

Virtualni novčanik beskontaktnih kartica

Gojanović, Jakov

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Šibenik University of Applied Sciences / Veleučilište u Šibeniku**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:143:395718>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-06**

Repository / Repozitorij:

[VUS REPOSITORY - Repozitorij završnih radova Veleučilišta u Šibeniku](#)



VELEUČILIŠTE U ŠIBENIKU
ODJEL POSLOVNE INFORMATIKE
PRIJEDIPLOMSKI STRUČNI STUDIJ POSLOVNE
INFORMATIKE

Jakov Gojanović

DIGITALNI NOVČANIK BESKONTAKTNIH
KARTICA

Završni rad

Šibenik, 2024.

VELEUČILIŠTE U ŠIBENIKU
ODJEL POSLOVNE INFORMATIKE
PRIJEDIPLOMSKI STRUČNI STUDIJ POSLOVNE
INFORMATIKE

DIGITALNI NOVČANIK BESKONTAKTNIH
KARTICA

Završni rad

Kolegij: Poslovna Informatika

Mentor: Marko Pavelić, mag. ing. inf. et comm. techn., pred.

Student: Jakov Gojanović

Matični broj studenta: 1219064598

Šibenik, rujan 2024.

DIGITALNI NOVČANIK BESKONTAKTNIH KARTICA

JAKOV GOJANOVIĆ

Matije Gupca 37, Šibenik; jgojanov@vus.hr

Ovaj rad bavi se analizom digitalnih novčanika za beskontaktnu karticu, tehnologijom koja omogućuje sigurno i praktično elektronsko plaćanje bez potrebe za fizičkim kontaktom. Digitalni novčanici postaju sve popularniji zahvaljujući svojoj jednostavnosti i brzini transakcija, a temeljni dio ovakvih sustava su beskontaktnu karticu bazirane primarno na NFC tehnologiji. Ova tehnologija omogućuje prijenos podataka između uređaja na malim udaljenostima te je posebno korisna u procesima plaćanja, identifikacije i autorizacije. Rad istražuje ključne aspekte digitalnog novčanika, uključujući programske jezike najčešće korištene u njihovoj kreaciji, načine sigurnosti, vrste elektroničkih plaćanja te EMV standard za beskontaktnu karticu. Praktični dio rada uključuje razvoj aplikacije za Android platformu koja koristi NFC za skeniranje beskontaktnih kartica, čitanje podataka u skladu s EMV standardima, te simulaciju NFC skenera. Zaključak rada ukazuje na značaj daljnjeg razvoja digitalnih novčanika, posebno u kontekstu sve većeg prihvaćanja beskontaktnih kartica u svakodnevnim transakcijama te njihov potencijal u oblikovanju budućnosti elektroničkog plaćanja.

(46 stranica / 19 slika / 2 tablice / 9 literaturnih navoda / jezik izvornika: hrvatski)

Rad je pohranjen u digitalnom repozitoriju Knjižnice Veleučilišta u Šibeniku

Ključne riječi: mobilna aplikacija, digitalni novčanik, elektronično plaćanje, NFC

Mentor: Marko Pavelić, mag. ing. inf. et comm. techn., pred.

Rad je prihvaćen za obranu dana: 30.09.2024.

BASIC DOCUMENTATION CARD

Polytechnic of Šibenik

Batchelor's Thesis

Department of

Undergraduate professional Studies of Business Informatics

DIGITAL WALLET OF CONTACTLESS CARDS

JAKOV GOJANOVIĆ

Matije Gupca 37, Šibenik; jgojanov@vus.hr

This paper deals with the analysis of digital wallets for contactless cards, a technology that enables safe and convenient electronic payment without the need for physical contact. Digital wallets are becoming more and more popular thanks to their simplicity and speed of transactions, and the fundamental part of such systems are contactless cards primarily based on NFC technology. This technology enables the transfer of data between devices over short distances and is particularly useful in payment, identification and authorization processes. The paper explores key aspects of digital wallets, including the programming languages most often used in their creation, security methods, types of electronic payments, and the EMV standard for contactless cards. The practical part of the work includes the development of an application for the Android platform that uses NFC to scan contactless cards, read data in accordance with EMV standards, and simulate an NFC scanner. The conclusion of the paper indicates the importance of the further development of digital wallets, especially in the context of the increasing acceptance of contactless cards in everyday transactions and their potential in shaping the future of electronic payments.

(46 pages / 19 figures / 2 tables / 9 references / original in Croatian language)

Thesis deposited in Polytechnic of Šibenik Library digital repository

Keywords: mobile application, digital wallet, electronic payment, NFC

Supervisor: Marko Pavelić, mag. ing. inf. et comm. techn., pred.

Paper accepted: 30.09.2024

SADRŽAJ

1. UVOD	1
2. OPERACIJSKI SUSTAV ZA MOBILNE UREĐAJE.....	2
2.1 Operacijski sustav Android	3
2.1.1 Prednosti	4
2.1.2 Nedostatci	5
2.1.3 Programski jezici za razvoj mobilnih aplikacija.....	6
2.2 Java	8
2.2.1. Prednosti	9
2.2.2. Nedostatci	10
2.3 Kotlin	11
2.3.1. Prednosti	12
2.3.2. Nedostatci	13
3. VRSTE ELEKTRONIČKOG PLAĆANJA	14
3.1 Plaćanje karticom	14
3.1.1 Kreditne kartice	15
3.1.2. Debitna kartica	16
3.1.3 Virtualne kartice	17
4. DIGITALNI NOVČANICI.....	21
4.1 Princip rada.....	21
4.2 Sigurnosne značajke	22
4.3 Tokenizacija	22
4.4 Plaćanje korištenjem QR kodova.....	23
4.4.1. Prednosti	24
4.4.2. Nedostatci	25
4.5 Plaćanje koristeći NFC.....	25
4.5.2 Nedostatci	28
4.6. MST.....	29
4.7 EMV	30
5. TESTIRANJE NFC TEHNOLOGIJE U PRIKUPLJANJU EMV PODATAKA	32
5.1 Postavljanje softvera (instalacija)	33
5.2 Implementacija NFC tehnologije	35
5.2.1 Manifest.....	35
5.2.2 MainActivity.....	36
5.2.3. NdefMessageParser	37

5.2.4. UriRecord.....	38
5.2.5 TextRecord.....	39
5.2.6 SmartPoster.....	40
5.2.7 Izgled aplikacije.....	41
6. ZAKLJUČAK.....	44
LITERATURA	45
PRILOZI	46

1. UVOD

Elektroničko plaćanje danas je postalo globalno prihvaćen način zamjene tradicionalnog plaćanja gotovinom. Za razliku od gotovine, elektroničko plaćanje pruža bolju sigurnost transakcija te uvodi novi koncept platnog sustava koji omogućuje trgovcima personalizaciju financijskih usluga. Ipak, s razvojem elektroničkih sustava plaćanja, i kriminalci su se prilagodili kako bi iskoristili njihove slabosti. Taj jednostavan koncept, koji je zahtijevao samo provlačenje kartice kroz naplatni uređaj, omogućavao je neovlaštenim osobama da koriste karticu koju su pronašli ili ukrali. Sofisticiraniji prevaranti razvili su metodu poznatu kao "skimming"¹, u kojoj su mogli skenirati karticu vlastitim čitačem kako bi lako i bez puno truda ukrali podatke vlasnika kartice. Ova nova prijetnja, zajedno s potrebom za standardiziranim sustavom plaćanja, postala je globalni problem. Kao odgovor na to, 1994. godine tri međunarodne korporacije – Europay, Mastercard i Visa – započele su razvoj globalne specifikacije čipova za platne sustave. Njihov trud je na kraju doveo do stvaranja novog standarda plaćanja, poznatog kao EMV. EMV tehnologija uvedena je kako bi unaprijedila sigurnost elektroničkih platnih transakcija, kombinirajući napredne hardverske zaštitne mjere i sofisticirane softverske mehanizme koji štite od neovlaštene upotrebe kreditnih i debitnih kartica. Ovaj jednostavan i siguran način plaćanja dodatno je evoluirao s pojavom digitalnih novčanika. Od 1999. godine, digitalni novčanici su se globalno proširili zahvaljujući svojoj jednostavnosti, praktičnosti i fleksibilnosti. Te su karakteristike, uz izazove koje je donijelo razdoblje pandemije COVID-19 od 2020. do 2023. godine, učinile digitalne novčanike ključnim alatom današnjeg potrošača.

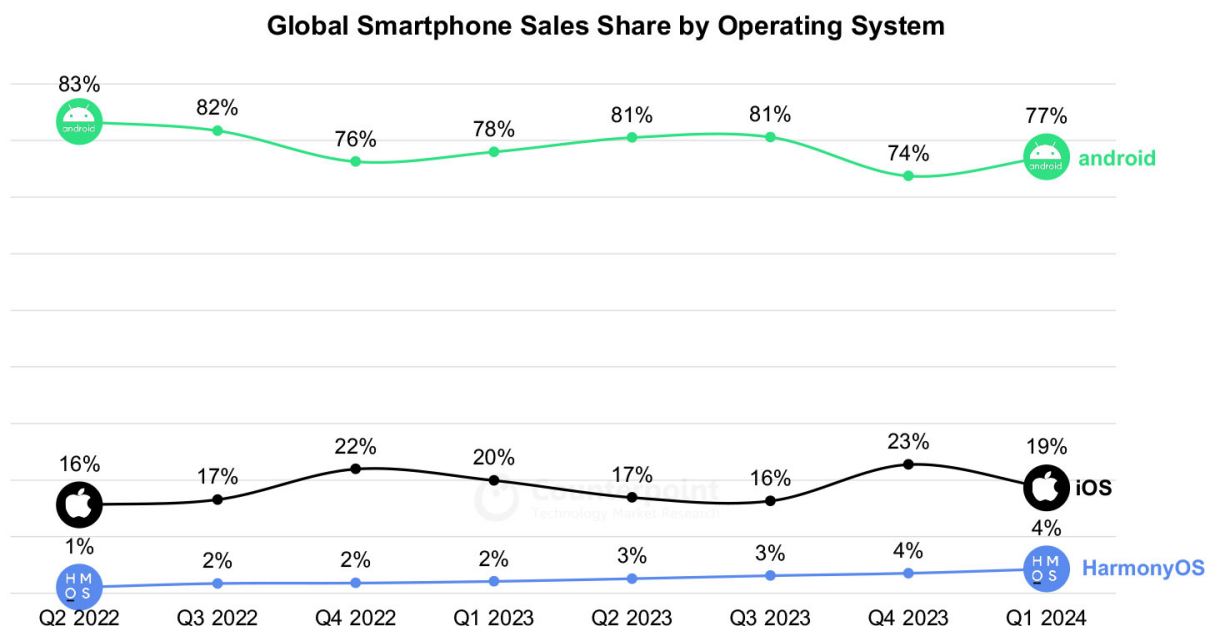
Ovaj rad analizira osnovne karakteristike digitalnog novčanika, uključujući EMV standard, NFC tehnologiju i tokenizaciju podataka. Također se istražuju sigurnosni protokoli koji se koriste za zaštitu podataka tijekom prijenosa, proces rada digitalnog novčanika, sigurnost i prednosti beskontaktnih kartica u odnosu na prave, te ranjivosti beskontaktnih plaćanja i virtualnih novčanika.

¹ Skimming je financijska prijevarena koja uključuje nezakonito hvatanje podataka o kartici žrtve pomoću uređaja poznatog kao skimmer. Nakon što prevarant preleti podatke o žrtvinoj kartici, može ih upotrijebiti za izradu kloniranih kartica ili ih koristiti za online transakcije.

