

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Allar Vendla 17599IDDR

# **Rakendus nähtavuse suurendamiseks sotsiaalmeedias krüptoraha abil**

Diplomitöö

Juhendaja: Kaido Kikkas  
Tehnikateaduste  
doktor

Kaasjuhendaja: Mart Mangus  
Rakenduskõrgharidus

Tallinn 2021

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Allar Vendla

16.05.2021

## **Annotatsioon**

Käesoleva diplomitöö eesmärgiks on luua rakendus, mis võimaldaks selle kasutajal suurendada enda nähtavust sotsiaalmeedias, premeerides teisi kasutajaid enda loodud sisule reageerimise eest krüptorahaga. Loodav lahendus peab olema kergesti kättesaadav ja pakkuma intuitiivset ning kasutajasõbralikku kasutajaliidest.

Diplomitöös leitakse lahenduse funktsionaalsed ja mittefunktsionaalsed nõuded, mida loodav rakendus peab täitma. Leitud nõuete põhjal kavandatakse planeeritava rakenduse arhitektuur ja leitakse sobilikud tööriistad ja -vahendid tarkvara realiseerimiseks ning ehitatakse valmis vajalikud komponendid. Töös analüüsitakse olemasolevaid lahendusi. Arenduse käigus testitakse lahenduse toimimist testrühma peal, peale mida võetaks rakendus kasutusse.

Töö lõpuosas analüüsitakse kasutuselevõetud rakenduse toimimist. Kasutusandmete tuginedes hinnatakse, kui edukalt suudeti täita püstitatud eesmärk. Lisaks käsitletakse rakenduse kasutusele võtmise järgselt ilmnunud probleeme ja tuuakse välja edasised võimalused lahenduse täiustamiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 26 leheküljel, 7 peatükki ja 7 joonist.

## **Abstract**

### **An Application for Increasing Social Media Presence Using Cryptocurrencies**

The aim of this thesis is to create an application that would allow its users to increase their social media presence by rewarding other users with cryptocurrency for engaging with their content. The created solution must be easily accessible and offer an intuitive and user-friendly interface.

This thesis finds the functional and non-functional requirements, which must be met by the application. Based on the requirements, the architecture of the planned application is designed and suitable technologies for software implementation are selected and the necessary components will be built. The work analyses the existing solutions. During the development of the application, its operation will be tested on a test group, after which the application will be made available to the public.

The final part of the work analyses the operation of the implemented application. Based on the collected usage data, it is assessed how successfully the set goal was achieved. In addition, the problems that were encountered after the implementation of the application will be addressed and further possibilities for improving the solution will be identified.

The thesis is in Estonian and contains 26 pages of text, 7 chapters and 7 figures.

## Lühendite ja mõistete sõnastik

Andmebaasi tabelite partitsioneerimine	Tabelite partitsioneerimine on protsess, kus suure hulga kirjetega andmebaasitabel jagatakse mitmeks väiksemaks osaks, mis aitab muuta sellest tabelist kirjete pärimise kiiremaks, kuna pole tarvis kogu tabelit üle käia [1]
API	<i>Application Programming Interface</i> – rakendustarkvara liides [2]
CAD	<i>Computer-Aided Design</i> – raalprojekteerimine on projekteerimistegevused (sealhulgas joonestamine ja illustreerimine), mille puhul andmetöötlussüsteeme kasutatakse selliseks otstarbeks nagu komponendi või toote projekteerimine, simuleerimine või täiustamine [3]
DAO	<i>Data Access Object</i> – andmebaasi abstraktse liidestamise muster [4]
GET	HTTP meetod allikast andmete pärimiseks [5]
Groovy	Javal põhinev staatiliselt kui ka dünaamiliselt tüübitud programmeerimiskeel, mis on Javast lihtsama süntaksiga [6]
HTTP	<i>Hypertext Transfer Protocol</i> – protokoll info edastamiseks arvutivõrkudes [5]
IDE	<i>Integrated Development Environment</i> – rakendus, mis pakub arendajatele keskkonda koodi kirjutamiseks
Java	Platvormist sõltumatu objektorienteeritud programmeerimiskeel [7]
JDBC	<i>Java Database Connectivity</i> – Java andmebaasipöördus protokoll, mis võimaldab Java rakendustel andmebaasiga ühendada ja sealt andmeid pärida [8]
JSON	<i>JavaScript Object Notation</i> – andmevahetusformaad [9]
Krüptoraha	Rahasüsteem, mida traditsiooniliselt iseloomustab krüptograafilistel alustel üles ehitatud turvalisus, detsentraliseeritud isereguleeruv maksete süsteem ning võrgu ja tarkvara läbipaistvus [10]
Manussüsteem	<i>Embedded system</i> – arvutisüsteem muu süsteemi (tehnoloogiline protsess, rakett, maksekaardisüsteem) lahutamatu funktsionaalse osana [11]
Nutileping	<i>Smart contract</i> – arvestusraamatu kirje, mis esitab kasutajate vahelist lepingut [12]

ODBC	<i>Open Database Connectivity</i> – Microsofti poolt loodud avatud andmebaasipöördus protokoll, mis võimaldab rakendustel ühenduda relatsiooniliste andmebaasi juhtimissüsteemidega [13]
Ploki ahela jälgija	<i>Blockchain explorer</i> – internetis kasutatav tööriist, mis on loodud ploki ahelat puudutava informatsiooni (ploki andmed, aadressid, toimunud ja pooleli olevad tehingud) vaatlemiseks [14]
RAD	<i>Rapid Application Development</i> – tarkvara kiirarendus mudel
RPC	<i>Remote Procedure Call</i> – kaugprotseduurikutse [3]
SDK	<i>Software Development Kit</i> – tarkvara arendamise tööriistade komplekt [15]
Spock	Käitumispõhine testide kirjeldamise raamistik Javas ja Groovys kirjutatud rakendustele [16]
SQL	<i>Structured Query Language</i> – keel andmete pärimiseks ja muutmiseks relatsiooniandmebaasidest [17]

## Sisukord

1	Sissejuhatus .....	10
1.1	Taust ja probleem .....	10
1.2	Metoodika.....	11
2	Planeeritava lahenduse nõuded.....	12
2.1	Eesmärk .....	12
2.2	Funktsionaalsed nõuded .....	12
2.3	Mittefunktsionaalsed nõuded.....	13
3	Olemasolevate lahenduste analüüs .....	15
3.1	Bounty0x .....	15
3.1.1	Platvormi tugevused .....	15
3.1.2	Platvormi nõrkused.....	16
3.2	Twitter Bounty.....	16
3.2.1	Platvormi tugevused .....	17
3.2.2	Platvormi nõrkused.....	17
4	Arendusvahendite valik .....	18
4.1	Programmeerimiskeel.....	18
4.1.1	Java .....	18
4.1.2	Python.....	19
4.1.3	Programmeerimiskeele valik .....	19
4.2	Andmebaas .....	20
4.2.1	Relatsiooniline andmebaas .....	20
4.2.2	HyperSQL ja SQLite .....	20
4.2.3	MySQL .....	20
4.2.4	Andmebaasi valik .....	21
4.3	Platvorm.....	21
5	Rakenduse arendus .....	23
5.1	Rakenduse arhitektuur .....	23
5.2	Vahemälu lahendus.....	23
5.3	Andmebaasi struktuur.....	24

5.3.1 APP_USER tabel.....	24
5.3.2 ACTIVITY tabel .....	25
5.3.3 COIN tabel.....	25
5.3.4 BALANCE tabel .....	25
5.3.5 BOUNTY tabel.....	25
5.3.6 CLAIMED_BOUNTY tabel .....	25
5.3.7 CHAT tabel .....	25
5.4 Väliste teenustega liidestamine .....	26
5.4.1 Krüptoraha plokiahela API.....	26
5.4.2 Telegram API .....	26
5.4.3 Twitter API.....	26
5.4.4 CoinGecko API .....	27
5.5 Rakenduse seadistamine .....	27
5.6 Rakenduse testid .....	28
6 Tulemuste analüüs .....	29
6.1 Statistika .....	29
6.2 Loodud lahenduse tõhusus.....	29
6.3 Tekkinud probleemid.....	30
6.3.1 Probleemid mitme löime kasutamisel .....	30
6.3.2 Probleemid Twitteri API-ga .....	31
6.3.3 Libakontod.....	32
6.3.4 Kasutajaliidese erisused erinevatel platvormidel .....	33
6.4 Võimalikud edasiarendused.....	34
7 Kokkuvõte .....	35
Kasutatud kirjandus .....	36
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	39
Lisa 2 – Rakenduse konfiguratsiooni näidis.....	40



## Jooniste loetelu

Joonis 1. Mitmetasandiline arhitektuur .....	23
Joonis 2. Andmebaasi andmemudel .....	24
Joonis 3. CEC App konto statistika .....	29
Joonis 4. Võimendatud säuts .....	30
Joonis 5. Võimendamata säuts .....	30
Joonis 6. Vastamisi libakontoga .....	32
Joonis 7. Vigane peamenüü iPhone'is .....	33

# 1 Sissejuhatus

Krüptoraha projektide rohkuse tõttu on uutel projektidel keeruline saada inimeste tähelepanu. Enamasti kaasatakse reklaamimisse olemasolevaid kogukonna liikmeid, keda premeeritakse erinevate ülesannete täitmise eest sotsiaalmeedias – näiteks Twitteri säutsude edasisäutsumise ja/või säutsude meeldivaks märkimise eest. Üldiselt lahendatakse osaliste premeerimine käsitööna – keegi aktiivsem kogukonnaliige või projekti looja vastutab selle eest, et tellitud tegevused saaksid tasustatud. Selle protsessi automatiseerimine võimaldaks tegevuste tasustamist kiirendada ning muuta see järjepidavamaks.

