Simulator trgovanja kriptovalutama

Bartulović, Ante

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:939800

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-20**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

SIMULATOR TRGOVANJA KRIPTOVALUTAMA

Završni rad

Ante Bartulović

Osijek, 2018.

Sadržaj

1. UVOD 1

1.1. Zadatak završnog rada 1

2. KORIŠTENE TEHNOLOGIJE 2

2.1. Android operacijski sustav 2

2.2. Android Studio 3

2.3. Kotlin 3

2.4. Aplikacije za trgovanje 3

2.5. Kriptovalute 4

2.6. Burze kriptovaluta 5

3. MOBILNA APLIKACIJA ZA TRGOVANJE KRIPTOVALUTAMA „EUPHORIA“ 6

3.1. MVP 6

3.2. API 8

3.3. Baza podataka 11

3.4. Kupovina i prodaja kriptovaluta 12

3.5. Portfelj 14

4. TESTIRANJE I USPOREDBA APLIKACIJE „EUPHORIA“ S VEĆ POSTOJEĆIM RJEŠENJIMA 16

4.1. Testiranje aplikacije 16

4.2. TradeHero 19

4.3. WallStreet Survivor 20

4.4. Coinbase 23

5. ZAKLJUČAK 24

LITERATURA 25

SAŽETAK.....27

ABSTRACT.....28

ŽIVOTOPIS.....29

1. UVOD

Tema završnog rada je osmisliti i implementirati mobilnu aplikaciju na operacijskom sustavu Android, koja omogućuje korisniku simulaciju trgovanja kriptovaluta korištenjem virtualnog novca. Osnove funkcionalnosti aplikacije su pregled cijena 10 najvećih kriptovaluta korištenjem coinmarketcap.com API-a, mogućnost kupovine i prodajte tih istih kriptovaluta korištenjem virtualnog novca, pregled portfelja korisnika s određenim karakteristikama.

Ovakva aplikacija omogućuje korisnicima jednostavno i sigurno sudjelovanje na virtualnom tržištu bez korištenja vlastitih financijskih sredstava. Ideja je omogućiti korisnicima Android operacijskog sustava trgovanje bez rizika, tj. predočiti im način rada burzi kriptovaluta većinom mlađim korisnicima koji bi jednog dana htjeli sudjelovati na otvorenom tržištu, te im na ovaj način omogućiti svojevrsnu edukaciju, odnosno pripremu.

U poglavlju 2. bit će opisane tehnologije korištene za uspješnu realizaciju zadatak završnoga rada, kratak opis i osnovne funkcionalnosti svih tehnologija koje se koriste u ovome radu, primjer korištenih biblioteka, modela i funkcija. Osim tehnologija, opisat će se i način rada kriptovaluta, te burzi i mjenjačnica kriptovaluta uz primjere najvećih burzi, odnosno mjenjačnica. U poglavlju 3. prikazat i opisat će se osnovne funkcije aplikacije izrađene u okviru završnog rada. Poglavlje 4. analizira realizaciju mobilne aplikacije pod nazivom „Euphoria“, testiranje i usporedbu sa postojećim rješenjima.

1.1. Zadatak završnog rada

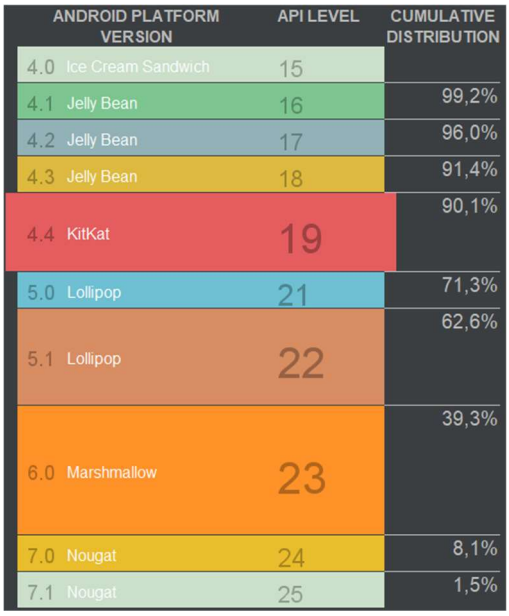
Koristeći Android Studio i programski jezik Kotlin, potrebno je osmisliti i napisati mobilnu aplikaciju za simulaciju trgovanja kriptovaluta. Osnovne funkcionalnosti aplikacije su spajanje na vanjskog poslužitelja te dohvaćanje podataka u stvarnom vremenu koristeći aplikacijsko programsko sučelje (API), prikazivanje cijena na stvarnom tržištu, omogućiti korisniku kupovinu i prodaju kriptovaluta po trenutnim cijenama na simuliranom tržištu, te pregled portfelja. Opisati osnovne principe rada Android Studija, te samog Android operacijskog sustava i tehnologije koje su korištene u izradi ovoga rada.

2. KORIŠTENE TEHNOLOGIJE

U ovom poglavlju ukratko će se nabrojati i opisati tehnologije korištene za realizaciju zadatka korištenjem aplikacije namijenjene Android operacijskom sustavu, konkretno Android pametnim telefonima s verzijom 4.4+ (KitKat i nasljednici). Kao integrirano razvojno okruženje (engl. *Integrated Development Environment*, IDE) korišten je Android Studio, službeni Android IDE. Programski kod pisan je u programskom jeziku Kotlin, a grafičko sučelje u jeziku XML (engl. *EXtensible Markup Language,* XML*)*. Za cijene, korišteno je aplikacijsko programsko sučelje (engl. *Application Programming Interface*, API) web stranice **coinmarketcap.com**, a za bazu podataka korišten je Room, biblioteka za konfiguriranje i korištenje baze podataka unutar Androida.

2.1. Android operacijski sustav

Android operacijski sustav, kao platforma za korištenje aplikacije, odabrana je radi najveće zastupljenosti na tržištu unutar posljednjih 12 mjeseci, statistika dostupna na [1], te radi jednostavnosti samog procesa kreiranja aplikacije uz pomoć Android Studija. Najniža verzija operacijskog sustava na kojoj će aplikacija raditi je Android KitKat 4.4 jer obuhvaća 90,1% svih uređaja baziranih na Androidu prema službenim podacima dostupnim na slici 2.1.



Sl. 2.1. Zastupljenost verzije Android operacijskog sustava na mobilnim uređajima.

2.2. Android Studio

Android Studio je službenoazvojno okruženje razvijeno za potrebe dizajniranja Android aplikacija od strane Googlea, kreatora Android operacijskog sustava, te JetBrainsa, kreatora IntelliJ IDEA razvojnog okruženja na kojem je baziran Android Studio. Osim Android Studija, na tržištu postoji veliki broj Android IDE-ova ovisno o korištenoj tehnologiji. Najpoznatiji IDE-ovi uz Android Studio su: Visual Studio, IntelliJ IDEA, Eclipse i NetBeans. Osim mogućnosti razvoja mobilnih aplikacija, Android Studio nudi i mogućnost razvoja aplikacija za pametne satove, tablete, pametne televizore i aplikacije za Android Auto. Prednosti Android Studija naspram spomenutih je osim službene podrške Googlea i mogućnost simuliranja aplikacije koristeći službeni Android emulator koji je ugrađen unutar Android Studija. Android Studio omogućuje korištenje i pregledavanje predložaka (engl. *Layout*), odnosno grafičkog korisničkog sučelja (engl. *Graphical User Interface*, GUI), dostupan je na mnogim platformama te se jednostavno integrira u Googleove servise.

2.3. Kotlin

Android Studio nudi opciju pisanja programskog koda aplikacije u više različitih jezika, primarno Java, no aplikacija u sklopu ovog rada će u potpunosti biti napisana u programskom jeziku Kotlin, razvijenog od strane JetBrainsa za potrebe Android operacijskog sustava, te u potpunosti podržan od strane Googlea. Glavne prednosti Kotlina nad Javom su sigurnost od nedostatka podataka (engl. *null safety*), te nedostatak bespotrebnog programskog koda (engl. *boilerplate code*), a na službenoj stranici Kotlina dostupnoj na [2], navodi se da je količina programskog koda potrebna za jednaku radnju manja do 40%. Kotlin je relativno novi jezik s brzo rastućom zajednicom kako u svijetu, tako i u Hrvatskoj. Izvršavanje Kotlina odvija se u Java virtualnom stroju (engl. *Java Virtual Machine*, JVM), te je u potpunosti tibilan s programskim jezikom Java, što omogućava kombiniranje Jave i Kotlina unutar istog projekta.