2. OPERACIJSKI SUSTAV ZA MOBILNE UREĐAJE

Operacijski sustav (engl. *Operating System, OS*) predstavlja potpuno integrirani skup specijaliziranih programa koji upravljaju svim operacijama računala. On kontrolira i nadzire izvršavanje svih drugih programa unutar računala, uključujući aplikacijske i sustavne softvere. Osim toga, operacijski sustav upravlja hardverskim resursima i pruža zajedničke usluge računalnim programima. Kao najvažnija vrsta sistemskog softvera, operacijski sustav omogućava računalima da obavljaju osnovne funkcije kao što su unos, pohrana, obrada i ispis podataka. Ovi sustavi također postoje na mobilnim uređajima, a tri glavna operacijska sustava koja dominiraju globalnim tržištem su operacijski sustav Android, operacijski sustav iOS i operacijski sustav Harmony.

Najpopularniji mobilni operacijski sustav na svijetu je operacijski sustav Android, koji zauzima 77% mobilnog tržišta. Iza njega slijedi operacijski sustav iOS s 19%, dok operacijski sustav HarmonyOS tvrtke Huawei drži 4% tržišnog udjela. Ovi podaci prikazani su na slici 1, koja ilustrira globalni udio prodaje pametnih telefona prema njihovim operacijskim sustavima.



Slika 1. Globalni udio u prodaji pametnih telefona, Izvor:

www.counterpointresearch.com

2.1 Operacijski sustav Android



Slika 2. Operacijski sustav Android, Logo, Izvor: ausdroid.net

Android OS je mobilni operacijski sustav koji je Google razvio 2005. godine, prvenstveno za uređaje sa zaslonom osjetljivim na dodir, mobilne telefone i tablete. Njegov dizajn omogućuje korisnicima intuitivno upravljanje uređajima pokretima prstiju koji odgovaraju uobičajenim radnjama poput stiskanja, povlačenja i dodirivanja zaslona. Google također implementira Android OS softver u televizorima, automobilima i ručnim satovima, pri čemu svaki uređaj ima jedinstveno korisničko sučelje. U svojoj srži, Android OS funkcionira kao softverski okvir koji korisnicima omogućava interakciju s uređajima, pristup raznim uslugama i nesmetano pokretanje različitih aplikacija. Njegova arhitektura je modularna i prilagodljiva, što omogućuje proizvođačima mobilnih uređaja i razvojnim programerima da prilagode korisničko iskustvo specifičnim hardverskim konfiguracijama i preferencijama korisnika.

2.1.1 Prednosti

Operacijski sustav Android predstavlja izvrstan primjer platforme otvorenog koda (engl. *open-source platform*)², licencirane pod Apache licencom³. Ova platforma nudi široko i prilagodljivo razvojno okruženje, omogućujući gotovo svakome da doprinese stvaranju aplikacija te rastu i razvoju cijelog Android OS ekosustava. Programerima je od posebne važnosti što imaju potpuni i neometan pristup izvornom kodu, što im pruža slobodu da prilagode svoje aplikacije specifičnim zahtjevima i potrebama korisnika. Nasuprot tome, operacijski sustav iOS postavlja stroga ograničenja i uvjete za pristup svom izvornom kodu, dok operacijski sustav Android potiče inovacije i prilagodbe, omogućujući programerima širu kreativnu slobodu. Dodatno, operacijski sustav Android programerima pruža iznimno visoku razinu slobode. Ne postoje suvišna ograničenja niti potrebe za dugotrajnim odobrenjima od strane operatera ili posrednika. Ovakvo oslobađajuće razvojno okruženje omogućuje programerima da bez prepreka razvijaju aplikacije raznih vrsta i funkcionalnosti, što operacijski sustav Android čini posebno konkurentnim u odnosu na druge operacijske sustave.

Jedna od ključnih karakteristika operacijskog sustava Android je mogućnost jednostavnog preuzimanja i instaliranja third-party widgeta, što korisnicima omogućuje personalizaciju početnih zaslona. Ovi widgeti pružaju različite korisne informacije, od vremenskih prognoza do vijesti i drugih funkcionalnosti, što dodatno doprinosi prilagodbi uređaja osobnim potrebama korisnika. Velika prednost Android OS ekosustava je i snažna podrška globalne zajednice. Operacijski sustav Android kontinuirano potiče razvoj aktivne zajednice programera diljem svijeta. Ta zajednica je vrlo angažirana te se temelji na opsežnoj suradnji i razmjeni povratnih informacija, što dodatno unapređuje sustav. Forumi, događanja, sastanci i radionice doprinose neprekidnom razvoju i inovacijama unutar Android OS zajednice, čime se osigurava stalan napredak koji koristi i programerima i korisnicima širom svijeta.

² Platforma otvorenog koda je svaka platforma koja dopušta pristup svom izvornom kodu svim drugim korisnicima ili programerima.

³ Apache licenca je licenca slobodnog softvera koju je napisala Apache Software Foundation. Omogućuje korisnicima korištenje softvera u bilo koju svrhu, njegovu distribuciju, modificiranje i distribuciju izmijenjenih verzija softvera prema uvjetima licence, bez brige o autorskim honorarima














2.1.2 Nedostaci

Aplikacije operacijskog sustava Android često rade u pozadini, trošeći značajne resurse uređaja, uključujući bateriju. Zaustavljanje ovih aplikacija može biti mukotrpan proces, osobito zato što su mnoge povezane sa sustavom i stalno aktivne. Kako bi smanjili potrošnju baterije, korisnici često pribjegavaju vanjskim aplikacijama za optimizaciju performansi, ali takve aplikacije mogu povećati potrošnju podataka, što nije uvijek poželjno. Uz to, operacijski sustav Android zahtijeva značajne resurse, što može dovesti do slabijih performansi na uređajima s nižim hardverskim specifikacijama. Ovi uređaji često rade sporije, a unaprijed instalirane aplikacije dodatno otežavaju situaciju, zauzimajući prostor za pohranu i smanjujući učinkovitost uređaja, što može rezultirati u pregrijavanju uređaja.

Povrh toga, mnogi korisnici smatraju operacijski sustav Android složenijim u usporedbi s drugim platformama, osobito za one koji nemaju mnogo iskustva s tehnologijom. Sustav nije uvijek intuitivan, a konfiguracija može predstavljati izazov, što može dovesti do neočekivanih problema tijekom prilagodbe uređaja. Kako bi se izbjegli takvi problemi, korisnici moraju imati određene tehničke vještine. Osim toga, za razliku od nekih drugih operativnih sustava, Android ne dolazi s ugrađenom zaštitom od virusa. Korisnici su stoga prisiljeni sami preuzeti i instalirati antivirusne programe kako bi osigurali sigurnost svojih uređaja i zaštitili ih od potencijalnih prijetnji.

2.1.3 Programski jezici za razvoj mobilnih aplikacija

Programski jezik predstavlja skup uputa koje programer piše kako bi dao naredbe računalu za izvršavanje određenog zadatka. Ove upute obično se prikazuju kao nerazumljiv kod, strukturiran prema specifičnoj sintaksi određenog programskog jezika. U današnje vrijeme postoji mnogo programskih jezika koji se mogu koristiti za razvoj mobilnih aplikacija, a izbor jezika ovisi o poslovnim ciljevima korisnika, mogućnostima odabranog operacijskog sustava, vrsti aplikacije koju korisnik želi stvoriti i zahtjevima same aplikacije. Programski jezici poput Objective-C i programskog jezika Swift koriste se za razvoj softvera za uređaje operacijskog sustava iOS; ovi jezici su objektno orijentirani i omogućuju developerima grupiranje sličnih radnji u jednu cjelinu tijekom kodiranja, što proces razvoja front-end-a mobilne aplikacije čini bržim i jednostavnijim. Za operacijski sustav Android češće se koriste programski jezik Java i programski jezik Kotlin, ali prisutni su i drugi programski jezici kao što su Python, C/C++, PHP i programski jezik JavaScript.

Nov 2021	Nov 2020	Change	Programming Language	Ratings	Change
1	2	▲	 Python	11.77%	-0.35%
2	1	▼	 C	10.72%	-5.49%
3	3		 Java	10.72%	-0.96%
4	4		 C++	8.28%	+0.69%
5	5		 C#	6.06%	+1.39%
6	6		 Visual Basic	5.72%	+1.72%
7	7		 JavaScript	2.66%	+0.63%
8	16	▲	 Assembly language	2.52%	+1.35%
9	10	▲	 SQL	2.11%	+0.58%
10	8	▼	 PHP	1.81%	+0.02%
11	21	▲	 Classic Visual Basic	1.56%	+0.83%
12	11	▼	 Groovy	1.51%	-0.00%
13	15	▲	 Ruby	1.43%	+0.22%

Slika 3. Pet najpopularnijih jezika za razvoj mobilnih aplikacija, Izvor: [fireart.studio](https://www.fireart.studio)

Danas se mnoge tvrtke suočavaju s izazovom razvoja mobilnih aplikacija za više platformi, posebno za operacijski sustav Android i operacijski sustav iOS. Zbog toga su cross-platform aplikacije postale jedan od najpopularnijih trendova u razvoju softvera. Ove aplikacije su dizajnirane da rade identično na različitim mobilnim platformama. Međuplatformski okviri (engl. *cross-platform frameworks*)⁴ omogućuju developerima pisanje koda koji se može dijeliti i ponovno koristiti, što olakšava razvoj takvih aplikacija. Ovaj pristup nudi nekoliko prednosti, uključujući učinkovitost u smislu vremena i troškova. Tvrtke koje žele osvojiti tržište aplikacija za operacijski sustav Android i operacijski sustav iOS često se odlučuju za korištenje višeplatformskih tehnologija koje omogućuju korištenje samo jedne baze koda za sve verzije aplikacija.

Tablica 1: Prednosti najpopularnijih programskih jezika

Programski jezik	Prednosti	Primjena
Kotlin	Sažetost	Razvoj Android aplikacija
Java	Podrška zajednice	Razvoj Android aplikacija
JavaScript	Interaktivnost	Razvoj web aplikacija
Ruby	Elegantan izgled koda	Razvoj web aplikacija
Python	Jednostavnost, svestranost	Strojno učenje

⁴ Cross-platform framework je okruženje za razvoj softvera koje programerima omogućuje stvaranje aplikacija koje se mogu izvoditi na više operativnih sustava (kao što su Windows, macOS, iOS i Android OS) koristeći jednu bazu koda.

2.2 Java



Slika 4. Android Java Logo, Izvor: theiconic.tech

Java je programski jezik i računalna platforma koju je prvi put objavio Sun Microsystems 1995. godine. Nastao iz skromnih početaka, ubrzo je postao jedan od najvažnijih programskih jezika u pokretanju velikog dijela današnjeg digitalnog svijeta, pružajući pouzdanu platformu na kojoj su izgrađene mnoge usluge i aplikacije. Više od dva desetljeća kasnije, inovativni proizvođači i digitalne usluge dizajnirane za budućnost i dalje se oslanjaju na ovaj programski jezik.

Programski jezik Java se koristi za razvoj aplikacija koje izvršavaju različite zadatke na različitim platformama. Aplikacije razvijene u Java programskom okruženju mogu raditi na desktop računalima, serverima, mobilnim telefonima, tablet uređajima, TV uređajima i mnogim drugima. Jedna od najvažnijih značajki Java programskog jezika je njegova neovisnost o platformi, što omogućuje izvršavanje koda napisanog u ovom programskom jeziku bez promjena u različitim okruženjima. Kod programskog jezika Java se prevodi u bytecode koji pokreće operacijski sustav, drugi program ili uređaji pomoću posebnog prevoditelja za programski jezik Javu. Bytecode nije izvršni kod, već visoko optimizirani skup instrukcija dizajniran za izvođenje unutar izvršnog sustava Java programskog jezika, poznatog kao Java virtualni stroj (JVM), koji služi kao prevoditelj za bytecode.