Eesmärgiks on valmis ehitada autonoomne rakendus, mis võimaldab selle kasutajal luua erinevaid sotsiaalmeediaga seonduvaid ülesandeid. Loodud ülesannete täitmiseks innustatakse teisi kasutajaid, premeerides neid ülesannete täitmise eest krüptorahaga. Rakenduse vahendusel ennast promovad kasutajad suurendavad selle abil enda jälgijate hulka, mille najal suureneks ka kõlapind nende poolt edastatavatel sõnumitel.

## 1.1 Taust ja probleem

Viimastel aastatel on suurenenud huvi digitaalraha, eriti just selle ühe vormi, krüptoraha vastu. Peamiselt kajastatakse meedias Bitcoiniga seonduvat, mis on enim tuntud krüptoraha. Kuid inimeste tähelepanu pärast võitlevad ka sajad krüptorahade sektoris alustavad ja juba tegutsevad projektid ning neil on keeruline saada sotsiaalmeedias enda toote turundamiseks piisavat kõlapinda.

Turundus oma traditsioonilisel kujul on muutumas ja vanamoodsad reklaamid ei toimi enam. Tänapäeva kärarikkas ja infoga üle ujutatud infovoos huvitab inimesi enamasti see, mida räägivad neid ümbritsevad sõbrad ja mõjuisikud, keda jälgitakse sotsiaalmeedias. Selleks, et ennast, oma ettevõtet või pakutavat toodet võimalikult paljude inimesteni viia, on pakutavat lugu tarvis levitada kiiresti ja paljude inimesteni.

[18]

Sotsiaalmeedias loevad klikid ja laigid ning selleks, et neid saada, on tarvis kasutada nii mikro- kui megamõjusikuid. Sellest tulenevalt on toodete turundamine samuti muutunud. Efektivsem on tekitada kõmu ja saada inimesed endast sotsiaalmeedias rääkima [19].

Tootemarkidel on hea võimalus mitte ainult suurimate mõjusikutega ühendusse astuda, vaid ka väiksemate mõjusikute, näiteks oma töötajate ja klientide jõudu kasutades. Igal inimesel on oma ringkonnas mõju ja sellistele võrgustikele ligi pääsemine võib kasvatada sinu edastatava sõnumi haarde mitmekordseks [18].

Kuid mõjusikute palkamine võib olla liialt kulukas ja alternatiivina suurema hulga tavainimeste premeerimine soovitud sõnumi levitamise eest väga keerukas. Väikestes summas osaliste premeerimine on traditsioonilist pangandust kasutades rakenduvate teenustasude tõttu kallis [20]. Sellest tulenevalt peaks ülesannete täitja lõpetama suurel hulgal ülesandeid enne, kui saab oma kogutud tasu välja võtta nii, et enamuse sellest ei läheks teenustasude katteks. See vähendab inimeste motivatsiooni sellise tuluteenimise võimaluse kasutamiseks.

## **1.2 Metoodika**

Käesoleva lõputöö raames selgitatakse välja loodava lahenduse funktsionaalsed ja mittefunktsionaalsed nõuded. Seejärel vaadeldakse olemasolevaid lahendusi, tuues välja nende süsteemide tugevused ja nõrkused.

Eelnevale analüüsile tuginedes valitakse töö realiseerimiseks sobilikud tööriistad ja vahendid. Järgnevalt kirjeldatakse rakenduse arendust täpsemalt. Lõpetuseks analüüsitakse kasutusse võetud rakenduse tõhusust ja probleeme, mis tekkisid peale kasutusele võtmise ning tuuakse välja rakenduse edasiarendamise võimalused.

Tarkvara arendamisel kasutatakse iteratiivset tarkvaraarendusmetoodikat. Arendus jagatakse iteratsioonideks ehk etappideks, kus igas etapis luuakse vaid osa rakenduse funktsionaalsusest. Arendusetapid toimuvad kiirete tsüklitena ja koosnevad loodava funktsionaalsuse nõuete seadmisest ja arhitektuuri projekteerimisest, realiseerimisest, testimisest ja integreerimisest. Tsükliline arendus kestab kuni rakenduse kõik nõuded on täidetud ja lahendus võetakse kasutusele. [21]

## **2 Planeeritava lahenduse nõuded**

Lahenduse loomise eelduseks on selle eesmärgi ja nõuete kirjeldamine. Funktsionaalsed nõuded kirjeldavad funktsioone, mida antud süsteem peab realiseerima ja mittefunktsionaalsed nõuded kirjeldavad seda, kuidas süsteem neid funktsioone täitma peab.

### **2.1 Eesmärk**

Eesmärgiks on luua rakendus, mille abil saavad selle kasutajad võimendada neile sobiliku sõnumi levikut ning seeläbi suurendada sõnumi enda kui ka sõnumi looja nähtavust sotsiaalmeedias. Selleks saavad rakenduse kasutajad luua sotsiaalmeedias sisu jagamisega seonduvaid ülesandeid. Neid ülesandeid saavad täita kõik rakenduse kasutajad.

Sõnumi laia auditooriumini viimiseks kasutatakse inimeste premeerimist selle levitamise eest, mis on uuringule [22] toetudes efektiivseks meetodiks. Selleks, et innustada teisi kasutajaid ülesannet täitma, määrab ülesande looja krüptorahalise preemia ülesande täitjale, mille viimane lunastab peale ülesande edukat täitmist.

### **2.2 Funktsionaalsed nõuded**

Rakenduse kasutajal peab olema võimalik vaadata kõiki rakenduses kasutatavaid funktsioone. Iga funktsiooni jaoks peab saadaval olema selle funktsiooni kirjeldus ja kasutamise juhend.

Rakenduse kasutajal peab olema täielik kontroll tema käsutuses olevate vahendite üle. Tal peab olema võimalik igal ajahetkel saada ülevaade kasutatavate vahendite jäägist. Kasutajal peab olema võimalik enda kontole vahendeid juurde kanda ning neid ka rakendusest välja kanda.

Kasutaja peab saama vaadata toetatud krüptorahade loetelu. Kuvatud peavad olema kasutatavate krüptorahade detailid – nimi, kauplemistähis, projekti veebileht, ploki ahela jälgija veebileht, rakenduse poolt rakendatavad tasud ja piirangud.

Kasutaja peab saama luua ülesandeid teistele kasutajatele täitmiseks. Ülesande loomisel saab kasutaja valida ülesande liigi, mis kirjeldab soovivat tegevust. Variantideks on postituste meeldivaks märkimine, edasipostitamine või siis määratud kasutaja jälgijaks hakkamine. Sõltuvalt ülesande liigist saab kasutaja täpsustada ülesande sihtmärki – meeldivaks märkimise ja edasipostitamise puhul on selleks postituse tunnus, jälgijaks hakkamise puhul aga kasutajatunnus. Ülesande loomisel saab kasutaja valida ülesande täitmisel preemiaks antava krüptoraha ja selle koguse. Preemia suurust piiratakse lähtuvalt kasutaja kontol olevate vahendite kogusest. Kasutaja saab ülesande loomisel valida ülesande täitmisel osalevate kasutajate arvu. Määratav osalejate arv peab olema piiritletud kasutatavate vahendite kogusest. Enne ülesande loomist näeb kasutaja ülevaadet seadistatud muutujatest ning saab vajadusel teha korrekture.

Peale ülesande loomist edastatakse selle kohane teavitus kõigile rakenduse kasutajatele. Kõik kasutajad, peale ülesande looja, saavad ülesande täitmise eest lunastada ülesandes määratud preemia, mis on koheselt kättesaadav ülesande täitja kontol. Loodud ülesannet saab tühistada ainult ülesande looja.

### **2.3 Mittefunktsionaalsed nõuded**

Loodav lahendus peab olema kättesaadav igalt platvormilt ja seadmetüübilt, et haarata võimalikult palju inimesi. Lahendus peab olema valmis koheselt kasutamiseks ilma suurema kasutajapoolse pingutuseta. Rakendus peab olema lihtsa ja mugava kasutajaliidesega ning pakkuma meeldivat kasutajakogemust – kasutatavad funktsioonid toimivad ootuspäraselt ja kasutamisel ei esine ebameeldivaid üllatusi.

Rakendus peab olema arendatud järgides levinud arendusmustreid ja häid tavasid ning olema teistele arendajatele mõistetava ja jälgitava struktuuriga ning arusaadava loogikaga. Rakendus peab olema kergesti edasiarendatav ja salvestatav logi peab olema informatiivne kerkivate probleemide lahendamiseks.

Rakendus ei tohi kasutajate kohta koguda rohkem andmeid, kui selle toimimiseks vajalik (vältitakse isikuandmete kogumist).

