

ИМПЛЕМЕНТАЦИЈА МЕХАНИЗМА АРБИТРАЖНОГ ТРГОВАЊА
КРИПТОВАЛУТАМА У iOS МОБИЛНОЈ АПЛИКАЦИЈИIMPLEMENTATION OF A CRYPTOCURRENCY ARBITRAGE MECHANISM IN AN
iOS MOBILE APPLICATION

Лука Ступар, Факултет техничких наука, Нови Сад

Oblast – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду је изложено једно решење механизма за извођење арбитражне трговине криптовалута засновано на технологијама за развој iOS мобилних апликација и развијено помоћу окружења Xcode. Апликација служи за праћење цена и извршавање трговине на различитим мењачницама. Након описа апликације дискутовани су резултати тестирања и изведени одговарајући закључци.

Кључне речи: мобилне апликације, iOS, блокчејн, криптовалуте, трговање

Abstract – This paper describes an implementation of mechanism for arbitrage trading of cryptocurrencies as an iOS mobile application developed using Xcode IDE. The application is intended for cryptocurrency price tracking and automatic trade execution on different exchanges. After the application description, test results are discussed, and appropriate conclusions are drawn.

Keywords: mobile applications, iOS, blockchain, cryptocurrencies, trading, arbitrage

1. УВОД

Арбитражно трговање је стратегија која се примењује на великом броју финансијских тржишта са циљем остварења профита. Овај вид трговине заснива се на скоро истовременој куповини и продаји финансијских инструмената на различитим тржиштима како би се уновчила разлика у ценама. Чест исход оваквог начина трговања је, поред прихода, изједначавање цена робе или финансијског инструмента на различитим тржиштима. Арбитражно трговање настало је још у античким временима, а попримило данашњи облик почетком XXI века [1]. Применом блокчејна, знатно се повећао број тржишта на којима је могуће уочити прилике за арбитражну трговину. Иако се арбитража сматра ниско ризичним начином трговања, вероватноћа настанка финансијских губитака постоји услед клизања цена¹, трошкова размене, недовољне брзине, као и грешака.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Душан Гајић, ванр. проф.

¹ Клизање цена означава разлику између очекиване и остварене цене при трговини.

2. ДИГИТАЛНА ИМОВИНА И БЛОКЧЕЈН

Дигитална имовина је сваки облик дигитално складиштене информације који има препознатљивост и који једном или више појединаца пружа вредност која, између осталог може бити материјална. Примери дигиталне имовине су: фотографије, документи, скупови података, метаподаци, криптовалуте, дигиталне валуте централних банака [2].

Блокчејн је технологија која служи за чување података намењена да отежа и онемогући фалсификовање ускладиштених података. Своју примену је пронашла у имплементацијама дистрибуираних регистара (енгл. DLT), што је омогућило напредне начине чувања дигиталне имовине.

Идеја блокчејна први пут настаје 1991. године, а 2008. године аутор под псеудонимом Сатоши Накамото објављује први предлог функционалне блокчејн мреже која решава проблем двоструке потрошње – Bitcoin, чија имплементација почиње са радом 2009 [3].

2.1 Мењачнице криптовалута

Крипто мењачнице су платформе намењене за куповину, продају и размену криптовалута. Многе мењачнице поред ових услуга нуде и могућност трговања финансијским дериватима, штедњу, позајмице и сличне услуге.

Најзначајнија подела ових платформи је на централизоване и децентрализоване. Док се централизоване мењачнице фокусирају на једноставност употребе, ниске трошкове конверзије криптовалута и корисничку подршку, децентрализоване мењачнице су технологија која користи блокчејн и паметне уговоре како би обезбедиле трговину без учешћа посредника, појачану безбедност, али уз ограничења по питању корисничке подршке [4].

2.2 Врсте арбитражног трговања криптовалута

Арбитража између мењачница – (енгл. Cross-exchange arbitrage) је метод трговања који захтева истовремено куповање и продају исте криптовалуте на две различите мењачнице.