2.2.1. Prednosti

Java je programski jezik koji se ističe po poboljšanim sigurnosnim značajkama, što ga čini omiljenim alatom za razvoj aplikacija visokog standarda. Među ključnim sigurnosnim elementima programskog jezika Java su enkapsulacija, nasljedstvo i apstrakcija. Enkapsulacija omogućuje skrivanje detalja implementacije unutar klase, dok nasljedstvo omogućuje stvaranje novih klasa iz postojećih, čime se postiže fleksibilnost. Apstrakcija dodatno olakšava rad skrivanjem složenih detalja objekata od krajnjih korisnika. Ove karakteristike, u kombinaciji sa strogim sigurnosnim pravilima ovog programskog jezika, stvaraju visoku razinu zaštite aplikacija.

Osim sigurnosnih prednosti, programski jezik Java se oslanja na objektno orijentirano programiranje (engl. *Object-oriented programming, OOP*), što omogućuje bolju organizaciju i održavanje koda. Korištenjem objektno orijentiranog programiranja, programeri definiraju objekte s pripadajućim strukturama podataka i funkcijama koje oponašaju od objekata iz stvarnog svijeta. Ovaj pristup pojednostavljuje rad na kompleksnim projektima, jer omogućuje podjelu koda na manje, preglednije dijelove. Za razliku od proceduralnog programiranja (engl. *Procedural programming*), koje se oslanja na niz uputa i može biti zamorno za održavanje, objektno orijentirano programiranje čini kod skalabilnijim i lakšim za razumijevanje. Unatoč tome što je visokoapstraktni programski jezik, programski jezik Java se može pohvaliti relativno jednostavnom sintaksom, zbog čega proces učenja nije dugotrajan. Ova dostupnost olakšava rad početnicima, ali i iskusnim programerima te im omogućuje da se brzo prilagode i usvoje vještine potrebne za rad u ovom programskom jeziku. U usporedbi s drugim programskim jezicima koji imaju složenije strukture, programski jezik Java nudi jednostavnost koja ne zahtijeva previše napora.

Veliku prednost programskom jeziku Java donosi i njezina aktivna i široka zajednica programera. Ova globalna mreža kontinuirano doprinosi razvoju ovog programskog jezika kroz projekte, biblioteke, alate za razvoj softvera i pružanje podrške drugim korisnicima. Forumi, tutorijali i drugi online resursi osiguravaju podršku kako za nove korisnike, tako i za one iskusne, čineći programski jezik Java jednim od najdostupnijih i najbolje podržanih jezika za programiranje.

2.2.2. Nedostatci

U usporedbi s drugim programskim jezicima, kao što je C++, performanse programskog jezika Java mogu biti inferiorne zbog korištenja virtualnog stroja i načina upravljanja memorijom. Faktori poput sakupljanja smeća, nepravilno konfigurirane predmemorije i pojave zastoja mogu negativno utjecati na ukupne performanse aplikacija razvijenih u ovom programskom jeziku, osobito ako se tim aspektima ne pristupa pravilno. Također, procesi interpretacije i Točno-Na-Vrijeme (engl. *Just-In-Time, JIT*) kompilacije mogu dovesti do problema s performansama, jer zahtijevaju dodatno vrijeme za interpretaciju i sastavljanje koda u trenutku izvođenja. Pored toga, aplikacije napravljene u programskom jeziku Java često troše više memorije u odnosu na one razvijene u programskim jezicima poput C ili C++, što može predstavljati izazov za uređaje s ograničenim resursima. Kada je potrebno optimizirati potrošnju memorije, programeri moraju biti svjesni da aplikacije programskog jezika Java općenito zahtijevaju više memorijskih resursa, što može postati problem u specifičnim okolnostima. Nadalje, programski jezik Java je poznat po opširnoj sintaksi koja zahtijeva pisanje velikog broja redaka koda čak i za jednostavne programe. Programski jezik Java se usredotočuje na to da bude kompaktan, ali u isto vrijeme, mora kompromizirati s veoma složenim kodom i dugim objašnjenjima za svaki pojam (TechVidvan). Ova složenost prisiljava programere da tijekom razvoja koriste kompleksne sintaktičke strukture, što može usporiti proces izrade i povećati napore potrebne za održavanje koda.

2.3 Kotlin



Slika 5. Android Kotlin Logo, Izvor: developer.android.com

Programski jezik Kotlin je besplatni open source programski jezik opće namjene, prvotno dizajniran za JVM (Java Virtual Machine) i operacijski sustav Android. Kombinira značajke objektno orijentiranog i funkcionalnog programiranja, s posebnim naglaskom na interoperabilnost, sigurnost i jasnoću. Razvijen je u JetBrainsu, tvrtki poznatoj po IntelliJ IDEA, 2010. godine, a javno je dostupan od 2012. godine. Budući da je većina njihovih proizvoda razvijena u programskom jeziku Java, odlučili su izgraditi programski jezik temeljen na programskom jeziku Java koji je sažetiji, istovremeno uvodeći nove konstrukcije kao što su funkcije visokog reda engl. *high-order functions*) (Walker, 2022.). Kao statički tipiziran jezik (engl. *Statically typed*), zahtijeva deklaraciju svih varijabli prije korištenja. Glavna prednost ovog programskog jezika leži u zaštiti od pogrešaka prilikom izvođenja programa i bržem vremenu izvršenja, jer prevoditelj zna točne tipove podataka i može generirati optimizirani strojni kod.

Iako je programski jezik Kotlin prvotno namijenjen za rad s programskim jezikom Java, njegov kod se može prevesti i u programski jezik JavaScript. Osim toga, izvorni programski jezik Kotlin omogućuje programerima kompajliranje koda ovog programskog jezika u izvorne binarne datoteke stvorene za izvođenje na određenom operacijskom sustavu, koje se mogu izvoditi bez JVM-a. Kotlinove izvorne binarne datoteke mogu se izvoditi na različitim platformama, uključujući operacijski sustav macOS, operacijski sustav iOS, operacijski sustav tvOS, operacijski sustav watchOS, operacijski sustav Linux, operacijski sustav Windows, te u razvojnom setu alata operacijskog sustava Android poznatom kao NDK (Android Native Development Kit). Na kraju, Kotlin proizvodi bytecode za programski jezik Javu, kod kompatibilan s programskim jezikom JavaScriptom ili izvorne binarne datoteke temeljene na ciljnoj platformi.

2.3.1. Prednosti

Programski jezik Kotlin podržava većina integriranih razvojnih okruženja (engl. *Integrated development environment, IDE*), uključujući Android Studio i razne druge SDK alate. Ova podrška pomaže povećati produktivnost programera, jer mogu raditi s ovim programskim jezikom koristeći alate s kojima su već upoznati i navikli raditi. Kod ovog programskog jezika je jasniji i kompaktniji od Java programskog jezika, što pomaže programerima da osiguraju stabilnost proizvodnog koda. Greške se otkrivaju tijekom faze kompajliranja, omogućujući programerima da isprave sve pogreške prije nego što se program izvrši. Jetpack Compose je moćan skup alata dostupan isključivo putem programskog jezika Kotlin. Nudi deklarativni način za učinkovito stvaranje korisničkog sučelja za aplikacije na operacijskom sustavu Android i drugim platformama, smanjujući količinu standardnog koda; baza koda se razvija na jednom jeziku bez potrebe za XML-om. Osim toga, upravljanje i ažuriranje korisničkog sučelja postaje lakše i pouzdanije.



Slika 6. Jetpack Compose Logo, Izvor: medium.com

2.3.2. Nedostatci

Programski jezik Kotlin se i dalje dekompilira u programski jezik Java, a njegove jezične značajke temelje se na onome što je dostupno u trenutnoj verziji JVM-a koju podržava operacijski sustav Android. Budući da operacijski sustav Android ne koristi najnoviju iteraciju programskog jezika Java, Kotlin programeri često moraju pronalaziti zaobilazna rješenja za implementaciju određenih značajki. Iako je programski jezik Kotlin stekao popularnost u razvoju aplikacija za operacijski sustav Android i backend sustave u suradnji s okvirima poput Spring Boota, njegova primjena u drugim domenama, poput poslovnog softvera ili naslijeđenih sustava, može biti ograničenija. To bi moglo utjecati na dostupnost radnih mjesta povezanih s ovim programskim jezikom ili na podršku zajednice koja se fokusira na ta područja programiranja. Vrijeme kompilacije u ovom programskom jeziku može biti sporije u usporedbi s drugim programskim jezicima poput programskog jezika Java, posebno za veće projekte. Iako ovo možda nije značajan problem za većinu aplikacija, može utjecati na tijek rada u razvoju određenih projekata, osobito tijekom učestalih ciklusa razvoja.

3. VRSTE ELEKTRONIČKOG PLAĆANJA

E-plaćanja ili digitalna plaćanja odnose se na online transakcije između platitelja i primatelja koristeći digitalne ili elektroničke metode plaćanja. Ove vrste plaćanja olakšavaju trenutne prijenose novca iz udobnosti doma platitelja, čineći ih jednostavnim i praktičnim. Razne vrste elektroničkih sustava plaćanja, uključujući debitne i kreditne kartice, obično se koriste za online transakcije. Virtualne platne kartice nude siguran način plaćanja bez dijeljenja osobnih podataka. Digitalni novčanici pružaju praktičnost pohranjivanjem podataka o plaćanju na mobilni uređaj.

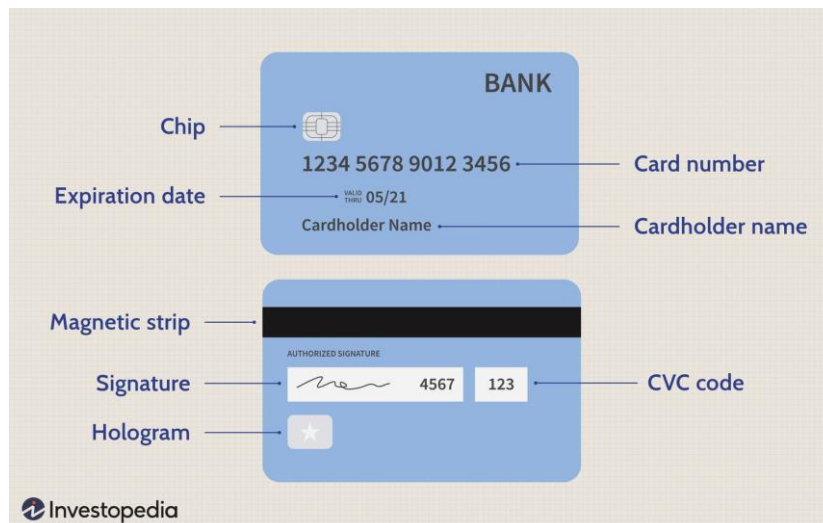
3.1 Plaćanje karticom

Kartično plaćanje je način bezgotovinskog plaćanja pri kojem kupci svoje transakcije izvršavaju svojom debitnom ili kreditnom karticom, bilo putem prodajnog mjesta (engl. *Point of Sale, POS*) ili online. Iznos se tereti izravno s njihovog bankovnog računa ili ako plaćaju kreditnom karticom, bit će im terećeno kasnije. Većina tvrtki prihvaća kartično plaćanje jer je brzo i jednostavno te ubrzava proces plaćanja. Postoji više vrsta platnih kartica, a najčešće su kreditne kartice, debitne kartice ili prepaid kartice. Vrlo često, platna kartica je elektronički povezana s računom ili računima vlasnika kartice. Ti računi mogu biti depozitni računi, zajmovi ili kreditni računi, a kartica je sredstvo za provjeru autentičnosti vlasnika kartice. Međutim, stored-value kartice pohranjuju novac na samoj kartici i nisu nužno povezane s vlasnikovim računom u financijskoj instituciji. Također postoje i pametne kartice koje sadrže svoj jedinstveni broj i neke sigurnosne informacije kao što je datum isteka ili s magnetskom trakom na svojoj poleđini, što različitim strojevima omogućuje čitanje i pristup informacijama. Ovisno o banci, izdavatelju i preferencijama klijenta, to može dopustiti da se kartica koristi kao ATM kartica, omogućavajući transakcije na bankomatima, ili kao debitna kartica, povezana s bankovnim računom klijenta i njome se može kupovati na prodajnom mjestu, ili kao kreditna kartica povezana s revolving kreditnom linijom koju izdaje banka.