2.4. Aplikacije za trgovanje

U posljednjih nekoliko godina s razvojem mobilnog sklopovlja i tehnologija korištenih za dizajniranje mobilnih aplikacija, omogućeno je kreiranje sofisticiranih programskih rješenja za trgovanje na raznim burzama korištenjem mobilnih uređaja. Sve veće tvrtke namijenjene trgovanju u svom portfelju posjeduju mobilne aplikacije za tu svrhu. Jedna od najpoznatijih aplikacija

namijenjena trgovanju je zasigurno Robinhood, koja je ove godine prema [3] prešla brojku od 4 milijuna korisnika, a omogućava trgovanje dionicama i kriptovalutama bez plaćanja provizije. Još od poznatijih mobilnih rješenja na tržištu su E*Trade sa 3,7 milijuna korisnika i TD Ameritrade Mobile. Osim aplikacija koje sudjeluju na stvarnom tržištu, postoje i aplikacije koje simuliraju tržište, te su većinom namijenjene edukaciji korisnika. Primjer takve mobilne aplikacije je TradeHero koja prema [4] broji 3,6 milijuna korisnika u cijelom svijetu. TradeHero je odličan primjer edukativne aplikacije i od nedavno nude simulaciju trgovanja kriptovalutama. Osim TradeHero-a, postoji još i Wall Street Survivor, web aplikacija koja simulira Wall Street burzu, dostupna na [5].

2.5. Kriptovalute

Svoju povijest kriptovalute započinju 1993. godine pojavom ideje o korištenju procesorske moći kako bi se izbjegli DDoS (engl. *Distributed Denial-of-service attack*) napadi na mrežu. Protokol za zaštitu nazvan je „Proof of Work“ ili „Proof of computational effort“, a osnovni princip rada je izvršavanje kompleksnog zadatka na procesoru računala koje je poslalo zahtjev za nekom uslugom na mreži, kratko opisana ideja rada dostupna je na [6]. Začetnici te ideje su Cynthia Dwork i Moni Naor, a tom idejom su omogućili kreiranje „Reusable Proofs of Work“ dostupnog na [7] protokol koji je kreiran od strane Hal Finneyja 2004. godine baziran na radu Nicka Szabe „Shelling Out: The Origins of Money“ koji je dostupan na [8]. Još kao inspiraciju za kriptovalute bitno je spomenuti rad Wei Dai-a iz 1998. godine „B-Money“ dostupnog na [9] koji opisuje osnovne principe rada današnjih kriptovaluta, te „Bit Gold“, rad Nicka Szabe iz 2005. godine, dostupnog na [10].

Prva prava kriptovaluta, naziva Bitcoin, nastaje 3.1.2009. godine izlaskom rada „Bitcoin: A Peer-to-Peer Electronic Cash System“, rad je dostupan u [11], od strane autora pod pseudonimom Satoshi Nakamoto, te rudarenjem Genesis bloka Bitcoin distribuirane javne knjige (engl. *Distributed Ledger, Blockchain*). Bitcoin koristi Blockchain tehnologiju za spremanje i arhiviranje svih transakcija ikad izvršenih na mreži. Blockchain se još može nazvati distribuiranom javnom knjigom (engl. *Distributed ledger*) transakcija. Najveće prednosti kriptovaluta su korištenje kriptografije javnog-privatnog ključa za sigurnost vlastitih sredstava na mreži, te rješavanje problema Bizantskih generala i duple potrošnje (engl. *Double spending*). Bitcoini se generiraju „rudarenjem“, a posao „rudara“ je potvrđivanje i zapisivanje transakcija unutar Blockchaina. Za proces rudarenja rudari su nagrađeni novootkrivenim Bitcoinima. Osnovne

prednosti Bitcoina su mogućnost jednostavnog korištenja, niski troškovi transakcija bilo gdje u svijetu, transparentnost transakcija, anonimnost, decentraliziranost, jednom poslanu transakciju u Blockchain nije moguće poništiti. Osim kriptovaluta, Blockchain tehnologija se može koristiti i u pravnim znanostima; primjer Pametni ugovori (engl. *Smart Contract*), dokaz o vlasništvu (engl. *Proof of Ownership*), te kao sustav za korištenje tijekom demokratskih procesa.

Od pojave Bitcoina pa do danas, na tržištu postoji preko 1 500 različitih kriptovaluta namijenjene različitim potrebama, prema podatcima dostupnim na [12].

2.6. Burze kriptovaluta

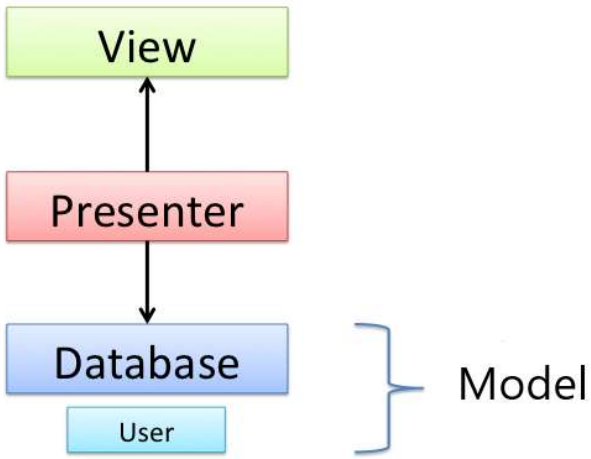
Trgovanje kriptovalutama obično se odvija na raznim burzama specijaliziranim za takvu vrstu razmjene, te za razliku od burzi dionica, burze kriptovaluta rade svaki dan u tjednu, bez prestanka. Postoje dvije kategorije burzi, to su mjenjačnice na kojima se kriptovalute (većinom Bitcoin, Ethereum i Litecoin) kupuju korištenjem tradicionalnih valuta (engl. *FIAT Money*) i burze na kojima se koristeći kriptovalute trguje drugim kriptovalutama bez korištenja tradicionalnih valuta. Većina burzi trguje isključivo kriptovalutama, dok neke nude i trgovanje tradicionalnim valutama, većinom Dolarom, Eurom i Yenom. Funkcionalnosti koje svaka mjenjačnica mora sadržavati su: mogućnost kupovine i prodaje kriptovaluta, mogućnost skladištenja kriptovaluta i/ili prebacivanja tih istih kriptovaluta na vlastiti digitalni novčanik (engl. *wallet*), te nalog za kupnju ili prodaju (engl. *stop limit*) koji omogućuje kupovinu ili prodaju kriptovaluta kada ona dostigne određenu vrijednost na tržištu. Sve navedene mogućnosti osim skladištenja i prebacivanja kriptovaluta, nudi brzorastuća američka tvrtka Robinhood, njene prednosti su neplaćanje provizije (engl. *comission*) za sudjelovanje na tržištu i jednostavnosti korištenja. Najveća i najkorištenija mjenjačnica za kriptovalute je Coinbase, nastala 2012. godine. Radi svog jednostavnog grafičkog sučelja, mobilne aplikacije i mogućnosti kupovine kriptovaluta sa tradicionalnim valutama, postala je najveća mjenjačnica s 13 300 000 registriranih korisnika, izvor broja korisnika je dostupan na [13]. Među burzama kriptovaluta, najpoznatije u svijetu su Bitstamp, Bitmex, Poloniex, Binance i Bitfinex, dok ih u svijetu postoji 210, zaključno sa 17.06.2018. godine, preuzeto sa [14]. Većina velikih burzi kriptovaluta također podržava trgovanje putem mobilnih aplikacija, što korisniku omogućuje stalno praćenje i trgovanje.

