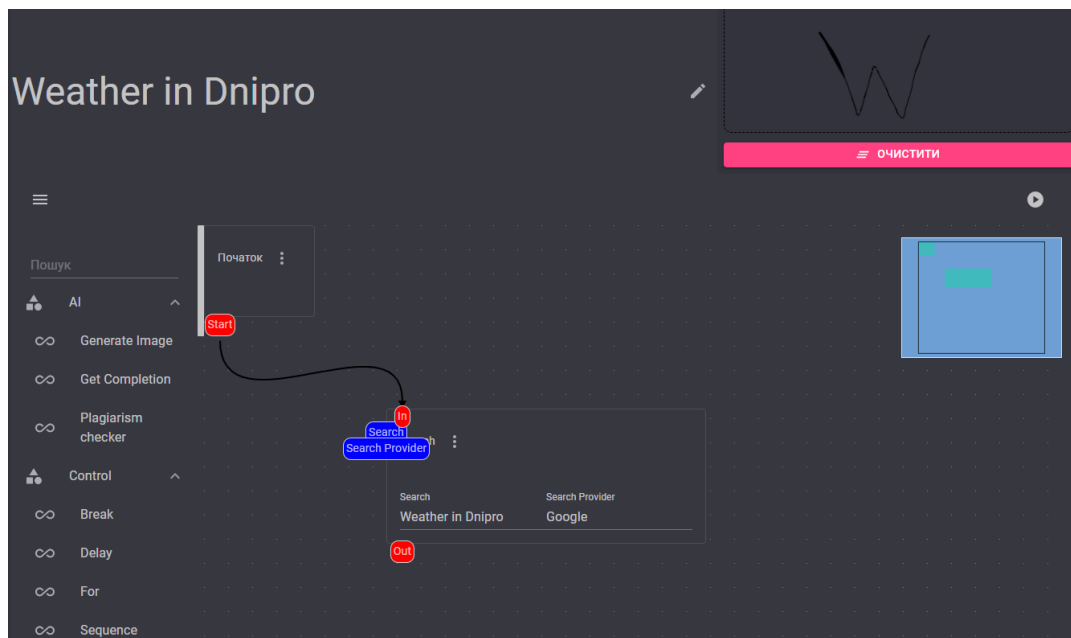


Рисунок 3.10 – Макрос для отримання інформації про погоду

Жест «W» легко асоціювати зі словом «Weather». Для збереження макросу необхідно натиснути кнопку «Зберегти».

Після збереження макросу, він з'явиться у списку (рисунок 3.11).

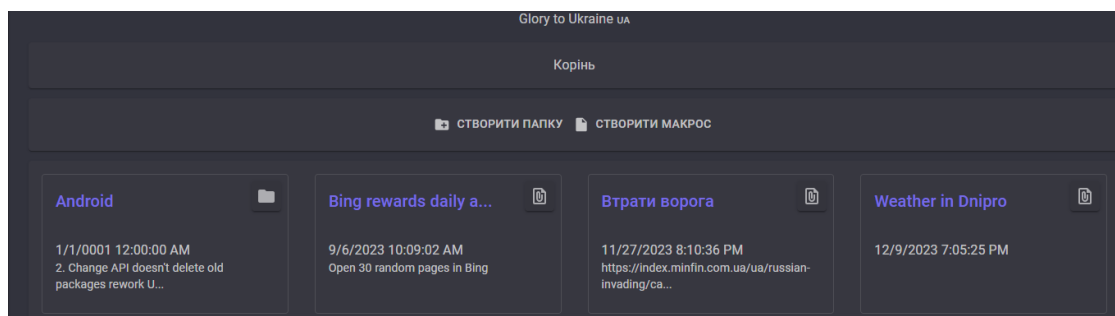


Рисунок 3.11 – Список папок і макросів

Для запуску макросу необхідно натиснути кнопку «Розпізнати макрос».

Намалювавши певний жест, відбувається пошук серед усіх макросів відповідного жесту і виконується запуск. На рисунку 3.12 зображений процес малювання графічного жесту у користувацькому інтерфейсі системи «Draw & GO». Користувачі взаємодіють з програмою, створюючи певні взірці рухів, що після розпізнавання системою виступають як команди для виконання заданих дій або активації макросів. Ця функціональність дозволяє виробляти більш натуральне та ефективне керування програмами через графічний інтерфейс, призначений для поліпшення користувацького досвіду та забезпечення швидкого доступу до часто використовуваних функцій. Малювання графічного жесту є інноваційним рішенням, яке відкриває нові можливості для інтерактивної автоматизації і спрощує процеси користувацького вводу для різноманітних операційних систем і пристроїв.

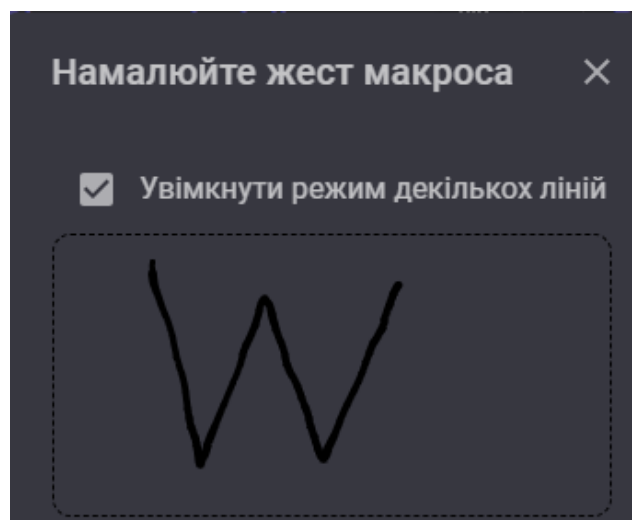


Рисунок 3.12 – Приклад малювання графічного жесту

При запуску макросу буде виконано пошук у Google по запиту «Weather in Dnipro».

У відповідному вікні відображається інформація про результати запуску макросу (рис 3.13).

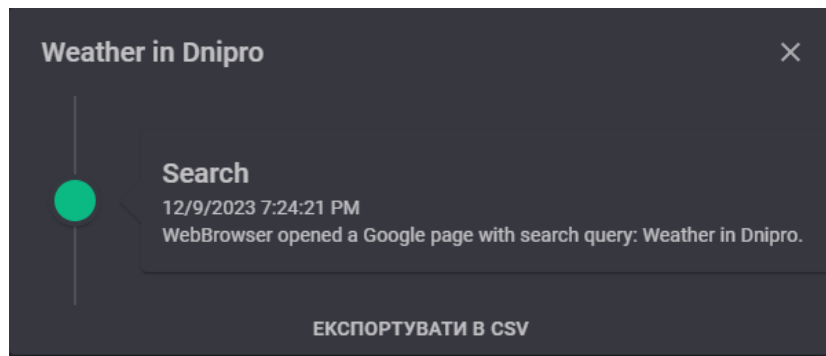


Рисунок 3.13 – Результат виконання макросу

Для збереження результатів користувач має натиснути кнопку «Експортувати в CSV». В результаті буде збережено CSV файл.

Після створення макросу з ним можна виконувати набір дій. Рисунок 3.14 демонструє набір дій, які можуть бути інтегровані в макроси за допомогою системи «Draw & GO».

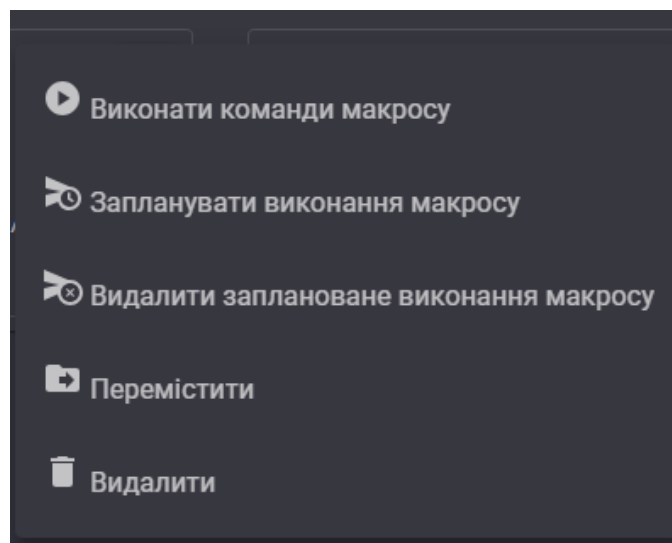


Рисунок 3.14 – Набір дій з макросами

Користувачі можуть вибрати з безлічі попередньо визначених опцій або створювати власні налаштовані дії, що відображають специфічні потреби робочого процесу. Функція набору дій є зручним інструментом для ефективного створення складних автоматизованих зв'язків та швидкого виконання багаторазових задач без необхідності вручну повторювати однакові кроки.

Кожен макрос можна додатково налаштувати для запланованого виконання. Підтримуються наступні графіки періодичності: щоденно, щотижня, щомісяця, щороку, ніколи, власний.

На рисунку 3.15 представлені варіанти планування виконання макросу в системі «Draw & GO», що дозволяють користувачам задавати графік активації різноманітних макросів. У вікні планування можна встановити одноразове виконання макросу в певний момент часу, або налаштувати періодичність, що забезпечить автоматизацію задач з визначеною регулярністю. Користувачі мають можливість вибору конкретних днів тижня чи часу доби для активації макросів, що робить планування гнучким і відповідним під робочі вимоги. Такі функції планування забезпечують ретельне управління процесом і допомагають забезпечити його безперебійність і оперативність.