Za plaćanje karticom putem POS-a, korisnik mora umetnuti karticu u čitač ili dodirnuti čitač s karticom. Čitač zatim čita sve podatke na kartici i šalje ih procesoru plaćanja. Procesor plaćanja obrađuje zahtjev s čvorišta (engl. *gateway*), autorizira transakciju i provjerava ima li kupac dovoljno novca na bankovnom računu. Ako je transakcija odobrena, novac se prenosi s računa kupca na račun tvrtke. Ovaj proces obično traje samo nekoliko sekundi. Platne kartice su obično plastične kartice, veličine 85,60 mm × 53,98 mm i zaobljenih kutova radijusa od 2,88–3,48 mm, u skladu s ISO/IEC 7810⁵ standardom. One obično imaju jedinstveni broj koji se izdaje u skladu sa standardom označavanja brojeva ISO/IEC 7812⁶, ime vlasnika kartice i datum isteka kartice, uz ostale sigurnosne značajke.

3.1.1 Kreditne kartice

Kreditna kartica je tanki pravokutni komad plastike ili metala izdan od strane banke ili tvrtke za financijske usluge. Ona omogućuje vlasniku kartice posudbu sredstava za plaćanje robe i usluga kod trgovaca koji prihvaćaju kartice kao način plaćanja. Kreditne kartice postavljaju uvjet da vlasnici kartice vrate posuđeni novac, plus sve primjenjive kamate, kao i sve dodatne dogovorene naknade, bilo u cijelosti do datuma naplate ili tijekom vremena.



Slika 7. Kreditna kartica, Izvor: www.investopedia.com

⁵ Međunarodna norma koja definira fizičke karakteristike za identifikacijske kartice.

⁶ Međunarodna norma koja definira sustav numeriranja za identifikaciju izdavatelja kartice, format identifikacijskog broja izdavatelja (engl. *Identification Number, IN*) i primarnog broja računa (engl. *Primary account number, PAN*)

Osim standardne kreditne linije, izdavatelj kreditne kartice također može odobriti zasebnu gotovinsku kreditnu liniju (LOC) vlasniku kartice, omogućujući mu da posuđuje novac u obliku gotovinskih predujmova kojima se može pristupiti putem bankovnih šaltera, bankomata ili kreditnih čekova. Takvi gotovinski predujmovi obično imaju drugačije uvjete, kao što su ukidanje razdoblja početka i više kamatne stope, u usporedbi s onim transakcijama koje pristupaju glavnoj kreditnoj liniji. Izdavatelji obično unaprijed postavljaju ograničenja zaduživanja na temelju kreditnog rejtinga pojedinca. Velika većina tvrtki dopušta kupcu plaćanje kreditnim karticama, koje su i danas jedna od najpopularnijih metoda plaćanja za kupnju potrošačkih dobara i usluga.

3.1.2. Debitna kartica

Debitna kartica je platna kartica koja skida novac izravno s tekućeg računa korisnika. Poznate su i pod nazivom "bankomat-kartica". Debitne kartice se mogu koristiti za kupnju robe ili usluga ili za podizanje gotovine preko bankomata. Debitne kartice služe da smanje korisnikovu potrebu za nošenjem gotovine, iako korištenje ovih kartica ponekad može zahtijevati određene naknade. Iako debitna kartica izgleda slično kreditnoj, njezin način rada se uvelike razlikuje. Iznos novca koji korisnik može potrošiti koristeći debitnu karticu određen je iznosom sredstava na njihovom računu, a ne kreditnim limitom s kojim dolaze kreditne kartice. Debitna kartica može biti elektronički povezana s korisnikovim računom ili može biti izvanmrežna kartica. Izvanmrežnim karticama potrebno je više vremena za obradu transakcija.

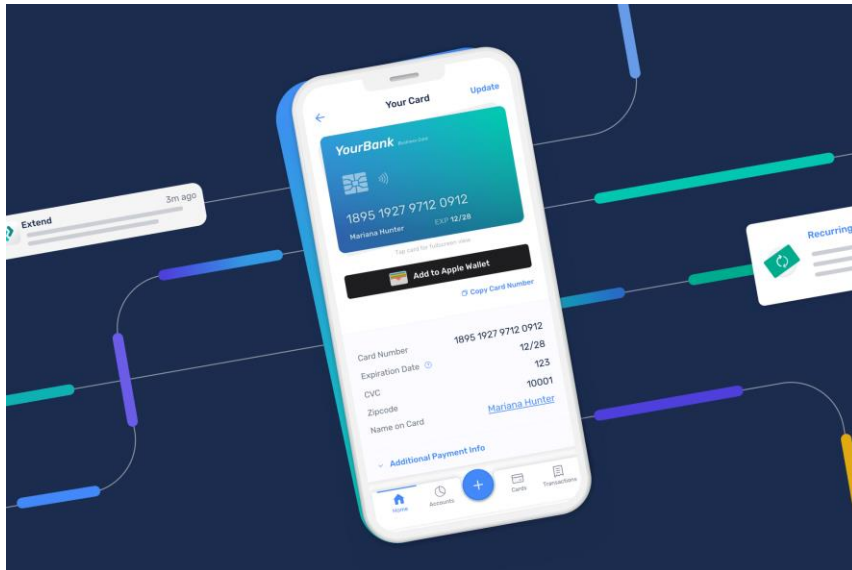


Slika 8. Debitna kartica, Izvor: m2pfintech.com

Za razliku od kreditne kartice, korisnik ne ulazi u dugove kada koristi ovu vrstu kartice jer se ona koristi za pristup sredstvima koja korisnik već posjeduje. Nije potrebno plaćati minimalne mjesečne uplate na debitnu karticu zato što nema duga koji se treba otplatiti. Debitne kartice odmah povlače sredstva s pridruženog računa.

3.1.3 Virtualne kartice

Virtualne kartice funkcioniraju slično tradicionalnim kreditnim karticama, ali to rade u obliku jednokratnog 16-znamenkastog broja i troznamenkastog CVV koda generiranog online, umjesto plastične ili metalne kartice koja se mora prisloniti POS uređaju. Virtualne kartice prihvaćaju se svugdje gdje se koriste tradicionalne fizičke kartice, ali dolaze s nizom značajki, kontrola i pogodnosti koje ih čine jedinstveno vrijednima za tvrtke. Oni mogu igrati ključnu ulogu u racionalizaciji načina na koji se obavljaju poslovna plaćanja dobavljačima ili prodavačima. Uklapajući se u postojeće strategije plaćanja, virtualne kartice proširuju rješenja obrtnog kapitala na krajnje dobavljače koji nemaju tendenciju prihvaćanja financiranja opskrbnog lanca ili dinamičkog diskontiranja. One također nude značajno poboljšane sigurnosne kontrole za trgovce, minimizirajući rizik od krivotvorenih transakcija i nepovratnih troškova. Koncept virtualnih kartica nije posebno nov, s nizom institucionalnih banaka koje već proizvode virtualne kartice. Međutim, novija virtualna kartična rješenja koja se besprijekorno integriraju s postojećim centraliziranim ERP sustavima i sustavima plaćanja nude znatno veću vrijednost tvrtkama koje žele poboljšati svoju sposobnost učinkovitijeg upravljanja obrtnim kapitalom.



Slika 9. Virtualna kartica, Izvor: taulia.com

Za razliku od tradicionalnih kreditnih kartica, koje imaju jedan 16-znamenkasti broj kartice i CVV kod koji se više puta koriste do datuma isteka kartice, virtualne kreditne kartice generiraju se digitalno za svaku pojedinačnu kupnju. U tipičnom postupku plaćanja neke usluge ili proizvoda, one djeluju slično tradicionalnim kreditnim karticama, uz nekoliko ključnih razlika:

1. Nakon odobrenja fakture, automatizirani ERP sustav šalje podatke o potrebnom plaćanju korisnika virtualne kartice
2. Izrađuje se račun za jednokratnu upotrebu s potrebnim dozvolama, a detalji računa virtualne kartice šalju se trgovcu kojem se plaća
3. Trgovac može unijeti podatke o računu virtualne kartice u svoj POS, a njegova banka to gotovo trenutačno autorizira, čime su sredstva dostupna na njegovom računu
4. Podaci o namiri uspoređuju se s uputama za plaćanje, a usklađivanje na strani kupca provodi se automatski u ERP sustavu
5. Platelj plaća banci koja je izdala virtualni kredit u skladu s uvjetima virtualne kartice, čuvajući obrtni kapital koji bi inače dospio na datum fakture

Nakon što se plaćanje virtualnom karticom izvrši gore opisanim postupkom, povezani račun više ne postoji. Podaci o računu više se ne mogu koristiti, a buduća plaćanja uključivat će stvaranje potpuno novog skupa podataka o virtualnoj kartici.

3.1.3.1 Prednosti virtualnih kartica

Virtualne kartice pružaju veću razinu sigurnosti u usporedbi s fizičkim kreditnim karticama jer se ne mogu izgubiti, zametnuti ili ukrasti. Također, smanjuju rizik od prijevara i neovlaštenog korištenja, što ih čini sigurnijom opcijom za korisnike. Jedna od glavnih prednosti je i to što se može koristiti bez potrebe za čekanjem da ona bude dostavljena korisniku – čim izdavatelj kartice generira virtualnu kreditnu karticu, ona je odmah spremna za korištenje. To je mnogo praktičnije od fizičkih kartica i uštedi vrijeme korisniku.

Virtualne kartice također olakšavaju praćenje troškova, jer korisnik može brzo pratiti i kategorizirati potrošnju zahvaljujući tome što su svi podaci o transakcijama dostupni online. Ta transparentnost omogućuje bolju kontrolu financija. Konačno, virtualne kartice su izuzetno jednostavne za korištenje – mogu se lako dodati u digitalni novčanik, kao što su Google Pay ili Apple Pay, ili koristiti za online plaćanje bez potrebe za fizičkom karticom.

3.1.3.2 Nedostatci virtualnih kartica

Virtualne kartice imaju određena ograničenja u svojoj upotrebi, jer se mogu koristiti isključivo za online transakcije. Za razliku od fizičkih kartica, ne mogu se koristiti za kupnju u trgovinama ili za podizanje gotovine na bankomatima, što može ograničiti njihovu funkcionalnost u određenim situacijama. Osim toga, virtualne debitne kartice nisu povezane s korisničkim računom na isti način kao fizičke kartice. Dok se kod fizičkih debitnih kartica sredstva izravno preuzimaju s bankovnog računa korisnika, kod instant virtualnih debitnih kartica sredstva dolaze s prepaid računa koji se unaprijed nadplaćuje. Time je osigurano da korisnik ne može potrošiti više nego što ima na prepaid računu, što može dovesti do neugodnosti ako se korisnik nalazi u situaciji gdje cijena proizvoda ili usluge nadmašuje maksimalnu dopuštenu količinu na računu.

