National Research University Higher School of Economics

Faculty of Computer Science
Bachelor's Programme in Data Science and Business Analytics

GROUP TERM PAPER
Software Project
Blockchain-Based Tips Service

Prepared by students: of Group 191 in Year 3,
Rybnikova Marianna Alekseevna,
Zotov Gleb Aleksandrovich,
Kurbatov Dmitry Aleksandrovich

Term Paper Supervisor:
PhD Associate Professor Yanovich Yury Aleksandrovich

Moscow
2022

Contents

Contents **2**

1 Key terms and definitions **4**

2 Introduction **10**

2.1 Abstract 10

2.2 Purpose and Objectives 10

2.3 Methods, algorithms, models and instruments for project implementation 11

2.3.1 Backend Stack 11

2.3.2 Frontend Stack 12

2.3.3 Token 13

2.4 Tokenomics 19

2.4.1 Tips schema 19

2.4.2 Business profit 19

2.4.3 Token distribution 20

2.4.4 Metamask insertion 20

2.5 Investment 21

2.6 Marketing 22

2.7 Main result 22

2.8 The role of members of the project 22

3 Comparative analysis **24**

3.1 Segment analysis 24

3.2 Competitors analysis 26

4 Project implementation **32**

4.1 Frontend functionality for web application 32

4.1.1 Design Specification 32

4.1.2 Developing and maintaining the user interface 34

4.1.3 Page Templates 38

4.2 Backend functionality 40

4.2.1 Database management 40

4.2.2 Project structure 41

4.2.3 User Authentication 43
4.2.4 Navigation 45
4.3 Token development 48
4.4 Web app and blockchain connection 51
4.4.1 Preliminary step 51
4.4.2 Transaction step 52
4.4.3 Profile data 53
4.5 Conclusion 54

5 References 56

6 Appendix 58

List of Figures 58

1 Key terms and definitions

Cryptography

That is a study of transferring information or a message totally securely, so that there is no possibility for a third party to access the containing information. During the last decades cryptography is widely used in the IT and is considered to be a basis for cryptocurrencies. One of the core cryptographic ways to send messages is hash functions usage.

Hash function

Hash function [1] is a function that converts an array of different symbols (it can be a set of numbers, lettering, words, combination of numbers and letters: any combination of symbols) to a specialized string called **hash**. As a rule, the length of a hash is determined in advance, depending on the specific algorithm of a hash function. There are several important corollaries of hash functions:

- They must be irreversible, thus, only hash given, it is impossible to obtain the initial array of data. In real life hash functions encode data in such a way that it is extremely challenging computationally and financially to decode the initial message.
- Hash functions must have minimal chance of the collision, that is the case when several distinct arrays are encoded in the same hash.

Public and private keys

Public and private keys in cryptography are used in order to complete the message transfer and ensure in the safety of the operation. Public key is accessible to everyone as it is seen from the name. In context of cryptocurrencies, public key are considered to be the sender‘s and receiver‘s addresses that are available for all parties. Private key is a special hash that is used by the sender to confirm the transaction. As it is private, it must be known **only** by the transmitter. In case someone accesses the foreigner private key, they are able to make any operation with the private key holder‘s wallet and, therefore, with money available. Such a key can be compared to the PIN and CVC code regarding debit and credit cards.

Cryptocurrency

Cryptocurrency is a virtual currency based on blockchain and cryptography methods so that transactions performed by cryptocurrencies are extremely safe. In addition, numerous cryptocurrencies today have an open source code and therefore, everyone may check out the code and ensure in its correctness and transparency.

Transaction

Transaction in context of cryptocurrencies is a transfer of the specific amount of money from one address to another at some moment of time. Transaction can be completed if and only if the operation is confirmed by the sender using his/her private key. Otherwise, the transaction will be rejected.

Blockchain

It is a virtual environment containing all information about each transaction or action on the specific blockchain [2]. Assume a cryptocoin was transferred from one crypto wallet to another. Then, in the blockchain information about the transmitter, receiver, amount and date and time will be encoded into a hash and put into the database. The same will happen with other transactions. Hashes of several transactions are also summarized, and a new hash is created with information of those several transactions. After each iteration a fresh hash is saved in the database. If there is a change in one of the first transactions, for example, someone desired to falsify the information of a completed transaction, its hash would also change and thus, the following hashes in the blocks will deviate from the original ones as well. Therefore, it is impossible to change the history of operations or delete information on some of them.