3. MOBILNA APLIKACIJA ZA TRGOVANJE KRIPTOVALUTAMA „EUPHORIA“

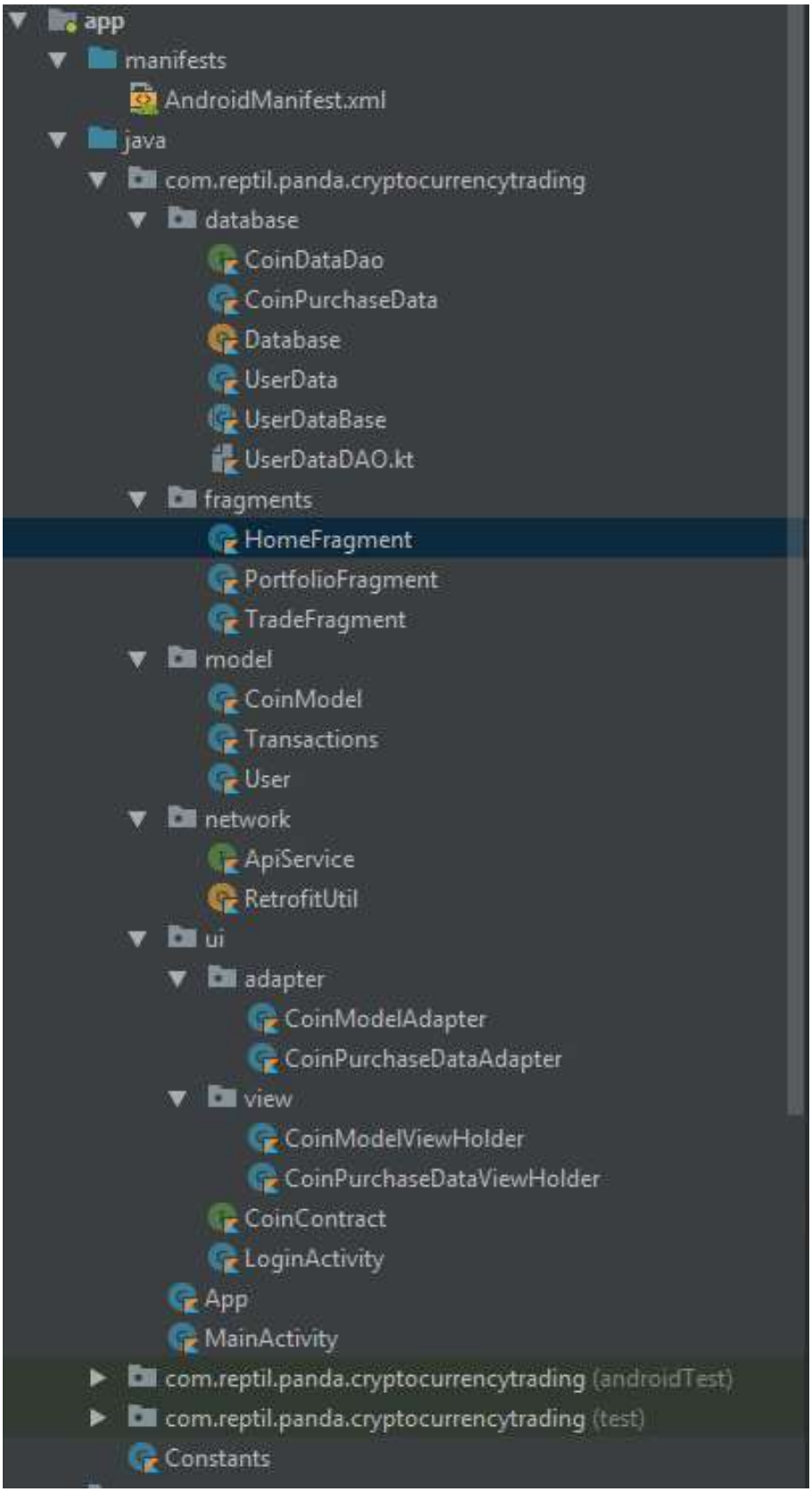
U ovom poglavlju bit će opisani arhitektura aplikacije Euphoria, potrebne funkcionalnosti aplikacije kako bi korisniku omogućila osnovne mogućnosti trgovanja na burzi, pregled portfelja, te prosječne cijene kriptovaluta preuzetih sa coinmarketcap.com stranice koristeći API servis.

3.1. MVP

MVP ili Model-view-presenter je oblikovni obrazac namijenjen Android aplikacijama. Koristi se za strukturiranje koda radi lakšeg uređivanja. Model je klasa sa opisom objekta, sadrži osnovne podatke o objeku ili bazi podataka. Pogled (engl. *View*) predstavlja GUI sloj baziran na Android platoformi. Presenter predstavlja logiku i rad sa podacima, odnosno sloj koji povezuje model i pogled. Shema obrasca prikazana je na slici 3.1., a struktura direktorija rada na slici 3.2.



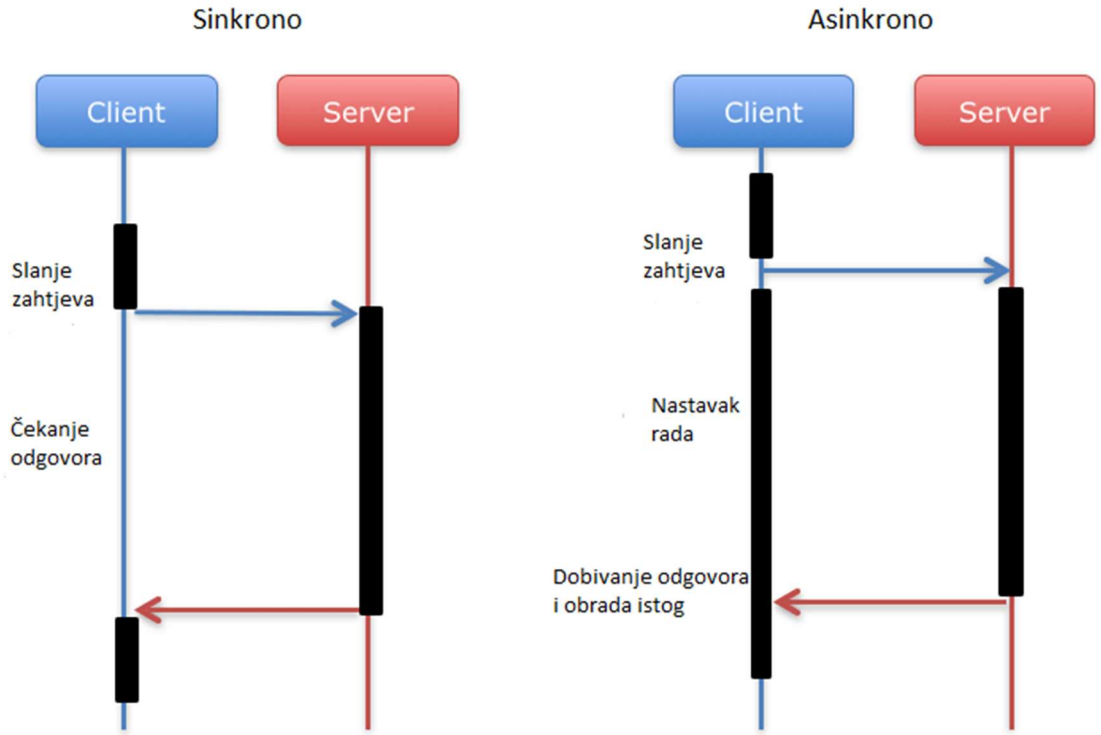
Sl. 3.1. Model View Presenter obrazac



Sl. 3.2. Struktura direktorija aplikacije Euphoria

3.2. API

API je definirano programersko sučelje koje omogućava korisnicima pristup uslugama i servisima napisanih od strane drugih programera ili tvrtki. API je skup mrežnih pravila koja opisuju komunikaciju između korisnika i računala, aplikacije ili poslužitelja, a najčešće dolazi u obliku biblioteke. Postoji nekoliko različitih tipova API-a, ali najkorišteniji su WebSocket i RESTful (engl. *Representational State Transfer*) API. Postoji više metoda za rad s aplikacijskim programskim sučeljem, no u ovom radu će se koristiti HTTP GET metoda za dohvaćanje podataka sa poslužitelja. API servis ima mogućnost rada na dva načina: sinkrono i asinkrono. Asinkrono za razliku od sinkronog tijekom čekanja odgovora poslužitelja nastavlja s radom, dok sinkroni čeka odgovor poslužitelja i nakon odgovora nastavlja raditi, kao što je prikazano na slici 3.3.



Sl. 3.3. Razlika modela sinkronog i asinkronog zahtjeva

Za početak, kako bi uopće postojala ikakva funkcionalnost aplikacije Euphoria, potrebno je spojiti se na API servis **coinmarketcap.com** web stranice. U slučaju spajanja na API servis coinmarketcap.com web stranice mobilna aplikacija je klijent, dok je **coinmarketcap.com** poslužitelj. Klijent šalje GET poziv za dohvaćanje podataka poslužitelju, poslužitelj prima zahtjev,

te klijentu šalje informacije. Za dohvaćanje podataka koristi se HTTP GET metoda unutar Retrofit biblioteke, čija je dokumentacija dostupna na [15]. Retrofit koristi HTTP API, te omogućava korištenje navedenog API-ja unutar Androida.

Kako bi se koristila biblioteka Retrofit potrebno je kreirati klase ApiService i RetrofitUtil koje dohvaćaju podatke sa poslužitelja CoinMarketCap.

a) Klasa ApiService.kt:

```
interface ApiService {

    @GET("/v1/ticker/?start=0&limit=10")
    fun getCoinData(): Flowable<List<CoinModel>>

    @GET("/v1/ticker/{id}")
    fun getCoin(@Path("id") coinId: String): Flowable<List<CoinModel>>
}
```

b) Klasa RetrofitUtil.kt:

```
object RetrofitUtil {

    private val retrofit by lazy { createRetrofit() }

    val apiService: ApiService by lazy { retrofit.create(ApiService::class.java) }

    private fun createRetrofit(): Retrofit {
        return Retrofit.Builder()
                .baseUrl(Constants.COIN_BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
                .build()
    }
}
```

c) Klasa Constants.kt:

```
const val COIN_BASE_URL = „https://api.coinmarketcap.com“
```

Sl. 3.4. a), b), c) Programski kod API servisa

Na slici 3.4. pod a) prikazano je dohvaćanje 10 najvećih kriptovaluta sa servisa, a to je opisano unutar GET naredbe. „start“ označava početak liste, a „limit“ količinu kriptovaluta koja se dohvaća, poredane po ukupnoj vrijednosti na tržištu (engl. *Market cap*).