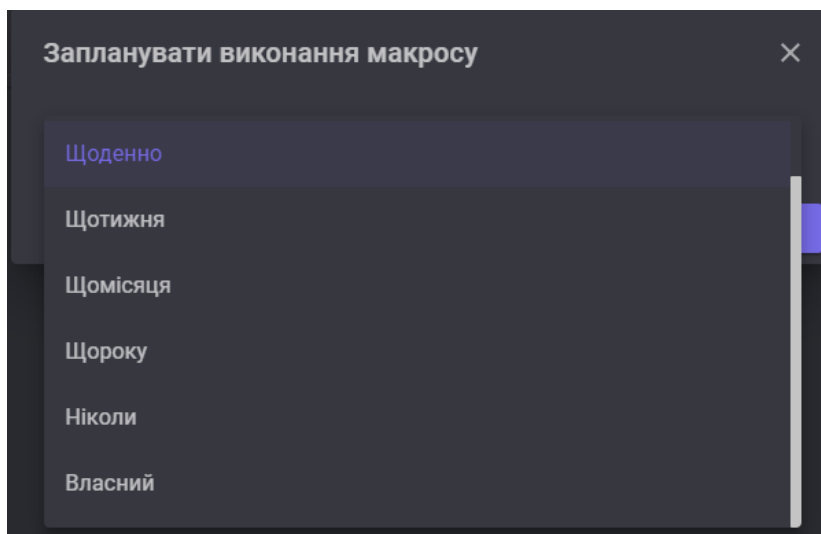


Рисунок 3.15 – Варіанти планування виконання макросу

Власний графік періодичності дозволяє запустити макрос, вписавши правильну CRON-команду у поле. До прикладу команда «15 14 1 * *» запускатиме макрос о 14:15 першого числа кожного місяця.

3.3 Розроблення бібліотеки розширень для «Draw & GO»

«Draw & GO» – це система, що підтримує плагіни та розширення. Розробники можуть створювати розширення, що додають нові функції або можливості до основної програми.

Для «Draw & GO» розроблено ключову бібліотеку для створення розширень `DrawGo.Plugin.Base`, що містить абстрактний клас `MacroCommand`, який є основою для усіх команд розширень [96].

Розширення можна створювати для різних платформ, включаючи Android, MacCatalyst, Windows, Tizen, та Web. Однак через обмеження iOS, створення розширень для цієї платформи може бути не потрібне, оскільки «Draw & GO» для iOS використовує вбудовані нединамічні розширення.

Послідовність розроблення розширення складається з наступних кроків:

1. Створення проєкту: розробник створює проєкт бібліотеки класів (.NET) з назвою, яка закінчується на «Extensions» і вибирає цільовий фреймворк, згідно з підтримкою платформи «Draw & GO».
2. Додавання пакету NuGet: розробник встановлює пакет NuGet `DrawGo.Plugin.Base`, який містить базові класи та інтерфейси для створення команд розширень.
3. Створення класу команди: розробник створює клас у своєму проєкті, який наслідує від `MacroCommand`, заповнюючи його специфічною логікою, необхідною для його розширення.
4. Використання `ExportAttribute` для класу команди: у кожному створеному класі команди розробник додає атрибут `ExportAttribute` з `DrawGo.Plugin.Base.Attributes`, вказуючи «Draw & GO» врахувати його як розширення.

5. Заміна абстрактних властивостей: клас команди, створений розробником, замінює абстрактні властивості, надаючи деталі та реалізацію, необхідні для інтеграції з «Draw & GO».

6. Компіляція та упаковка розширення: Розробник компілює розширення та упаковує його у фабричний NuGet-пакет, переконуючись, що всі залежності правильно включені.

7. Локальне тестування: перед публікацією розробник тестує розширення в локальному середовищі, забезпечуючи його правильну роботу та відсутність конфліктів.

8. Документація розширення: розробник пише докладну документацію для розширення, яка включає описи, параметри, приклади використання та іншу корисну інформацію.

9. Публікація та поширення: розробник завантажує NuGet-пакет розширення на вебсайт, забезпечуючи доступність цього розширення для інших розробників та кінцевих користувачів.

10. Підтримка та оновлення: розробник повинен забезпечувати підтримку свого розширення, реагуючи на зворотній зв'язок від користувачів та виправляючи помилки. Також це передбачає оновлення розширення відповідно до нових версій «Draw & GO» або зовнішніх залежностей.

11. Моніторинг використання: розробник може включити запис даних та аналітику до свого розширення, щоб отримати інформацію про те, як і якою мірою його розширення використовується. Це може допомогти у поліпшенні продукту.

12. Забезпечення відповідності ліцензії: перед публікацією свого розширення розробник повинен вибрати ліцензійну угоду, що регулює умови використання, модифікації та поширення розширення. Це допоможе вберегти інтелектуальні права розробника та уточнити правила для кінцевого користувача.

13. Дотримання стандартів безпеки: оскільки розширення можуть взаємодіяти з даними користувача, розробник повинен забезпечити їх безпеку відповідно до найкращих практик та стандартів індустрії.

14. Співпраця зі спільнотою: розробник може взаємодіяти зі спільнотою «Draw & GO», публікуючи розширення на форумах, у блогах або на конференціях, а також використовуючи відкритий код для сприяння спільній роботі.

Весь код розширень має дотримуватися стандартів безпеки та якості коду, також повинен бути оптимізований для кращої продуктивності та сумісності з «Draw & GO».

3.4 Впровадження системи «Draw & GO»

Одним з основних етапів створення системи «Draw & GO» було розроблення плану впровадження системи, що включає стратегію комерціалізації, механізми технічної підтримки та оновлення, а також методи забезпечення безпеки та конфіденційності даних користувачів, з урахуванням потреб усіх зацікавлених сторін.

Стратегія комерціалізації включає:

1. Маркетинг і просування. Стратегія містить наступні кроки:
 - a. цільова аудиторія: викладачі, проєктні менеджери;
 - b. канали просування: соціальні мережі (LinkedIn), освітні форуми, конференції, SEO та контекстна реклама;
 - c. цінова політика: безкоштовна версія з можливістю придбання платних плагінів з розширеними можливостями.
2. Партнерства:
 - a. освітні заклади: пропозиція спеціальних умов для університетів та шкіл;

- b. бізнеси: співпраця з компаніями для інтеграції в їх робочі процеси.

3. Монетизація:

- a. платні підписки на плагіни: користувач орендує ті плагіни, які потрібні для його задач;
- b. реклама: обмежена реклама для користувачів;
- c. інтеграції: платні інтеграції з іншими популярними сервісами.

Механізми технічної підтримки та оновлення включають:

1. Технічна підтримка:

- a. цілодобова підтримка: чат-боти для миттєвих відповідей та система тікетів для складніших питань;
- b. база знань: документація, відео-уроки, FAQ на сайті;
- c. форуми та спільнота: форум користувачів для обміну досвідом та допомоги.

2. Оновлення:

- a. регулярні оновлення: місячні релізи з виправленнями багів та новими функціями;
- b. зворотний зв'язок: впровадження функцій на основі відгуків користувачів;
- c. бета-тестування: залучення користувачів для тестування нових функцій перед релізом.

Методи забезпечення безпеки та конфіденційності даних користувачів включають:

1. Безпека:

- a. шифрування даних: використання SSL/TLS для захисту даних під час передачі;
- b. регулярні аудити: проведення внутрішніх і зовнішніх аудитів безпеки.

2. Конфіденційність:

- а. політика конфіденційності: чітка та прозора політика щодо зберігання та використання даних;
- б. користувацькі налаштування: можливість для користувачів налаштовувати рівень доступу до своїх даних;
- с. видалення даних: забезпечення права користувачів на повне видалення своїх даних з системи.

Урахування потреб зацікавлених сторін включають:

- 1. користувачі:
 - а. регулярні опитування та зворотний зв'язок;
 - б. впровадження користувацьких запитів у дорожню карту продукту.
- 2. інвестори:
 - а. прозора фінансова звітність;
 - а. регулярні оновлення про досягнення та майбутні плани.

Розробка та впровадження цих стратегій забезпечить успішний запуск та стабільний розвиток системи Draw & GO.

Draw & GO швидко завойовує популярність серед користувачів різного рівня. За перший місяць кількість зареєстрованих користувачів досягла 70 осіб, причому спостерігається стабільний ріст нових реєстрацій на рівні 10% щотижня.

Рисунок 3.16 демонструє статистику використання системи. Щодня система обробляє в середньому 150 запитів.

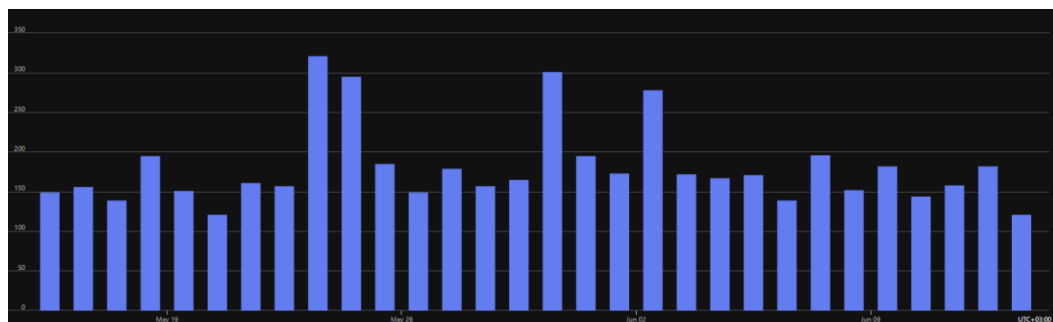


Рисунок 3.16 – Кількість запитів на сервер

На рисунку 3.17 відображено час відповіді на запити. Середній час відповіді на запит складає 20 мс, що забезпечує швидке та безперебійне користування.