4. DIGITALNI NOVČANICI

Digitalni novčanik, također poznat kao e-novčanik ili mobilni novčanik, je softverski sustav ili aplikacija koja radi na bilo kojem uređaju povezanom na internet koji jednoj strani omogućuje elektroničke transakcije s drugom stranom razmjenu jedinica digitalne valute za robu ili usluge. Digitalni novčanici rade primarno na mobilnim uređajima, ali im se može pristupiti i s računala. Jedan podskup digitalnih novčanika, zvan mobilni novčanici, prvenstveno se koriste na mobilnim uređajima. Digitalni novčanici omogućuju korisniku da prilikom kupnje plaća pomoću svog uređaja tako da ne mora nositi kartice sa sobom. Korisnik samo treba unijeti podatke o svojoj kreditnoj kartici, debitnoj kartici ili bankovnom računu, pohraniti ih u digitalni novčanik i zatim može nesmetano koristiti svoj uređaj za kupovinu. Osim kartica namijenjenih plaćanju, digitalni novčanici mogu pohranjivati i poklon kartice, članske iskaznice, kupone, kartice vjernosti, zrakoplovne karte, vozačke dozvole, identifikacijske kartice te automobilske ključeve.

4.1 Princip rada

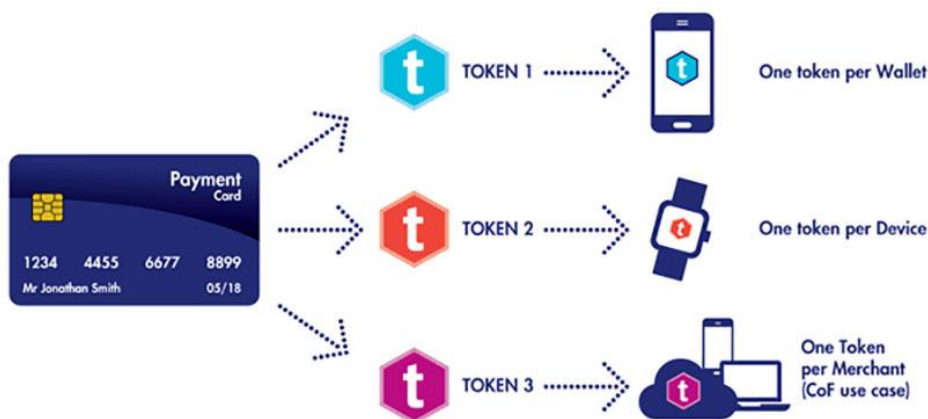
Digitalni novčanici su aplikacije dizajnirane kako bi iskoristile mogućnosti mobilnih uređaja za poboljšanje pristupa financijskim proizvodima i uslugama. Svrha digitalnih novčanika je da eliminira potrebu za nošenjem fizičkog novčanika pohranjujući sve potrošačke podatke o plaćanju na siguran i kompaktan način. Digitalni novčanici koriste bežične mogućnosti mobilnog uređaja kao što su Bluetooth, WiFi i magnetski signali za siguran prijenos podataka o uplati s korisnikovog uređaja na prodajno mjesto dizajnirano za čitanje podataka i povezivanje putem tih signala. Podaci o kartici koji su pohranuti u digitalni novčanik prenose se s korisnikovog uređaja na terminal prodajnog mjesta koji je povezan s procesorima plaćanja. Zatim se putem procesora, čvorišta ili prihvatitelja uključenog u transakciju plaćanje usmjerava kroz mrežu kreditnih kartica i banke kako bi se izvršilo plaćanje. Trenutno, digitalni novčanici koriste tri tehnologije za plaćanje, a to su QR, NFC i MST.

4.2 Sigurnosne značajke

Digitalni novčanici nude višestruke slojeve zaštite za svaku transakciju. Na prvoj razini, svaka transakcija napravljena korištenjem digitalnog novčanika zaštićena je tehnologijom koja se zove tokenizacija. Ovaj postupak kodira podatke o debitnoj ili kreditnoj kartici korisnika tako da se brojevi nikada ne dijele s trgovcem. U slučaju da trgovac bude hakiran, broj kreditne ili debitne kartice korisnika neće biti ugrožen. Za razliku od pravih kartica, korisnik neće trebati otkazivati svoju karticu i zatražiti novi broj računa u slučaju povrede podataka. Na sljedećoj razini, digitalni novčanici imaju dodatni sloj zaštite koji se aktivira kada korisnik planira izvršiti transakciju. Neki novčanici koriste biometrijske podatke kao što su otisak prsta ili prepoznavanje lica kako bi potvrdili da transakciju stvarno obavlja vlasnik novčanika. Konačno, digitalni novčanici su dizajnirani samo za interakciju s određenim aplikacijama na korisnikovom telefonu. Ova izolacija može spriječiti zlonamjerni softver koji može utjecati na druge aplikacije na uređaju da se infiltrira u novčanik i domogne financijskih podataka.

4.3 Tokenizacija

Tokenizacija je postupak zaštite osjetljivih podataka kartice, kao što su primarni broj računa (engl. *Primary account number, PAN*), datum isteka i sigurnosni kod, zamjenom jedinstvenim, algoritamski generiranim brojem koji se zove token. To je temeljna metoda osiguravanja obrade mobilnog plaćanja, zato što se digitalna kartica ne može dodati u digitalni novčanik osim ako nije prvotno tokenizirana. Kada korisnik doda kreditnu ili debitnu karticu u svoje digitalni novčanik i upotrijebi je za plaćanje, token se šalje putem mreže na obradu, umjesto stvarnih podataka na kartici. Ova metoda osigurava da čak i ako haker presretne token, ne može ga koristiti za lažne transakcije, budući da tokenizirani brojevi nemaju bitno značenje ili vezu s izvornim detaljima kartice. Nedostatak ove metode je to što tokeniziranim karticama je općenito potrebno više vremena za pokretanje jer zahtijevaju koordinaciju, pregled i odobrenja više strana. Propuštanje određenih rokova također može dodatno produžiti vremenski rok.



Slika 10. Primjer tokenizacije, Izvor: www.thalesgroup.com

4.4 Plaćanje korištenjem QR kodova

QR kod, skraćenica od "Quick Response code", je tip matičnog bar-koda koji pohranjuje informacije kao niz piksela. Kada se skenira na pametnom telefonu, korisnik može izvršiti brzu uplatu trgovcu, prenijeti novac ravnopravnoj osobi, donirati novac u dobrotvorne svrhe, prijaviti se za plaćenu pretplatu ili izvršiti niz drugih vrsta transakcija. Glavni način upotrebe QR koda je traženje uplate od korisnika putem mobilnog plaćanja. To može imati različite oblike, kao što je pokretanje zahtjeva iz kupčeve aplikacije za plaćanje s potrebnim podacima o trgovcu već ispunjenim. Drugi trgovac može upotrijebiti QR kod za otvaranje stranice za plaćanje u web pregledniku kupca, gdje kupac odabire željeni način plaćanja. Osnovna struktura QR koda uvijek je kvadrat, jer odražava mrežu društvene igre Go, koja je inspirirala njenog kreatora Masahira Hara (Dougall, 2024.). Popularnost mu je eksplodirala 2017. godine, kada je Apple omogućio iPhone kameri da skenira QR kodove. Posebno su popularni u gospodarstvima u razvoju gdje se mobilno plaćanje češće koristi od kreditnih kartica ili gotovine.



Slika 11. QR kod, Izvor: www.fiverr.com

4.4.1. Prednosti

Korisnički QR kod s dinamičkom vrstom koda je poput jednokratne lozinke i mora biti skeniran od strane trgovca. Dinamički QR kod ima nekoliko prednosti, od kojih je jedna dobra sigurnost. Budući da dinamički QR kod vrijedi kratko vrijeme, ne može se više koristiti u budućnosti, čak i ako ga netko ukrade za svoje potrebe. To će se dogoditi čak i ako se transakcija nije dogodila prije isteka koda. QR kod tehnologija ne zahtijeva poseban hardver ili infrastrukturu, što je čini isplativim i pristupačnim rješenjem za trgovce i potrošače. Za razliku od NFC-a, za koji su potrebni uređaji i terminali koji podržavaju NFC, QR kodovi se mogu skenirati kamerom pametnog telefona, što znači da se mogu koristiti na širokom rasponu uređaja bez dodatnih ulaganja. Trgovci mogu koristiti QR kodove za niz scenarija plaćanja, od malih transakcija poput kupnje šalice kave do većih transakcija poput plaćanja automobila. QR kodovi ne zahtijevaju nikakvu posebnu obuku za izradu ili korištenje, a budući da mogu pohraniti mnogo informacija, trgovci mogu smanjiti troškove ispisa i marketinga preusmjerenjem kupca na internetsku stranicu.

4.4.2. Nedostatci

QR kodovima se može lako manipulirati, a prevaranti mogu stvoriti lažne QR kodove kako bi ukrali korisničke podatke. QR kodovi također ne mogu funkcionirati bez internetske veze. Osobe sa starijim modelom pametnog telefona mogu imati slab signal ili nemaju pristup Wi-Fi-ju, što im onemogućuje pristup onome što se nalazi iza QR koda. QR kodovi zahtijevaju pametni telefon s mogućnošću skeniranja koda; nekolicina modernih telefona imaju tu mogućnost ugrađenu u kameru, ali često biva da korisnik mora sam preuzeti aplikaciju. Neki se korisnici također mogu oslanjati na telefone koji nisu pametni ili jednostavno nemaju svoj telefon na sebi u tom trenutku. Međutim, najveći nedostatak QR koda je upravo njegov način rada. QR kod je jednosmjerni komunikacijski alat; nije sposoban uspostaviti komunikaciju između kupca i trgovca u isto vrijeme. Također, pretjerano oslanjanje na QR kod može dovesti do prekida između kupca i trgovca do kojeg on pokušava doći.

4.5 Plaćanje koristeći NFC

NFC, skraćenica za Near Field Communication, je tehnologija koja uređajima poput pametnih telefona i pametnih satova omogućuje razmjenu malih bitova podataka s drugim uređajima i čitanje kartica opremljenih NFC-om na relativno malim udaljenostima. Tehnologija iza NFC-a vrlo je slična radiofrekvencijskoj identifikaciji (engl. *Radio-frequency identification, RFID*) koja se obično koristi u sigurnosnim karticama i privjescima za ključeve koji se koriste za identifikaciju pri ulasku u poslovni ured. U stvarnosti, NFC je evolucija RFID-a koja nudi naprednije značajke i bolju sigurnost, omogućavajući brzu dvosmjernu komunikaciju između uređaja. Radi na frekvenciji od 13,56 MHz i može prenositi podatke brzinom do 424 kbit/s. Ova mogućnost omogućuje NFC-u da podržava razne aplikacije, od beskontaktnog plaćanja do razmjene podataka i kontrole pristupa, na kratkim udaljenostima. Kao i kod većine novih tehnologija, postojale su mnoge rane aplikacije za NFC koje nikada nisu zaživjele. Jedna od tih aplikacija se zvala Android Beam, koju je izmislio Google. Svrha te aplikacije je bila razmjena podataka između pametnih telefona. Međutim, Google je ubrzo otkrio da većina ljudi ne želi držati svoje telefone u neposrednoj blizini onoliko vremena koliko je potrebno za prijenos fotografija i videozapisa preko relativno spore NFC veze.