Varijabla „COIN_BASE_URL“, prikazana na slici 3.4. pod c) čuva glavnu adresu poslužitelja koja se koristi unutar klase „RetrofitUtil.kt“, koja se nalazi na slici 3.4. pod b), dok se putanja na kojoj se nalaze potrebne informacije stavlja unutar „GET“ naredbe.

Odgovor poslužitelja dobiven je u JSON (engl. *JavaScript Object Notation*) obliku, te ga je potrebno rasčlaniti (engl. *parsing*). Uloga rasčlanjivanja je prevođenje podataka u nativni kod. Na

slici 3.5., pokazat će se primjer dolazne informacije korištenjem koda navedenog na slikama 3.4 a), b) i c).

```
{
        "id": "bitcoin",
        "name": "Bitcoin",
        "symbol": "BTC",
        "rank": "1",
        "price_usd": "7036.03005894",
        "price_btc": "1.0",
        "24h_volume_usd": "4581314548.84",
        "market_cap_usd": "121275737040",
        "available_supply": "17236387.0",
        "total_supply": "17236387.0",
        "max_supply": "21000000.0",
        "percent_change_1h": "-0.3",
        "percent_change_24h": "4.11",
        "percent_change_7d": "9.03",
        "last_updated": "1535475026"
    }
```

Sl. 3.5. Informacije dobivene od poslužitelja

Kako bi se omogućilo korištenje varijabli preuzetih sa servisa, potrebno ih je napraviti klasu koja služi kao model, odnosno napraviti rasčlanjivanje. Kod rasčlanjivanja potrebno je paziti na upotrebu pravilnog tipa podatka za varijable, kao što je navedeno kodu dostupnom na slici 3.6.

```
class CoinModel {

    @SerializedName("id")
    val id: String? = null
    @SerializedName("name")
    val name: String? = null
    @SerializedName("symbol")
    val symbol: String? = null
    @SerializedName("price_usd")
    val priceUsd: Double? = null
    @SerializedName("24h_volume_usd")
    val volume_24h: String? = null
    @SerializedName("market_cap_usd")
    val market_cap_usd: String? = null
    @SerializedName("percent_change_1h")
    val percent_change_1h: String? = null
    @SerializedName("percent_change_24h")
    val percent_change_24h: String? = null
    @SerializedName("percent_change_7d")
    val percent_change_7d: String? = null

}
```

Sl. 3.6. Model kriptovalute napisan u Kotlinu

Pomoću navedenih kodova, omogućeno je korištenje API servisa, te rada s dostupnim podacima. Kako bi se pozvale varijable dobivene pomoću API servisa, koristimo rasčlanjene varijable. Primjerice za cijenu u dolarima koristimo varijablu „*priceUsd*“.

3.3. Baza podataka

Kako bi se korisniku omogućila kupovina, prodaja kriptovaluta i pregled vlastitog portfelja, potrebno je implementirati bazu podataka. Baza podataka implementirana je pomoću biblioteke Room koja koristi SQL naredbe za rad.

Za početak potrebno je kreirati tablicu pod nazivom „*userData*“, prikazana na slici 3.7., koja sadržava informacije o korisniku. Osnovne informacije potrebne za rad su „*id*“ korisnika, odnosno identifikacijski broj, te „*currentFiatAmount*“ odnosno trenutna raspoloživa količina novca iskazana u FIAT valuti dolaru.

```
1.  @Entity(tableName = "userData")
2.  data class UserData(@PrimaryKey(autoGenerate = true) var id: String = "",
3.                      @ColumnInfo(name = "USD Amount") var currentFiatAmount: Double =
     10000.0)
```

Sl. 3.7. Tablica „userData“

Kako bi se dobile informacije o korisniku, koristimo funkciju „*getUserDao*“, funkcija je prikazana na slici 3.8.

```
1.  fun getUserDao() = database.userDataDao()
```

Sl. 3.8. Funkcija „getUserDao“

Osim tablice sa podacima korisnika, potrebno je kreirati još jednu tablicu pod nazivom „coinData“ koja sadržava „id“ odnosno identifikaciju korisnika, „coinId“ odnosno identifikaciju kriptovalute, „*currentPrice*“ koji nam označava trenutnu cijenu kriptovalute u američkom dolaru, „amount“ koji nam označava količinu kriptovalute, te „timestamp“ odnosno vrijeme transakcije. Programski kod tablice dostupan je na slici 3.9.

```
1.  @Entity(tableName = "coinData")
2.  data class CoinPurchaseData(@PrimaryKey(autoGenerate = true) var id: Long? = null,
3.                              var coinId: String? = null,
4.                              var coinName: String? = null,
5.                              var userId: Long = 0,
6.                              @ColumnInfo(name = "USD price") var currentPrice: Double
     = 0.0,
7.                              @ColumnInfo(name = "Amount") var amount: Double = 0.0,
8.                              var timestamp: Long = 0)
```

Sl. 3.9. Tablica „coinData“

Kako bi se dobile informacije o transakciji, koristimo funkciju „*getCoinDao*“ koja je prikazana na slici 3.10.

```
1.  fun getCoinDao() = database.coinDataDao()
```

Sl. 3.10. Funkcija „getCoinDao“

3.4. Kupovina i prodaja kriptovaluta

Za kupovinu i prodaju kriptovaluta potrebno je korisniku dodijeliti novčana sredstva kako bi mogao sudjelovati na simuliranom tržištu. Burze imaju definirane cijene trgovanja, no unutar ove simulacija cijena trgovanja ne postoji, odnosno koristimo cijene dobivene sa API-ja koji je opisan u prethodnom poglavlju. Aplikacija prilikom pokretanja kreira novog korisnika te mu dodjeljuje virtualnih 10000 dolara za trgovanje.

```
1.  val userDao = Database.getUserDao()
2.
3.          val users = userDao.getAll()
4.
5.          if (users.isEmpty()) {
6.              userDao.insert(userData = UserData(UUID.randomUUID().toString(), 10000.0
    ))
7.
8.              val intent = Intent(this, MainActivity::class.java)
9.
10.             startActivity(intent)
11.         }
```

Sl. 3.11. Provjera postojanosti korisnika

Prilikom pokretanja aplikacije provjerava se postoji li već korisnik unutar memorije mobilnog telefona, ako korisnik ne postoji, kreira se novi korisnik te mu se dodjeljuju gore navedena sredstva. Provjera je prikazana na slici 3.11.

U slučaju da korisnik već postoji, aplikacija prelazi na „*MainActivity*.kt“, odnosno glavni prozor aplikacije kao što je prikazano na slici 3.12.

```
1.  if (users.isNotEmpty()){
2.
3.      val intent = Intent(this, MainActivity::class.java)
4.
5.      startActivity(intent)
6.  }
```

Sl. 3.12. Provjera prije prelaska na „MainActivity.kt“

Kupovina kriptovaluta je moguća ako korisnik ima dovoljno sredstava na računu kako bi mogao izvršiti transakciju.

```
1.  private fun buyCoin(coinModel: CoinModel, amount: Double, coinName: String) {
2.          val user = userDao.getAll().first()
```

```
3.
4.          val newPurchase = CoinPurchaseData(coinId = coinModel.id,
5.                  userId = user.id,
6.                  coinName = coinName,
7.                  amount = amount,
8.                  currentPrice = coinModel.priceUsd?.toDouble() ?: 0.0,
9.                  timestamp = System.currentTimeMillis()
10.         )
11.
12.         coinDataDao.storePurchase(newPurchase)
13.     }
```

Sl. 3.13. Funkcija „buyCoin“

Funkcija „*buyCoin*“, prikazana na slici 3.13., koristi se za kupovanje kriptovaluta korištenjem sredstava na računu korisnika. Klasa sadrži identifikaciju kriptovalute, identifikaciju korisnika, količinu kriptovalute i trenutnu cijenu, te vrijeme transakcije.

Funkcija „*buyCoin*“ pokreće se unutar funkcije „*onBuyClicked*“ koja pokreće kupovinu. Funkcija „*onBuyClicked*“ prikazana je na slici 3.14.

```
1.  private fun onBuyClicked() {
2.          val coin = coinName.text.toString()
3.          val amount = tradeAmount.text.toString().toDouble()
4.          val tradePrice = tradePrice.text.toString()
5.
6.          if (!tradePrice.isBlank()) {
7.              val user = userDao.getAll().first()
8.              val totalCost = tradePrice.toDouble() * amount
9.
10.
11.             if (user.currentFiatAmount > totalCost) {
12.                 user.currentFiatAmount = user.currentFiatAmount - totalCost
13.                 userDao.insert(user)
14.
15.                 currentCoin?.run { buyCoin(this, amount, coin) }
16.             }
17.         }
18.     }
```

Sl. 3.14. Funkcija „onBuyClicked“

Kako bi se trgovina izvršila, potrebno je zadovoljiti količinu novaca koju korisnik posjeduje, odnosno količina novaca koju korisnik posjeduje mora biti veća od cijene same transakcije. Ako je uvjet ispunjen, korisniku se omogućuje kupovina zadane količine kriptovaluta, u protivnom korisniku se ispisuje poruka na ekran da nije u mogućnosti kupiti navedenu količinu kriptovaluta, odnosno da nema dovoljno novčanih sredstava na računu kako bi izvršio transakciju. Na slici 3.14. prikazan je primjer provjere stanja korisnika.