Рисунок 3.17 – Час відповіді на запити

Аналіз помилок показав, що найчастішими є:

1. Помилка 404 (Not Found), що виникає, коли користувачі намагаються отримати доступ до неіснуючих ресурсів.
2. Помилка 500 (Internal Server Error), що пов'язана з внутрішніми проблемами сервера.

Часовий розподіл помилок вказує на їх пікову кількість в години пік – з 02:00 до 05:00, коли найбільша кількість користувачів активна (рисунок 3.18).

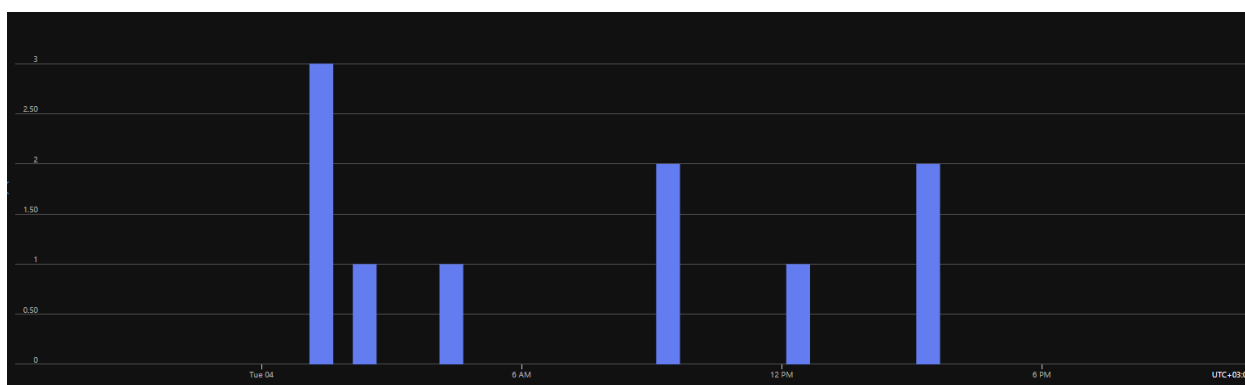


Рисунок 3.18 – Найчастіші помилки та час їх виникнення

З огляду на поточну динаміку, Draw & GO має високий потенціал для подальшого зростання. Очікується, що до кінця року кількість користувачів може перевищити 1000, а щоденна кількість запитів зросте до 3500.

Проте, аби витримати таке навантаження можуть знадобитися зміни в архітектурі. Для забезпечення стабільності та масштабованості сервісу рекомендується:

- Оптимізація серверної частини: впровадження балансування навантаження та масштабування ресурсів.
- Моніторинг та логування: покращення системи моніторингу для швидкого виявлення та усунення проблем.
- Резервування даних: забезпечення надійного резервного копіювання даних користувачів.

Draw & GO демонструє значний потенціал, проте для досягнення максимальних результатів необхідно продовжувати роботу над оптимізацією та вдосконаленням системи. Завдяки цим заходам, платформа зможе задовольнити потреби зростаючої аудиторії та залишатися лідером у своїй галузі.

3.5 Висновки до розділу 3

У третьому розділі було розглянуто особливості та основні концепції розроблення програмного забезпечення системи «Draw & GO». Система надає інтуїтивний графічний інтерфейс, що включає перетягування компонентів для створення індивідуалізованих блок-схем процесів, зменшуючи навантаження на користувача та забезпечуючи гнучкість у вирішенні конкретних завдань.

Реалізовано систему макросів та макрокоманд. Макроси, які вводяться через програмний засіб для автоматизації повторюваних дій, стали

ключовими елементами у розробленій системі «Draw & GO». Ця концепція дозволяє користувачам ефективно маніпулювати процесами через візуальний інтерфейс, підвищуючи продуктивність роботи.

Запропоновано нову архітектуру для програмних систем автоматизації процесів Clean Microkernel Architecture, яка поєднує в собі архітектуру мікроядра та чисту архітектуру. Архітектура мікроядра дозволяє динамічно підключати власні розширення, розширюючи список доступних макросів, які можна запускати. Завдяки чистій архітектурі вдалося видалити дублюючий код моделей та інтерфейсів, реалізувавши їх для кожної платформи.

У системі «Draw & GO» реалізовано підтримку плагінів, розроблено бібліотеку для створення розширень, що вже містить широкий набір макрокоманд для автоматизації та оптимізації робочого процесу, забезпечуючи швидкість та зручність використання, а також надає можливості для користувачів створювати власні розширення, що додають нові функції до основної програми.

Запропоновано план впровадження системи, що включає стратегію комерціалізації, механізми технічної підтримки та оновлення, а також методи забезпечення безпеки та конфіденційності даних користувачів, з урахуванням потреб усіх зацікавлених сторін. Також надано аналітичні графіки, що демонструють поточний стан системи.

РОЗДІЛ 4. ПРАКТИЧНЕ ЗАСТОСУВАННЯ СИСТЕМИ «DRAW & GO»

Система «Draw & GO» має широкий спектр практичного застосування для ефективної автоматизації процесів та взаємодії з комп'ютеризованими пристроями в найрізноманітніших задачах і предметних галузях. Підрозділ 4.1 надає детальну інструкцію по використанню системи та її можливостей, описуючи покрокові дії. Підрозділи 4.2–4.6 описують різні приклади застосування системи, зокрема підрозділ 4.2 описує процес інтеграції з комп'ютеризованими пристроями на прикладі управління роботами-пилососами, у підрозділ 4.3 розглянута задача імітації трафіку на сайт, підрозділ 4.4 ілюструє використання у навчальному процесі, а підрозділ 4.5 – автоматизацію повсякденних дій, у підрозділ 4.6 наведено інші приклади застосування системи. Висновки за розділом подані у підрозділі 4.7.

4.1 Посібник користувача

У додатку «Draw & GO» інтерфейс було створено так, щоб максимально спростити дії користувача і позбавити його від зайвої роботи. «Draw & GO» надає інтуїтивно зрозумілий, єдиний для всіх платформ графічний інтерфейс користувача на основі перетягування та скидання, які можуть використовувати платформу для створення складних систем робочого процесу. Користувачі можуть створювати та налаштовувати макроси за допомогою компонентів-макрокоманд, а платформа дозволяє їм створювати індивідуальні програми, які відповідають конкретним потребам бізнесу.

Як демонструє рисунок 4.1, інтерфейс додатку «Draw & GO» був спроектований з акцентом на мінімалізм і при цьому включає всі необхідні функції для забезпечення легкого й швидкого доступу до інструментів

автоматизації. Центральним елементом є інтуїтивна панель керування, що дозволяє користувачам здійснювати дії простим перетягуванням і відпусканням елементів у візуальному редакторі. Функціональність інтерфейсу зорієнтована на практичність та ефективність, дозволяючи користувачам легко створювати та налаштовувати комплексні робочі процеси через вбудовування макросів і компонентів без необхідності глибокого досвіду в програмуванні. В такий спосіб, «Draw & GO» служить потужним інструментом, що робить робочі процеси більш еластичними і відповідними до вимог бізнесу.

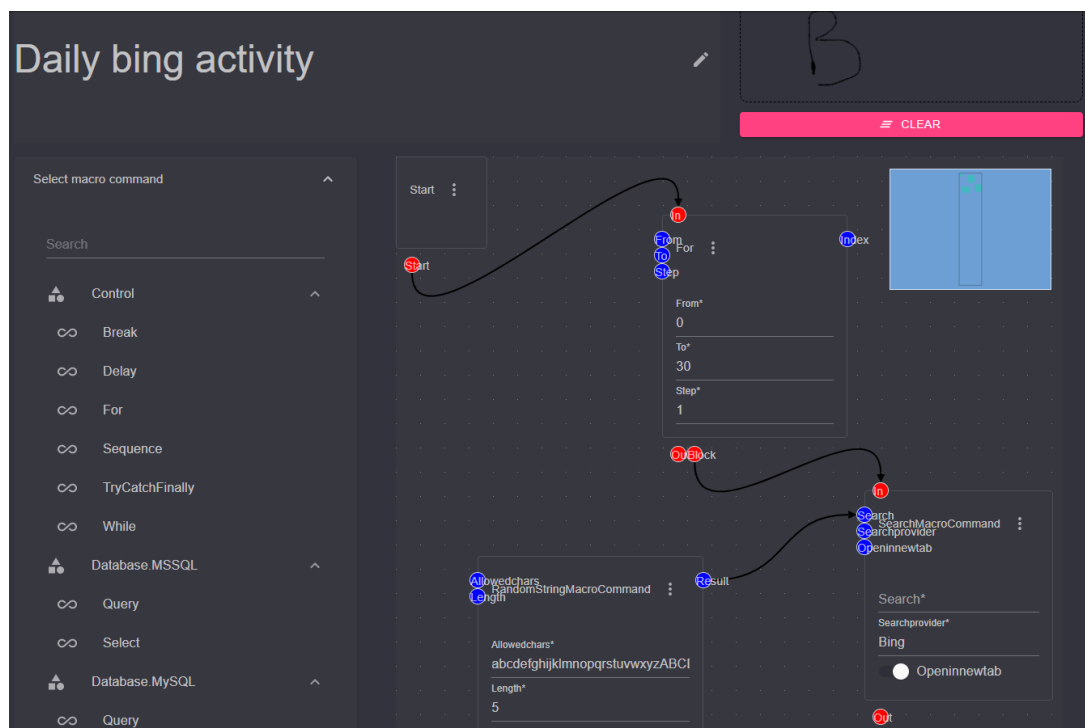


Рисунок 4.1 – «Draw & GO» Macro Editor

Робота користувача починається з реєстрації у додатку. «Draw & GO» дозволяє зареєструватися, використовуючи Google або Microsoft (рис 4.2).