Android Beam je kasnije zamijenjen aplikacijom Fast Share, kasnije preimenovan u Nearby Share, kako bi iskoristio Bluetooth i Wi-Fi, koji su nudili bolji domet i veće brzine prijenosa. Nakon što se prvotno uzbuđenje zbog ugradnje NFC-a u pametne telefone malo smirilo, tvrtke su se počele fokusirati na praktičnije primjene za koje je ova beskontaktna tehnologija mnogo prikladnija. Platne kartice i sigurnosne oznake već su godinama koristile RFID tehnologiju, a budući da je NFC izgrađen na istim temeljima, pokazalo se da je gotovo savršen za integraciju tih značajki u pametne telefone i pametne satove. Počevši u 2015. godini, Apple započinje nuditi NFC mobilna plaćanja u svojim pametnim uređajima otkako su izdali svoj prvi Apple Watch. U današnje vrijeme, mobilna plaćanja se mogu obaviti s bilo kojeg modernog pametnog telefona koji podržava Apple Wallet, Google Wallet ili Samsung Pay, kao i bilo kojeg Apple Watch i mnogih Wear OS pametnih satova.



Slika 12. NFC Logo, Izvor: stock.adobe.com

Uređaj koji podržava NFC može implementirati tri glavna načina rada:

1. *Reader/Writer Mode* – Ovaj način rada omogućuje prijenos podataka za NFC omogućenu aplikaciju pametnog telefona. Također omogućuje čitanje ili pisanje pasivnih NFC oznaka ili naljepnica pomoću uređaja koji podržava NFC. NFC čitač će očitati podatke s NFC oznaka ili naljepnica kada je u načinu rada čitača. U načinu rada pisača, uređaj je sposoban pisati podatke na oznake ili naljepnice.
2. *Peer-to-Peer Mode (P2P Mode)* – P2P način rada provode dva aktivna uređaja s omogućenim NFC-om koji mogu međusobno komunicirati radi dijeljenja podataka. Poruke, dokumenti i druge vrste podataka mogu se razmjenjivati između dva uređaja s koji podržavaju NFC.
3. *Card Emulation Mode (CE Mode)* – U CE načinu rada pametni telefon s omogućenim NFC-om djeluje kao beskontaktna pametna kartica. U ovom načinu rada telefon emulira fizičku karticu, kao što je kreditna/debitna kartica ili pristupna kartica, dopuštajući da se koristi s NFC čitačem za plaćanja, izdavanje karata ili kontrolu pristupa u prostor koji bi inače bio izvan granica pristupa korisnika. Postoje dvije vrste emulacije NFC kartica, Host-based Card Emulation (HCE) i Secure Element Card Emulation (SE). U oba načina, NFC čitač tretira telefon kao da je fizička beskontaktna kartica

Tablica 2: Sažetak rada NFC tehnologije

Način rada	Definicija
Reader/Writer Mode	NFC uređaj djeluje kao čitač za čitanje informacija iz odredišta.
P2P Mode	Ovaj način rada omogućuje razmjenu podataka, uključujući dokumente i poruke, između dva NFC uređaja.
CE	<p>NFC uređaj emulira karticu s dovoljno informacija koje treba razmijeniti. To omogućuje uređaju da djeluje kao virtualna kartica koju može čitati NFC čitač kartica.</p> <ul style="list-style-type: none"> • SE način: Ovaj način rada koristi SE (sigurni element) čip za pohranu informacija potrebnih za emulaciju kartice. SE čip djeluje kao posrednik između NFC uređaja i NFC čitača. • HCE način: U ovom načinu rada NFC čitač izravno komunicira s glavnom aplikacijom na NFC uređaju. Ovo eliminira potrebu za zasebnim SE čipom.

4.5.1 Prednosti

NFC plaćanja omogućuju brze transakcije samo dodiranjem telefona, eliminirajući potrebu za nošenjem gotovine ili platnih kartica. Plaćanje korištenjem NFC-a je mnogo puta brže od tradicionalnih metoda plaćanja, što smanjuje vrijeme čekanja na blagajni. Osim što je jednostavan način plaćanja, NFC omogućuje trgovcima prihvaćanje transakcije bez potrebe da zaposlenici rukuju fizičkim karticama. Ovo ne samo da ubrzava proces naplate, već i smanjuje fizički kontakt, prednost koja je postala sve važnija u današnjem svijetu, gdje je zabrinutost za zdravlje i sigurnost porasla (Seaman, 2024.). NFC plaćanja mogu se integrirati s programima vjernosti (engl. *Loyalty program*) i digitalnim novčanicima, omogućujući korisnicima da automatski ostvaruju nagrade vezane uz svoj program vjernosti.

4.5.2 Nedostatci

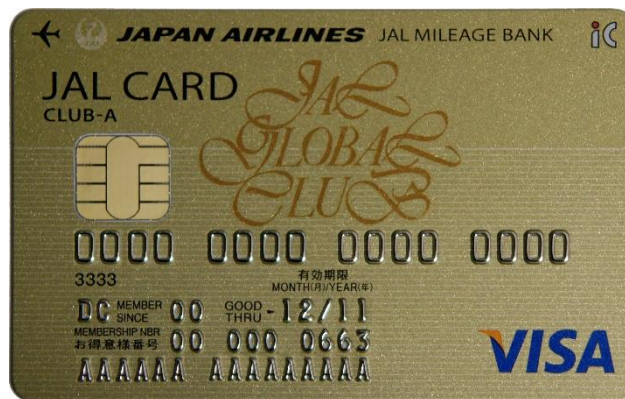
Iako su općenito sigurni za korištenje, još uvijek postoji zabrinutost zbog potencijalnih uređaja za hakiranje ili skimming napada koji bi mogli uhvatiti NFC signale i tako omogućiti neovlaštenim osobama pristup osobnim podacima korisnika. Osim toga, može se dogoditi da određeni pametni mobilni uređaji ne podržavaju NFC, što može ograničiti upotrebljivost za neke korisnike. Korisnici također moraju biti svjesni sigurnosnih mjera koje mogu poduzeti kako bi zaštitili svoje podatke, poput korištenja sigurnosnih aplikacija i redovitog ažuriranja softvera. NFC-ov kratki doomet također može biti njegov nedostatak. Na primjer, NFC tehnologija nije dobar odabir ako korisnik treba skenirati oznake na dugom nizu proizvoda koji se proteže do zadnje strane police. Osim toga, NFC oznake obično su skuplje od mnogih vrsta RFID oznaka zbog dodatnih troškova enkripcije. Zbog toga su NFC oznake manje prikladne, na primjer, za maloprodaju u kojoj svaki odjevni predmet treba oznaku (Lawton, 2022.).

4.6. MST

MST, skraćeno od Magnetic Secure Transmission, je vrsta tehnologije koja pametnim telefonima omogućuje bežično plaćanje putem tradicionalnih magnetskih sustava i modernih terminala. Ova tehnologija radi na bazi magnetskih signala, koji se šalju s pametnog uređaja na POS terminal kako bi se ostvarila uplata. Emulira provlačenje fizičke kartice kroz terminal bez potrebe za nadogradnjom softvera ili hardvera terminala, za razliku od naprednije tehnologije, što ga čini veoma korisnim za ostvarivanje beskontaktnog plaćanja. Za razliku od NFC tehnologije, MST tehnologija je kompatibilna s gotovo svim terminalima za plaćanje koji posjeduju čitač magnetskih traka, što ga čini šire prihvaćenim od ostalih usluga mobilnog plaćanja koje se oslanjaju isključivo na NFC. MST je izvorno razvio LoopPay, koji je Samsung kupio 2015. godine i uključio ga u svoju uslugu Samsung Pay. Kao rezultat toga, nedostatak MST-a je to što je u vlasništvu Samsunga i njime se upravlja isključivo sa Samsung uređajima. MST tehnologija je doživjela svoj kraj u 2021. godini. Zbog rasta NFC tehnologije u svijesti tipičnog korisnika, zahvaljujući Appleu i Googleu, Samsung je preusmjerio svoje napore na integraciju Samsung Pay-a sa NFC tehnologijom, počevši s njihovim portfeljem Galaxy uređaja. Iako se MST još uvijek može koristiti na mobilnim uređajima koji već imaju tu tehnologiju, budući Samsung uređaji više neće uključivati ovu tehnologiju (Hager, 2021.)

4.7 EMV

EMV je tehnologija plaćanja koja koristi maleni, moćni čip ugrađen u kreditne i debitne kartice kako bi kartične transakcije bile sigurnije. Razvijen je sredinom 1990-ih godina i od tada je postao standard za sigurno kartično plaćanje. U vrijeme prije EMV čipa, sve transakcije kreditnim ili debitnim karticama uključivale su korištenje magnetske trake ili mehaničkog otiska za čitanje i snimanje podataka o računu, te potpis kupca za potrebe provjere identiteta. Kupac bi svoju karticu predao blagajniku na prodajnom mjestu, a on bi zatim karticu provukao kroz magnetski čitač ili napravio otisak uzdignutog teksta kartice. Takve metode nisu bile veoma sigurne, što je omogućilo lopovima i prevarantima da se lagano domognu privatnih podataka kupca, bilo krađom same kartice, ili brisanjem i zamjenom legitimnog potpisa ili krivotvorenjem istog. Kako bi riješile ovaj problem prijevara s kreditnim karticama i omogućile jednostavniji način plaćanja između trgovca i kupca, tvrtke su se počele okretati čipovima. Tri poznate kompanije, Europay, MasterCard i Visa, predvodile su razvoj i široku primjenu ove tehnologije. Ime EMV je skraćenica stvorena od prvih slova ovih triju kompanija. Standardom sada upravlja EMVCo, konzorcij koji je kontrolu nad EMV-om ravnopravno podijelio između Visa, Mastercard, JCB, American Express, China UnionPay i Discover (Stripe, 2023.).



Slika 13. EMV Visa kartica, Izvor: en.wikipedia.org

EMV kartice su kreditne i debitne kartice s ugrađenim malim računalnim čipom. Upravo taj čip, a ne magnetska traka (engl. *magstripe*) koja se nalazi na poleđini kartice, prenosi podatke o plaćanju čitaču kartice tijekom transakcije. Iako EMV čip provodi transakcije bez pomoći magnetne trake kartice, kartice s čipom obično su još uvijek opremljene magnetskom trakom.

Postoje dvije vrste EMV čip kartica:

1. *Chip-and-PIN* – Ove kartice, koje se smatraju sigurnijom opcijom, zahtijevaju od vlasnika kartice da kreiraju svoj PIN i unesu ga na prodajnom mjestu za autentifikaciju transakcije. Plaćanje se ne može nastaviti bez PIN-a.

2. *Chip-and-signature* – Ove kartice zahtijevaju od vlasnika kartice da se potpiše nakon svake transakcije, kako bi potvrdili svoj identitet.

EMV čipovi su eksponencijalno sigurniji od magnetskih traka na karticama, velikim dijelom zato što ne prenose pravi broj kartice tijekom transakcije. Umjesto toga, oni generiraju jedinstveni kod za svaku kupnju i šalju taj kod čitaču kartica. Kodovi koje generiraju EMV kartice ne mogu se replicirati, koristiti više od jednom ili lako krivotvoriti, čime se EMV kartice štite od sigurnosnih propusta koji zadese plaćanja magnetskom trakom (Thales).