Prodaja kriptovaluta je moguća uz provjeru stanja računa korisnika, ako korisnik ima količinu željnu za prodaju na svome računu omogućuje mu se prodaja kriptovaluta, uz to da je potrebno platiti troškove trgovanja na burzi. Troškovi trgovanja unutar ove simulacije ne postoje.

```
1.  private fun onSellClicked() {
2.          val coin = coinName.text.toString()
3.          val amount = tradeAmount.text.toString().toDouble()
4.          val tradePrice = tradePrice.text.toString()
5.
6.          if (!tradePrice.isBlank()) {
7.              val user = userDao.getAll().first()
8.              val myCoin = coinDataDao.getCoinsForUser(user.id).firstOrNull { it.coinI
    d == coin }
9.                      ?: return
10.
11.             if (myCoin.amount >= amount) {
12.                 val totalCost = tradePrice.toDouble() * amount
13.
14.                 user.currentFiatAmount += totalCost
15.                 userDao.insert(user)
16.
17.                 myCoin.amount -= amount
18.
19.                 coinDataDao.storePurchase(myCoin)
20.             }
21.         }
22.     }
```

Sl. 3.15. Funkcija „onSellClicked“

Na slici 3.15. prikazana je provjera količine kriptovaluta potrebna za izvršenje transakcije.

3.5. Portfelj

Nakon što je korisnik u mogućnosti kupiti i prodati određenu kriptovalutu, potrebno mu je omogućiti pregled stanja svih kriptovaluta koje posjeduje. Portfelj sadrži trenutno količinu sredstva izraženu u američkim dolarima, naziv i količinu kriptovaluta. Kako bi se koristio portfelj, potrebno je kreirati model portfolia. Za to je korištena biblioteka Room. Programski kod tablice „*coinData*“ pisan je u SQL-u i Kotlin, te je prikazan na slici 3.16.

```
1.  @Entity(tableName = "coinData")
2.  data class CoinPurchaseData(@PrimaryKey(autoGenerate = true) var id: Long? = null,
3.                              var coinId: String? = null,
4.                              var coinName: String? = null,
5.                              var userId: Long = 0,
6.                              @ColumnInfo(name = "USD price") var currentPrice: Double
     = 0.0,
7.                              @ColumnInfo(name = "Amount") var amount: Double = 0.0,
8.                              var timestamp: Long = 0)
```

Sl. 3.16. Tablica „coinData“

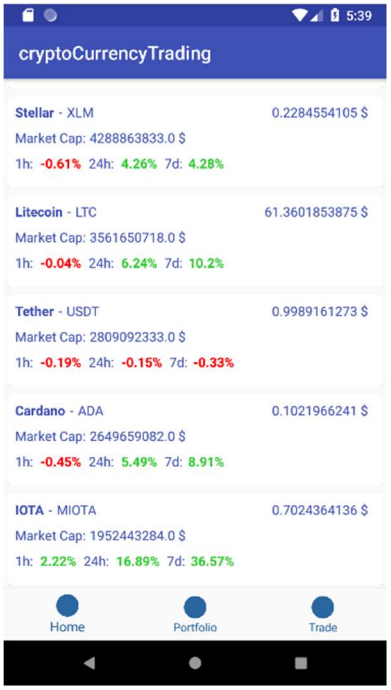
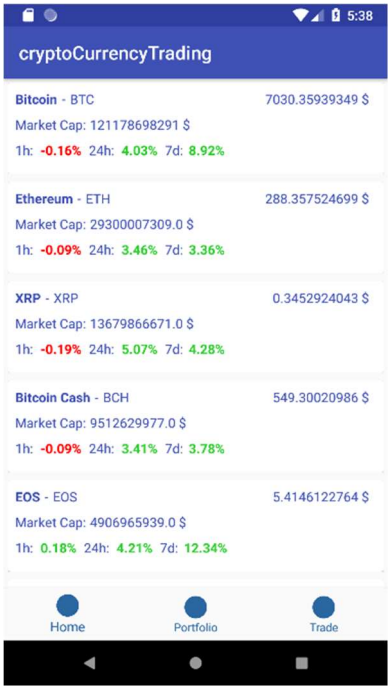
Unutar portfelja prikazane su informacije za svaku kriptovalutu koju korisnik posjeduje. Informacije koje se mogu dobiti su naziv, cijena kupovine, količina kriptovalute koju korisnik posjeduje, te vrijeme transakcije.

4. TESTIRANJE I USPOREDBA APLIKACIJE „EUPHORIA“ S VEĆ POSTOJEĆIM RJEŠENJIMA

U ovom poglavlju, analizirat će se već postojeća rješenja aplikacija za simuliranje trgovanja kriptovalutama prisutna na tržištu te njihove mogućnosti, nedostatke i moguće nadogradnje već postojećih sustava. Od postojećih rješenja na tržištu koja trenutno rade, dvije aplikacije nude simulaciju trgovanja kriptovalutama, TradeHero i WallStreet Survivor. Uz prethodno dvije nabrojane, spomenut ćemo i Coinbase mobilnu aplikaciju kao primjer jednostavne i brze aplikacije za sudjelovanje na stvarnom tržištu.

4.1.Testiranje aplikacije

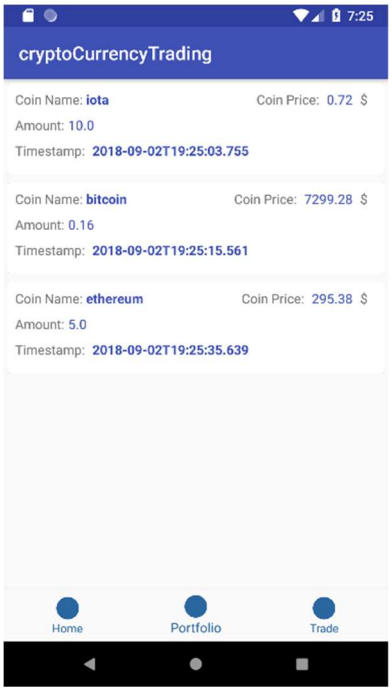
Testiranje vizalnom dijela aplikacije, prikaz podataka prikupljenih sa API servisa, vizulani prikaz portfelja iz baze podataka korisnika, te vizualni prikaz trgovanja, odnosno uspješnosti trgovanja. Prvi pogled aplikacije prikazuje informacije o kriptovalutama. Korištenjem API servisa, prikupili smo podatke o nazivu, kratici, cijeni, vremenskoj promjeni, te ukupnoj tržišnoj vrijednosti deset najvećih kriptovaluta.

a)	b)	c)
		

Sl. 4.1. a), b), c) Prikaz podataka prikupljenih sa API servisa

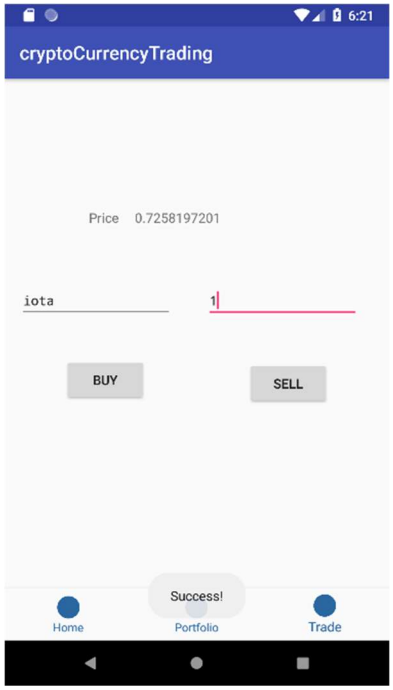
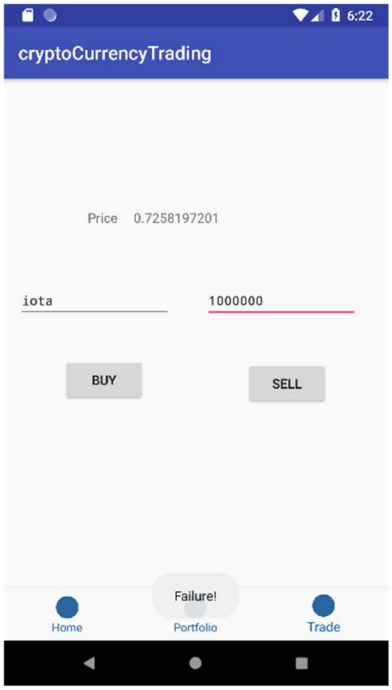
Na slikama 4.1. pod a), b) i c) mogu se vidjeti uspješno prikupljeni i prikazani podaci, te obojane vremenske promjene cijena kriptovaluta. Zelena boja označava pozitivnu promjenu cijene, dok crvena boja označava negativnu promjenu cijene.

