

<https://doi.org/10.31891/2307-5740-2023-316-2-24>

УДК 338.242

Катерина ШАПОВАЛОВА

Дніпровський національний університет імені Олеся Гончара
e-mail: shapovalova.kateryna23@gmail.com

Поліна ШЕРЕМЕТА

Дніпровський національний університет імені Олеся Гончара
e-mail: polinasheremetta@gmail.com

NFC ТЕХНОЛОГІЯ ТА ПЕРСПЕКТИВИ ЇЇ РОЗВИТКУ

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 отриманні та передачі даних від двох кінцевих клієнтів системи до однієї з платіжних систем в певному порядку, що дозволяє безпечно виконувати платіжні операції.

Ключові слова: Python, Kivy, еквайринг, мікросервіс.

Kateryna SHAPOVALOVA, Polina SHEREMETA

Oles Honchar Dnipro National University

NFC TECHNOLOGY AND ITS DEVELOPMENT PROSPECTS

The article examines a system for making contactless payments using NFC technology. The practice of acquiring as a trade method that increases the attractiveness of business for the client and simplifies trade processes for owners and employees is considered. The relevance of such a system in Ukraine and the prospects for its development have been determined. The presence on the market of such systems, combinations of systems that allow to obtain the same attractiveness for business and customers was analyzed.

Despite the fact that such systems attract customers to the business, the number of services that can provide relevant services to the customer or business remains insignificant, and ready-made solutions combining functionality for the customer and the seller at the same time have not been found to date. The purpose 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 of operation. It is indicated that the principle of operation consists in receiving and transmitting data from two end customers of the system to one of the payment systems in a certain order, which allows for safe payment transactions.

Keywords: Python, Kivy, acquiring, microservice.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Торгівля є найвідомішим процесом і двигуном соціального розвитку та розвитку людей у світі. По суті, це стимулює пошук рішень, які розширюють попит і збільшують пропозицію. Торгівля як процес існувала з давніх часів і протягом всієї історії використовувала найсучасніші методи обміну вартості. З точки зору налагодження зв'язків і обміну культурними звичаями, торгівля забезпечувала об'єднуючий рух для різних і раніше не пов'язаних народів 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 [1, с. 1-4].

Ще в 1800-х роках почалися дискусії про відмову від використання або принаймні альтернативи розрахункам готівкою. Сучасна ера кредитних карток почалася в 1949 році з появою кредитної картки ресторану Diners Club. Коли гості приходять до ресторану, вони можуть пред'явити цю картку замість готівки, і ресторан надасть копію рахунку в Diners Club, а Diners Club виставить гостю загальний рахунок наприкінці місяця. Сьогодні роздрібна торгівля переживає активну модернізацію, переходячи від готівкових до безготівкових способів оплати.

До недавнього часу завдання ідентифікації платіжних даних носили лише контактні банківські картки, проте зараз з'явилися 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.

Аналіз останніх досліджень та публікацій

Сьогодні використання систем безготівкових розрахунків стало звичайною практикою. Аналізуючи сценарій безготівкового придбання, ми отримуємо набір дій, які виконуються під час будь-якої безготівкової операції [2, с. 143-146]. Саме для цієї роботи необхідно розробити систему користування продавцем, тобто передачу даних від споживача, обробку та зберігання продавцем і передачу даних банку-еквайру.

Проте найбільший прорив у використанні безконтактних платежів стався завдяки виходу на український ринок Google Pay та Apple Pay як альтернативи безконтактним банківським картам. Вище зазначені додатки (Google Pay і Apple Pay) працюють подібно до безконтактних банківських карток, але отримання та збереження платіжних токенів вивільняється від завдань банку-емітента, який вводить ці дані безпосередньо в картку в смартфон або будь-яку іншу систему безконтактних платежів на підтримуваних пристроях. Для Google Pay користувач реєструє кредитну або дебетову картку в програмі 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.