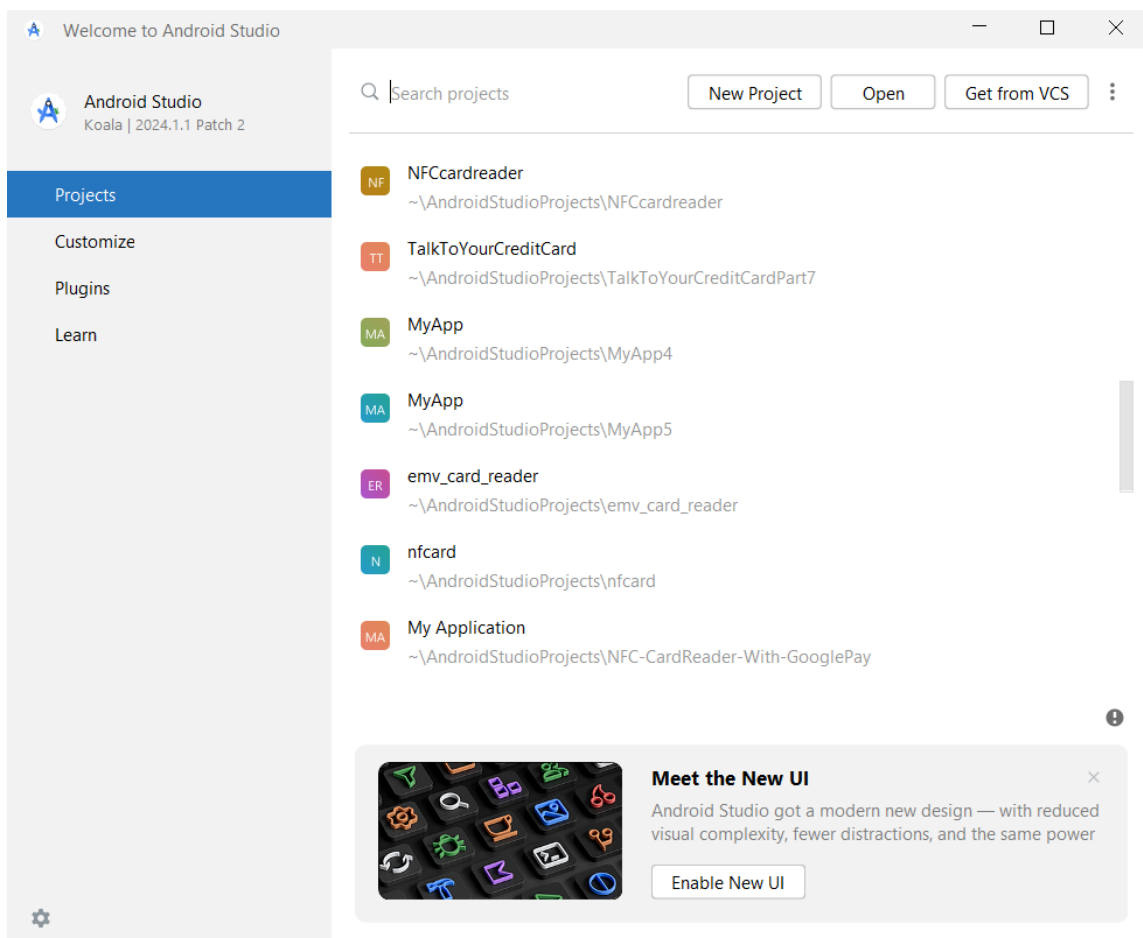
U današnje vrijeme, mnoge EMV kartice imaju mogućnost beskontaktnog plaćanja korištenjem NFC tehnologije. Sve što je potrebno za ostvariti EMV mobilno plaćanje je uređaj koji podržava NFC te digitalni novčanik koji je povezan s korisnikovim bankovnim računom. Kada kupac poželi platiti, jednostavno svojim uređajem dodirne terminal koji prihvaća EMV mobilna plaćanja. Terminal čita kod s mobilnog uređaja i provjerava ga kod banke ili izdavatelja kartice. Transakcija se zatim obrađuje i dovršava u nekoliko sekundi. Za kupca, to pruža brz, praktičan i siguran način plaćanja bez nošenja gotovine ili kartica. Za trgovce, oni smanjuju rizik od prijevare i povrata uplate jer provjeravaju autentičnost svake transakcije dinamičkim kodom. Osim toga, trgovcima omogućuju prikupljanje više podataka i uvida u ponašanje i preferencije kupaca, što im može pomoći da poboljšaju svoje marketinške i prodajne strategije (Bedoya).

5. TESTIRANJE NFC TEHNOLOGIJE U PRIKUPLJANJU EMV PODATAKA

Za ispitivanje rada NFC tehnologije u skeniranju beskontaktnih kartica korišten je jedan pametni telefon koji podržava NFC, te je na njemu instalirana mobilna aplikacija "NFCcardreader" kojoj je namjena emulirati NFC čitač. U edukativne svrhe izvodi se izvlačenje podataka sa VISA kartica i prikaz istih u aplikaciji. Ovakav pristup omogućuje uvid u podatke koji se koriste pri komuniciranju digitalnog novčanika s POS terminalom.

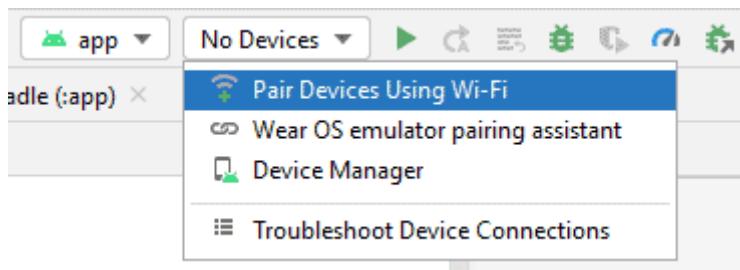
5.1 Postavljanje softvera (instalacija)

Mobilna aplikacija je kreirana u Android Studiu, službenom integriranom razvojnom okruženju za Googleov operativni sustav Android. Prije svega, potrebno je prvo preuzeti Android Studio s browsera i instalirati ga na računalo. Također je potrebno pripremiti pametni telefon koji podržava NFC. Zbog nedostatka podrške za NFC u AVD-u, aplikacija ne može raditi u emulatoru. Nakon što smo završili postavljanje Android Studia, preuzet ćemo aplikaciju na računalo te ćemo ju otvoriti na početnoj stranici, prikazano na slici 13.



Slika 14. Otvaranje projekta

U gornjoj desnoj strani ćemo odabrati “Device manager” i povezati pametni telefon s Android Studio pomoću “Pair devices using Wi-Fi”. Završetkom ovog koraka, sva konfiguracija je gotova i možemo kreirati aplikaciju.



Slika 15. Povezivanje s stvarnim pamtnim telefonom pomoći WiFi-a

5.2 Implementacija NFC tehnologije

Ovaj odjeljak će raspravljati o implementaciji NFC-a u mobilnoj aplikaciji uz pomoć slika. NFC dolazi s puno načina i funkcija, kao što su plaćanje, prijenos podataka ili kontrola pristupa. U ovom projektu, NFC je korišten za hvatanje i čitanje podataka s EMV kartica kako bi nam pomogao vidjeti podatke koji se prenose s kartice na POS terminal tijekom procesa plaćanja.

5.2.1 Manifest

Za implementaciju NFC modula potrebno je zatražiti neka dopuštenja. Unutar Android Studia, dopuštenje je potrebno zatražiti unutar datoteke AndroidManifest.xml. Redak 6 implementira dopuštenje koje omogućuje aplikaciji korištenje NFC-a. Redak 7 označava da je NFC obavezna značajka za aplikaciju. Ako je `android:required="false"`, aplikacija se i dalje može instalirati na uređaje bez NFC-a, ali sama funkcija ne bi bila dostupna.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:installLocation="auto">

  <uses-permission android:name="android.permission.NFC" />
  <uses-feature android:name="android.hardware.nfc"
android:required="true" />
```

Kod 1. Postavljanje dopuštenja u AndroidManifest.xml

5.2.2 MainActivity

Klasa MainActivity služi kao središte “NFCcardreader” aplikacije. Ova klasa je dizajnirana za otkrivanje i čitanje NFC oznaka pomoću Androidovog NFC okvira. Podržava različite vrste NFC oznaka, izvlači i prikazuje njihove podatke i pruža jednostavan način brisanja prethodnih informacija o NFC oznakama putem korisničkog sučelja. Aplikacija obrađuje NFC namjere, izdvaja podatke o oznakama i formatira ih za prikaz u stvarnom vremenu.

```
private fun resolveIntent(intent: Intent) {
    val validActions = listOf(
        NfcAdapter.ACTION_TAG_DISCOVERED,
        NfcAdapter.ACTION_TECH_DISCOVERED,
        NfcAdapter.ACTION_NDEF_DISCOVERED
    )
    if (intent.action in validActions) {
        val rawMsgs =
intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES)
        val messages = mutableListOf<NdefMessage>()
        if (rawMsgs != null) {
            rawMsgs.forEach {
                messages.add(it as NdefMessage)
            }
        } else {
            val empty = ByteArray(0)
            val id = intent.getByteArrayExtra(NfcAdapter.EXTRA_ID)
            val tag = intent.parcelable<Tag>(NfcAdapter.EXTRA_TAG) ?:
return
            val payload = dumpTagData(tag).toByteArray()
            val record = NdefRecord(NdefRecord.TNF_UNKNOWN, empty, id,
```

Kod 2. Otkrivanje, izdvajanje I prikazivanje podataka na NFC kartici

5.2.3. NdefMessageParser

Klasa NdefMessageParser analizira sadržaj NDEF poruka razlažući ju na pojedinačne zapise i kategorizirajući ju kao URI, tekst, Smart Poster ili neobrađene podatke. Raščlanjeni zapisi pretvaraju se u objekte ParsedNdefRecord, koji se prikazuju u korisničkom sučelju aplikacije u jednostavnom formatu.

```
fun getRecords(records: Array<NdefRecord>): List<ParsedNdefRecord> {
    val elements = mutableListOf<ParsedNdefRecord>()
    for (record in records) {
        if (UriRecord.isUri(record)) {
            elements.add(UriRecord.parse(record))
        } else if (TextRecord.isText(record)) {
            elements.add(TextRecord.parse(record))
        } else if (SmartPoster.isPoster(record)) {
            elements.add(SmartPoster.parse(record))
        } else {
            elements.add(object : ParsedNdefRecord {
                override fun getView(
                    activity: Activity,
                    inflater: LayoutInflater,
                    parent: ViewGroup,
                    offset: Int
                ): View {
                    val text = inflater.inflate(R.layout.tag_text,
parent, false) as TextView
                    text.text = String(record.payload)
                    return text
                }
            })
        }
    }
}
```

Kod 3. Pronalazak zapisa na kartici

5.2.4. UriRecord

Klasa UriRecord upravlja raščlanjivanjem i prikazivanjem NFC zapisa koji sadrže URI. Raščlanjuje NFC podatke kako bi izdvojio URI, a zatim ga prikazuje u tekstualnom modelu.

```
class UriRecord private constructor(private val uri: Uri) :  
    ParsedNdefRecord {  
    override fun getView(  
        activity: Activity,  
        inflater: LayoutInflater,  
        parent: ViewGroup,  
        offset: Int  
    ): View {  
        val text = inflater.inflate(R.layout.tag_text, parent,  
false) as TextView  
        text.autoLinkMask = Linkify.WEB_URLS or  
Linkify.EMAIL_ADDRESSES or Linkify.PHONE_NUMBERS  
        text.text = uri.toString()  
        return text  
    }  
}
```

Kod 4. Raščlanjivanje URI-ja

5.2.5 TextRecord

Klasa TextRecord odgovorna je za rukovanje NFC zapisima koji sadrže tekst. Raščlanjuje zapis kako bi izdvojio tekst, uzimajući u obzir njegovo kodiranje i jezik, te pruža metodu za prikaz ovog teksta u TextViewu za prikaz u korisničkom sučelju. Companion object uključuje metode za analiziranje i provjeru je li zapis tekstualni zapis.