Web3

Development of public Internet or Web can be compared to the industrial revolutions. There were several drastic changes in web development and nowadays humanity is located on the edge of two epochs. First one, Web1, was the introduction to the wireless network as we know it today, however, all pages there were static and in case someone desired to create own website, they should have studied plenty of manuals, documentation and etc. Web2 introduced high-level dynamic web pages on which one can find numerous interesting features that seemed unreal in Web1 epoch. However, with development of Web2 people started trusting websites their passwords, private data and other personal information that we do not want to share with the entire world. Unfortunately, this fact implies data is often being stolen and used in sordid motives and that is one of the most important and crucial issues in the modern world. In order to remedy the situation the concept of new web system was introduced. Web3 [3] is based on better and higher confidence of any data available online with the help of blockchain technology and decentralization.

Decentralized Finance (DeFi)

As it was already mentioned above, the basic concept of blockchain technology is decentralized finance. Humanity has been living in a system of centralized finance for the entire period: perennially some regulator existed that had been setting definite rules and conditions for monetary assets movements and thus, these regulators could interfere into the system and manage it in the way they want. The general scheme remains even today as there is a banking system (including central banks) that often adjusts pricing policy according to their own willing and as a rule, this policy does not aim ordinary citizens. Therefore, DeFi‘s goal is to solve this problem. People desiring to transfer a particular amount of money do not interact with third parties embodied in banks, brokers and others, the only necessary condition is to have a special software for transactions.

Obviously, there is nothing ideal in the world and even a good idea of decentralization has its own drawbacks and nuances, but primarily the positive aspects will be discussed.

- Anyone has an effortless access to the services provided by the finance system.
- DeFi is based on the smart-contracts that have open-source code and can be analysed by anyone in the Network. In addition, totally automated process minimizes human factor and consequently, eventually neglects errors.
- Every participant is an entirely fledged user of the system.
- No third parties needed implying higher processing velocity and better production performance.
- There is a huge room for improvement and development such that anyone can invent a new program or an app and deploy it without any difficulties.

However, every coin has both sides, and there are enough disadvantages of the system that do not guarantee totalitarian superiority of DeFi above conventional centralized financial system. Here are some crucial negative moments:

- As mentioned before, all users have the same rights and therefore, no one is responsible for other users, therefore, in case of a major force it is impossible to get support and ask for a help.
- As the system is quite raw at the moment, all numeric values, for example, exchange rates and interest rates are enough fluctuating and therefore, unstable.
- Rawness of the system affects additionally its security. Smart-contracts are still available for interfering implying embezzlement. As an example, more than 10 billion dollars assets were stolen in 2021 from the decentralised financial systems.
- Some services and platforms set high fees for smart-contracts deploying.

- System is quite chaotic and unordered that leads to challenging information or app seek.

To sum up, the idea of decentralised systems is breakthrough and extremely in long-run perspective. However, instantly it needs deeper development, especially in context of security. Very likely because of this fact nowadays the quantity of unique users of the system is very low, it hardly exceeds 3 million participants. Nevertheless, the sphere is completely fresh, it can be said the first development wave took place few years ago, in 2020 in the fervor of Covid-19 pandemic, and likely in the next decade a significant part of today existing financial institutions will become a part of the decentralised financial system.

Token

As any currency, cryptocurrencies also have their own unit of exchange. That is a token. However, with development of the sphere, not only currencies but platforms tend to appear. These days there are two main and most popular cryptocurrencies: BTC (Bitcoin) and ETH (Ethereum) but the difference is that Ethereum is a platform for other currencies as well. Moreover, there are several standards for smart-contracts based on this currency, for example, ERC20 which is one of the most commonly used one. Using such standards one may create their own token on different platforms. That is exactly what we have done in the project.

Altcoin

Bitcoin is still the most popular cryptocurrency with the highest daily volume among others. It was the first cryptocurrency ever made and very largely because of this fact it is so popular today. However, it is quite challenging to create a totally new product and to have the best quality for more than a decade. Bitcoin has several drawbacks or lowlights and in order to solve the problem alternative cryptocurrencies were made and they are called "Altcoins" (Alternative coins). The most popular one is Ethereum (ETH) that was founded in 2015. As Bitcoin generally operates as only a currency with the main goal of transferring tokens, altcoins explore other potential functions and application domains. There exist several common types of altcoins:

- Payment coin - tokens operating as a simple currency, thus, similarly to Bitcoin. The greatest share of altocoins are payment coins as plenty of cryptocurrency owners would like to earn money by selling their tokens since they cost something.

- Utility coin [4]- needed to provide services within a network. In other words, there are usually used for operation tasks. For example, a so called Filecoin is a utility coin for information security and storage management.

