Vol. 4, No. 1, 2022

DOI: https://doi.org/10.23939/csn2022.01.028

УДК 681.3

СИСТЕМА БЕЗКОНТАКТНИХ ПЛАТЕЖІВ НА ОСНОВІ ТЕХНОЛОГІЇ NFC

І. М. Жолубак, П. В. Курман

Національний університет "Львівська політехніка",
кафедра електронних обчислювальних машин
E-mail: Ivan.M.Zholubak@lpnu.ua, *pavlo.kurman.ki.2018@lpnu.ua*

У статті досліджено систему для проведення безконтактних платежів методами технології NFC. Розглянуто практику еквайрингу як методу торгівлі, що збільшує привабливість бізнесу для клієнта та спрощує процеси в торгівлі для власників та працівників. Визначено актуальність такої системи в Україні та перспективи її розвитку. Проаналізовано наявність на ринку таких систем, комбінації систем, що дозволяють отримати таку ж привабливість для бізнесу та клієнтів.

Незважаючи на те, що такі системи приваблюють клієнтів до бізнесу, кількість сервісів, що можуть надавати відповідні послуги клієнту чи бізнесу, залишається незначною, а готових рішень, що об'єднують функціонал для клієнта і продавця одночасно, до сьогодні знайдено не було. Метою статті є подання етапів розробки систем безконтактних платежів, що передають дані між клієнтськими аплікаціями за допомогою технології NFC, та сервера на базі мікросервісної архітектури.

У статті представлено систему безконтактних платежів на основі технології NFC, її структуру та алгоритм роботи. Вказано, що принцип роботи полягає у отримані та передачі даних від двох кінцевих клієнтів системи до однієї із платіжних систем в певному порядку, що дозволяє безпечно виконувати платіжні операції.

Ключові слова: Python, Kivy, еквайринг, мікросервіс.

Вступ

Торгівля – найвідоміший процес і двигун розвитку суспільств і народів у світі. Вона в своїй основі спонукає шукати рішення для розширення попиту та збільшення пропозиції. Торгівля як процес існувала ще з давніх часів, і завжди в своїй історії послуговувалась найсучаснішими методами обміну цінностями.

З погляду налагодження зв'язків та обміну культурними практиками саме торгівля дала поштовх для об'єднання розрізнених та до цього не пов'язаних народностей навколо торгових шляхів.

Проте ця галузь, що спонукала розвиток упродовж тисячоліть існування людства технологічно майже не змінювалась з часів першої промислової революції, окрім використання нових інструментів доставки у вигляді авто та залізниці, до самого двадцятого століття.

Перші розмови про відмову від використання, або хоча б альтернативу готівковим розрахункам велись ще у дев'ятнадцятому столітті. Ера сучасних кредитних карток почалась у 1949 році з появою ресторанної кредитної карти Diners Club. Приходячи у ресторан, клієнт міг пред'явити замість готівки цю карту, ресторани передавали копії рахунків Diners Club, а той в кінці місяця виставляв клієнту загальний рахунок.

Сьогодні роздрібна торгівля зазнає активної модернізації, переходячи від готівкових до безготівкових методів оплати за товари. До недавнього часу завдання ідентифікації платіжних даних носили лише контактні банківські картки, проте зараз з'явились технології, що дають змогу виконувати передачу цих даних безконтактно, використовуючи свій персональний смартфон, годинник чи навіть NFC-мітку, прикріплену до будь-якого зручного користувачеві об'єкту. Проте складність залучення та обслуговування підштовхує до пошуку нових альтернатив, щоб торгівля знову стала джерелом інновацій.

1. Аналіз останніх досліджень та публікацій

Сьогодні використання систем безготівкових платежів увійшло до загального вжитку. Аналізуючи схему еквайрингу без використання готівки, отримуємо набір дій, що виконуються під час будь-якої безготівкової операції [1]. Саме для цієї роботи необхідно розробити систему, використовувану продавцем, тобто передачі даних від споживача, обробки та зберігання продавцем та передачі даних банку-екваєру.

Проте найбільший прорив у використанні безконтактних платежів стався завдяки виходу на український ринок Google Pay та Apple Pay як альтернативи банківським безконтактним карткам.