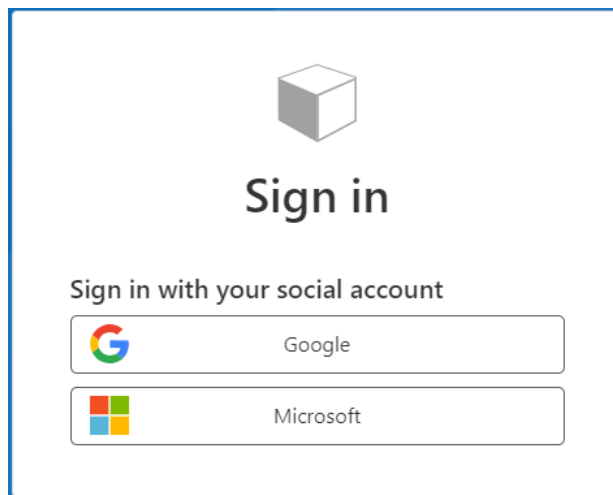


Рисунок 4.2 – Вибір OAuth провайдера для авторизації

Після успішного створення аккаунту, користувач переадресовується у менеджер макросів.

Менеджер – помічник у керуванні кодами та макросами. Менеджер організовано схожим чином до файлової системи. Макроси можна організувати, використовуючи папки.

Для створення папки потрібно натиснути кнопку «Створити папку». У спливаючому вікні ввести ім'я папки та натиснути кнопку «ОК» (рисунок 4.3). Для створення макросу потрібно натиснути кнопки «Створити макрос».

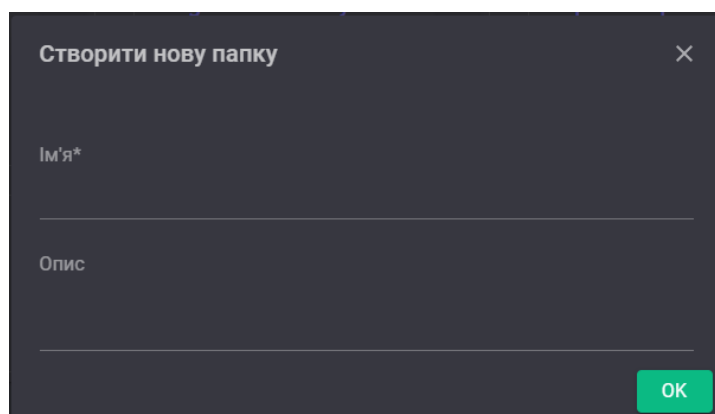


Рисунок 4.3 – Створення папки для організації макросів

Кожен користувач має власний профіль, де можна побачити роль користувача – User, Developer, Administrator. Роль User видається усім

користувачам автоматично. Роль Developer видається користувачам, що бажають створювати команди для «Draw & GO». Роль Administrator має лише власник продукту.

Також у профілі можна керувати доступністю. При включеному режимі доступності «Draw & GO» можуть використовувати люди з обмеженими властивостями.

Профіль дозволяє керувати мовами інтерфейсу. Доступні англійська та українська. Рисунок 4.4 візуалізує інтерфейс профілю користувача в системі «Draw & GO», куди входить персональна інформація та налаштування. Ця область системи дозволяє користувачам керувати своїми особистими даними, налаштовувати параметри безпеки і вибирати варіанти налаштування робочого середовища. Централізоване місце для керування персональними налаштуваннями спрощує управління користувацькими ресурсами і дозволяє забезпечити більш ефективну і безпечну взаємодію з автоматизованими системами.

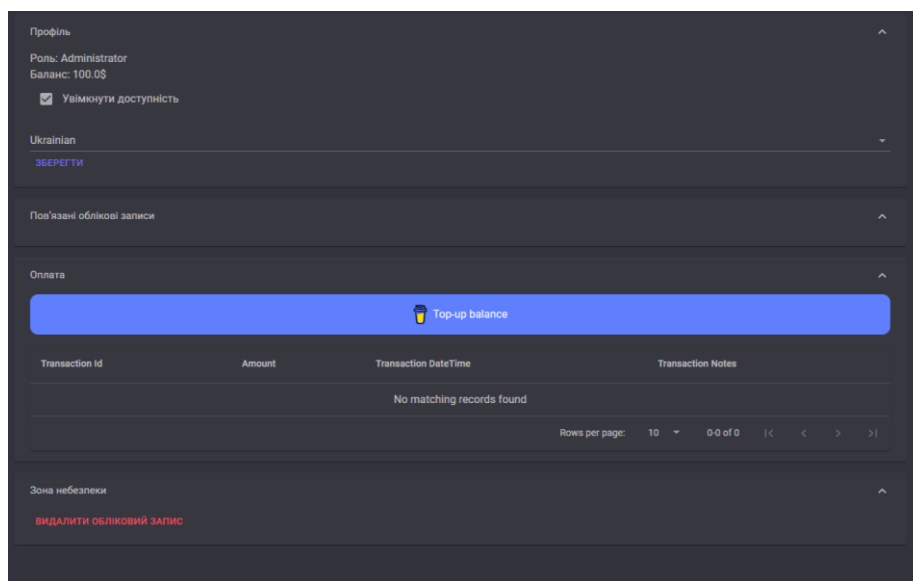


Рисунок 4.4 – Профіль користувача

Передбачена можливість інтеграції з обліковими записами. Наразі підтримується Telegram. Користувач може керувати макросами та розширеннями з месенджера Telegram.

Рисунок 4.5 ілюструє інтерфейс інтеграції системи «Draw & GO» з месенджером Telegram. Ця функція дозволяє користувачам здійснювати віддалене керування макросами та взаємодіяти з процесами безпосередньо через бота в Telegram. Установка тригерів для повідомлень, автоматичне виконання задач при отриманні команд через чат, а також відправлення повідомлень з результатами виконання процесів – все це можна легко налаштувати в інтерфейсі інтеграції. Завдяки цій особливості, «Draw & GO» відкриває додаткові можливості для мобільної взаємодії та посилює комунікаційні здібності програми, роблячи автоматизацію доступною і зручною навіть на ходу.

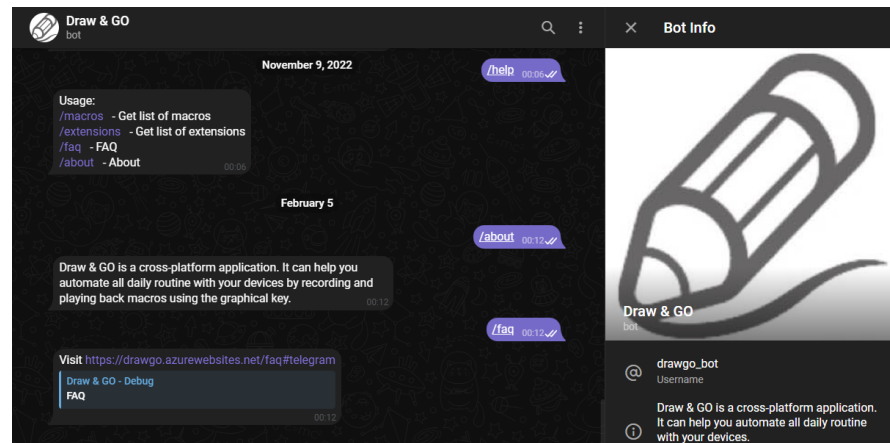


Рисунок 4.5 – Інтеграція з Телеграм

У профілі користувач може побачити список транзакцій та поповнити рахунок.

За вимогою GDPR про права людини кожен користувач має право на повне видалення акаунту. Це включає повне видалення усіх папок та макросів користувача, історії запусків, налаштувань макросів за розкладом, профілю користувача. Повне видалення користувача з системи виконується протягом 30 днів.

Сторінка адміністратора. Адміністратор має повний доступ до системи та може бачити та керувати списком користувачів, налаштовувати банери на

сайті, редагувати Sitemap, робити backup/restore для бази даних, чистити кеш, тощо.

На рисунку 4.6 показано інтерфейс адміністрації банерів в системі «Draw & GO», який надає адміністраторам повний контроль над розміщенням і управлінням рекламних та інформаційних банерів. В цьому розділі можна створювати, редагувати, планувати покази та видаляти банери, а також аналізувати статистику їх переглядів і взаємодії з користувачами. Наявність детальних фільтрів і налаштувань дозволяє адміністраторам ефективно сегментувати та націлювати вміст, тим самим забезпечуючи відповідність рекламних повідомлень встановленим цілям і критеріям. Інструмент адміністрації банерів стає важливою складовою системи з точки зору маркетингової комунікації та управління користувацьким досвідом.

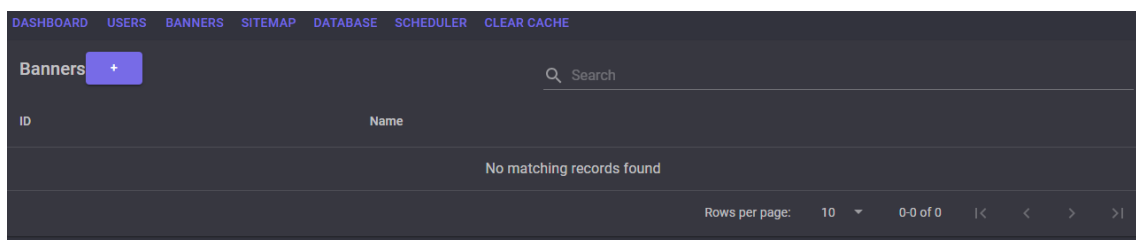


Рисунок 4.6 – Адміністрація банерів

4.2 Інтеграція з комп'ютеризованими пристроями

Система «Draw & GO» окрім роботи з пристроєм на якому вона запущена може інтегруватися і керувати іншими пристроями. Для цього існує декілька розширень. Розглянемо на прикладі управління роботами-пилососами. Плагін дозволяє користувачам створювати плани прибирання, керувати роботами-пилососами дистанційно та отримувати інформацію про їхній стан.