Rakenduse töötamise ajal peavad olema kasutatavad krüptorahakotid krüpteeritud ja nende võtit ei tohi teha rakenduse väliselt kättesaadavaks.

Rakendus peab olema skaleeruv ning suutma teenindada sadu samaaegseid kasutajaid ilma, et esineks märgatavaid viiteid ja tõrkeid rakenduse toimimises.

## 3 Olemasolevate lahenduste analüüs

Loodavale rakendusele on üsna keeruline leida tõsisemalt võetavaid alternatiive. Üks suurimaid ja edukamaid platvorme selles valdkonnas on Bounty0x. Lisaks nimetatule on võimalik leida GitHub keskkonnast erinevaid Twitter APIga suhtlevaid rakendusi, millest ühte ka lähemalt vaadeldi.

### 3.1 Bounty0x

Tegu on platvormiga, mis pakub võimalust selle kasutajatel luua erinevaid ülesandeid, mida saavad platvormi teised kasutajad täita. Lisaks ülesannete loojatele ja täitjatele saab platvormi kasutaja olla ka ülesannete täitmise kontrollija rollis, kes kinnitab, et seatud ülesanne on edukalt täidetud ning kõik märgitud osalised on reaalselt ülesande täitmisesse panustanud.

Platvormi kasutamiseks tuleb end kasutajaks registreerida ja täita isikutuvastusankeet. Peale ankeedi täitmist tuvastatakse, kas registreerunud kasutajal on õigus platvormi kasutamiseks või milliseid piiranguid sellele kasutajale peab kohaldama vastavalt kasutaja päritoluriigi kehtestatud regulatsioonidele.

Ülesande loomisel saab täpsustada erinevaid tingimusi ja reegleid, mida on vaja ülesande täitjatel järgida. Tingimusteks võivad olla näiteks nõutud jälgijate arv soovitud sotsiaalmeedia platvormil või näiteks loodava sisu pikkus ja kasutatavad märksõnad või käsitletavat teemasid. Ülesande täitjat ei tasustata juhul, kui ülesanne on täidetud seatud tingimustele mittevastavalt.

Bounty0x platvormil tasustatakse ülesande täitjaid ja kontrollijaid alles peale seda, kui ülesande täitmise tingimused on edukalt täidetud. Sellele, kas ülesanne on edukalt täidetud või mitte, annavad kinnituse kontrollijad. Tasu ülesande täitmise või kontrollimise eest laekub kasutaja kontole mõne nädala jooksul peale ülesande edukat lõppu.

#### 3.1.1 Platvormi tugevused

Bounty0x platvormi peamiseks tugevuseks võrreldes loodava lahendusega on võimalus defineerida rohkem tingimusi ülesande täitmiseks. See võimaldab jagatavaid preemiaid

suurendada, kuna limiteerib sobilike osalejate arvu, ning suurem preemia motiveerib ülesannete täitjad ülesannet tõsisemalt võtma.

Lisaks pakub platvorm tasuta kasutaja nõustamise teenust, mis on suunatud äriklientidele ja selle käigus aidatakse klienti ülesande sõnastamisel. Selle teenuse abil leitakse ülesandele võimalikult asjakohased täitjad ning kliendi poolt edastatav sõnum saab võimalikult suure võimenduse.

### **3.1.2 Platvormi nõrkused**

Bounty0x platvormi üheks nõrkuseks võrreldes loodava rakendusega on see, et enne, kui kasutaja saaks hakata platvormi kasutama, on tarvis kasutajaks registreeruda ja läbida isikutuvastus. See kahandab märgatavalt potentsiaalsete kasutajate hulka, kuna kasutajakonto loomine ja isikuandmete jagamine kolmanda osapoolega on paljudele inimestele vastumeelne [23] või puuduvad isikutuvastuseks vajalikud dokumendid [24].

Teise punktina võib tuua välja pika perioodi, mis eelneb ülesande täitmise eest tasu saamisele. Lisaks võib peale mitme nädalast ootamist selguda, et ülesandes seatud tingimuse vastu on eksitud ja tasu jääb saamata.

Veel võib ära märkida pakutavas lahenduses väga suurt rolli omava inimfaktori. Ülesande täitmise tingimused paneb paika ülesande looja ja tal on lubatud seda teha vastavalt oma äranägemisele. See aga võib tekitada olukorra, kus nii ülesande täitja kui ka kontrollija tõlgendavad kirjeldatud valesti ja selle tulemusena võib kasutajate premeerimine ebaõnnestuda.

## **3.2 Twitter Bounty**

Twitter Bounty - A Ethereum Based Twitter Bounty [25] näol on tegemist avatud lähtekoodiga rakendusega, mis võimaldab selle kasutajal hõlpsasti luua Twitteris säutsumise ülesandeid ning kontrollib ja tasustab ülesande täitjaid. Ülesannete loomine ja nende täitmise kontroll toimub plokiahela nutilepingu vahendusel.

Ülesande loomiseks peab kasutaja sisestama veebilingi säutsule, mida soovitakse levitada. Lisaks peab ülesande looja saatma ülesandeks reserveeritava tasu ülesandega seotud plokiahela aadressile, peale mida on ülesande loojal võimalik ka ülesanne tühistada. Eduka tühistamisel järgselt tagastatakse reserveeritud krüptoraha.



Ülesandes pakutava preemia lunastamiseks sisestab platvormi kasutaja sobiliku säutsu aadressi. Sisestatud säutsu sobivust kontrollitakse nutilepingu abil ja säutsu sobivusel saab kasutaja sisestada preemia kättesaamiseks enda krüptorahakoti aadressi. Kui sisestatud säuts ei sobi, siis kuvatakse kasutajale veateade.

### **3.2.1 Platvormi tugevused**

Selle platvormi peamisteks tugevusteks võib lugeda kogu äri loogika asumist plokiahelas nutilepingutes. See muudab lahenduse läbipaistvaks ja võimaldab kergesti hinnata kasutatava platvormiga seonduvaid ohte. Lisaks on tegemist põhjaliku dokumentatsiooniga avatud lähtekoodiga projektiga, mis muudab selle teise arendaja jaoks kergesti edasiarendatavaks.

### **3.2.2 Platvormi nõrkused**

Praegusel kujul on tegu rohkem prototüübiga kui reaalseks kasutamiseks mõeldud lahendusega, kuna luua saab ainult ühte tüüpi ülesandeid. Puudu on ka süsteem, mis annaks kasutajatele märku uue ülesande loomisest. Lisaks peab kasutaja preemia lunastamiseks iga kord sisestama käsitsi enda plokiahela aadressi.

Üheks platvormi nõrkuseks on see, et preemia lunastamisel kontrollitakse ainult säutsu sisu ja seda, kas selle tunnusega säuts on juba varasemalt preemia saanud. Sellest tulenevalt võivad ülesannete täitjatena esineda need, kes säutsu tegelikult teinud ei ole ja kasutavad teiste säutse, mis vastavad soovitud säutsu sisule.

## 4 Arendusvahendite valik

Arendusvahendite valikul lähtuti esmajoonest sellest, et kasutatavad vahendid oleks juba varasemalt tuttav, mis kiirendaks lahenduse realiseerimise loomist. Teiseks lähtepunktiks oli see, et lahenduse loomisel omandatud teadmisi oleks võimalik ka tulevikus rakendada nii professionaalse karjääri edendamiseks kui ka eraprojektide juures.

### 4.1 Programmeerimiskeel

Programmeerimiskeelte osas kahandas töö autor valiku kahele kõrgtaseme keelele – Javale ja Pythonile. Selle kitsenduse põhjuseks oli see, et need on kaks keelt, millega autor on kõige rohkem kokku puutunud ning mõlemale valitud keelele saadaval rohkesti teeki, mis lihtsustavad oluliselt arendustööd.

#### 4.1.1 Java

Java on klassipõhine, rangelt tüübitud objektorienteeritud kõrgtaseme programmeerimiskeel. Esimese Java versiooni avaldas Sun Microsystems 1995. aastal. Olemasolevaid Java tuginevaid rakendusi on suurel hulgal ja päev-päevalt lisandub uusi. Java on kiire, turvaline ja töökindel. Java töötab kõikjal, alustades sülearvutitest ja mängukonsoolidest kuni teaduslike superarvutite ja nutitelefonideni. [26]

Java lähtekood kompileeritakse baitkoodi, mida oskab käivitada iga Java virtuaalmasin sõltumata platvormist. Tänapäeval sisaldab JVM reaaliajalist kompilaatorit – programmi käivitamisel kompileeritakse baitkood masinkoodiks, mis töötab otse protsessoril ilma vajaduseta baitkoodi jooksvalt interpreteerida. Mõned kompilaatorid võimaldavad Java lähtekoodi või baitkoodi otse käivitavaks programmiks kompileerida. Leidub ka riistvara, mis suudab Java baitkoodi otse käivitada. [27]

Javat kasutatakse erinevates valdkondades. Paljud Androidi ja teiste Javat toetavate mobiilsete süsteemide rakendused on kirjutatud Javas. Java on levinud programmeerimiskeel äripõhiste lahenduste loomiseks. Seda kasutatakse teaduspõhiste

rakenduste loomisel ja suurandmete analüüsiks. Lisaks on Java olulisel kohal veebirakenduste serveripoolsete lahenduste loomisel. [28]

#### **4.1.2 Python**

Python on objektorienteeritud kõrgtaseme programmeerimiskeel. Tihti eelistatakse seda teistele kõrgtasemekeeltele sellesse sisse ehitatud andmestruktuuridele ja dünaamiliselt tüübitud semantikale, mistõttu sobib see väga hästi RADi rakendamisel, ja samuti skriptimiskeelena erinevate olemasolevate komponentide ühendamiseks.