Троугаона (или кружна) арбитража – (енгл. Triangular arbitrage) је стратегија трговања која користи разлике у ценама између три (у неким случајевима и више) криптовалуте.

Временска арбитража - (енгл. *Time arbitrage*) захтева праћење цене једне криптовалуте на једној мењачници како би се искористиле краткорочне промене у цени.

Арбитража унутар мењачнице – (енгл. *Inter-exchange arbitrage*) је стратегија која користи разлике у ценама различитих криптовалута на истој мењачници. Захтева идентификацију парова криптовалута који имају корелацију у цени и брзо извршавање трговина када дође до одступања у цени у односу на корелацију [5].

3. МОБИЛНЕ АПЛИКАЦИЈЕ

Мобилне апликације су наменски програми дизајнирани за рад са преносивим уређајима као што су паметни телефони. Појединачно омогућавају ограничен скуп функционалности које корисник може да комбинује и прилагоди својим потребама. [6]

3.1. Развој и испитивање мобилних апликација

Развој наменских апликација је поступак израде програмске подршке за преносиве уређаје узимајући у обзир могућности, ограничења физичке архитектуре и системског софтвера за дату врсту уређаја као што су: верзија оперативног система, количина меморије, процесорска моћ, величина екрана, итд. Процес развоја апликација подразумева дефинисање захтева, употребе и алата неопходних за њену имплементацију, дизајн корисничког интерфејса, дефиницију архитектуре и модела комуникације.

3.2 iOS и iOS мобилне апликације

iOS је оперативни систем намењен за преносиве уређаје који је развила компанија *Apple* и потекао је од *Mac OS X*. iOS подржава ограничен број уређаја у односу на друге мобилне оперативне системе, што му даје бољу прилагођеност физичкој архитектури, безбедност, као и само корисничко искуство [7].

iOS апликација је појам који обухвата широку групу програма израђених искључиво за уређаје које производи компанија *Apple* (*iPod*, *iPhone*, *iPad*). У већини случајева iOS апликације су затвореног кода. Одликује их једноставност употребе Пишу се у *Swift* и *Objective-C* програмским језицима. Велики број апликација је написан уз помоћ *UIKit* радног оквира који омогућава употребу елемената корисничког интерфејса и препознавање корисничких команди, али све је више популаран *SwiftUI* радни оквир.

4. SWIFT ПРОГРАМСКИ ЈЕЗИК

Swift је програмски језик отвореног кода и опште намене објављен од стране компаније *Apple* 2014. године. Главне одлике овог програмског језика су брзина извршавања, висок ниво апстракције као и дескриптивна синтакса приликом писања програма. Поред тога, сигурност типа података (енгл. *Type safety*) чини овај програмски језик погодним за писање мобилних апликација.

Данас се овај програмски језик најчешће користи за израду апликација на *iOS*, *macOS* и осталим платформама компаније *Apple*, али и за развој сервера у рачунарским мрежама [8].

4.1 SwiftUI

SwiftUI је радни оквир намењен за израду корисничког интерфејса који омогућава развој употребом декларативне синтаксе и напредних алата за дизајн. *SwiftUI* је направљен тако да функционише упоредо са старијим радним оквирима као што су *UIKit* и *AppKit*, што додатно олакшава његову употребу како у новим, тако и у постојећим пројектима [9].

4.2. CoreData

Core Data је граф објеката и радни оквир намењен за обраду и стално складиштење података на *macOS* и *iOS* платформама. Користи се за привремено и трајно чување података на, синхронизацију података између више уређаја и сл. Као базу података користи *SQLite*, а имплементација овог радног оквира омогућава једноставну манипулацију објектима без директним управљањем базама података [10].