Dugi pogled aplikacije prikazuje portfelj korisnika preuzet iz baze podataka. Pogled sadrži trenutnu količinu virtualnog novca u vlasništvu korisnika, te nazive, količinu i trenutnu vrijednost kriptovaluta. Pogledu se pristupa dodirom na „*Portfolio*“, a vizualni prikaz dostupan je na slici 4.2.



Sl. 4.2. Prikaz podataka portfelja preuzetih iz baze podataka

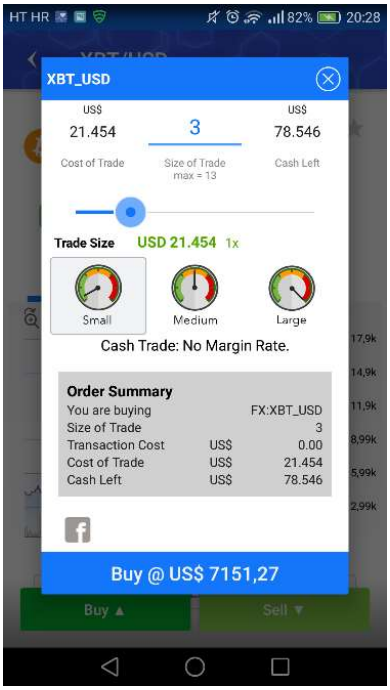
Treći pogled aplikacije prikazuje sučelje za trgovanje. Sučelje za trgovanje omogućava korisniku unos naziva kriptovalute i količine kriptovalute koju želi kupiti ili prodati koriteći gumbove „BUY“ ili „SELL“. Uz to sučelje prikazuje uspjeh ili neuspjeh transakcije u obliku poruka „Success!“ ili „Failure“. Na slikama 4.3. pod a) i b) prikazan je primjer odgovora koji korisnik dobiva nakon pokušaja trgovanja.

a)	b)
	

Sl. 4.3. a), b) Prikaz sučelja trgovanja, te uspjeha odnosno neuspjeha transakcije

4.2. TradeHero

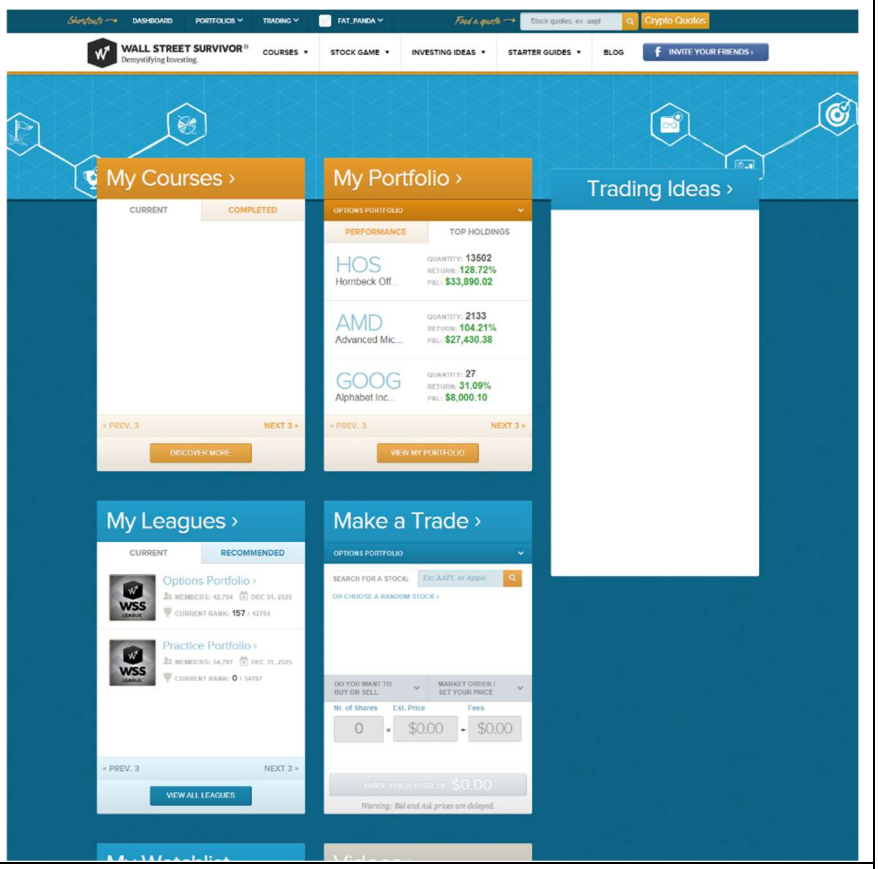
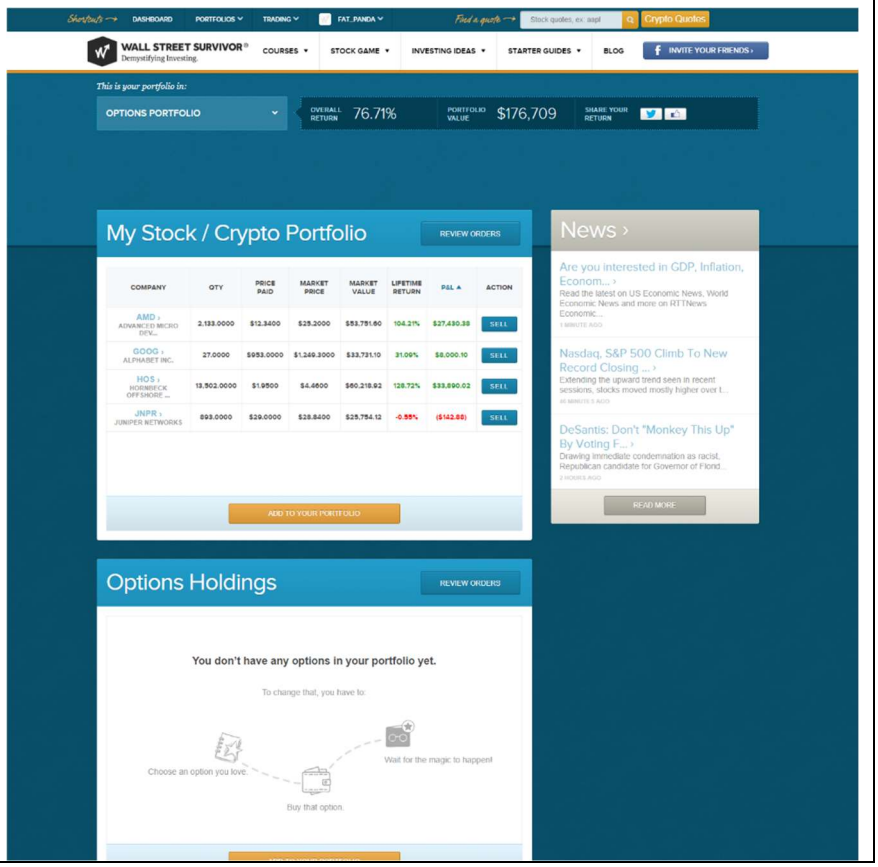
TradeHero je mobilna aplikacija namjena simulaciji trgovanja dionicama, resursima i određenim kriptovalutama. Aplikacije ima veliki broj funkcionalnosti poput edukacijskog segmenta, natjecateljskog segmenta, grafova i svijeća cijena resursa. TradeHero vrlo je jednostavna aplikacija za korištenje s puno mogućnosti. Prednosti aplikacije su mogućnost trgovanja dionicama, tradicionalnim valutama i plemenitim metalima. Osim navedenim mogućnosti, korisniku se nudi pregled povijesnih cijena trgovanja, pregled vijesti vezanih uz trgovanja i mogućnost diskusije s drugim korisnicima. Najveći nedostatak aplikacije je ne mogućnost trgovanja kriptovalutama u vrijeme kada je burza zatvorena, što se protivi osnovnim principima rada mjenjačnica odnosno burzi kriptovaluta. Prema slici 4.4. pod a). vidimo parove kriptovaluta koji su mogući za trgovanje, dok na slici 4.4. pod b) imamo prikaz informacija, rasprave, vijesti i povijesti para za trgovanje. Kod slika 4.4. pod c) prikazano je sučelje za odabir količine kriptovalute kojom se trguje, kliznik (engl. *slider*) je jedna od mogućih nadogradnji za aplikaciju Euphoria.

a)	b)	c)
		