Pythoni lihtne – kergesti õpitav – süntaks, mis rõhub loetavusele, vähendab oluliselt Pythonis kirjutatud programmide hoolduskulusid. Python toetatud moodulite ja pakside struktuur soodustab programmi modulaarsust ja koodi taaskasutust. Tänu laialdasele kasutusele on suur hulk Pythoni teekes tasuta saadaval ja neid võib ka piiranguteta levitada.

Tarkvaraarendajad hindavad Pythoni puhul selle kompileerimise vajaduse puudumise tõttu. Tänu sellele on muuda-testi-silu tsüklil eriti kiire ja sellest tulenevalt saavutatakse suurem produktiivsus. [29]

Tänu suurele hulgale erinevatele ja tasuta kättesaadavatele raamistiketele on Pythoni kasutusala loetelu pikk ja selle tõttu võib Pythoni rakendamist kohata pea igal elualal. Pythonit kasutatakse veebi- ja mänguarenduses, masinõppe puhul, tehisintellekti loomisel, andmete töötlemiseks ja visualiseerimiseks, töölauarakenduste kasutajaliideste loomiseks, veebist andmete kogumiseks, ärilahenduste, audio- ja videorakenduste ja CAD rakenduste loomisel. Lisaks on võimalik näiteks Raspberry Pi platvormil Pythoniga luua manussüsteeme juhtivaid rakendusi. [30]

#### **4.1.3 Programmeerimiskeele valik**

Loodava rakenduse realiseerimiseks kirjutamiseks valis töö autor peamiseks programmeerimiskeeleks Java. Kuigi Pythoni kirjutamine on lihtsam ja loodud kood on tihti kergemini loetav, vähendab Java tüübikindel süntaks vigade tekkimise ohtu. Lisaks on Java – tänu kompileeritavale koodile – oluliselt kiirem kui Python [31]. Java on ka autori põhiliseks arenduskeeleks töö juures ning selle rakendamine lahenduse loomisel aitab keelt veel paremini tundma õppida.

## 4.2 Andmebaas

Andmebaasi valikul kaaluti kolme erinevat relatsioonilist andmebaasi juhtimise süsteemi (ABJS-i): SQLite'i, HyperSQLi ja MySQLi. Eelnevalt väljatoodute puhul on tegemist vabavaraliste ja avatud lähtekoodiga toodetega. Lisaks on kõigi valikus olevate ABJS-ide andmestruktuuride loomis- ja kirjeldamiskeeleks SQL, mis on autorile juba varasemalt tuttav.

### 4.2.1 Relatsiooniline andmebaas

Relatsiooniliste andmebaaside puhul, kasutatakse põhikomponentidena tabeleid, vaateid, rolle, kitsendusi ja salvestatud protseduure. Tabel koosneb ridadest ja veergudest. Tagamaks, et iga kirje (rida tabelis) oleks unikaalne, kasutatakse primaarvõtit, mida omakorda saaks kasutada rist-viitena (välisvõtmena) teistest tabelitest sellele kirjele. [32]

### 4.2.2 HyperSQL ja SQLite

HyperSQL ja SQLite on olemuselt üsna sarnased. Mõlemad on manussüsteemid, ning ei vaja käitamiseks eraldi serverit, vaid töötavad peaprogrammis. Seetõttu on need kompaktsed ja nende üles seadmine ei vaja suuremat konfiguratsiooni loomist. Nii HyperSQLi, kui ka SQLite'i puhul salvestatakse kogu andmebaasi sisu ühte faili, mis muudab andmebaasi taastepunkti loomise ning hilisema taastamise lihtsaks ja kiireks. [33]

HyperSQLi ja SQLite'i peamiseks erinevusteks on see, et HyperSQLi on võimalik kasutada ainult nendes programmeerimiskeeltes, mis toetavad JDBC/ODBC draivereid. Seevastu SQLite'i tugi on pea kõigis enimlevinud programmeerimiskeeltes. HyperSQL toetab mitme lõime kasutamist ja seega on võimalik luua mitu samaaegset ühendust andmebaasiga, SQLite'il puuduvad need võimalused. Mõlema süsteemi puhul võib välja tuua tabeli andmete partitsioneerimise võimaluse puudumise. Seega saab suuremahulistest tabelitest päringute tegemiselt ja päringute optimeerimisel tugineda peamiselt andmete indekseerimisele. [33]

### 4.2.3 MySQL

Erinevalt teistest valikus olevatest ABJS-idest on MySQL andmebaasi käitamiseks vaja eraldi serverit. Seega nõuab selle üles seadmine veidi rohkem tööd, kui manussüsteemidena toimivad ABJS-id. Lisaks on ka andmebaasist taastepunkti loomine

ajamahukam protsess, kuna see genereeritakse andmebaasi serveris eraldi protsessina. [33]

MySQL ABJS-i tugevusena võib välja tuua väga hea toe enimlevinud programmeerimiskeeltele ja väga hea kasutajaliidese. See muudab varasema kogemuseta arendaja jaoks andmebaasi haldamise oluliselt lihtsamaks, kuna kogu info on hästi visualiseeritav ja ei pea nii palju tuginema SQL süntaksi tundmisele. Ja erinevalt SQLite'ile ja HyperSQLile on võimalik MySQL puhul andmebaasi tabelites kasutada partitsioneerimist. [33]

#### **4.2.4 Andmebaasi valik**

Töö autor eelistas ABJS-i valikul HyperSQLi. Valik lähtus loodava rakenduse suurusest ja sellest, et valitud ABJS ühilduks valitud programmeerimiskeelega. Lisaks sellele pidas autor oluliseks, et valitud andmebaas toetaks mitut samaaegset andmebaasiühendust ning ABJS-i ülesseadmine oleks võimalikult lihtne ja andmebaasi taastepunkti loomise protsess oleks kiire.

### **4.3 Platvorm**

Rakenduse platvormi valikul kaalus töö autor erinevaid variante. Valikus oli luua nii veebirakendus, töölauarakendus kui ka mobiilirakendus. Neist variantidest tundus veebirakenduse loomine kõige parem. Veebirakendus oleks kergesti kättesaadav igalt platvormilt ja seadmelt sõltumata operatsioonisüsteemist. Veebirakenduse arendamine on töö autorile ka varasemalt tuttav.

Kuid lisaks veebirakendusele oli ka teine variant – ära kasutada mõnda olemasolevat, laialt levinud suhtlusrakendust, millel on vestlusrobotite loomiseks avalik API. See pakuks veebirakendusele sarnaselt mitme platvormi tuge. Puuduks ka vajadus registreerimisvormi täitmiseks enne rakenduse kasutamisele võtmist. Kasutaja võiks kohe peale vestlusroboti käivitamist lisada rakenduse kasutajate hulka ning kogu rakenduse funktsionaalsus oleks talle sellega kättesaadav.

Lõpliku platvormi valikul vaatles töö autor kahte erinevat vestlusrakendust – Discordi (discord.com) ja Telegrami (telegram.org). Mõlema näol on tegemist laialt levinud suhtlusplatvormidega, mida kasutavad aktiivselt ka krüptorahade kogukonnad. Mõlemal

eelmainitud rakendusel on põhjalik API tugi vestlusrobotite loomiseks koos põhjaliku dokumentatsiooniga. Samuti leiab mõlema rakenduse jaoks avatud lähtekoodiga teeke erinevate programmeerimiskeelte jaoks.

Üheks erinevuseks Discordi ja Telegrami vestlusrobotite API toe juures on see, et Telegrami puhul on võimalik luua nuppudega graafiline liides vestlusrobotile käskude andmiseks. Discordi puhul on ainsaks vestlusrobotile käskude andmise meetodiks käskude trükkimine vestlusaknasse.