- Stablecoins - This is a specific type of tokens which total supply is secured by the fiat money pool. It can be any currency like euro or dollars or even rubles, the logic of stablecoins is that for each unit from the pool the one unit of token is created. Let's provide an example, assume that the token creator has 50 000$, it means that the total supply of the stablecoin is also 50 000, and the price of each coin is 1$.

  It is important to note that the stable exchange rate of 1 coin to 1 dollar is not so simple. If token holders would like to exchange stablecoins back to the fiat, then these stablecoins should be burned, which provides stability of the total supply to the security fund. So the implementation of the smart-contract is simple, but there are a lot of rules and restrictions on stablecoins from a legal point of view.

- Governance coins - these tokens provide the owners rights to make decisions in cryptocurrency management. Generally speaking, owners of such tokens can be compared to stakeholders in a company. They may vote for several decisions and participate in currency governance.

There are other types of coins that are not as popular and are usually created as an experiment. For example, meme coins exist in crypto universe in order to gain popularity in a short-run prospect.

Crypto wallet

Obviously, cryptocurrency has to be stored somewhere, and for this crypto wallets are created. Crypto wallet [5] is a program or a software that contains data on the users, such as public and private keys, balance and etc. In addition, they allow users not only to store money but additionally make transactions to other accounts.

Wallets can be divided into two groups: cold and hot. A cold crypto wallet is an external storage like a USB adapter containing information about all transactions in the world of the specific cryptocurrency. This implies a wallet can also be an additional data source of the blockchain. Cold wallets are safe as they do not demand internet connection and thus are not possible for hacking. However, such a wallet may be lost, and there is no opportunity to restore its content. Most popular cold crypto wallets are Ledger, Trezor and KeepKey.

Hot crypto wallets have connection to the Internet and therefore, store information there. This makes transactions and operations extremely simple, convenient and rapid. Hot crypto wallets creators have put in a lot of effort in order to make their wallets safe from external strikes and the most popular ones today inspire confidence and millions of people have hot crypto wallets. The most popular wallets are Bitcoin Wallet, Blockchain Wallet, MetaMask and Jaxx.

Smart-contract

That is a protocol that defines parameters and rules for token transactions, including token name, index, maximal possible volume, different features such as burning and etc. Actually, smart-contracts connect created token to a blockchain and set transaction algorithms. Smart-contracts have different standards depending on the platform they will work with. Taking an example above, if a token is deployed on Ethereum platform smart-contracts will have to be written with respect to Ethereum protocols like ERC20 [6] or others.

2 Introduction

2.1 Abstract

Universe of cryptocurrencies and blockchain these days is considered to be the bleeding edge of technology and for that reason our team decided to delve into the sphere.

In many countries with mature economy touchless tips have been raising the popularity during the last several years due to several reasons that would be discussed more toughly later. However, the number of them based on blockchain is exiguous, especially in context of Russia. That is why we decided to create the tips service based on blockchain. A fully-featured website was made containing the database of both tippers and tippees, profile pages, transaction page, "about us" page and etc. By clicking the "Tip" button on the website and choosing the receiver‘s crypto wallet address with the amount one desire to send, a connection with the smart-contract is being set and thus, a transaction through the blockchain is made. In order to have own currency we have created an Ethereum based token "OnlyTipsToken" (OTT), smart-contracts for which are ERC20 standardized. Additionally, we have implemented several features for the token that are going to be discussed later.

To sum up, we have created a working blockchain based tips service that is aimed at a concept of a waiter-visitor relation. In other words, for now we have a service for restaurants and cafes where visitors may tip waiters for their well done job.

2.2 Purpose and Objectives

The goal of the project is to explore the sphere of blockchain technology and its configurations, create a service for sending tips using cryptocurrencies via the web app, obtain invaluable knowledge in crypto and contemplate about further potential development of the service.

Objectives of the project are:

- Analyse the crypto sphere in Russia, identify possibility of setting the startup.
- Seek for competitors and analogs available in the market, define their advantages and downsides, select interesting and perspective features that can be implemented in the service.
- Create a web application for making transactions in a convenient way, containing databases with information on users.
- Create and deploy a token for the service.

- Connect the web application with token API.
- Analyse the accomplished work and state possible ways for service improvement.

2.3 Methods, algorithms, models and instruments for project implementation

2.3.1 Backend Stack

Python

Python is the object-oriented, high-level programming language with dynamic semantics. This coding language has become extremely popular among a variety of applications. Web development with python is widely used. For instance, such services like Netflix, Google, Pinterest, Instagram were created with Python. It is very productive and includes dozens of predefined modules and packages with the solutions for all types of problems.