Банки, що обслуговують точки продажу, і банки-емітенти карток використовують доступну інформацію про клієнтів і їхню розшифровану платіжну інформацію для здійснення транзакцій. Процес здійснення безготівкової безконтактної оплати включає наступні етапи:

- Користувач отримує токен з картки. Користувач додає картку в Google Pay. Після цього платіжний токен зберігається на його мобільному пристрої, зашифрований одноразовим або багаторазовим ключем [3, с. 46-49].

- Продавець отримує токен. Коли користувач у магазині наближає свій пристрій до терміналу 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.

- Обслуговуючий банк обробляє платіж. Банк, який обслуговує торгову точку, обробляє дані картки, отримані через NFC, з відповідною платіжною мережею.

- Постачальник маркерів перетворює маркер. Постачальник маркерів перевіряє пароль і перетворює маркер на фактичний номер картки користувача. Банк-емітент 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 транзакції.

- 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 транзакцію.

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 [4, с. 17-22].

Рішення базується на платіжній технології Tap to Phone, яка перетворює сучасний смартфон або планшет, оснащений чіпом NFC, на повноцінний POS-термінал [6]. Нова послуга позбавляє суб'єктів господарювання необхідності використовувати банківські термінали під час продажу товарів чи послуг. В якості терміналів використовуються смартфони на базі ОС Android версії 8.0 і вище. «Термінал» не є заміною Google Pay або Apple Pay, оскільки він робить саме 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'єднувати в собі функціонал обох систем.

Формулювання цілей статті

Мета цієї статті – виокремити NFC технологію та перспективи її розвитку.

Виклад основного матеріалу

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 дозволити використовувати ресурси достатньо ефективно, щоб обробляти велику кількість запитів на сервер [5, с. 1199-1201].

Для виконання таких завдань було обрано неідеальне, але цілком задовільне рішення – мова Python [6, с. 248-251]. Він широко використовується при реалізації серверів, обробки та передачі звичайних і великих обсягів даних, підтримки більшості популярних баз даних. Його основними недоліками є, ймовірно, слабка підтримка роботи в ОС Android, слабка розробка основних бібліотек, що дозволяють працювати на цій ОС, а також обмежений і складний 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, с. 1215-1259].

На відміну від інших проектів з відкритим кодом, таких як Apache, FreeBSD або MySQL, PostgreSQL не контролюється жодною компанією, і його розробка стала можливою завдяки багатьом людям, які хотіли використовувати цю СУБД і впроваджувати в неї останні розробки, а також співпраці з елементом, з яким найчастіше взаємодіє користувач, є сервіс графічного інтерфейсу.

Зручність та функціонал цього інтерфейсу буде вирішальним для користувача під час вибору між подібними системами [8, с. 159-164]. 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 разом з деякими додатковими рішеннями дозволяє побудувати графічний інтерфейс для ОС Android, забезпечуючи високу продуктивність, оскільки всі ресурсомісткі процеси виконуються в коді C. Ця бібліотека забезпечує доступ до камери, буфера обміну, мікрофону та інших ресурсів смартфона. Оскільки вся банківська безконтактна платіжна система складається з певної кількості незалежних паралельних мікросервісів, необхідно підбирати систему віддаленого виклику процедури для певного зв'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 нами була обрана друга через свою новизну, вищу ефективність та перспективність. Логічним кроком перед розробкою системи буде створення структурної схеми, оскільки проект 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 проекту належить сервіс обробки даних DMS, що переформатовує дані у зручну для зберігання або відправки банківській системі форму [8, с. 159-164].

При створенні елемента необхідно враховувати відступи країв з атрибутом `size_hint` і положення елемента в заданому наборі полів. Щоб запустити завершене вікно, виконайте метод `run` створеного вище класу `RegApp`, який успадковує батьківський клас `App`. Серверна частина системи розділена на чотири мікросервіси, які взаємодіють між собою за технологією gRPC. Використання цього методу вимагає створення протофайлу, який містить налаштування для передачі даних між службами.