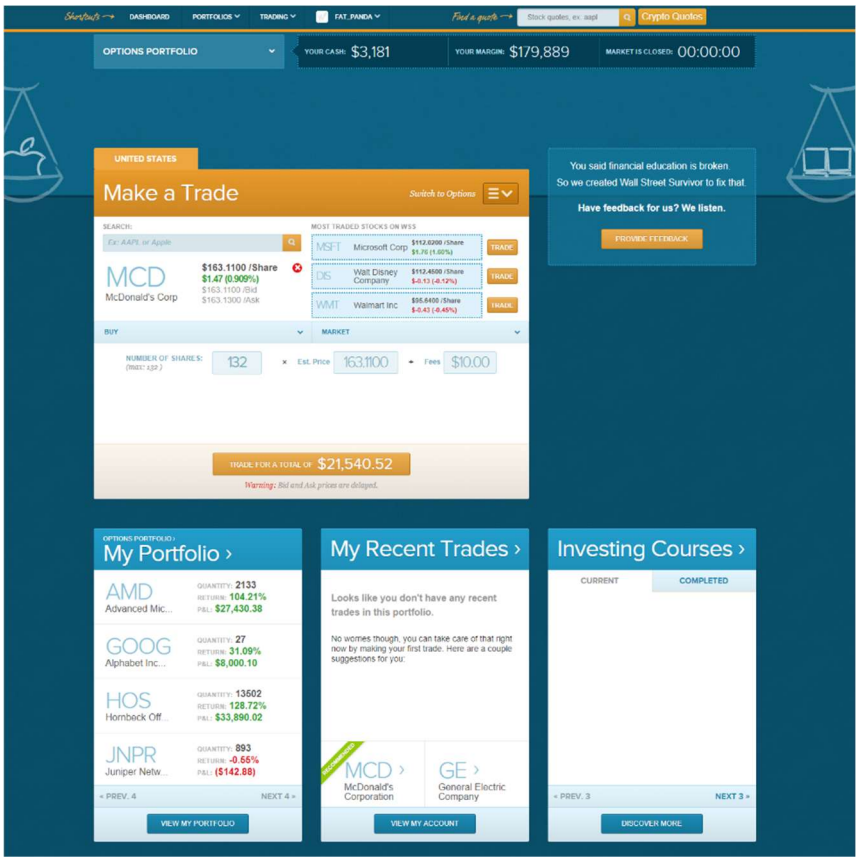
Sl. 4.4. a), b), c) Prikaz grafičkog sučelja aplikacije TradeHero

4.3. WallStreet Survivor

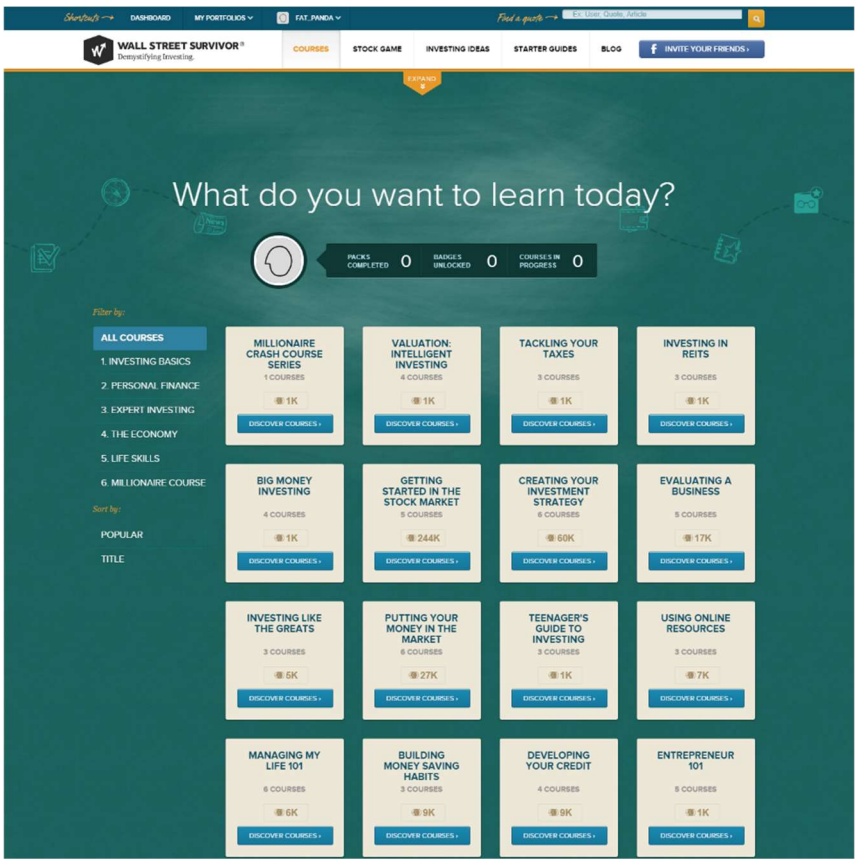
WallStreet Survivor nudi mogućnost simulacije trgovanja kriptovalutama i trgovanja dionicama na WallStreet burzi. Iako ne postoji zasebna mobilna aplikacija, web aplikacija prilagođena je korištenju na mobilnom uređaju. Aplikacija ima vrlo jednostavno i brzo korisničko sučelje uz veliki niz funkcionalnosti i mogućnosti. Nudi pregled portfelja, svih dosad odrađenih trgovanja, pregled tržišta i natjecanje u različitim vremenskim periodima i ligama. Osim simulaciji, aplikacija ima poseban dio posvećen edukaciji korisnika. Svrha same aplikacije, kako navode kreatori, je edukacija.

a)	 WALL STREET SURVIVOR Demystifying Investing. COURSES STOCK GAME INVESTING IDEAS STARTER GUIDES BLOG INVITE YOUR FRIENDS My Courses › CURRENT COMPLETED DISCOVER MORE My Portfolio › PERFORMANCE TOP HOLDINGS HOS Hornbeck Off... AMD Advanced Mic... GOOG Alphabet Inc... VIEW MY PORTFOLIO Trading Ideas › My Leagues › CURRENT RECOMMENDED Options Portfolio › Practice Portfolio › VIEW ALL LEAGUES Make a Trade › SEARCH FOR A STOCK: Nr. of Shares Est. Price Fees
b)	 WALL STREET SURVIVOR Demystifying Investing. This is your portfolio in: OPTIONS PORTFOLIO OVERALL RETURN 76.71% PORTFOLIO VALUE $176,709 My Stock / Crypto Portfolio REVIEW ORDERS ADD TO YOUR PORTFOLIO News › Are you interested in GDP, Inflation, Econom... › Nasdaq, S&P 500 Climb To New Record Closing ... › DeSantis: Don't "Monkey This Up" By Voting F... › READ MORE Options Holdings REVIEW ORDERS You don't have any options in your portfolio yet. To change that, you have to: Choose an option you love. Buy that option. Wait for the magic to happen!

c)



d)

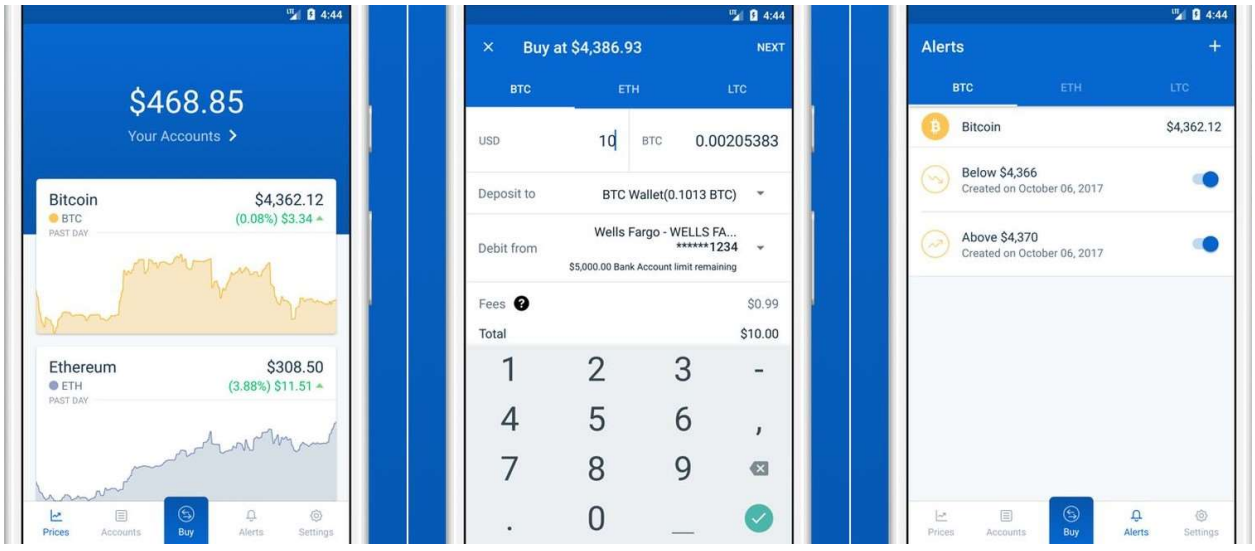


Sl. 4.5. a), b), c), d) Grafički prikaz WallStreet Survivor web aplikacije

Slika 4.5. pod a) prikazuje početni pogled aplikacija unutar kojega se nalaze putanje prema ostalim segmentima aplikacije. Na slici 4.5. pod b) nalazi se pregled portfelja korisnika sa prikazom količine dionica, odnosno kriptovaluta, kupovnom i trenutnom cijenom, količinom i postotnom promjenom. Kod kupovine i prodaje korisnik sam unosi varijable kao što su naziv kriptovalute i količina koju želi kupiti ili prodati, proces je prikazan na slici 4.5. pod c). Edukacija, kao glavni dio web aplikacije prikazana je na slici 4.5. pod d). Edukacija sadržava tekstulne datote i video datoteke, te bi u sklopu aplikacije Euphoria bila veliki dodatak kako bi se korisncima pružile osnovne informacije o principima rada tržišta.