Плагін для підключення до робота-пилососа – це програмна компонента, яка дозволяє системі «Draw & GO» взаємодіяти з конкретним

типом робота-пилососа. Плагін забезпечує доступ до таких функцій робота-пилососа, як керування рухом, керування режимами прибирання, отримання інформації про стан робота-пилососа.

Плагін для підключення до робота-пилососа складається з двох основних компонентів:

1. клієнтський компонент – це компонент, який виконується на сервері системи «Draw & GO». Він відповідає за обробку запитів від користувачів і передачу команд роботу-пилососу.

2. протокол – це стандарт, який використовується для обміну інформацією між клієнтським компонентом і роботом-пилососом.

Клієнтський компонент плагіна реалізований на мові C#. Він використовує протокол HTTP для обміну інформацією з роботом-пилососом.

Для впровадження плагіна для підключення до робота-пилососа необхідно вказати IP адресу пристрою та команду, яку необхідно виконати.

У майбутньому плагін для підключення до робота-пилососа може бути розширений для підтримки наступних можливостей: автоматичне виявлення роботів-пилососів та підтримка декількох роботів-пилососів.

Ці можливості зроблять систему «Draw & GO» ще більш зручним і функціональним інструментом для управління комп'ютеризованими пристроями.

4.3 Імітація трафіку на сайт

Даний макрос виконує відкриття 30 сторінок з випадковим текстом на сайті Bing.com. Результат відкриття кожної сторінки записується. Після відкриття сторінок відбувається очікування 5000 мс і закриття всіх відкритих сторінок.

На рисунку 4.7 представлено макрос імітації трафіку створений в системі «Draw & GO», який демонструє автоматизований процес генерації штучного користувацького трафіку для тестування інфраструктури або аплікацій. Цей макрос налаштовується для ініціювання запитів або дій користувачів в спосіб, котрий імітує реальний користувацький трафік, дозволяючи розробникам і тестувальникам оцінювати поведінку системи під навантаженням. Функціонал включає можливість встановлення параметрів запитів, тривалість сесії, проміжки між діями та кількості користувацьких сесій, що мають бути симульовані. Цей інструмент є незамінним для верифікації масштабованості та стабільності сервісів перед їх запуском у виробництво.

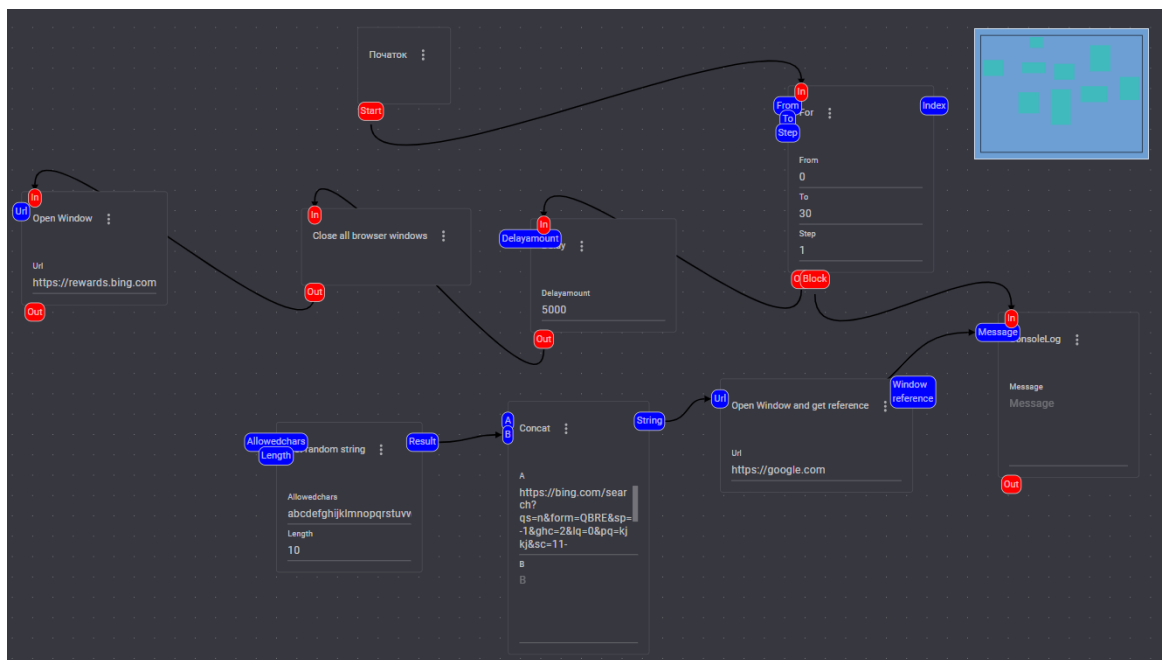


Рисунок 4.7 – Макрос імітації трафіку

Даний приклад показує можливості програми для виконання навантажувального тестування для сайту, що може бути корисно QA-інженерам.

4.4 Використання у навчальному процесі

«Draw & GO» може широко використовуватись у навчальному процесі, а саме:

1. Інтерактивні лекції та семінари: використання «Draw & GO» у лекціях та семінарах дозволяє викладачам створювати інтерактивні заняття. За допомогою жестів викладачі можуть керувати презентаціями, малювати схеми та графіки у реальному часі, що значно покращує візуалізацію матеріалу та залучення студентів.
2. Спільна робота: інструмент підтримує функції спільної роботи, що дозволяє студентам працювати над груповими проектами, обмінюватися ідеями та спільно розв'язувати задачі. Це сприяє розвитку навичок командної роботи та комунікації, що є важливими у сучасному освітньому середовищі.
3. Інтерактивні завдання та тести: викладачі можуть створювати інтерактивні завдання та тести, які студенти можуть виконувати на своїх пристроях. «Draw & GO» забезпечує миттєвий зворотний зв'язок, дозволяючи студентам швидко виправляти помилки та засвоювати матеріал.
4. Персоналізоване навчання: завдяки гнучкості плагінів, «Draw & GO» може бути налаштований для персоналізованого навчання. Інструмент адаптує матеріали та завдання до індивідуальних потреб кожного студента, забезпечуючи оптимальні умови для їхнього розвитку та успіху.
5. Дистанційне навчання: в умовах дистанційного навчання «Draw & GO» є незамінним інструментом. Він дозволяє викладачам і студентам ефективно взаємодіяти незалежно від місця знаходження, зберігаючи високу якість навчального процесу.
6. Візуалізація складних концепцій: за допомогою розпізнавання жестів викладачі можуть малювати складні діаграми та схеми у реальному часі, що допомагає студентам краще розуміти складні концепції. Наприклад,

вивчення математичних функцій, хімічних реакцій або фізичних явищ стає значно легшим завдяки візуалізації.

«Draw & GO» представляє собою потужний інструмент для освітнього процесу, який об'єднує інноваційні технології та сучасні методи навчання. Використання цього інструменту сприяє залученню студентів, покращенню їх навчальних результатів та розвитку навичок, необхідних у сучасному цифровому світі. Завдяки своїм можливостям, «Draw & GO» відкриває нові горизонти для інтерактивного та ефективного навчання, роблячи освітній процес більш захоплюючим та результативним [101].

4.5 Автоматизація повсякденних дій

Система «Draw & GO» дозволяє автоматизувати повсякденні дії для підвищення ефективності та економії часу. Розглянемо на прикладі моніторингу курсу валют. Даний макрос налаштований на автоматичний щоденний запуск о 10:00 і виконує пошук курсу валют.

На рисунку 4.8 зображено конфігураційне вікно заданого макросу у системі «Draw & GO», який програмований на автоматичний запуск щоденно о 10:00 ранку. Цей макрос спрощує процес моніторингу інформації, автоматизуючи відкриття пошукової сторінки для збору даних про курс валют. Функція макросу зводиться до автоматизації задачі, яка б вимагала вручну введення пошукових запитів користувачем, а тепер відбувається без його безпосередньої участі. Подібний інструмент адаптується для регулярного збору інформації, що може бути використана для подальшого аналізу, підготовки звітів або інформування прийняття рішень.

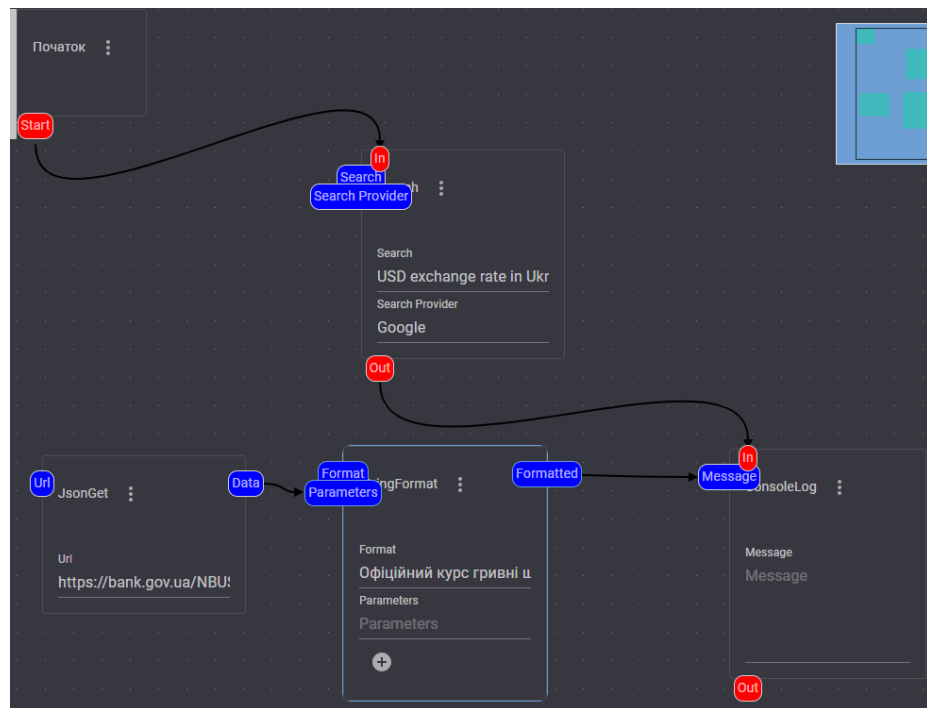


Рисунок 4.8 – Приклад щоденного макросу

4.6 Інші приклади застосування системи

Розглянемо застосування «Draw & GO» з точки зору більш пріоритетних сфер, де автоматизація може мати значний вплив на ефективність та безпеку.