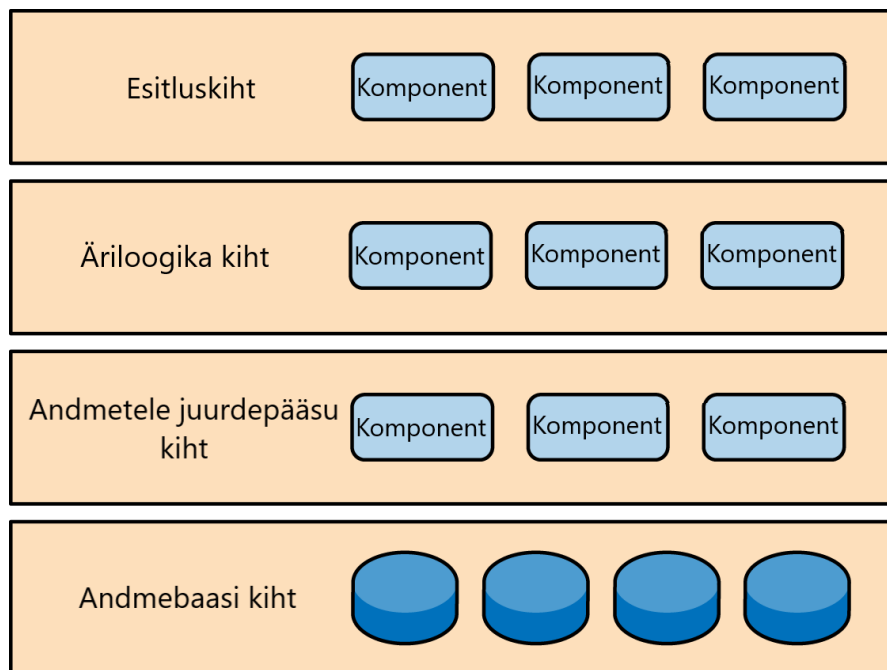
Töö autor valis rakenduse platvormiks Telegrami vestlusrakenduse, mille API peale vestlusrobot luua. Telegrami kasuks tehtud valiku peamise põhjusena võib välja tuua võimaluse luua lihtsamat laadi graafiline liides vestlusrobotile käskude andmiseks. Lihtsustatud kasutajaliides oli oluline nõue lahenduse loomisel. Teiseks Telegrami valiku põhjuseks oli autori varasem kokkupuude Telegrami vestlusroboti API-ga.

## 5 Rakenduse arendus

Järgnevalt käsitletakse lahenduse loomisel järgitud põhimõtteid ja kirjeldatakse rakenduse erinevaid aspekte.

### 5.1 Rakenduse arhitektuur

Rakenduse arendamisel rakendati mitmetasandilise arhitektuuri põhimõtteid. See on üks enimlevinud struktuure Java rakenduste arendamisel. Mitmetasandilise arhitektuuri puhul jaguneb rakendus mitmeks eri kihiks (Joonis 1). Igal kihil on oma kindel roll ja vastutusala ning sisaldab ainult neile vastavat loogikat. See lihtsustab rakenduse arendamist, testimist ja hilisemat hooldamist ning täiendamist. [34]



Joonis 1. Mitmetasandiline arhitektuur

Rakenduse andmebaasiga suhtlemiseks kasutati DAO mustrit. See võimaldab eraldada äri loogika andmete küsimise ja salvestamise loogikast, muutes erinevad komponendid üksteisest sõltumatuteks ja kergemini testitavateks [4].

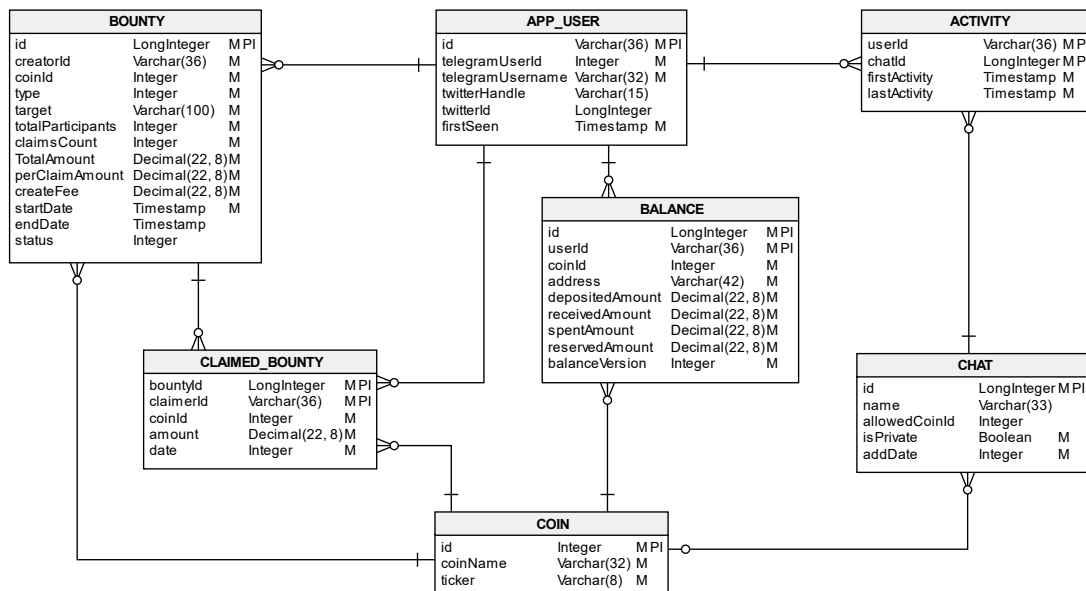
### 5.2 Vahemälu lahendus

Kuna rakenduse disain eeldab, et põhiline suhtlus kasutaja ja rakenduse vahel käib läbi kolmanda osapoolse suhtlusrakenduse, sai väga suurt tähelepanu võimalikult kiire

andmevahetus rakenduse äriloogika ja andmebaasi kihi vahel. Selle saavutamiseks kasutas töö autor Google Guava teeki, mille abil oli võimalik luua üsna paindlikke vahemälu lahendusi. Peaaegu kogu rakenduse ja kliendi vahelises suhtluses kasutatud info käis läbi loodud vahemälu lahenduse, mis võimaldas kiiret kasutaja ja rakenduse vahelist andmevahetust.

### 5.3 Andmebaasi struktuur

Rakenduses kasutatavate andmete visualiseerimiseks koostati andmemudel (Joonis 2), selle põhjal kirjutati andmebaasi loomiseks vajalikud SQL laused. Andmebaasi salvestatakse kasutajate andmed, nende kontode seisud ja aktiivsus erinevates Telegrami vestlusgruppides ning info loodud ja lunastatud ülesannete kohta.



Joonis 2. Andmebaasi andmemudel

#### 5.3.1 APP\_USER tabel

Telegrami rakenduse kasutajatunnus (32bitine täisarv) ja kasutajanimi salvestatakse automaatselt kasutaja esimesel kokkupuutel vestlusrobotiga tabelisse APP\_USER. Koheselt täidetakse ka väli, millal kasutaja loodi. Andmebaasi salvestatakse kasutaja kohta ainult rakenduse toimimiseks vajaminev info, mis võimaldaks kasutajat teistest kasutajatest eristada, kuid samas ei rikuks tema privaatsust. Telegrami kasutaja on samuti



võimalik siduda Twitteri kontoga, sellisel juhul salvestatakse Twitteri tunnus (64bitine täisarv) ja kasutajanimi, see on eelduseks loodud ülesannete täitmiseks.

### **5.3.2 ACTIVITY tabel**

ACTIVITY tabel on kasutaja aktiivsuse jälgimiseks erinevates vestlusgruppides. Tabelisse märgitakse kasutaja esimese ja kõige hilisema tegevuse aeg grupivestluses, mida kasutatakse grupivestluses aktiivsete kasutajate premeerimiseks.

### **5.3.3 COIN tabel**

COIN tabel võimaldab siduda seadistatud krüptorahad kasutajate kontode ning loodud täidetud ülesannetega. Tabelisse salvestatakse krüptoraha nimi ja selle kauplemissümbol.

### **5.3.4 BALANCE tabel**

Tagamaks, et rakenduses toimuvad vahendite liikumised oleksid kiired ja neilt ei oleks vaja maksta plokiahela tehingutasusid, salvestatakse kasutajate krüptoraha kontode seisud BALANCE tabelisse. Lisaks hoitakse seal vastava krüptoraha arvele lisamiseks vajaminevat krüptorahakoti aadressi – see on iga krüptoraha ja kasutaja jaoks unikaalne kood.

### **5.3.5 BOUNTY tabel**

Ülesannete ja selle detailide – ülesande liik, ülesande sihtmärk (Twitteri säutsu tunnus või Twitteri kasutaja kasutajanimi), preemiaks kasutatav krüptoraha, preemia suurus, ülesande täitjate arv – hoidmiseks kasutatakse BOUNTY tabelit.

### **5.3.6 CLAIMED\_BOUNTY tabel**

CLAIMED\_BOUNTY tabelit kasutatakse ülesannete täitmise kontrollimiseks ja hilisemaks arvepidamiseks. Tabelisse salvestatakse iga ülesande täitmise andmed – viide ülesandele ja kasutajale ning viide preemiaks olevale krüptorahale ja lunastatud preemia suurus ning lunastamise aeg.

### **5.3.7 CHAT tabel**

Kasutajate teavitamiseks uute ülesannete loomiste kohta, salvestatakse iga kasutaja ning vestlusroboti vahelise vestluse tunnus andmebaasi tabelisse CHAT. Lisaks vestluse tunnusele täidetakse CHAT tabelis vestluse nimi ja määratakse, kas tegu on privaat- või

grupivestlusega. Täiendavalt on võimalik tabelisse salvestada grupivestluses kasutamiseks lubatav krüptoraha.

## **5.4 Väliste teenustega liidestamine**

Loodud rakendus suhtleb paljude väliste teenustega. Järgnevalt on väljatoodud nende kirjeldused ja ülevaade, millist rolli nad rakenduses omasid.