Принцип роботи названих додатків – Google Pay та Apple Pay – подібний до безконтактних банківських карток, проте отримання та збереження токена для платежів переноситься з задачі банку-емітента карти, що вносить ці дані в карту напряму до системи безконтактних платежів на смартфоні чи будь-якому іншому підтримуваному пристрої.

В випадку Google Pay користувач реєструє кредитну або дебетову картку в додатку Google Pay. Система надсилає банку-емітенту цієї картки запит на присвоєння токена, який використовуватиметься замість доданої картки [2]. Його отримання означає, що картці «присвоєно токен», тобто з нею зв'язано унікальний ідентифікаційний номер. Google Pay шифрує цю картку з присвоєним токеном, щоб її можна було використовувати для оплати.

Банк, який обслуговує торгову точку, і банк-емітент картки використовують наявну інформацію про клієнта та його дешифровану платіжну інформацію для здійснення трансакції.

Процес виконання безготівкового безконтактного платежу складається із таких етапів:

- Користувач отримує токен для картки.

Користувач додає картку в Google Pay. Після цього на його мобільному пристрої зберігається токен для платежів, зашифрований за допомогою ключа одноразового або кількаразового використання.

- Продавець отримує токен.

Коли користувач у магазині наближає свій пристрій до термінала з увімкненим зв'язком малого радіуса дії (NFC), пристрій надсилає на термінал цей токен, дату закінчення терміну дії та криптограму через протокол NFC.

- Продавець обробляє платіж. На основі даних картки продавець обробляє платіж через банк, який обслуговує цю торгову точку.
- Примітка. Платіж потрібно позначити як безконтактну трансакцію через систему торгової точки або платіжний термінал.
- Обслуговуючий банк обробляє платіж. Банк, який обслуговує торгову точку, обробляє дані картки, отримані через NFC, за допомогою відповідної платіжної мережі.
- Постачальник токенів перетворює токен. Постачальник токенів перевіряє криптограму й перетворює токен на фактичний номер картки користувача.
- Банк-емітент картки отримує інформацію про власника картки. Мережа надсилає банку-емітенту номер картки користувача, дату закінчення терміну дії й індикатор про те, що постачальник токенів виконав перевірку за її дорученням.
- Мережа отримує дозвіл на авторизацію. Банк-емітент картки виконує перевірку на рівні рахунку та перевірку повноважень і надсилає мережі дозвіл на авторизацію.

- Термінал повідомляє про отримання або помилку дозволу на авторизацію трансакції. Платіжна мережа через обслуговуючий банк надсилає дозвіл на авторизацію в точку продажу, а з неї – користувачеві. На платіжному терміналі користувач і касир бачать повідомлення про отримання або помилку дозволу на трансакцію.

Тобто, якщо взяти до уваги все сказане вище, ця система продовжує використовувати інфраструктуру банків у вигляді їхніх терміналів і допоміжного обладнання [3]

Продовжуючи пошук подібних рішень, можна зіткнутись із сервісом «Термінал у смартфоні» кооперації ПриватБанку й Visa. Це рішення засноване на основі Tap to Phone – платіжної технології, що перетворює сучасні смартфони або планшети, оснащені NFC-чіпом, у повноцінний POS-термінал [6].

Цей новий сервіс дозволяє анулювати потребу бізнесу в банківському терміналі під час продажу товарів чи послуг. Терміналом слугує смартфон на базі ОС Android версії 8.0 і новішої. «Термінал» не є заміною Google Pay чи Apple Pay, оскільки виконує зовсім інші, паралельні функції, не відсилаючи дані про користувача, а їх отримуючи.

Проте, комбінація рішень від ПриватБанку та Google Pay не дозволить здійснити жодного взаємозв'язку для передачі додаткових даних про куплені товари чи послуги від однієї системи до іншої, що зумовлює певні складнощі та перспективу для систем, які будуть об'єднювати в собі функціонал обох систем.

2. Постановка задачі

Метою статті є розробка систем безконтактних платежів на основі технології NFC. Розробка проводиться на мові Python з використанням функціоналу фреймворку Kivy Project [4], що буде отримувати дані про платіж, надсилати серверу, обробляти і зберігати, та передавати платіжній системі дані для проведення платежу. Потрібно розглянути методи та інструменти розробки мікросервісної архітектури для ОС Android та еквайрингу, для реалізації системи безконтактних платежів на основі технології NFC, розглянути типові аналоги цієї системи, їхні комбінації, визначити їхні ключові особливості, недоліки та переваги. Проаналізувати доцільність розвитку створеної технології.