Промисловість:

- управління виробничими процесами: автоматизація контролю за виробничими лініями, моніторинг стану обладнання, оптимізація робочих циклів, а також прогнозування та попередження поломок, застосування візуалізації даних для надання операторам та керівникам чіткої картини стану виробничого процесу [38].

Інформаційні технології:

- моніторинг інфраструктури: автоматизація збору даних про стан серверів, мереж та інших компонентів ІТ-інфраструктури, а також сповіщення про помилки або нестандартну поведінку систем.

- розгортання та управління програмним забезпеченням: автоматизація процесів випуску оновлень, розгортання додатків та управління конфігураціями.

Енергетика:

- управління енергоспоживанням: автоматизація моніторингу та аналізу споживання енергії, оптимізація навантаження на мережі та підвищення енергоефективності.

- управління відновлюваними джерелами енергії: автоматизація операцій з вітровими, сонячними та іншими видами відновлюваних джерел енергії для оптимального використання природних ресурсів.

Розумний дім і Розумне місто:

- автоматизація будинків: інтеграція систем керування освітленням, опаленням, вентиляцією та безпекою для підвищення комфорту і енергоефективності.

- управління міськими послугами: автоматизація управління вуличним освітленням, водопостачанням, транспортними потоками та іншими аспектами інфраструктури міста.

Автоматизація в цих областях може призвести до значного зниження витрат, підвищення продуктивності та покращення якості послуг.

Наразі проводиться дослідження з метою ідентифікації потенційних застосувань системи «Draw & GO» у сфері військових потреб. Проводиться аналіз специфічних вимог та викликів, з якими стикаються військові підрозділи, щоб визначити, як продукт може сприяти підвищенню ефективності, тактичної координації або інших критично важливих процесів. Перед нами стоїть завдання адаптувати та розширити функціонал «Draw & GO» з урахуванням строгих вимог щодо безпеки та надійності, які є характерними для військових застосувань.

4.7 Висновки до розділу 4

У розділі показано, що система «Draw & GO» має широкий спектр практичного застосування, забезпечує ефективну автоматизацію найрізноманітніших процесів та взаємодію з комп'ютеризованими пристроями в різних предметних галузях.

Надана детальна інструкція користувача, що описує покрокові дії для роботи з системою для кожного типу користувачів, демонструє легкість та інтуїтивність у роботі з системою та застосування її для вирішення різних задач, наприклад, таких як:

- інтеграція з комп'ютеризованими пристроями на прикладі управління роботами-пилососами;
- імітації трафіку на сайт для виконання навантажувального тестування для сайту, що може бути корисно QA-інженерам;
- можливості використання системи у навчальному процесі;
- автоматизація повсякденних дій та огляд інших сфер застосування таких, як промисловість, інформаційні технології, енергетика, розумний дім тощо.

ВИСНОВКИ

У дисертації розв'язано актуальну наукову задачу створення ефективної та гнучкої кросплатформної системи автоматизації процесів та взаємодії з комп'ютеризованими пристроями «Draw & GO», яка забезпечує управління на основі графічних жестів, застосування штучного інтелекту для автоматичної генерації макросів, інтеграцію з будь-якими існуючими сторонніми службами та додавання розширень.

Основні наукові та практичні результати дисертаційної роботи:

1. Проведено огляд та аналіз літературних джерел відповідно до теми дисертаційного дослідження, існуючих програмних продуктів для автоматизації процесів, технологій взаємодії з комп'ютеризованими пристроями на основі запису макросів. В результаті проведеного дослідження виявлено актуальні проблеми та перспективні напрями, які були враховані при створенні програмного продукту «Draw & GO».

2. Розроблено систему макросів та макрокоманд. Макроси, які вводяться через програмний засіб для автоматизації повторюваних дій, стали ключовими елементами у розробленій системі «Draw & GO». Ця концепція дозволяє користувачам ефективно маніпулювати процесами через візуальний інтерфейс, підвищуючи продуктивність роботи. Інтегрований підхід до налаштування індивідуальних завдань забезпечує можливість пристосувати систему до унікальних потреб без додаткових технічних навичок, що значно підвищує доступність і комфорт використання.

3. Запропоновано концепцію асоціативного графічного інтерфейсу у системах автоматизації процесів на основі застосування графічних ключів для визначення команд, що дозволяє підвищити інтуїтивність та ефективність управління процесами, що є особливо актуальним у контексті збільшення складності та об'єму задач. Удосконалено метод розпізнавання графічних жестів, що дозволило розпізнавати жести, які складаються з

кількох ліній, тим самим розширюючи кількість можливих асоціацій для більш складних команд. Враховуючи, що зв'язувати макроси з графічним кодом дуже зручно через психологічну схильність людини до асоціативної пам'яті, вдалося поєднати мінімальність інтерфейсу з максимальною продуктивністю та інтуїтивністю в управлінні, оскільки тепер можна звести рутинні дії до трьох кроків: перший – запис, другий – збереження, третій – відтворення.

4. Запропоновано технологію автоматичної побудови макросів у системах автоматизації процесів на основі застосування мовної моделі ІІІ, що дозволяє користувачам будувати робочі процеси, описуючи лише вимоги до них з можливістю голосового вводу, без необхідності поглиблюватись у знання програмування або навіть самої системи.

5. Розроблено Clean Microkernel Architecture – архітектуру програмного забезпечення для автоматизації процесів, яка базується на поєднанні мікроядра плагінів та чистої архітектури, що забезпечує масштабованість, дозволяє оптимізувати структуру програмного коду, зменшуючи залежності та дублювання, що сприяє підвищенню читабельності, обслуговуваності та тестованості коду, а також забезпечує покращення метрик якості, включаючи зниження цикломатичної складності та підвищення індексу підтримуваності.

6. Удосконалено методологію інтеграції програмного забезпечення з іншими бізнес-системами та сервісами за допомогою розроблення модульних кросплатформних з'єднувачів, які значно розширюють можливості автоматизації та інтегрованості корпоративного програмного забезпечення. Система може інтегруватися з широким колом сторонніх служб, таких як Slack, Amazon S3 Files, SendGrid тощо, що дозволяє розширити її функціональність та можливості.

7. Реалізовано підтримку плагінів, розроблено бібліотеку для створення розширень, що вже містить широкий набір макрокоманд для

автоматизації та оптимізації робочого процесу, забезпечуючи швидкість та зручність використання, а також надає можливості для користувачів створювати власні розширення, що додають нові функції до основної програми.

8. Реалізовано синхронізацію в режимі реального часу, та забезпечено питання безпеки.

«Draw & GO» – це інноваційний кросплатформний механізм робочого процесу, створений, використовуючи фреймворки .NET MAUI та Blazor, щоб дозволити розробникам швидко та ефективно розгортати програму та керувати нею. Розроблений програмний продукт є завершеним з точки зору логістики, тобто доступний для будь-якого користувача, що володіє навичками роботи з комп'ютером. Ознайомитись з функціональними можливостями програмного продукту «Draw & GO» та дізнатися про доступні ліцензії можна на офіційній сторінці <https://drawgo.azurewebsites.net>. Ефективність «Draw & GO» була визначена шляхом ретельної перевірки та тестування, які продемонстрували її високу адаптивність до різних умов роботи, гнучкість до вдосконалень та оновлень, а також здатність до швидкого впровадження універсальних рішень на різних платформах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A Cross-Platform Mobile Application for Learning Programming Basics / A. Yassine et al. Int. J. Interact. Mob. Technol. 2018. Vol. 12. P. 139-151. URL: <https://api.semanticscholar.org/CorpusID:53783039>.
2. A Shoulder Surfing Resistant Graphical Authentication System / H.-M. Sun et al. IEEE Transactions on Dependable and Secure Computing. 2018. Vol. 15. P. 180-193. URL: <https://api.semanticscholar.org/CorpusID:3878795>.
3. A survey on the use of graphical passwords in security / H. Gao et al. J. Softw. 2013. Vol. 8. P. 1678-1698. URL: <https://api.semanticscholar.org/CorpusID:10124745>.
4. A Systematic Literature Review of Graphical Password Schemes / T. I. Shammee et al. J. Comput. Sci. Eng. 2020. Vol. 14. P. 163-185. URL: <https://api.semanticscholar.org/CorpusID:234458710>.
5. Adhikari S., Jones B. A. A Modular Plugin Architecture for Literate Programming Editors. 2019 SoutheastCon. 2019. P. 1-4. URL: <https://api.semanticscholar.org/CorpusID:212645098>.
6. Al-Debagy O., Martinek P. A Comparative Review of Microservices and Monolithic Architectures. 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI). 2018. P. 000149-000154. URL: <https://api.semanticscholar.org/CorpusID:159041638>.
7. Andre P., Stievenart Q., Ghafari M. Developers Struggle with Authentication in Blazor WebAssembly. 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME). 2022. P. 389-393. URL: <https://api.semanticscholar.org/CorpusID:251224006>.
8. Antonyuk V. A., Sydorova M. G. A cross-platform mobile development for accelerating software development lifecycle. Actual problems of automation and information technology. 2022. URL: <https://api.semanticscholar.org/CorpusID:268995306>.