Створивши для кожної взаємодії між сервісами такі конфігурації, отримуємо готове середовище для взаємодії сервісів. 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 проектних файлів у контейнер;
- 3) обробка залежностей проекту;
- 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 всіх служб запускається оркестратор служб, який надсилає команди запуску іншим службам. Крім того, ця служба перевіряє, чи встановлено з'єднання з усіма службами, і перезапускає службу, якщо з'єднання немає. Для створення будь-якої служби потрібно зв'язатися з докер-клієнтом і

передати йому такі параметри, як ім'я, зображення, налаштування доступу до мережі, режим, обмеження та точки монтування в параметрах функції `dclient.services.create`. Однак в рамках цієї роботи не вдалося розробити сервіси для банківської справи через відсутність загальнодоступних API для доступу до платіжних і 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 [8, с. 159-164].

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямку

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 розробки технології. Проект реалізовано на мові програмування Python з використанням технології віддаленого виклику процедур Google gRPC та фреймворку Kivy Project для взаємодії з операційною системою Android. Виберіть базу даних PostgreSQL. Під час аналізу створюються структурні діаграми, діаграми баз даних і схеми системного алгоритму програмного продукту.

Література

1. Падмапрія Н., Таміларасі К., Каніможі П., Кумар Т. А., Раджмохан Р. та Адеола А. С. Безпечна торгова система з використанням алгоритму віртуальної машини високого рівня (HLVM). Обчислювальна техніка : міжнародна конференція 2022 р. з інтелектуальних технологій і систем для наступного покоління (ICSTSN), 2022. С. 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. С. 143–146. DOI: 10.1109/SOLI.2008.4686380.
3. Перрі Т. С. Електронний банкінг виходить на ринок. IEEE Spectrum. том. 25. № 2. С. 46–49, лютий 1988. DOI: 10.1109/6.4510.
4. Гад А. Ф. NumPyCNNAndroid: бібліотека для прямої реалізації згорткових нейронних мереж для пристроїв Android. 2019 Міжнародна конференція з інноваційних тенденцій у комп'ютерній інженерії (ITCE), 2019. С. 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. P. 1199–1201. DOI: 10.1109/ASE51524.2021.9678749.
6. Кумар А. та Панда С. П. Опитування: як Python виступає в IT-світі. 2019 Міжнародна конференція з машинного навчання, великих даних, хмарних і паралельних обчислень (COMITCon), 2019. С. 248–251. DOI: 10.1109/COMITCon.2019.8862251.
7. Zhengdong X. and Tianming B. The implementation of flash-aware buffer replacement algorithms in PostgreSQL. 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2015. P. 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. IEEE 17th Conference on Business Informatics, 2015. P. 159–164. DOI: 10.1109/CBI.2015.22.

References

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 of intelligent technologies and systems for the next generation Computing technology (ICSTSN), 2022. P. 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. С. 143–146. DOI: 10.1109/SOLI.2008.4686380.
3. Perry T.S. Electronic Banking Takes the Market,” in IEEE Spectrum. volume. 25. No. 2. P. 46–49, February 1988. DOI: 10.1109/6.4510.
4. Gad A.F. NumPyCNNAndroid: a library for direct implementation of convolutional neural networks for Android devices. International Conference on Innovative Trends in Computer Engineering (ITCE), 2019. P. 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. P. 1199–1201. DOI: 10.1109/ASE51524.2021.9678749.
6. Kumar A. and Panda S. P. Survey: How Python Performs in the IT World. International Conference on Machine Learning, Big Data, Cloud and Concurrent Computing (COMITCon), 2019. P. 248–251. DOI: 10.1109/COMITCon.2019.8862251.
7. Zhengdong X. and Tianming B. The implementation of flash-aware buffer replacement algorithms in PostgreSQL. 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2015. P. 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. IEEE 17th Conference on Business Informatics, 2015. P. 159–164. DOI: 10.1109/CBI.2015.22.