```
companion object {
    fun parse(record: NdefRecord): TextRecord {
        require(record.tnf == NdefRecord.TNF_WELL_KNOWN)
        require(Arrays.equals(record.type,
NdefRecord.RTD_TEXT))
        return try {
            val payload: ByteArray = record.payload
            val textEncoding = if (payload[0].toInt() and 128
== 0) StandardCharsets.UTF_8 else StandardCharsets.UTF_16
            val languageCodeLength = payload[0].toInt() and
63

            val text = String(
                payload,
                languageCodeLength + 1,
                payload.size - languageCodeLength - 1,
                textEncoding
            )
            TextRecord(text)
        } catch (e: UnsupportedEncodingException) {
            throw IllegalArgumentException(e)
        }
    }

    fun isText(record: NdefRecord): Boolean {
        return try {
```

Kod 5. Dekodiranje teksta iz NFC zapisa

5.2.6 SmartPoster

Klasa SmartPoster pruža strukturu za rukovanje NFC zapisima koji predstavljaju SmartPostere. Može sadržavati URI i izborni naslov, prikazujući oboje u izgledu prilagođenom korisniku. Uključuje metode za raščlanjivanje zapisa SmartPostera iz NdefRecord objekata i provjeru valjanosti je li dani zapis doista SmartPoster.

```
class SmartPoster private constructor(  
    private val uri: UriRecord,  
    private val title: TextRecord?  
) : ParsedNdefRecord {  
  
    override fun getView(  
        activity: Activity,  
        inflater: LayoutInflater,  
        parent: ViewGroup,  
        offset: Int  
    ): View {  
        return if (title != null) {  
            val container = LinearLayout(activity)  
            container.orientation = LinearLayout.VERTICAL  
            container.layoutParams = ViewGroup.LayoutParams(  
                ViewGroup.LayoutParams.MATCH_PARENT,  
                ViewGroup.LayoutParams.WRAP_CONTENT  
            )  
            container.addView(title.getView(activity, inflater,  
container, offset))  
            inflater.inflate(R.layout.tag_divider, container)  
            container.addView(uri.getView(activity, inflater,  
container, offset))  
            container  
        } else {  
            uri.getView(activity, inflater, parent, offset)  
        }  
    }  
}
```

Kod 6. Kreiranje okomitog izgleda za prikaz URI-ja

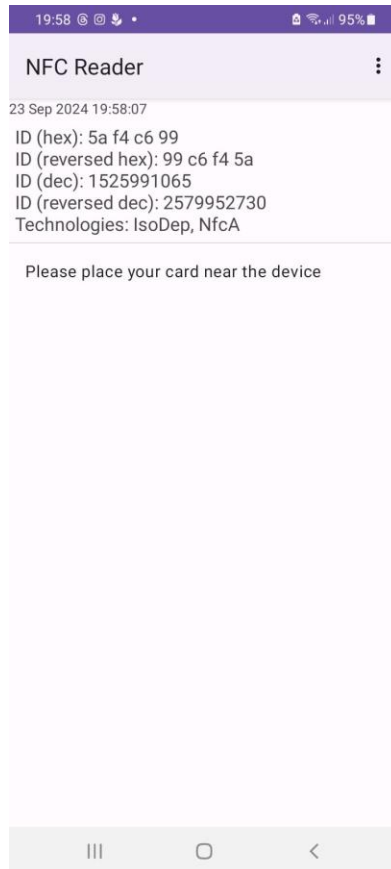
5.2.7 Izgled aplikacije

Pri pokretanju aplikacije, pojavljuje se korisničko sučelje. Ako korisnik pokuša instalirati aplikaciju na mobilnom uređaju koji ne podržava NFC ili u Android OS emulatoru, pojavit će se obavijest u obliku iskočnog prozora. Klikom na taj prozor, aplikacija će se zatvoriti. Na mobilnom uređaju koji podržava NFC, korisnik će vidjeti sučelje koje je prikazano na Slici 17.



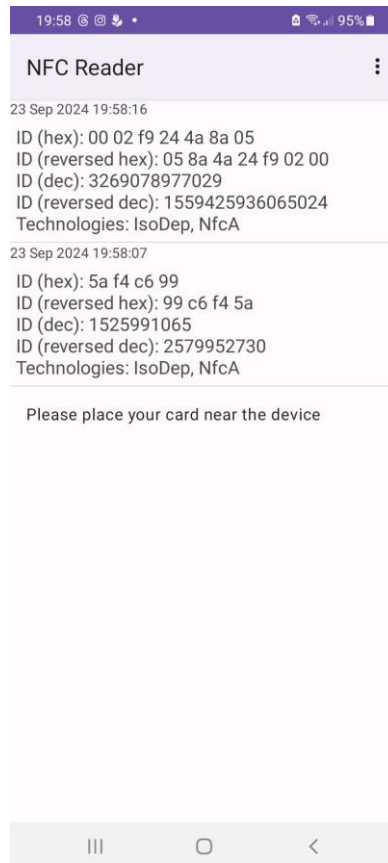
Slika 16. Početni zaslon

Interakcija sa sučeljem je jednostavna: korisnik samo treba približiti svoju beskontaktnu karticu kameri mobilnog uređaja na udaljenosti ne većoj od četiri centimetra. Nakon nekoliko sekundi, aplikacija će očitati podatke s kartice i prikazati ih na sučelju, kao što je prikazano na Slici 18.



Slika 17. Rezultat skeniranja kartice

Aplikacija može očitati podatke iz različitih vrsta beskontaktnih kartica. Ako korisnik želi obrisati prikazane podatke, može kliknuti na gumb u gornjem desnom kutu, čime će se otvoriti izbornik s opcijom "Clear all". Pritiskom na taj gumb, svi podaci na korisničkom sučelju bit će izbrisani.



Slika 18. Rezultat skeniranja više kartica

6. ZAKLJUČAK

Rad pruža opsežan pregled definicije, funkcija i sigurnosnih značajki digitalnih novčanika, te prenosi fokus na različite vrste elektroničkog plaćanja koje se danas koriste u svijetu, s fokusom na prednosti i nedostatke ove tehnologije i načina plaćanja. S obzirom na brz razvoj digitalne ekonomije u današnje vrijeme te sve veću uporabu beskontaktnih kartica, pa tako i sve veću potrebu za zaštitom trgovca i potrošača, digitalni novčanici postaju ključni alati za poboljšanje korisničkog iskustva pri plaćanju. Jedan od glavnih, a može se smatrati i najvećim izazovom koji stoji na putu ovoj tehnologiji da potpuno zamjeni tradicionalnu vrstu plaćanja, je nepovjerenje ljudi te briga o mogućem nestanku, krađi ili zlouporabi osobnih podataka. Ali kako vrijeme prolazi, šansa da će digitalni novčanici uskoro potpuno nadmašiti svoje trenutne sposobnosti integracijom nove tehnologije, poput umjetne inteligencije (engl. *Artificial Intelligence, AI*), u svoje usluge postaje sve veća. Može se smatrati da će u bliskoj budućnosti broj korisnika digitalnih novčanika znatno porasti. Ključni doprinos popularnosti digitalnih novčanika je postojanje NFC tehnologije. Pomoću NFC-a, transakcije između potrošača i trgovca su jednostavnije nego ikad. Korisnici digitalnih novčanika uživaju pogodnost pri plaćanju, upravo zbog prisutnosti ove tehnologije. NFC plaćanja ostavila su veliki trag na globalnoj razini, i taj trag neće nestati u skoro vrijeme.

Zaključno, transakcije putem digitalnih novčanika budućnost su digitalnog plaćanja. A plaćanja NFC-om nije samo prolazni trend, već temeljna tehnologija koja će oblikovati budućnost električnih transakcija. Uz NFC, trgovci su sposobni učiniti iskustvo svojih klijenata izuzetno glatkim, praktičnim i sigurnim.

LITERATURA

- Bedoya, U. (n.d.). *What are the benefits and challenges of adopting EMV mobile payments in different markets and regions?* Dohvaćeno iz LinkedIn: <https://www.linkedin.com/advice/0/what-benefits-challenges-adopting-emv-mobile-payments>
- Dougall, S. (28. Svibanj 2024.). *A guide to QR code payments.* Dohvaćeno iz checkout.com: <https://www.checkout.com/blog/a-quick-guide-to-qr-code-payments>
- Hager, R. (14. Siječanj 2021.). *Samsung's Galaxy S21 won't support MST for contactless payments in the US.* Dohvaćeno iz Android Police: <https://www.androidpolice.com/2021/01/14/samsung-pay-is-losing-its-one-remaining-good-feature-on-the-galaxy-s21/>
- Lawton, G. (08. Kolovoz 2022.). *RFID vs. NFC: Learn the pros and cons of each.* Dohvaćeno iz TechTarget: <https://www.techtarget.com/searcherp/tip/RFID-vs-NFC-Learn-the-pros-and-cons-of-each>
- Seaman, M. (30. Kolovoz 2024.). *The Benefits of NFC Mobile Payment: A Complete Guide.* Dohvaćeno iz Swipesum: <https://www.swipesum.com/insights/the-benefits-of-nfc-mobile-payment-a-complete-guide>
- Stripe. (2. Veljača 2023.). *What are EMV chip cards?* Dohvaćeno iz Stripe: <https://stripe.com/en-hr/resources/more/what-are-emv-chip-cards>
- TechVidvan. (n.d.). *Advantages and Disadvantages of Java.* Dohvaćeno iz TechVidvan: <https://techvidvan.com/tutorials/pros-and-cons-of-java/>
- Thales. (n.d.). *Are contactless cards safe?* Dohvaćeno iz thalesgroup.com: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/cards/contactless/how-it-works>
- Walker, K. (20. Svibanj 2022.). *What exactly is Kotlin?* Dohvaćeno iz codeop: <https://codeop.tech/what-exactly-is-kotlin/>

PRILOZI

Popis kodova

<i>Kod 1. Postavljanje dopuštenja u AndroidManifest.xml</i>	35
<i>Kod 2. Otkrivanje, izdvajanje I prikazivanje podataka na NFC kartici</i>	36
<i>Kod 3. Pronalazak zapisa na kartici</i>	37
<i>Kod 4. Raščlanjivanje URI-ja</i>	38
<i>Kod 5. Dekodiranje teksta iz NFC zapisa</i>	39
<i>Kod 6. Kreiranje okomitog izgleda za prikaz URI-ja</i>	40

Popis tablica

<i>Tablica 1: Prednosti najpopularnijih programskih jezika</i>	7
<i>Tablica 2: Sažetak rada NFC tehnologije</i>	27

Popis slika

<i>Slika 1. Globalni udio u prodaji pametnih telefona, Izvor: www.counterpointresearch.com</i> ...	2
<i>Slika 2. Operacijski sustav Android, Logo, Izvor: ausdroid.net</i>	3
<i>Slika 3. Pet najpopularnijih jezika za razvoj mobilnih aplikacija, Izvor: fireart.studio</i>	6
<i>Slika 4. Android Java Logo, Izvor: theiconic.tech</i>	8
<i>Slika 5. Android Kotlin Logo, Izvor: developer.android.com</i>	11
<i>Slika 6. Jetpack Compose Logo, Izvor: medium.com</i>	12
<i>Slika 7. Kreditna kartica, Izvor: www.investopedia.com</i>	15
<i>Slika 8. Debitna kartica, Izvor: m2pfintech.com</i>	16
<i>Slika 9. Virtualna kartica, Izvor: taulia.com</i>	18
<i>Slika 10. Primjer tokenizacije, Izvor: www.thalesgroup.com</i>	23
<i>Slika 11. QR kod, Izvor: www.fiverr.com</i>	24
<i>Slika 12. NFC Logo, Izvor: stock.adobe.com</i>	26
<i>Slika 13. EMV Visa kartica, Izvor: en.wikipedia.org</i>	30
<i>Slika 14. Otvaranje projekta</i>	33
<i>Slika 16. Povezivanje s stvarnim pamtnim telefonom pomoći WiFi-a</i>	34
<i>Slika 17. Početni zaslon</i>	41
<i>Slika 18. Rezultat skeniranja kartice</i>	42
<i>Slika 19. Rezultat skeniranja više kartica</i>	43