Advantages:

- Easy-to-learn and English-like syntax is the main benefit which allows one to get focused on particular tasks without any understanding struggles.
- Easy to track the error. A really convenient debugger allows developers to inspect the variables. Moreover, a code is executed line-by-line and if any mistake occurs, a programmer will easily figure it out.

Disadvantages:

- Python programs are slowly executed when compared to other languages. The line-by-line execution is not that efficient.
- Python is not memory efficient. Since this language is rather simple, it does not provide enough flexibility for managing memory storage. Moreover, the memory cannot be explicitly allocated.

Django

Django [7] is a high-level Python web framework that is used for web development and design. The choice was made on this tool for several reasons. Firstly, the other part of the coursework was done with the usage of Python, meaning that together with Django Framework the process of combining all the work together will be extremely easy and gentle. Secondly, our team has a great knowledge of Python tools. As a result, even at the first steps of coding with

Django some of the patterns will be familiar to a developer and this will speed up the process of coding. Now let us consider several advantages and disadvantages of this framework.

Advantages:

- Django concepts are easy to understand, and thus web development becomes as quick as possible.
- Django provides a variety of techniques for security handling. A junior development can quickly build a totally secure web app avoiding many common mistakes.
- Django is scalable. It handles almost every hardware addition and can build a web structure that will be integrated for completely different devices.

Disadvantages:

- Requires huge blocks of coding.
- A junior developer needs some time to get familiar with the structure of Django projects and read some documentation about some of the predefined functions and variables.

2.3.2 Frontend Stack

Both tools will be described in the Frontend section. Now we will focus on their advantages and disadvantages.

HTML

Advantages:

- This tool is easy-to-learn as well as the Django Framework. The Hypertext Markup Language is so widely used that all sorts of study materials and techniques are found on the internet and can be studied in a short period of time.
- HTML is browser-friendly and works properly in a wide range of browsers.
- Is integrated with other languages like JavaScript or CSS in one line of code.

Disadvantages:

- HTML is a static language and cannot produce an output on its own. That is why we used this tool for the creation of a web page only. All the operations with the variables were done with Python.

- Too much code should be written for huge projects. For instance, the website that was created during this coursework is not that big, however, we have more than 20 HTML files as a result. At some point it becomes less convenient and readable and takes some time to find a particular part in it.

- HTML can provide us with the static web pages only. When some up-to-date styling or animation is added, some other tools need to be used.

CSS

Advantages: [8]

- Provides a variety of styling and formatting tools.

- Together with HTML tool, CSS makes the first much more readable. CSS files can contain all the options concerning styles of the used elements, while HTML files are responsible for the elements that should be present on a page.

Disadvantages:

- CSS is not secure, since everyone can access a CSS file and make some changes on a local version of a web page.

- Users can face problems when using different browsers, since a web page with the CSS inclusion may be shown in slightly different ways. Therefore, a developer must conduct a test of a website using different browsers to make sure that everything works properly as it should be.

2.3.3 Token

1. Token smart contract selection

Defining the right token model is one of the key tasks in token development. First of all, the properties of a token are the most important thing, which will be the main factors in choosing an appropriate way to develop a token.

For our goals the ERC-20 [6] standard token was picked, for several reasons. The ERC-20 is a standard for smart contracts on the Ethereum blockchain. It is a set of rules that must be followed when developing a contract responsible for the parameters and issue of a new user token. In other words, the task of ERC is to establish agreements that simplify the interaction of applications and contracts with each other.

So, applying the existing standard, developers save time and effort. Thanks to ERC-20, they can create new tokens and not worry about compatibility and security. Also, ERC-20 tokens will be compatible with all services and software that support this standard, and there are a lot of services right now which work with ERC-20 as one of the most popular standards. Here are some examples of ERC-20 tokens:

- Maker (MKR)
- Tether (USDT)
- Chainlink (LINK)
- The Sandbox (SAND)
- The Graph (GRT)
- Uniswap (UNI)
- Axie Infinity (AXS)
- Aave (AAVE)

Some of them are huge DeFi projects which provide their services as a business set up, some of them are used as in-game currency like AXS and some of them even represent NFT industry, like SAND. The ERC-20 tokens must include six mandatory functions: `totalSupply`, `balanceOf`, `transfer`, `transferFrom`, `approve` and `allowance`. Also, some additional functions such as name, symbol and decimal can be added. Now let's look at these functions, realized at the Solidity language created for Ethereum:

1. **totalSupply**

   ```
   function totalSupply() public view returns (uint256)
   ```

   This function returns the total number of tokens in the contract.