5. ДИЗАЈН И РАЗВОЈ КОРИСНИЧКОГ ИНТЕРФЕЈСА

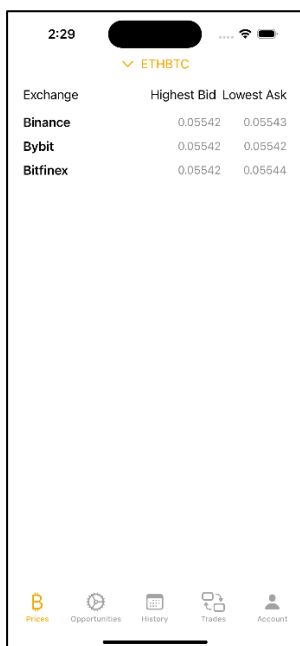
Кориснички интерфејс апликације је подељен на пет главних целина које омогућавају кориснику ефикасан преглед и конфигурацију апликације. Прва целина је приказ цена, односно екран који има задатак да прикаже цену изабраног пара криптовалута на свим подржаним тржиштима. У другој целини корисник има могућност да конфигурише понашање апликације, док трећа и четврта целина представљају прегледе историје цена, односно извршених трговина током арбитражног трговања. Четврта целина садржи додатна подешавања апликације.

5.1. MVVM шаблон

Model-View-ViewModel је структурни шаблон који се користи за раздвајање програмске логике од корисничког интерфејса. *MVVM* је користан приликом организације кода и рашчлањивања програма на модуле како би се поједноставила и убрзала надоградња и поновна употреба постојећег кода. Често се користи приликом израде мобилних и web апликација. На слици 1 може се видети изглед једне компоненте корисничког интерфејса [11].

Програмски код је организован у три компоненте:

- Приказ – (енгл. *View*) је скуп елемената корисничког интерфејса задужен за приказивање и прикупљање корисничких команди.
- Модел – садржи програмску логику и приступа му модел приказа приликом пријема корисничких команди.
- Модел Приказа – компонента која се налази између модела и приказа која садржи контроле за интеракцију са приказом и повезује елементе приказа са подацима који се налазе у моделу.



Слика 1- Приказ Дела Корисничког Интерфејса

6. ОРГАНИЗАЦИЈА ПРОГРАМСКОГ КОДА

Програмски код је подељен на сегменте који омогућавају његову прегледност и проширивост. Посебно су издвојене компоненте које обухватају структуре података неопходне за анализу цена, парова криптовалута, типова понуда итд.

6.1 Singleton образац

Singleton је структурни шаблон који се користи за креирање глобално јединствене инстанце класе или структуре података. Користан је када је потребно обезбедити синхронизован приступ и обраду података током конкурентног извршавања програма [12].

Компоненте које користе овај образац су:

- *KeychainManager* – Класа задужена за безбедно чување, добављање и брисање кључева дигиталних новчаника као и кључева за аутентификацију приликом комуникације са мењачницама.
- *LoginManager* – Класа која омогућава регистрацију, пријаву и одјаву корисничког налога
- *SettingsManager* – Структура података чија јединствена инстанца служи за управљање подешавањима.
- *DatabaseManager* – Ова класа служи за операције над базом података како би се обезбедила функционалност у деловима апликације који захтевају трајно складиште. Ослања се на *CoreData*

6.2. Решење

Решење предложено у овом раду представља апликацију израђену помоћу технологија наведеним у претходном поглављу. Имплементација рада дата је у програмском језику Swift. За имплементацију коришћено је развојно окружење *Xcode*, а за само испитивање решења коришћен је *iOS* симулатор.

Тренутни захтев за правилан рад апликације је уређај који подржава *iOS* верзију 16.4 или вишу, или уређај који подржава *macOS Catalist* технологију.

Поред приступачног корисничког интерфејса који пружа, решење преузима понуде криптовалута на подржаним мењачницама, затим анализира куповну и продајну цену проналазећи прилике за трговину.

Тренутна имплементација подржава арбитражу између мењачница и кружну арбитражу на једној мењачници.

По проналаску на довољно велику разлику у ценама, апликација обавештава корисника путем корисничких обавештења *iOS* оперативног система, чува евиденцију о догађају у бази података, и уколико је то дозвољено, шаље захтев за аутоматску објављивање куповине и продаје одабраним мењачницама.