3. Огляд системи безконтактних платежів

Система банківських платежів із заданим функціоналом повинна мати дві аплікації для продавця і покупця, комунікувати між собою завдяки NFC та безпечно зберігати дані про трансакції.

Ці вимоги обумовлюють вибір клієнт-серверної архітектури, оскільки дані про платежі на пристрої клієнта не є захищеними від злому, є незручними та створюють ризики витоку цих даних до зловмисників, або уможливлюють отримання доступу до банківської системи.

Клієнтські частини через наявність покупця та продавця повинні бути різними, та взаємодіяти з сервером уніфіковано, надаючи достатньо даних для ідентифікації та передачі банківській системі. Окрім цього, система може бути нерівномірно навантажена та масштабована, що відкидає можливість використання монолітної архітектури.

Як наслідок, ми отримуємо клієнт-серверну мікросервісну архітектуру [5]. Ця архітектура передбачає використання мікросервісів для виконання бізнес-задач незалежно один від одного, та поєднуваних лише взаємними викликами API.

4. Реалізація проєкту

В першу чергу для розробки необхідно обрати мову програмування, на інструментах якої будуть реалізовуватись всі або частина системи (у випадку використання кількох мов – компоненти).

Для системи безконтактних платежів необхідна підтримка однієї, або краще декількох технологій для кожної з задач, ця мова повинна підтримувати ООП та мати можливість запуску на

мобільних пристроях з ОС Android. Необхідно дозволяти використовувати ресурси достатньо ефективно, щоб обробляти велику кількість запитів на сервер.

Для виконання таких задач було обрано не ідеальне але цілком задовільне рішення – мову Python [6]. Вона широко застосовується для реалізації серверів, обробки і передачі як звичних, так і великих об'ємів даних, підтримує більшість популярних баз даних.

Основним недоліком її може бути слабка підтримка роботи в ОС Android, основні бібліотеки, що дозволяють працювати з цією ОС, слабко розвинені та дають дещо обмежений і важкий у реалізації функціонал.

Наступним елементом для таких систем є база даних.

База даних – абстракція над файловою системою ОС, яка значно спрощує зберігання, обробку та оновлення даних у розроблених програмах. На високому рівні програми зберігають дані та демонструють їх користувачеві в зручному вигляді.

Однією з найрозвиненіших та таких, що найкраще підходять за своєю продуктивністю та масштабованістю, є база даних PostgreSQL.

PostgreSQL – об'єктно-реляційна система керування базами даних з відкритим вихідним кодом [7]. Порівняно з іншими проєктами з відкритим кодом, наприклад, Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.

Елементом, з яким найчастіше буде взаємодіяти користувач, є сервіс графічного інтерфейсу. Зручність та функціонал цього інтерфейсу буде вирішальним для користувача під час вибору між подібними системами [8].

Оскільки Python сьогодні не є популярним рішенням для розробки на ОС Android, то й вибір бібліотек для цієї системи є не широким. Проте є одне рішення, що задовольняє більшість потреб банківської платіжної системи – бібліотека Kivy.

Ця бібліотека з певними додатковими рішеннями дозволяє будувати графічний інтерфейс для ОС Android, надаючи високу продуктивність, оскільки всі ресурсоємні процеси виконуються кодом мови С. Бібліотека надає прямий доступ до камери, буфера обміну, мікрофону та інших ресурсів смартфону.

Оскільки вся система банківських безконтактних платежів складається з певної кількості незалежних і працюючих паралельно мікросервісів, необхідно вибрати систему віддаленого виклику процедур для певного взаємозв'язку між ними. Основними ресурсами для такої взаємодії сьогодні є REST API та gRPC. Для розробки цієї системи підходять як перша так і друга система, проте мною була обрана друга через свою новизну, вищу ефективність та перспективність.

Логічним кроком перед розробкою системи буде створення структурної схеми, оскільки проєкт передбачає взаємодію певної кількості самостійних елементів.