4.4. Coinbase

Iako Coinbase nije aplikacija namijenjena trgovanja kriptovaluta, već mjenjačnica kriptovaluta, važno ju je spomenuti kao aplikaciju s najjednostavnijim i vrlo brzim grafičkim sustavom prikazanim na slici 4.6. Aplikacija je namjena svim korisnicima kojima je potrebna jednostavnost i brzina pri korištenju mjenjačnice. Ne posjeduje određene funkcionalnosti kao burza, no to nije niti njena namjena. Prilagođena je za rad na iOS i Android operacijskom sustavu, te kao webaplikacija za web preglednike. Iako se ne koristi za simuliranje tržišta, aplikacija služi kao ogledni primjer grafičkog dizajna i brzine rada mobilne aplikacije.



Sl. 4.6. Grafički prikaz Coinbase mobilne aplikacije slika uređena i preuzeta sa [16]

5. ZAKLJUČAK

U radu su opisane tehnologije korištene za dizajniranje i implementiranje simulatora trgovanja kriptovaluta. Aplikacija je namijenjena korisnicima Android operacijskog sustava, te većinom mlađim osobama. Tijekom izrade rada, bilo je potrebno obratiti pažnju na jednostavnost korištenja aplikacije. Tehnologije korištene za izradu završnoga rada su industrijske poznate i većinom najčešće korištene tehnologije, što je rezultiralo vrlo velikom količinom potrebnih informacija za implementaciju.

Prednosti korištenja ovakve aplikacije su edukacijske prirode. Aplikacija ima svrhu naučiti korisnika osnovnim funkcionalnostima trgovanja, te principe ponude i potražnje na tržištu koji rezultiraju cijenom resursa, u ovom slučaju 10 najvećih kriptovaluta. Kao i što je navedeno, aplikacija je najviše usmjerena mlađim generacijama, srednjoškolcima, zbog jednostavnosti i ne mogućnosti vlastitog sudjelovanja na raznim tržištima.

Tijekom izrade rada bilo je potrebno naučiti osnovne principe rada Android operacijskog sustava i sam grafički dizajn sustava i aplikacija, programskog jezika Kotlin, Model View Presenter obrasca, kreiranje baze podataka i rad sa istom, te pozivanje i korištenje APIa. Najviše pažnje posvećeno je kreiranju i korištenju baze podataka, te proučvanju načina rada i implementaciji API servisa unutar Android operacijskog sustava. Uz to, vrlo širok spektar mogućih biblioteka za korištenja kod rješavanja problema implementacije baze podataka ili prikupljanjem informacija od poslužitelja oduzima veliku količinu vremena programerima koji se upoznavanju s platformom.

Dodatne mogućnosti aplikacije, koje nisu uvedene radi složenosti rješenja, su edukacijski videozapisi i/ili članci, ljestvica korisnika po zaradi i mogućnost pregleda stanja i transakcija drugih korisnika. Edukacijski videozapisi i/ili članci donjeli bi korisniku osnovne informacije o načinu rada same burze i trgovanja.

LITERATURA

[1] StatCounter, [online], StatCounter, dostupno na: http://gs.statcounter.com/os-market-share/mobile/worldwide [24. lipnja 2018.]

[2] Kotlinlang, [online], JetBrains, 2017. dostupno na: https://kotlinlang.org/docs/reference/faq.html [24. lipnja 2018.]

[3] Fortune, [online], Meredith Corporation, 2018. dostupno na: http://fortune.com/2018/05/10/robinhood-users-trading-app-tops-etrade [24. lipnja 2018.]

[4] Enterprise Singapore, [online], Enterprise Singapore, dostupno na: https://ie.enterprisesg.gov.sg/Venture-Overseas/SgGoesGlobal/TradeHero [24. lipnja 2018.]

[5] WallStreet Survivor [online], Stok-Trak, dostupno na: http://www.wallstreetsurvivor.com/ [24. lipnja 2018.]

[6] Wikimedia Foundation, [online] , Wikimedia Foundation, 2018. dostupno na: https://en.wikipedia.org/wiki/Proof-of-work_system [24. lipnja 2018.]

[7] Satoshi Nakamoto Institute, [online], Satoshi Nakamoto Institute, 2004. dostupno na: https://nakamotoinstitute.org/finney/rpow/ [24. lipnja 2018.]

[8] Nick Szabo, Shelling Out: The Origins of Money, 2002. dostupno na: https://nakamotoinstitute.org/shelling-out/ [24. lipnja 2018.]

[9] Wei Dai, B-Money, 1998. dostupno na: http://www.weidai.com/bmoney.txt [24. lipnja 2018.]

[10] Nick Szabo, Bit Gold, 2005. dostupno na: https://nakamotoinstitute.org/bit-gold/ [24. lipnja 2018.]

[11] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. dostupno na: https://bitcoin.org/bitcoin.pdf [24. lipnja 2018.]

[12] CoinMarketCap, [online], CoinMarketCap, 2013. dostupno na: https://coinmarketcap.com/all/views/all/ [24. lipnja 2018.]

[13] Wikimedia Foundation, [online] , Wikimedia Foundation, 2018. dostupno na: https://en.wikipedia.org/wiki/Coinbase [24. lipnja 2018.]

[14] CoinMarketCap, [online], CoinMarketCap, 2013. dostupno na: https://coinmarketcap.com/exchanges/volume/24-hour/all/ [24. lipnja 2018.]

[15] Square, Inc, [online], Github, 2013. dostupno na: http://square.github.io/retrofit/ [24. lipnja 2018.]

[16] Best Apps Guru, [online], Best Apps Guru, 2018. dostupno na: http://bestappsguru.com/best-bitcoin-apps/ [23. kolovoz 2018.]

SAŽETAK

Sa sve većom prisutnosti kriptovaluta, pojedinac osjeća potrebu sudjelovanja u trgovanju i korištenju kriptovaluta. S obzirom na visoke troškove i rizike ulaganja, javlja se potreba za edukacijom i treningom korisnika prije sudjelovanja na financijskom tržištu. U sklopu ovog rada, izrađena je mobilna aplikacija Euphoria namijenjena upravo simuliranom trgovanju. Programski kod aplikacije pisan je u programskom jeziku Kotlin, baza podataka u SQL-u, a dizajn u XML-u. Pregled cijena kriptovaluta u stvarnom vremenu postignut je spajanjem na coinmarketcap.com API servis uz pomoć Retrofit biblioteke i HTTP GET metode. Aplikacija omogućuje pregled cijena kriptovaluta i njihovu kupovinu i prodaju, te izradu portfelja korisnika. Baza podataka same aplikacije implementirana je uz pomoć Room biblioteke. Uz implementaciju baze podataka, omogućeno je korištenje i pregled portfelja korisnika.

Ključne riječi: kriptovalute, Android, trgovanje, API, baza podataka

ABSTRACT

Cryptocurrency Trading Simulator

With an increasing presence of cryptocurrencies, one feels the need to participate in trading and using cryptocurrencies. Given the high costs and investment risks involved, there is a need for education and training before one participates in the financial market. As a part of this bachelor thesis, a mobile application named Euphoria used for cryptocurrency market simulation was designed and developed. The main application code was written in Kotlin programming language, the database in SQL and design was written using XML. To recieve real market prices of cryptocurrencies the coinmarketcap.com API service was used along with the Retrofit library and HTTP GET method. The application allows a user to check cryptocurrency prices, to trade them, and an ability to create portfolio. The application database has been implemented using the Room library. In addition to implementing the database, the user is given the ability to use and view its own portfolio.

Keywords: cryptocurrency, Android, trading, API, database

ŽIVOTOPIS

Ante Bartulović rođen je 29. prosinca 1994. godine u Zagrebu. OŠ Ivan Mažuranić završava u Vinkovcima u kojima živi sve do upisa na fakultet. U Vukovaru 2009. godine upisuje Matematičku Gimnaziju, maturira 2013. godine, te iste te godine upisuje Elektrotehnički fakultet u Osijeku, današnji Fakultet elektrotehnike, računarstva i informacijskih tehnologija, smjer Računarstvo.