9. Antonyuk V., Sydorova M. Integration and use of artificial intelligence for automated macros creation. *System technologies*. 2024. Vol. 5 No. 154. P. 16-23.
10. Antonyuk V., Sydorova M. Synthesis of software architectures for cross-platform application development. *Actual problems of automation and information technology*. 2021. Vol. 25. P. 3-12.
11. Antonyuk V., Sydorova M. The concept of associative graphical interface in the workflow automation system. *System technologies*. 2023. Vol. 5 no. 148. P. 133-140.
12. Antonyuk V.A., Segeda N.E. Recognition of graphical passwords as the means to accelerate the work with software projects. *Сучасні науково-технічні дослідження у контексті мовного простору*. Т.2, С. 5-6.
13. Arora A., Mohana M. Architectural and Functional Differences in Dot Net Solutions. 2022 International Conference on Edge Computing and Applications (ICECAA). 2022. P. 1617-1622. URL: <https://api.semanticscholar.org/CorpusID:253423954>.
14. AutoIt. URL: <https://www.autoitscript.com/site/autoit/>.
15. Azure OpenAI Service REST API reference. URL: <https://learn.microsoft.com/en-us/azure/cognitive-services/openai/reference>.
16. Balalaie A., Heydarnoori A., Jamshidi P. Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture. *IEEE Software*. 2016. Vol. 33. P. 42-52. URL: <https://api.semanticscholar.org/CorpusID:18802650>.
17. Beltrao A. C., Farzat F. d. A., Travassos G. H. Technical Debt: A Clean Architecture Implementation. *Anais Estendidos do XI Congresso Brasileiro de Software: Teoria e Pratica (CBSOFT 2020)*. 2020. URL: <https://api.semanticscholar.org/CorpusID:234479966>.
18. Bhushan B., Muthiah P. Anti-smudge screening apparatus for electronic touch screens. *Microsystem Technologies*. 2013. Vol. 19, no. 8. P. 1261–1263. URL: <https://doi.org/10.1007/s00542-013-1856-2>.

19. Bianchi A., Oakley I., Kim H. PassBYOP: Bring Your Own Picture for Securing Graphical Passwords. IEEE Transactions on Human-Machine Systems. 2016. Vol. 46, no. 3. P. 380–389. URL: <https://doi.org/10.1109/thms.2015.2487511>.
20. Brown B. Save Time and Increase Accuracy With Zapier. 2018. URL: <https://api.semanticscholar.org/CorpusID:67741114>.
21. C#/.NET SDK for accessing the OpenAI APIs. URL: <https://github.com/OkGoDoIt/OpenAI-API-dotnet>
22. Chiang H.-Y., Chiasson S. Improving user authentication on mobile devices: a touchscreen graphical password. International Conference on Human-Computer Interaction with Mobile Devices and Services. 2013. URL: <https://api.semanticscholar.org/CorpusID:5075319>.
23. Clean Architecture of Client-Side Software Development for Smart Furniture Control / M. Bukovcan et al. 2022 11th Mediterranean Conference on Embedded Computing (MECO). 2022. P. 1-4. URL: <https://api.semanticscholar.org/CorpusID:249928502>.
24. Code Project. Processing Global Mouse and Keyboard Hooks in C#. URL: <http://www.codeproject.com/Articles/7294/Processing-Global-Mouse-and-Keyboard-Hooks-in-C>
25. Combining the Best of Two Worlds: Microservices and Micro Frontends as Basis for a New Plugin Architecture / F. Buhler et al. Symposium and Summer School on Service-Oriented Computing. 2022. URL: <https://api.semanticscholar.org/CorpusID:252819625>.
26. Component Management Framework for Operating System Based on Microkernel Architecture / W. Hui et al. 2022 China International Conference on Electricity Distribution (CICED). 2022. P. 218-222. URL: <https://api.semanticscholar.org/CorpusID:253251426>.
27. Cued-Click Point Graphical Password Using Circular Tolerance to Increase Password Space and Persuasive Features / K. Patra et al. Procedia

Computer Science. 2016. Vol. 79. P. 561-568. URL: <https://api.semanticscholar.org/CorpusID:59087905>.

28. Dunphy P., Yan J. Do background images improve “draw a secret” graphical passwords? Conference on Computer and Communications Security. 2007. URL: <https://api.semanticscholar.org/CorpusID:10922169>.

29. Esposito D., Saltarello A. Microsoft .NET: Architecting Applications for the Enterprise, Second Edition. 2014. URL: <https://api.semanticscholar.org/CorpusID:16772226>.

30. Evaluating Large Language Models Trained on Code / M. Chen et al. ArXiv. 2021. Abs/2107.03374. URL: <https://api.semanticscholar.org/CorpusID:235755472>.

31. Everything You Need to Know About Business Process Automation. URL: <https://www.smartsheet.com/understanding-evolution-and-importance-business-process-automation>

32. ExperD: Web-based Support for Laboratory Class Workflow Design and Execution / K. v. d. Kolk et al. International Conference on Computer Supported Education. 2013. URL: <https://api.semanticscholar.org/CorpusID:43359218>.

33. Fowler M. Patterns of Enterprise Application Architecture. 2002. URL: <https://api.semanticscholar.org/CorpusID:16457441>.

34. From Monolith to Microservice: Measuring Architecture Maintainability / M. H. Hasan et al. International Journal of Advanced Computer Science and Applications. 2023. URL: <https://api.semanticscholar.org/CorpusID:259141188>.

35. Goldman M. .NET MAUI in Action. Manning Publications, 2022. 350 p.

36. Goyal N., Singh H. Workflow Automation for Implementing Customer Service Request Desk in Hotel Industry. 2021 6th International

Conference on Signal Processing, Computing and Control (ISPCC). 2021. P. 25-28. URL: <https://api.semanticscholar.org/CorpusID:244405466>.

37. g-RAT | A Novel Graphical Randomized Authentication Technique for Consumer Smart Devices / M. A. Khan et al. IEEE Transactions on Consumer Electronics. 2019. Vol. 65. P. 215-223. URL: <https://api.semanticscholar.org/CorpusID:86674240>.

38. Hoffmann-Walbeck T. Workflow Automation. SpringerBriefs in Applied Sciences and Technology. 2021. URL: <https://api.semanticscholar.org/CorpusID:244177933>.

39. Hombergs T. Get Your Hands Dirty on Clean Architecture. 2019. URL: <https://api.semanticscholar.org/CorpusID:212898997>.

40. IFTTT vs. Zapier: A Comparative Study of Trigger-Action Programming Frameworks / A. Rahmati et al. ArXiv. 2017. Abs/1709.02788. URL: <https://api.semanticscholar.org/CorpusID:8958182>.

41. IntelliCode compose: code generation using transformer / A. Svyatkovskiy et al. Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. New York, NY, USA, 2020. P. 1433–1443. URL: <https://doi.org/10.1145/3368089.3417058>.

42. Izadi M., Gismondi R., Gousios G. CodeFill: Multi-token Code Completion by Jointly learning from Structure and Naming Sequences. 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE). 2022. P. 401-412. URL: <https://api.semanticscholar.org/CorpusID:246823166>.

43. Jankovic Z., Vesin B. Clean Genetic Algorithm Architecture for Improved Modularity and Extensibility. 2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES). 2023. P. 1-4. URL: <https://api.semanticscholar.org/CorpusID:266745888>.

44. Jesse Liberty, Rodrigo Juarez, Maddy Montaquila (Leger). .NET MAUI for C# Developers: Build cross-platform mobile and desktop applications. 2023, 296 p.
45. JetBit Macro Recorder. URL: <https://www.jetbit.com/macro-recorder/>
46. Language models are better than humans at next-token prediction / B. Shlegeris et al. ArXiv. 2022. Abs/2212.11281. URL: <https://api.semanticscholar.org/CorpusID:254974215>.
47. Language Models are Few-Shot Learners / T. B. Brown et al. ArXiv. 2020. Abs/2005.14165. URL: <https://api.semanticscholar.org/CorpusID:218971783>.
48. Large Language Models are Zero-Shot Reasoners / T. Kojima et al. ArXiv. 2022. Abs/2205.11916. URL: <https://api.semanticscholar.org/CorpusID:249017743>.
49. Lawrence S. Introducing .NET MAUI: Build and Deploy Cross-platform Applications Using C# and .NET Multi-platform App UI, 2023. 481p.
50. Leung T. Using Power Automate. 2021. URL: <https://api.semanticscholar.org/CorpusID:234146177>.
51. LocPass: A Graphical Password Method to Prevent Shoulder-Surfing / L. Y. Por et al. Symmetry. 2019. Vol. 11. P. 1252. URL: <https://api.semanticscholar.org/CorpusID:204911169>.
52. Macro Maker. URL: <https://macromaker.en.softonic.com/>
53. Macro Recorder. URL: <https://www.macrorecorder.com/>
54. Make. URL: <https://www.make.com/en>
55. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. 2017. URL: <https://api.semanticscholar.org/CorpusID:86710992>.
56. MazeLock URL: <https://eusing.com/mazelock/pclock.htm>