Головну роль відіграють додатки на ОС Android, що будуть передавати технологією NFC між собою дані про оплату та банківський модуль, що безпосередньо звертається до банківської системи для здійснення нею оплати.

Модулі між собою можна умовно поділити на основні та допоміжні. До основних сервісів проєкту належать:

- Сервіс обробки даних DMS –переформатовує дані у зручну для зберігання або відправки банківській системі форму. Окрім цього, цей модуль напряму керує базою даних.
- Сервіс банківських операцій BOS, що використовує отримані дані для здійснення необхідних запитів до банківської системи. Зв'язується за наданиит банком або платіжною системою методами та передає необхідні цьому банку або платіжній системі для оформлення платежу дані.
- Сервіс клієнта та сервіс продавця. В дечому подібні сервіси, що використовуються для взаємодії з користувачем завдяки сервісам графічного інтерфейсу, між собою завдяки технології NFC та з серверною частиною через сервіс передачі даних клієнта. Окрім цього, відповідає за криптографію повідомлень з боку клієнта.

- Сервіси передачі даних клієнта та сервера. Виконують задачу передачі даних між клієнтською та серверними частинами.
- Сервіс-оркестрант. Виконує роль управління роботою сервісів обробки даних та сервісом банківських операцій. Перевіряє стан з'єднання сервісу передачі даних сервера з клієнтом та відповідає за криптографію даних з боку сервера.
-

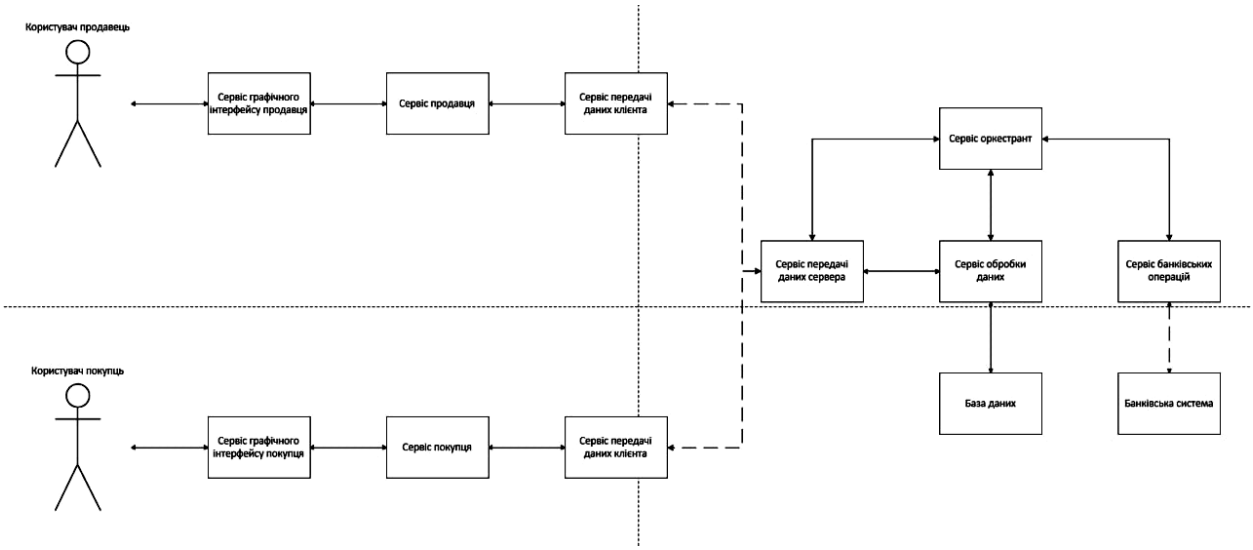

Рис. 1. Структурна схема проєкту

Основною задачею на початку розробки системи, виходячи зі структурної схеми, є створення міжсервісного зв'язку та серверної частини, або окремо клієнтської частини.

Користувачі системи безконтактних платежів поділяються на дві категорії за задачами, які вони виконують, використовуючи цю систему. Це купівля та продаж товарів чи послуг.

Виходячи з цього, вищеназваними двома категоріями є покупець та продавець. Кожен з них виконує свою задачу, тому алгоритми системи для цих користувачів будуть різними.