### **5.4.1 Krüptoraha plokiahela API**

Bitcoin koodibaasil põhinevate plokiahelatega suhtlemiseks kasutas töö autor Bitcoin RPC Client [35] Java teeki. See võimaldab kergesti Java rakendusel suhelda erinevate Bitcoinil põhinevate krüptorahakottidega RPC abil. Selle abil saab rakendus hankida infot plokiahelas toimunud tehingute kohta ning luua plokiahelasse uusi aadresse ja sooritada uusi tehinguid.

### **5.4.2 Telegram API**

Telegrami vestlusrakenduse vestlusroboti loomise tugi on väga põhjalik ja põhjalikult dokumenteeritud. Lihtsamat sorti roboti loomine on võimalik ka ilma ühtegi abistavat teeki kasutamata ja võimaldab vähendada rakenduse sõltuvusi välistest teekidest. Keerukamate rakenduste puhul on aja kokkuhoiu mõttes mõistlikum kasutada olemasolevaid teeke.

Töö autor kasutas Telegrami API-ga liidestamiseks Telegram Ability Bot [36] Java teeki. See teek võimaldab arendajal kergesti luua Telegrami vestlusrakenduses töötavaid vestlusroboteid. Teegil on ka väga hea dokumentatsioon ja piisavalt näiteid, tänu millele on robotile antavate käskude loomine lihtne. Sellest tulenevalt saab arendaja keskenduda robotile antavate käskude töötlemise loogika ning vastava äri loogika kirjutamisele. Lisaks oli kasutatud teegi suureks tugevuseks see, et kirjutatud loogika testimine oli tehtud väga mugavaks tänu lihtsasti jälgitavatele juhistele, mis oli suureks abiks põhjalikult katvate komponenttestide loomiseks.

### **5.4.3 Twitter API**

Rakenduse sidumiseks Twitteriga kasutati Twitteri API-t, millel on põhjalik dokumentatsioon ja mis pakub rohkesti võimalusi rakenduse toimimiseks vajalike andmete pärimiseks.

Twitteri API-ga liidestamisel lisas keerukust asjaolu, et Twitter arendab hetkel uut API versiooni, mis on eelmisest versioonist kiirem ja efektiivsem. Kuna v2 arendamine alles käib, siis ei ole veel võimalik kõike infot uutest otspunktidest saada ning mõnede päringute jaoks peab kasutama Twitteri vana API-t. Lisaks on v1 ja v2 tagastatav andmestruktuur erinev ja rakendus peab mõlema versiooni vastuste tõlgendamisega hakkama saama.

#### **5.4.4 CoinGecko API**

Kasutatavate krüptorahade turuväärtuse leidmiseks liidestati rakendus CoinGecko API-ga. CoinGecko platvorm agregeerib kõigi teadaolevate krüptorahade kauplemissplatvormide andmed, kuvades kauplemiseks olevate krüptorahade kauplemissajalugu ja hetke hindu.

CoinGecko platvormi eelistati konkureerivatele platvormidele tasuta pakutava API teenuse tõttu. Lisaväärtust andis CoinGecko pakutav API lihtne struktuur ja asjakohane dokumentatsioon. Näiteks piisas mitme krüptoraha hinna küsimiseks lihtsa HTTP GET päringu tegemisest, kus parameetriks tuli lisada komaga eraldatud soovivate krüptorahade nimed.

### **5.5 Rakenduse seadistamine**

Et arenduse ajal testimist ning hiljem rakenduse kasutussevõtmisel rakenduse seadistamist lihtsustada, otsustas töö autor luua rakenduse seadistamiseks konfiguratsioonifaili. Rakenduse konfiguratsioonifail on JSON formaadis.

Konfiguratsioonifail võimaldab kiirelt muuta rakenduse toimimiseks olulisi parameetreid. Näiteks saab seal defineerida Telegrami vestlusroboti API võtit ja samamoodi ka Twitter API kasutamiseks vajalikku võtit. Konfiguratsioonifaili kaudu saab kergesti lisada või eemaldada ka kasutatavaid krüptorahakottide konfiguratsioone, kus on kirjeldatud nii krüptorahade üldised andmed, kui ka rahakotiga üle RPC suhtlemiseks vajalik autentimisinfo. Veel hoitakse konfiguratsioonifailis andmebaasi seadeid ning rakenduse poolt kasutatavate ühenduste hoidla suurust.

## 5.6 Rakenduse testid

Ühiktestide loomiseks kasutati JUnit raamistikku. Ühiktestid kirjutati kõigi DAO klasside ja vahemälu lahenduste testimiseks, mille eesmärgiks oli tagada isoleeritud meetodite ootuspärane toimimine.

Lisaks ühiktestidele loodi Spocki raamistiku abil komponenttestid. Komponenttestide kirjutamise eesmärgiks oli, et veenduda kasutajalugude toimimises. Komponenttestide loomisel kasutati Groovy't, mis võimaldab luua kergesti loetavaid teste. Komponenttestide puhul kasutati väliste otspunktide simuleerimiseks Mockito raamistikku, mis võimaldas testida rakenduse toimimist väga erinevates stsenaariumides.

## 6 Tulemuste analüüs

Tulemuste analüüsis vaadeldakse loodud rakenduse kasutusele võtmise järgset statistikat ja tuginedes Twitteri platvormile tuuakse välja näited lahenduse tõhususest. Lisaks kirjeldatakse, millised probleemid tekkisid peale rakenduse kasutusele võtmist ja vaadeldakse edasiarenduse võimalusi.

### 6.1 Statistika

Intellij IDE pistikrakendusel Statistic [37] pakutavatele andmetele tuginedes kirjutati rakenduse loomiseks ligi 13 500 rida koodi (ilma tühje ridu arvestamata). Sellest ligi 8750 rida programmi lähtekoodi, umbes 2000 rida ühikteste ja üle 2750 rea komponenttete. Kokku kirjutati 161 testi, millest 98 on komponenttestid ja 63 on ühiktestid.

25. aprilli seisuga on rakendusel ligi 500 kasutajat, kellest pea 150 on aktiivsed ülesannete täitjad ja umbes 10 on loonud teistele kasutajatele täitmiseks ülesandeid. Kokku on loodud 45 ülesannet.

### 6.2 Loodud lahenduse tõhusus

Töö autor lähtub lahenduse tõhususe hindamisel rakenduse promomiseks loodud Twitteri konto ([https://twitter.com/cec\\_app](https://twitter.com/cec_app)) statistikast, mida pakub Twitter Analytics keskkond.

CEC App Twitteri konto loodi 1. aprillil 2021 – samal päeval, kui rakendus kasutusele võeti. Sellest hetkest alates on kontolt tehtud 10 säutsu, mille tulemusena on kontot külastatud ligi 2000 korda ja selle aja jooksul on konto kogunud 90 jälgijat (Joonis 3).



Joonis 3. CEC App konto statistika

Neist 10 säutsust on olnud 9 säutsu, mida võimendati läbi rakenduse loodavate ülesannetega. Neist 9st kõige edukamat (Joonis 4) on Twitteri kasutajate poolt nähtud (*impressions*) ligi 4500 korral ning säutsule on reageeritud (*engagements*) 205 korral.

#### Tweet activity

<b>CEC App</b> @cec_app Here's a #demo on how #easy it is to #claim #Twitter #bounties using the CEC #Telegram #bot @ <a href="http://t.me/CEC_bounty_bot">http://t.me/CEC_bounty_bot</a> . Currently there are 5 #Bitcoin based #cryptocurrencies available for #rewarding users: @Dashpay @dogecashcoin @ElectraProtocol @litecoin and @vergecurrency. <a href="pic.twitter.com/S5Fll3nxVJ">pic.twitter.com/S5Fll3nxVJ</a>	Impressions	4,658
	Media views	420
	Total engagements	205
	Likes	65
	Retweets	60
	Detail expands	41
	Media engagements	23
	Profile clicks	7
	Hashtag clicks	5
	Link clicks	4

Joonis 4. Võimendatud säuts

Võrdluseks on lisatud läbi rakenduse võimendamata säuts (Joonis 5). See sai vaid 3 reageeringut ja jõudis kõigest 75 Twitteri kasutaja säutsu/otsingu voogu.

#### Tweet activity

<b>CEC App</b> @cec_app SOON™ <a href="https://twitter.com/allavett/status/1385138124859260929">https://twitter.com/allavett/status/1385138124859260929</a> ...	Impressions	75
	Total engagements	3
	Retweets	1
	Likes	1
	Detail expands	1

Joonis 5. Võimendamata säuts

## 6.3 Tekkinud probleemid

Peale rakenduse avalikult kasutusele võtmist ilmnisid mõned probleemid, millest enamik on tehnilist laadi ja kergesti lahendatavad. Ilmnes ka probleem (libakontode probleem), millele head lahendust on keeruline leida.

### 6.3.1 Probleemid mitme lõime kasutamisel

Preemia lunastamisel tekkis probleem mitme lõime kasutamisel ja sama kasutaja sai ühte preemiat mitu korda lunastada. Probleem tulenes sellest, et peale vajalike kontode lukustamist ei kontrollitud uuesti, kas kasutaja oli juba preemia lunastanud. Ehk siis võis tekkida olukord, kus esimene lõim hakkas preemiat lunastama, ja samal ajal hakkas ka

teine lõim seda tegema. Teine lõim ootas ära, kuni esimene lõpetas – vahendid liigutati kontode vahel, preemia arvestus uuendati ja preemia lunastaja lisati preemia saajate hulka ja salvestati andmebaasi. Peale seda jooksis sama protsess ka teisel lõimel lõpuni, kontode vahel liigutati vahendeid sarnaselt esimesele, kuid preemia seisu ja preemia lunastamise infot ei uuendatud. Seega tekkis olukord, kus preemia lunastaja võis saada mitmekordselt premeeritud ja samuti võeti preemia looja kontolt mitmekordne kogus vahendeid. Selle tulemusena ei jagunud viimaste preemiate lunastamiseks enam vahendeid ja ülesanne jäi katkisenä üles.