57. Meng W. RouteMap: A Route and Map Based Graphical Password Scheme for Better Multiple Password Memory. International Conference on Network and System Security. 2015. URL: <https://api.semanticscholar.org/CorpusID:30984398>.
58. Microkernel in Operating System: Architecture, Advantages URL: <https://www.guru99.com/microkernel-in-operating-systems.html>
59. Mohan D. K., Jothi M. J. S., Ajay M. B. Workflow Automation in Business Process Management. International Journal of Advanced Research in Science, Communication and Technology. 2022. URL: <https://api.semanticscholar.org/CorpusID:249450173>.
60. Nguyen T. V., Sae-Bae N., Memon N. D. DRAW-A-PIN: Authentication using finger-drawn PIN on touch devices. Comput. Secur. 2017. Vol. 66. P. 115-128. URL: <https://api.semanticscholar.org/CorpusID:39890868>.
61. Nijs P. The MVVM Pattern in .NET MAUI. 2023. 386p.
62. Nugroho Y. N., Kusumo D. S., Alibasa M. J. Clean Architecture Implementation Impacts on Maintainability Aspect for Backend System Code Base. 2022 10th International Conference on Information and Communication Technology (ICoICT). 2022. P. 134-139. URL: <https://api.semanticscholar.org/CorpusID:253047166>.
63. Nyman H. J. Modularizing Robot Driver Framework with a Plugin Architecture. 2018. URL: <https://api.semanticscholar.org/CorpusID:195450590>.
64. Ortinau D. Introducing .NET MAUI – One Codebase, Many Platforms. 2022. URL: <https://devblogs.microsoft.com/dotnet/introducing-dotnet-maui-one-codebase-many-platforms>
65. Othman A., R. M. Solution- Architecture in ASP.Net Core. International Journal of Computer Applications. 2020. URL: <https://api.semanticscholar.org/CorpusID:215869409>.

66. Pace Z. Window 8: signing in with a picture password, 2011. URL: <https://moviesgamesandtech.com/2011/12/17/windows8-signing-in-with-a-picture-password/>.
67. PassApp: My App is My Password! / H. Sun et al. Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services. 2015. URL: <https://api.semanticscholar.org/CorpusID:1782383>.
68. Por L. Y., Ku C. S., Ang T. F. Preventing Shoulder-Surfing Attacks using Digraph Substitution Rules and Pass-Image Output Feedback. Symmetry. 2019. Vol. 11. P. 1087. URL: <https://api.semanticscholar.org/CorpusID:202938913>.
69. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing / P. Liu et al. ACM Computing Surveys. 2021. Vol. 55. P. 1 - 35. URL: <https://api.semanticscholar.org/CorpusID:236493269>.
70. Prompt Engineering Guide, URL: <https://www.promptingguide.ai/>
71. Radford A., Narasimhan K. Improving Language Understanding by Generative Pre-Training. 2018. URL: <https://api.semanticscholar.org/CorpusID:49313245>.
72. Raj V., Ravichandra S. A service graph based extraction of microservices from monolith services of service-oriented architecture. Software: Practice and Experience. 2022. Vol. 52. P. 1661 - 1678. URL: <https://api.semanticscholar.org/CorpusID:247493466>.
73. RecGPT: Generative Personalized Prompts for Sequential Recommendation via ChatGPT Training Paradigm / Y. Zhang et al. 2024. URL: <https://api.semanticscholar.org/CorpusID:269148580>.
74. Recording and replaying system specific, source code transformations / G. Santos et al. 2015 IEEE 15th International Working Conference on Source

Code Analysis and Manipulation (SCAM). 2015. P. 221-230. URL: <https://api.semanticscholar.org/CorpusID:4737830>.

75. Reynolds L., McDonell K. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. 2021. URL: <https://api.semanticscholar.org/CorpusID:231925131>.

76. Rocha H. F. O. Moving from a Monolith to an Event-Driven Architecture. Practical Event-Driven Microservices Architecture. 2021. URL: <https://api.semanticscholar.org/CorpusID:244025637>.

77. Roehrig N. Applying Clean Architecture to Legacy Software Systems. Proposed for presentation at the Tri-Lab Advanced Simulation \& Computing Sustainable Scientific Software Conference held May 24-26, 2022 in Albuquerque, NM. 2022. URL: <https://api.semanticscholar.org/CorpusID:265010023>.

78. Secure Trusted Operating System Based on Microkernel Architecture / D. Li et al. International Congress on Information and Communication Technology. 2021. URL: <https://api.semanticscholar.org/CorpusID:239219841>.

79. Shahzad M., Liu A. X., Samuel A. Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it. Proceedings of the 19th annual international conference on Mobile computing \& networking. 2013. URL: <https://api.semanticscholar.org/CorpusID:207206090>.

80. Shoulder-surfing-proof graphical password authentication scheme / T.-S. Wu et al. International Journal of Information Security. 2013. Vol. 13. P. 245 - 254. URL: <https://api.semanticscholar.org/CorpusID:17555270>.

81. Software Architecture Patterns – Microkernel Architecture. URL: <https://priyalwalpita.medium.com/software-architecture-patterns-microkernel-architecture-97cee200264e>

82. Stonis M. Enterprise Application Patterns using .NET MAUI, 2022. 101 p.
83. Su R., Li X. Modular Monolith: Is This the Trend in Software Architecture?. ArXiv. 2024. Abs/2401.11867. URL: <https://api.semanticscholar.org/CorpusID:267068668>.
84. Tao H., Adams C. M. Pass-Go: A Proposal to Improve the Usability of Graphical Passwords. Int. J. Netw. Secur. 2008. Vol. 7. P. 273-292. URL: <https://api.semanticscholar.org/CorpusID:651758>.
85. Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review / B. Chen et al. ArXiv. 2023. Abs/2310.14735. URL: <https://api.semanticscholar.org/CorpusID:264426395>.
86. Weiss R., Luca A. D. PassShapes: utilizing stroke based authentication to increase password memorability. Nordic Conference on Human-Computer Interaction. 2008. URL: <https://api.semanticscholar.org/CorpusID:2698470>.
87. What is a Workflow Engine? URL: <https://www.ibm.com/cloud/blog/workflow-engine>
88. Workflow automation. URL: <https://www.techtarget.com/searchcontentmanagement/definition/workflow-automation>
89. Workflow Core. URL: <https://workflow-core.readthedocs.io/en/latest/>
90. YAGP: Yet Another Graphical Password Strategy / H. Gao et al. 2008 Annual Computer Security Applications Conference (ACSAC). 2008. P. 121-129. URL: <https://api.semanticscholar.org/CorpusID:10199>.
91. Ye R. .NET MAUI Cross-Platform Application Development: Leverage a first-class cross-platform UI framework to build native apps on multiple platforms, 2023. 400p.
92. Zayas-Caban T., Okubo T. H., Posnack S. Priorities to accelerate workflow automation in health care. Journal of the American Medical Informatics

Association: JAMIA. 2022. URL:
<https://api.semanticscholar.org/CorpusID:252994473>.

93. Ziliak E. THE BOOKSHELF: The Mathematics of Secrets: Cryptography from Caesar Ciphers to Digital Encryption by Joshua Holden. Math Horizons. 2017. Vol. 25. P. 27. URL:
<https://api.semanticscholar.org/CorpusID:125453573>.

94. Антонюк В.А., Сидорова М.Г. Microkernel architecture у розробці сучасних мобільних додатків. Сучасні інформаційні системи та технології: тези доповідей III Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених. Херсон, 30 листопада 2020 р., С. 96-97.

95. Антонюк В.А., Сидорова М.Г. Автоматизація аналізу та форматування коду. Інформаційні технології – 2022: тези доповідей IX Всеукраїнської науково-практичної конференції молодих науковців, Київ, 19 травня 2022 р., С. 7-9.

96. Антонюк В.А., Сидорова М.Г. Використання візуальних елементів керування для покращення взаємодії користувача. Проблеми математичного моделювання: тези доповідей всеукраїнської науково-методичної конференції, Кам'янське, 25-27 травня 2022 р., С.69-70.

97. Антонюк В.А., Сидорова М.Г. Використання ідеї асоціативного мислення людини у програмному забезпеченні. Інформаційні технології – 2021: тези доповідей VIII Всеукраїнської науково-практичної онлайн конференції молодих науковців, Київ, 20 травня 2021 р., С. 112-114.

98. Антонюк В.А., Сидорова М.Г. Використання штучного інтелекту у розробці клієнт-орієнтованих додатків. Інформаційні технології – 2023: тези доповідей X Всеукраїнської науково-практичної конференції молодих науковців, Київ, 18 травня 2023 р., С. 5-6.

99. Антонюк В.А., Сидорова М.Г. Жестова взаємодія як засіб прискорення роботи користувача з комп'ютером. Проблеми прикладної

математики та інформаційних технологій: тези доповідей наукової конференції за підсумками науково-дослідної роботи ДНУ за 2020 рік. Дніпро, 2021, С. 7.

100. Антонюк В.А., Сидорова М.Г. Застосування стенографії для захисту програм від несанкціонованого доступу. Інформаційна безпека та інформаційні технології (ІБІТ-2022): тези доповідей IV Міжнародної науково-практичної конференції, Львів, 30 листопада 2022 р., С. 140-142.

101. Антонюк В.А., Сидорова М.Г. Можливості використання програмного продукту «Draw & GO» у навчальному процесі. Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022): тези доповідей XX міжнародної науково-практичної конференції, Дніпро, 23-25 листопада 2022 р., С. 15-16.

102. Антонюк В.А., Сидорова М.Г. Нова архітектура у .NET MAUI як еволюція XAMARIN.FORMS. Сучасні інформаційні системи та технології: матеріали IV Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених, Херсон, 30 листопада 2021 р., С. 11-12.

103. Антонюк В.А., Сидорова М.Г. Порівняльна характеристика систем автоматизації програмного забезпечення у розробці власного додатку. Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2021): тези доповідей XIX міжнародної науково-практичної конференції, Дніпро, 17-19 листопада 2021, С. 5-6.