Першочергово необхідно розуміти, що платіжні дані не можна транспортувати без надійного шифрування, оскільки це надзвичайно чутливі дані користувача. Необхідно забезпечити їхню максимальну захищеність.

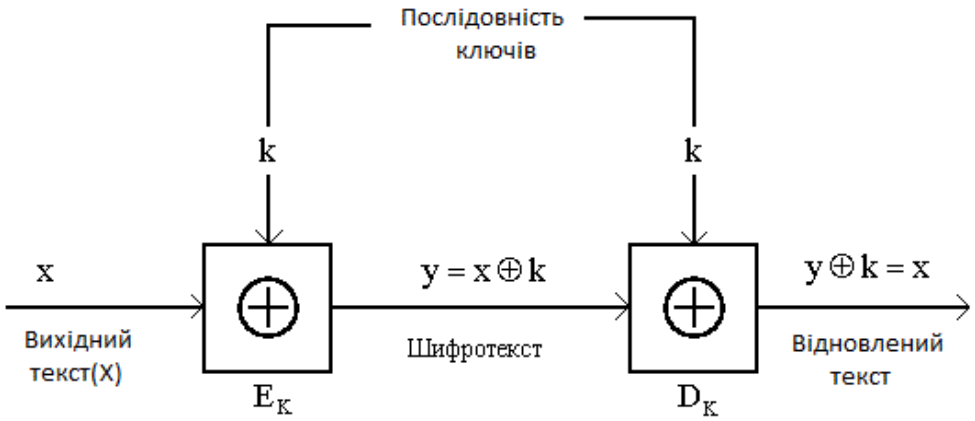

Рис. 2. Загальна схема процесу шифрування

Для ініціалізації програми на основі фреймворку Kivy необхідно імпортувати всі елементи з бібліотеки, що будуть використовуватись, створити клас, що буде успадкований від kivy.app.App.

Елементи необхідно додавати у певні layout, які формують стиль, за яким будуть розташовані елементи. Для цього вікна було обрано BoxLayout.

Створюючи елементи, потрібно враховувати як відступи від країв атрибутом size_hint, так і розташування елементів у заданому налаштуванні поля.

Для запуску готового вікна необхідно виконати метод run створеного вище класу RegApp, успадкований від батьківського класу App.

Серверна частина в цій системі ділиться на чотири мікросервіси, що взаємодіють між собою за допомогою технології gRPC. Використання такого методу зобов'язує створити .proto файл з налаштуваннями передачі даних між сервісами.

Створивши для кожної взаємодії між сервісами такі конфігурації, отримуємо готове середовище для взаємодії сервісів.