Probleemi lahenduseks oli lisada täiendav kontroll lunastaja poolt lunastatud preemiatele peale seda kui kontod ja preemia seis oli lukustatud. Sellisel juhul, kui isegi mõni teine lõim oli samaaegselt preemia edukalt lunastanud, oli võimalik esimesel lõimel tegevuse edasine töötlemine peatada ja kuvada kasutajale vastav teade.

Sarnaselt lunastamise probleemile esines mitme lõimega samaaegselt andmete töötlemisel häireid ka loodud ülesannete tühistamisel. Sellisel juhul olid küll loogikavea tagajärjed piiratud ainult ülesande looja kontoga. Probleem tekkis looja konto reserveeritud vahendite arvestamisel olukorras, kus ülesande loojal oli mitu käimasolevat ülesannet ja siis võis juhtuda, et ühe ülesande tühistamisel tagastati kogu reserveeritud vahendite konto summa ning alles jäänud ülesande lunastamisel ei olnud enam võimalik preemiat ülesande täitmise eest välja maksta.

Eelnevalt väljatoodud olukorras oli abiks peale lukkude rakendamist ülesande info värskendamine ja uuesti kontrollimine, kas tegu on endiselt käimasoleva ülesandega. Sellega garanteeriti, et sama ülesannet ei olnud võimalik mitut korda tühistada.

### **6.3.2 Probleemid Twitteri API-ga**

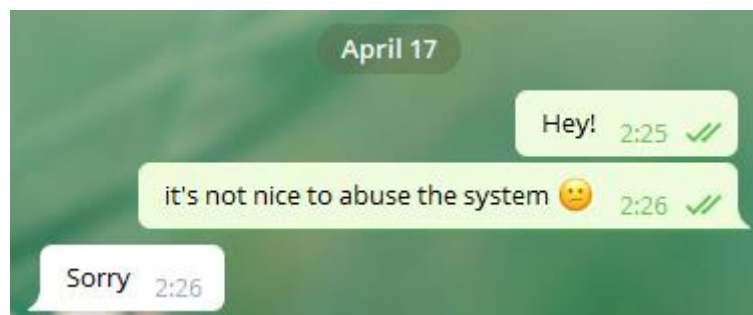
Twitteri API-t kasutades tekkis olukord, kus kasutaja ajajoone kohta infot küsides ei olnud koheselt saadaval kõige ajakohasem sisu. Viite suurus – kasutaja tehtud uuenduse ja API-lt päritud info kokkulangevuse osas – oli umbes 10 sekundit. Sellises olukorras oli keeruline hinnata, kas kasutajal polnudki tegelikult ajajoonel uuendusi või oli lihtsalt API otspunkti info uuenemata.

Selle probleemi lahendamiseks sai algul proovitud API otspunktilt info uuesti küsimist kuni info kättesaamiseni. See ei olnud õige lahendus, kuna otspunktilt saadud info võis

tegelikult olla ajakohane. Lisaks sellele ei olnud võimalik kasutajat teavitada veel töös olevast andmete küsimisest. Selle olukorra lõpplahenduseks sai veateate kuvamine kasutajale, et Twitteri otspunktil info uuendamine alles käib ja kasutaja peaks mõne aja pärast uuesti proovima või kasutaja pole ülesannet veel täitnud.

### 6.3.3 Libakontod

Rakenduse populaarsemaks muutumisel tekkis kasutajatel suurem huvi lahenduse kuritarvitamiseks. Hakati looma libakontosid (Joonis 6). Praeguse hetkeni on tegu probleemiga, millele head lahendust ei ole. Kasutajate seire toimub manuaalselt, peamiselt hinnates uute kasutajate käitumist ja Twitteri kontodel saada olevat infot.



Joonis 6. Vastamisi libakontoga

Uute kasutajate käitumise all peab töö autor silmas seda, kuidas kasutaja rakenduses liigub ja milliseid tegevusi esmajärjekorras teeb. Lahenduse kasutusele võtmisel esitasid uued kasutajad tihti küsimusi, mida ja kuidas nad tegema peavad, et ülesandeid edukalt täita ja seejärel preemiaid lunastada.

Seega pidas autor libakontoks peamiselt selliseid uusi kasutajaid, kes esmakordsel rakenduse kasutamisel esimese 10-20 sekundi jooksul asusid oma kasutajat siduma Twitteri kontoga, ilma, et nad oleks kordagi menüüs teinud mõnda teist valikut. Selle põhjal võis oletada, et nad olid rakendust juba varasemalt kasutanud ja tegu pole kasutaja esimese kontoga.

Lisaks võib välja tuua selle, et kui enamasti jõuti rakenduseni läbi mõne teise vestlusgrupi. Siis libakonto kahtlusega uued kasutajad ühendusid otse rakendusega, ilma et oleks eelnevalt külastanud gruppi, kus rakendus kasutuses oli.

Täiendavalt oli märgata, kuidas eelnevalt mainitud tunnustega kasutajatel olid laisalt loodud Twitteri kontod. Täpsemalt peab autor siin silmas seda, et oli märgata



kasutajanimede järel pikka numbrite rida, mida Twitter sageli uue konto loomisel ise pakub, aga mida on hõlbus muuta.

### 6.3.4 Kasutajaliidese erisused erinevatel platvormidel

Peale seda, kui rakendus kasutusele võeti, ilmnes, et vanemate iPhone'ide ja MacBookide kasutajatele kuvatakse Telegrami rakenduses vestlusroboti peamenüü vigaselt (Joonis 7). Sellist probleemi ei esinenud ei Androidi ega Windowsi töölaarakenduses ja samuti toimis kõik õigesti ka uuemas iPhone'is.



Joonis 7. Vigane peamenüü iPhone'is

Lähtekoodi poolelt oli vea põhjuseks töö autori poolt suure hulga tühikute ja nähtamatu märgi rakendamine menüü pealkirja järel kuvamaks kõiki menüüga elemente sama laiana. Sellise lähenemise tingis asjaolu, et Telegrami vestlusroboti API ei võimalda määrata kuvatavate nuppude laiust ja sellest tulenevalt on igas vaates menüü erineva laiussega, mis muudab kasutajaliidese ebaühtlaseks. Sellise olukorra vältimiseks kasutati eelmainitud meetodit, mis aga vanemate Apple'i toodetega ei toimunud ootuspäraselt, nimelt ei toimunud neis seadmetes ridade murdmist ja selle tõttu ei mahtunud kogu kuvatav sisu ekraanile ära.

## 6.4 Võimalikud edasiarendused

Rakenduse edasiarenduse võimalusi on mitmeid, nii erinevate sotsiaalmeedia platvormide liidestamise, kui ka teistele vestlusrakendustele vestlusroboti loomise näol. Rakenduse kasutajad on tundnud huvi Facebooki, Instagrami ja YouTube'iga seonduvate ülesannete loomise vastu.

Rakendus on algselt ka sedasi üles ehitatud, et edaspidine teiste vestlusrakenduste põhiste vestlusrobotite loomine oleks võimalikult lihtne. Näiteks tunti huvi Discordi ja Twitteri platvormidel toimiva vestlusroboti toe võimalikkuse kohta.

Edasiste ideede all võiks välja tuua arenduse, mille võimaluse kohta küsisid mitmed kasutajad - kuidas rakendus oleks võimeline toetama ka Ethereumi ja Binance'i ning teisi, Ethereumi sarnasel plokiahelal toimivatel, nutilepingutel põhinevaid krüptorahasid. See võimaldaks oluliselt suurendada toetatud krüptorahade valikut, ilma et oleks iga lisanduva krüptoraha jaoks vajalik rakenduse serverile lisada järjekordne krüptorahakott.

Kasutajate hulga kasvades tuleks suuremat tähelepanu pöörada libakontode probleemile. Mõtlemata peaks võimalike vastu meetmete loomiseks, kui rakendust hakatakse ära kasutama pahatahtlike kasutajate poolt loodud robotite abil.

## 7 Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua autonoomne rakendus, mis võimaldab selle kasutajal luua erinevaid sotsiaalmeediaga seonduvaid ülesandeid. Loodud ülesannete täitmiseks innustatakse teisi kasutajaid premeerides neid ülesannete täitmise eest krüptorahaga. Rakenduse vahendusel ennast promovad kasutajad suurendavad selle abil enda jälgijate hulka, mille najal suureneks ka kõlapind nende poolt edastatavatel sõnumitel.

Diplomitöö käigus analüüsiti rakendusele esitatavaid nõudeid ja tuginedes vajadustele valiti lahenduse realisatsiooniks vajalikud tööriistad ning arendusvahendid. Rakenduse loomisel rakendati mitmetasandilise arhitektuuri põhimõtteid ning jälgiti, et edasine rakenduse laiendamine oleks kergesti saavutatav.