Тренутна изведба апликације подржава следеће мењачнице:

- *Binance*
- *Bybit*
- *Bitfinex*
- Децентрализоване мењачнице на *Hive* блокчејну

Испитивање апликације своди се на аутоматско тестирање појединачних модула програмског кода, и на ручно тестирање корисничког интерфејса на симулатору више различитих уређаја.

Такође, велики изазов за овакву врсту мобилне апликације јесте потреба за константном интернет везом, што није увек могуће због варијација у мрежном сигналу мобилних уређаја.

Поред тога, учестане промене цена и ниска учестаност појављивања прилика за арбитражно трговање са високом стопом профита учинили су време испитивања рада апликације дужим. Понашање апликације је посматрано током различитих стања мреже и самог оперативног система како би се обезбедила њена стабилност и робусност.

Додатни изазов је представљала чињеница да појединци и организације које се баве развојем алата за арбитражно трговање неретко одлучују да резултате свог рада не деле јавно како би задржали предност на тржишту, што као последицу има мањак истраживања и јавних имплементација решења, нарочито у области мобилних апликација.

7. ЗАКЉУЧАК

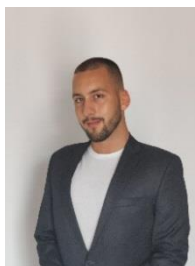
Почетна верзија апликације за арбитражно трговање пружа могућност једноставног аутоматизованог трговања на ограниченом броју тржишта.

Иако услови које арбитражна трговина криптовалутама често захтева испуњавање услова које није увек могуће задовољити на мобилним уређајима, овај рад показује да је могуће извршити аутоматизацију детекције и извршавања трговине такве да се током рада апликације оствари приход.

8. ЛИТЕРАТУРА

- [1] Geoffrey Poitras: *Arbitrage: Historical Perspectives*, Simon Fraser University, 2010
- [2] *Digital Asset: Meaning, Types, and Importance* – investopedia, приступано 19.11.2023. Доступно на: <https://www.investopedia.com/terms/d/digital-asset-framework.asp>
- [3] Сатоши Накамото, „*Bitcoin: A peer-to-peer electronic cash system*“, 2008, Приступано 21.11.2023. Доступно на: <https://bitcoin.org/bitcoin.pdf>
- [4] crypto exchange – PC mag, приступано 20.11.2023. Доступно на: <https://www.pcmag.com/encyclopedia/term/crypto-exchange>
- [5] Crypto Arbitrage Trading - Coindesk, приступано 20.11.2023. Доступно на: <https://www.coindesk.com/learn/crypto-arbitrage-trading-what-is-it-and-how-does-it-work/>
- [6] Лука Ступар, Дипломски рад, Имплементација механизма за логовање на ELK сервер у мобилним апликацијама – iOS, Факултет техничких наука, Универзитет у Новом Саду, 2021.
- [7] Процес развоја мобилних апликација, приступано 19.11.2023. Доступно на: <https://www.invonto.com/insights/mobile-app-development-process/>
- [8] *Swift*, приступано 21.11.2023. Доступно на: <https://www.swift.org/about/>
- [9] *SwiftUI – Apple developer*, приступано 21.11.2023. Доступно на: <https://developer.apple.com/xcode/swiftui/>
- [10] *CoreData – Apple developer*, приступано 21.11.2023. Доступно на: <https://developer.apple.com/documentation/coredata/>
- [11] *Model-View-ViewModel (MVVM) – Techtarget*, приступано 21.11.2023. Доступно на: <https://www.techtarget.com/whatis/definition/Model-View-ViewModel>
- [12] Singleton Design Pattern, Приступано 21.11.2023. Доступно на: <https://www.geeksforgeeks.org singleton-design-pattern/>

Кратка биографија:



Лука Ступар рођен је у Новом Саду 1998. год. Дипломски рад на Факултету техничких наука из области Електротехника и рачунарство – Рачунарство и аутоматика одбрањено је 2021. године.
контакт: stuparbluka@protonmail.com