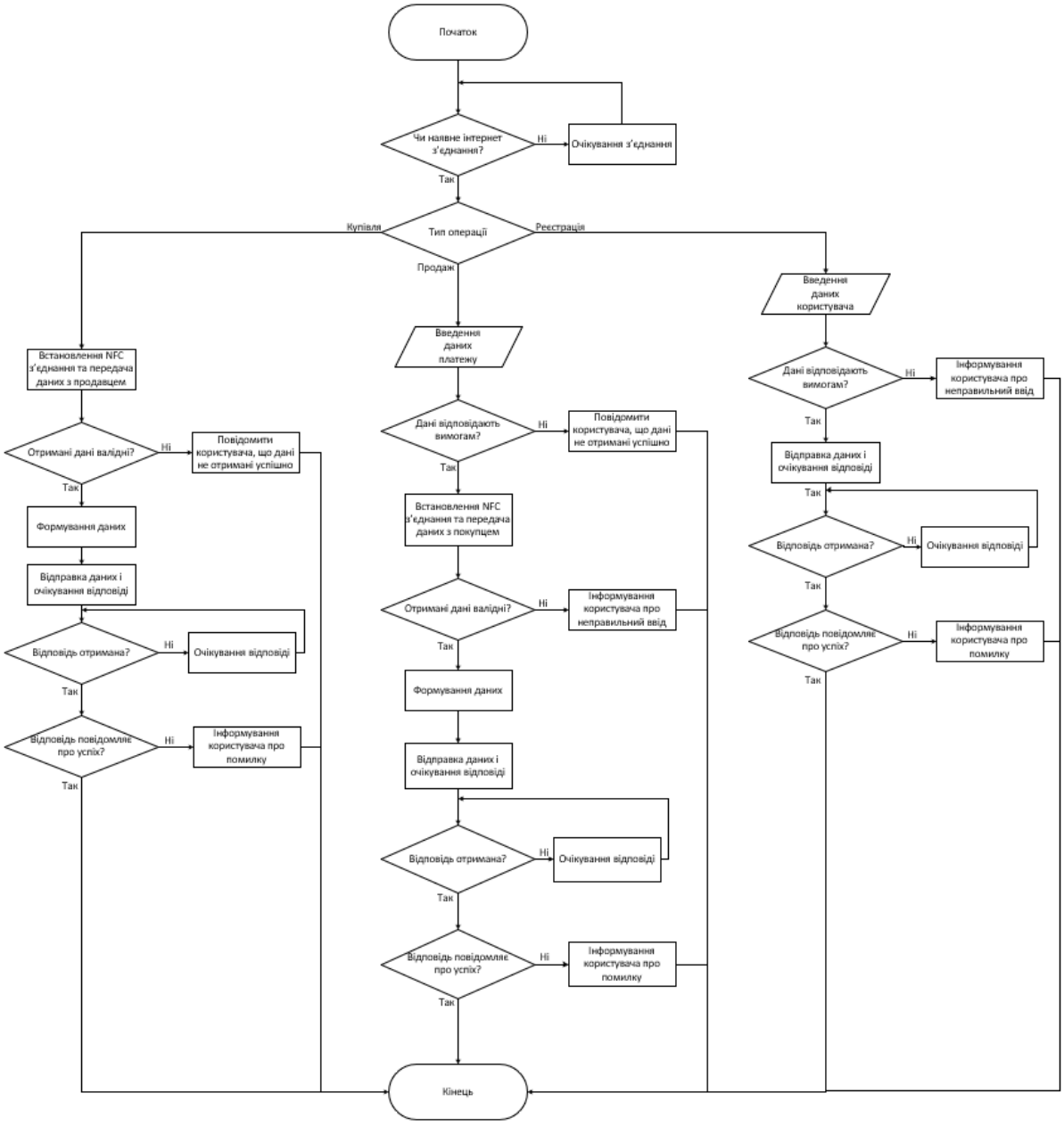

Рис. 3. Алгоритм роботи клієнтської частини

Для автоматичного запуску сервісів, необхідно використовувати контейнери Docker, та докерфайл, в якому зазначити правила розгортання та запуску тих чи інших аплікацій.

Концептуально він складається з декількох етапів, а саме:

1) вибір базової платформи контейнера;
2) перенесення проєктних файлів у контейнер;
3) обробка залежностей проєкту;
4) запуск сервісу.

Сервіс DMS реалізовує опрацювання отриманих даних, їх підготовку, зберігання та передачу. Для виконання цього функціоналу необхідно створити механізми отримання даних від gRPC, обробки цих даних та методи запису й читання з бази даних.

Отримання даних реалізовано класом DataStub, що у конструкторі отримує аргументом канал grpc.Channel для отримання даних.

Для передачі даних необхідно реалізувати клас DataServicer, який під час ініціалізації grpc буде перезаписувати метод Put. Також необхідно створити функцію add_DataServicer_to_server, яка буде повідомляти grpc про DataServicer.

Для роботи з базою даних необхідно створити з'єднання з базою та отримати курсор, що дає прямий доступ для управління базою даних.

Для запуску усіх сервісів запускається сервіс-оркестрант, що передає команду іншим сервісам на запуск. Також цей сервіс перевіряє наявність встановленого з'єднання з усіма сервісами, та у разі відсутності зв'язку перезапускає сервіси.

Створення будь-якого сервісу потребує звернення до клієнта docker та передачі йому в аргументах функції dclient.services.create таких параметрів, як назва, образ, налаштування мережних доступів, режиму, обмеження, точки монтування.

Однак в рамках цієї роботи неможливою є розробка сервісу банківських операцій через відсутність загальнодоступних API для доступу до платіжних та банківських систем. Загальнопоширеним рішенням є вбудовані вікна оплати в вебсторінку або аплікацію, проте рішення server–server, що якраз і має на меті проведення платежу зі сторони розробника системи, і що дозволяє виконувати платежі автоматично, є жорстко контрольовані та потребують високого рівня сертифікації PCI DSS.