Loodud rakendus leidis kiiresti väga korraliku kasutajaskonna ja hetkel saadaolevate andmete põhjal täitis edukalt seatud eesmärgi.

## Kasutatud kirjandus

- [1] DataShack, „Database table partitioning in SQL Server,“ 4 aprill 2014. [Võrgumaterjal]. Available: <https://www.sqlshack.com/database-table-partitioning-sql-server/>. [Kasutatud 22 aprill 2021].
- [2] RedHat, „What is an API?,“ [Võrgumaterjal]. Available: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>. [Kasutatud 31 märts 2021].
- [3] Eesti Keele Instituut, „IT terministandardi sõnastik,“ [Võrgumaterjal]. Available: <http://www.eki.ee/dict/its>. [Kasutatud 15 aprill 2021].
- [4] Oracle, „Data Access Object,“ [Võrgumaterjal]. Available: <https://www.oracle.com/java/technologies/data-access-object.html>. [Kasutatud 31 märts 2021].
- [5] RFC Editor, „Hypertext Transfer Protocol,“ [Võrgumaterjal]. Available: <https://www.rfc-editor.org/rfc/rfc2616.txt>. [Kasutatud 31 märts 2021].
- [6] „Groovy,“ Apache Groovy, 2021. [Võrgumaterjal]. Available: <https://groovy-lang.org/>. [Kasutatud 24 aprill 2021].
- [7] Guru99, „What is Java? Definition, Meaning & Features of Java Platforms,“ 2021. [Võrgumaterjal]. Available: <https://www.guru99.com/java-platform.html>. [Kasutatud 30 märts 2021].
- [8] JavaTPoint, „Java JDBC Tutorial,“ 2021. [Võrgumaterjal]. Available: <https://www.javatpoint.com/java-jdbc>. [Kasutatud 2021 aprill 2021].
- [9] ECMA International, „The JSON Data Interchange Syntax,“ [Võrgumaterjal]. Available: [https://www.ecma-international.org/wp-content/uploads/ECMA-404\\_2nd\\_edition\\_december\\_2017.pdf](https://www.ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf). [Kasutatud 31 märts 2021].
- [10] Eesti Krüptoraha Liit, „Mis on krüptoraha?,“ [Võrgumaterjal]. Available: <http://www.kryptoraha.ee/tehnoloogia/>. [Kasutatud November 2017].
- [11] A. Tavast ja V. Hanson, Arvutikasutaja sõnastik inglise-eesti, Tallinn: Kirjastus Ilo, 2008.
- [12] A. Ansper, A. Buldas ja J. Willemson, „Krüptograafiliste algoritmide elutsükli uuring 2017,“ Riigi Infosüsteemi Amet, Tallinn, 2018.
- [13] Microsoft, „Microsoft Open Database Connectivity,“ 19 jaanuar 2017. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/odbc/microsoft-open-database-connectivity-odbc?view=sql-server-ver15>. [Kasutatud 21 aprill 2020].
- [14] IntelliPaat, „What is blockchain explorer?,“ 2021. [Võrgumaterjal]. Available: <https://intellipaat.com/blog/tutorial/blockchain-tutorial/block-explorer/>. [Kasutatud 10 aprill 2021].

- [15] RedHat, „What is an SDK?“, [Võrgumaterjal]. Available: <https://www.redhat.com/en/topics/cloud-native-apps/what-is-SDK>. [Kasutatud 5 aprill 2021].
- [16] „Spock - the enterprise ready specification framework“, 2021. [Võrgumaterjal]. Available: <https://spockframework.org/>. [Kasutatud 14 aprill 2021].
- [17] W3Schools, „Introduction to SQL“, [Võrgumaterjal]. Available: [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp). [Kasutatud 5 aprill 2021].
- [18] C. Gil, Turunduse lõpp: kuidas muuta oma tootemark sotsiaalmeedia ja tehisintellekti ajastul inimlikuks, Tallinn: Äripäev, 2020.
- [19] M. Hughes, Buzzmarketing: kuidas panna inimesed asjast rääkima, Tartu: Kirjastus Hermes, 2006.
- [20] R. McCrary, „How To Avoid The Potential Pitfalls Of Making Peer-To-Peer Payments“, Forbes Media, 14 oktoober 2019. [Võrgumaterjal]. Available: <https://www.forbes.com/sites/forbesbusinesscouncil/2019/10/14/how-to-avoid-the-potential-pitfalls-of-making-peer-to-peer-payments/>. [Kasutatud 24 aprill 2021].
- [21] Tutorials Point, „SDLC - Iterative Model“, 2021. [Võrgumaterjal]. Available: [https://www.tutorialspoint.com/sdlc/sdlc\\_iterative\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm). [Kasutatud 10 mai 2021].
- [22] Y. Zou, „Buying Love Through Social Media: How Different Types Of Incentives Impact Consumers' Online Sharing Behavior“, Old Dominion University, Norfolk, 2016.
- [23] Advertising Research Foundation, „1st Annual ARF Privacy Study: What's the Value of Trust and Privacy?“, New York, 2018.
- [24] World Bank, „Global ID Coverage, Barriers, and Use by the Numbers: Insights from the ID4D-Findex Survey“, World Bank, Washington, 2019.
- [25] S. Tabrizi, 2021. [Võrgumaterjal]. Available: <https://github.com/shawntabrizi/Ethereum-Twitter-Bounty>. [Kasutatud 27 aprill 2021].
- [26] Java™, „What is Java?“, 2021. [Võrgumaterjal]. Available: [https://java.com/en/download/help/whatis\\_java.html](https://java.com/en/download/help/whatis_java.html). [Kasutatud 16 aprill 2021].
- [27] M. Villum, „Java“, [Võrgumaterjal]. Available: <https://2moodul.weebly.com/>. [Kasutatud 16 aprill 2021].
- [28] Guru99, „Java platform“, [Võrgumaterjal]. Available: <https://www.guru99.com/java-platform.html>. [Kasutatud 17 aprill 2021].
- [29] Python.org, „What is Python?“, 2021. [Võrgumaterjal]. Available: <https://www.python.org/doc/essays/blurp/>. [Kasutatud 14 aprill 2021].
- [30] Edureka, 2021. [Võrgumaterjal]. Available: <https://www.edureka.co/blog/python-applications/>. [Kasutatud 16 aprill 2021].
- [31] B. Okuneu, „Java Vs Python Tried and true Vs "modern and new"“, 20 august 2018. [Võrgumaterjal]. Available: <https://belitsoft.com/java-development-services/java-vs-python-tried-and-true-vs-modern-and-new>. [Kasutatud 24 aprill 2021].
- [32] M. Sarig, „MariaDB Vs MySQL: Compatibility, Performance, And Syntax“, 4 mai 2019. [Võrgumaterjal]. Available: <https://blog.panoply.io/a-comparative-vmariadb-vs-mysql>. [Kasutatud 20 aprill 2021].

- [33] DB-Engines, „System Properties Comparison HyperSQL vs. MySQL vs. SQLite,“ 2021. [Võrgumaterjal]. Available: <https://db-engines.com/en/system/HyperSQL%3BMySQL%3BSQLite>. [Kasutatud 22 aprill 2021].
- [34] M. Richards, Software Architecture Patterns, O'Reilly Media, Inc., 2015.
- [35] A. Polverini, 2021. [Võrgumaterjal]. Available: <https://github.com/Polve/bitcoin-rpc-client>. [Kasutatud 27 aprill 2021].
- [36] R. Bermudez, 2021. [Võrgumaterjal]. Available: <https://github.com/rubenlagus/TelegramBots/tree/master/telegrambots-abilities>. [Kasutatud 16 aprill 2021].
- [37] T. Topinka, „JetBrains Marketplace - Statistic plugin,“ 2021. [Võrgumaterjal]. Available: <https://plugins.jetbrains.com/plugin/4509-statistic>. [Kasutatud 18 aprill 2021].

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Allar Vendla

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Rakendus nähtavuse suurendamiseks sotsiaalmeedias krüptoraha abil“, mille juhendaja on Kaido Kikkas ja kaasjuhendaja on Mart Mangus.
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

16.05.2021

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Rakenduse konfiguratsiooni näidis

```
{
  "telegramBotToken": "****",
  "telegramBotName": "CEC_bot",
  "telegramBotId": ***,
  "telegramBotCreatorId": ***,
  "twitterBearerToken": "****",
  "chatActivityTime": 8,
  "limitWithdrawTime": 1,
  "bountyMaximumParticipants": 1000,
  "database": {
    "url": "./data/ce_db",
    "poolSize": 4
  },
  "coins": [
    {
      "ticker": "WSP",
      "name": "Wispr",
      "website": "N/A",
      "minimumTransactionAmount": 0.5,
      "withdrawalFee": 1,
      "bountyFee": 100,
      "multiplier": 0.001,
      "confirmationsCount": 6,
      "rpcConfiguration": {
        "username": "username",
        "password": "****",
        "host": "127.0.0.1",
        "port": 17005
      },
      "blockchainExplorer": {
        "url": "http://127.0.0.1:3001",
        "txPrefix": "tx",
        "addressPrefix": "address"
      }
    }
  ]
}
```