5. Алгоритм роботи

Алгоритм роботи системи зображено на рис. 3 та 4.

– Користувач починає роботу з системою через одну з аплікацій клієнтської частини. Першочергово користувач повинен зареєструватись. Після цього користувач може повноцінно використовувати систему.
– Продавець вказує вартість та деталі покупки, покупець підтверджує платіж.
– Клієнтські аплікації перевіряють дані один одного, шифрують та передають дані серверу.
– Бот аналізує повідомлення клієнтських частин, зберігає дані та проводить платіж.

6. Висновки

Розроблено систему безконтактних платежів на основі технології NFC. Розглянуто методи та інструменти розробки мікросервісної архітектури для ОС Android та еквайрингу для реалізації системи безконтактних платежів на основі технології NFC. Розглянуті типові аналоги цієї системи, їхні комбінації, визначено їхні ключові особливості, недоліки та переваги. Проаналізовано доцільність розвитку створеної технології.

Проєкт реалізовано мовою програмування Python з використанням технології віддаленого виклику процедур Google gRPC та фреймворку Kivy Project, для взаємодії з операційною системою Android. Обрано базу даних PostgreSQL. Під час аналізу було створено структурну схему програмного продукту, схеми бази даних, а також схеми алгоритмів роботи системи.

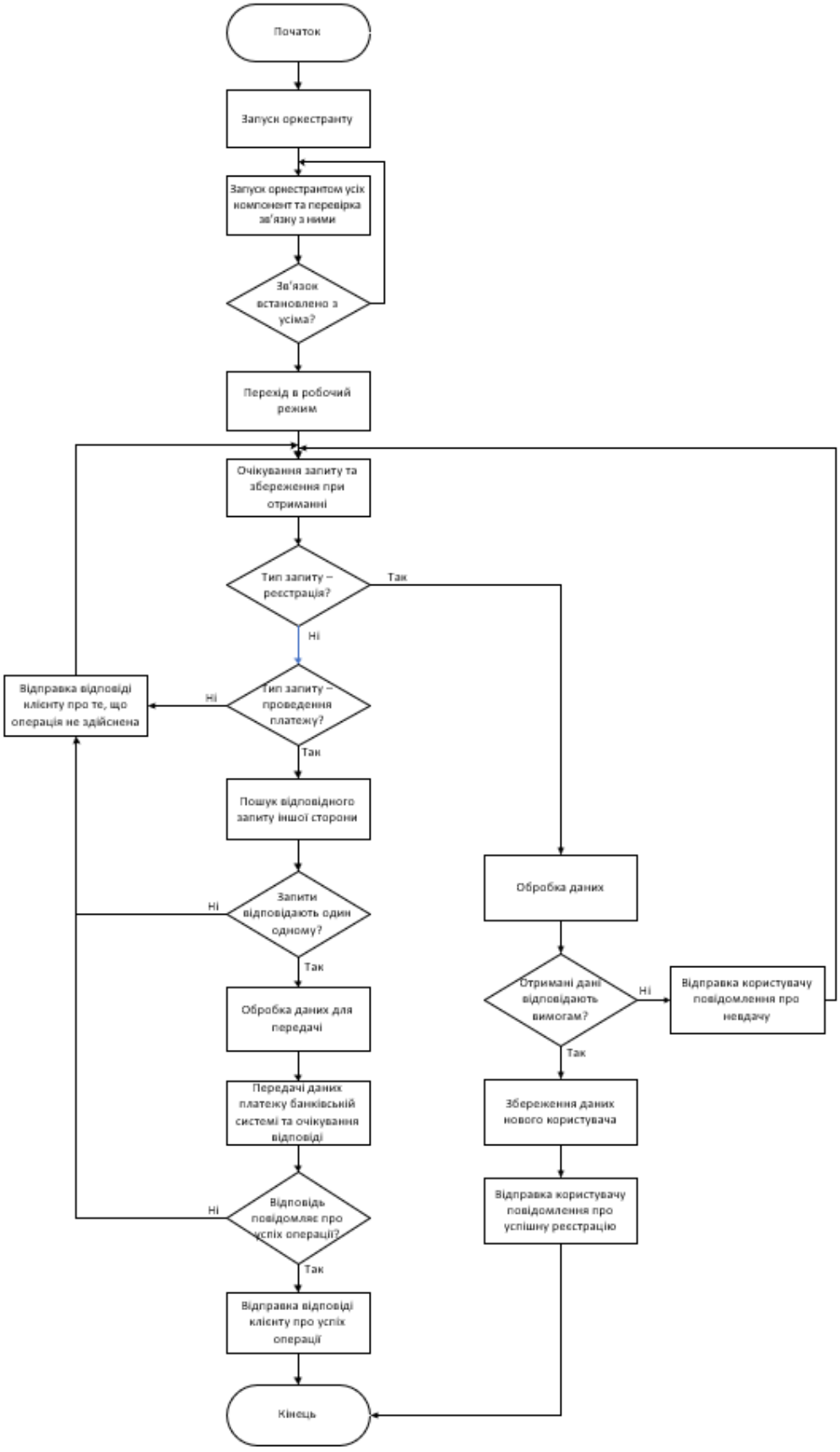

Рис. 4. Алгоритм роботи серверної частини

Список літератури

1. *Padmapriya N., Tamilarasi K., Kanimozhi P., Kumar T. A., Rajmohan R. and Adeola A. S. «A Secure Trading System using High level Virtual Machine (HLVM) Algorithm»,* 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), *2022. Pp. 1–4. DOI: 10.1109/ICSTSN53084.2022.9761326.*

2. *Zhonglin L., Minghua Z., Yin W. and Dong J. «Banking Intelligence: Application of data warehouse in bank operations»,* 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, *2008. Pp. 143–146. DOI: 10.1109/SOLI.2008.4686380.*

3. *Perry T. S. «Electronic banking goes to market», in* IEEE Spectrum. *Vol. 25. No. 2. Pp. 46–49, Feb. 1988. DOI: 10.1109/6.4510.*

4. *Gad A. F. «NumPyCNNAndroid: A Library for Straightforward Implementation of Convolutional Neural Networks for Android Devices»,* 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), *2019. Pp. 17–22. DOI: 10.1109/ITCE.2019.8646653.*

5. *Bushong V., Das D., Al Maruf A. and Cerny T. «Using Static Analysis to Address Microservice Architecture Reconstruction»,* 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), *2021. Pp. 1199–1201. DOI: 10.1109/ASE51524.2021.9678749.*

6. *Kumar A.and Panda S. P. «A Survey: How Python Pitches in IT-World»,* 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), *2019. Pp. 248–251. DOI: 10.1109/COMITCon.2019.8862251.*

7. *Zhengdong X. and Tianming B. «The implementation of flash-aware buffer replacement algorithms in PostgreSQL»,* 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), *2015. Pp. 1215–1219. DOI: 10.1109/FSKD.2015.7382115.*

8. *Aleksy M., Domis D., Sehestedt S. and Ulrich M. «Utilizing Business Process Analysis and Feature Analysis in Software Product Assessment»,* 2015 IEEE 17th Conference on Business Informatics, *2015. Pp. 159–164. DOI: 10.1109/CBI.2015.22.*

CONTACTLESS PAYMENT SYSTEM BASED ON NFC TECHNOLOGY

I. Zholubak, P. Kurman

Lviv Polytechnic National University,
Computer Engineering Department

The article investigates the system for contactless payments using NFC technology. The practice of acquiring as a method of trade that increases the attractiveness of the business for the client and simplifies the process of trade for owners and employees. The relevance of such a system in Ukraine and prospects for its development are determined.

The availability of such systems on the market, combinations of systems that allow to obtain the same attractiveness for business and customers are analyzed. Although such systems attract customers to the business, the number of services that can provide relevant services to the customer or business remains small, and ready-made solutions that combine functionality for both customer and seller have not yet been found. The aim of the article is to present the stages of development of contactless payment systems that transfer data between client applications using NFC technology and a server based on microservice architecture.

The article presents a system of contactless payments based on NFC technology, its structure and algorithm. It is stated that the principle of operation is to receive and transmit data from two end customers of the system to one of the payment systems in a certain order, which allows secure execution of payment transactions.

Keywords: Python, Kivy, acquiring, microservice.