2. **balanceOf**

   ```
   function balanceOf (address _owner) public view returns (uint256 _balance)
   ```

   This function uses a parameter, which is an address of the user. It shows the balance of the address tokens on request. Since accounts on the Ethereum network are publicly available, anyone can request the balance of any user if you know his address.

3. **transfer**

```
function transfer(address _to, uint256 _value)
public returns (bool success)
```

This function transfers tokens from one user to another. The parameters are the recipient's address and the transfer amount. It is important to know the transfer function triggers an event, which indicates to the blockchain the need to include a link to the transaction.

4. **transferFrom**

```
function trasferFrom(address _from, address _to, uint256 _value)
public returns (bool success)
```

The transferFrom function is a more convenient alternative to the transfer, which provides greater programmability in decentralized applications. Like `transfer`, it is used to move tokens, but they do not necessarily have to belong to the person accessing the contract.
In other words, you can authorize another person or another contract to transfer funds on your behalf. Another use case includes automatic payment for subscription-based services, in case you don't want to manually send payments every day/week/month. The program will do it for you.
This function triggers the same event as transfer.

5. **approve**

```
function approve(address _spender, uint256 _value)
public returns (bool success)
```

This is another useful function for programmability. Applying it, it is possible to limit the number of tokens that a smart contract can withdraw from your balance. In its absence, there is a risk of improper use of the contract, since someone can exploit it for their own purposes or steal all balance.

6. **allowance**

```
function allowance (address _owner, address _spender)
public view returns (uint256 remaining)
```

The allow function can be used together with the approve function. By giving the contract permission to manage its tokens, you can use this feature to check how many tokens it can still write off.

Other functions like `name`, `symbol`, and `decimal` are not mandatory, but they can make ERC-20 contracts more clear and neat. They make it possible to add a well-readable name, set a symbol (for example OTC, ETH, DOGE) and specify how many decimal places tokens can be divided into.

2. Token‘s working principle

The key feature of ERC-20 tokens in comparison to the ETH itself, is that the tokens are not stored in accounts. They exist only inside the contract, which is a kind of autonomous database, which has columns on symbol, name, decimals and balance of the particular token on the particular address.

The way how tokens are transferred is also non-standard. When the transaction is sent, smart-contract puts the request into the usual ETH transaction, but it takes 0 ETH. Then, the transaction is completed, commission is also taken, because it should be put into the blockchain, but we transfer 0 ETH and the request to add some amount of tokens to receiver balance, removing the same amount from the request senders address.

3. Token Type

As was mentioned above, the utility token is the internal digital currency of the company or project, which provides advantages for users using the products of this company. In other words, utility token should simplify the service usage or give some ability for users who own tokens. These tokens can not be used as an investment to gain profit in the future. The simplest and obvious example of utility token is in-game currency, which users used to buy some in-game accessories or features.

To distinguish utility token from security token [9], our team applied the Howey Test. This test was developed in 1946 during the trial on appeal of the SEC decision, the court used 4 question to define financial instrument to be considered a security and be under the competence of the SEC, the instrument must meet the following criteria:

- Investments
- Profit expectations

- Investments should be in a joint venture
- Profit from the sale of the asset invested in the company

As the goal of our project is the utility token which will be the currency for our service, let's check all these parameters to be sure that OnlyTips Token is not a security token:

- Yes, our token assumes investments
- No, the profit is gained from other sources
- Yes, we attract external investments
- No, the profit from sold tokens will be used for exchange token to fiat

So, since there is a negative answer in 2 of 4 key questions the OTT is a utility token.

4. ERC-20 advantages and disadvantages

The main advantage of ERC20 is the standardization of tokens. Previously, unique tokens were created for projects based on the Ethereum blockchain, which made it difficult to interact between platforms and exchanges.

In order to ensure compatibility between them, additional mechanisms had to be developed. Which in turn caused the need to create other software layers for interacting with tokens. As a result, the process became very complicated and time-consuming. The solution to the problem was the creation of ERC-20 tokens.

Despite the significant advantage, the standard protocol is functionally insufficient. It is the main basis for the creation of Ethereum tokens and in no way guarantees the benefit or its operability.

Tokens can be modified, supplemented and at the same time maintain compatibility with the established ERC-20 standard. And at the technical level, it is quite simple to make a deployment. Which is definitely a disadvantage. As a result, many similar tokens appeared, thereby making the selection process more complicated for potential customers.

As with many cryptocurrency networks, Ethereum is not immune from scalability issues. In its current form, the network does not have high enough bandwidth. Attempting to send a transaction at peak times leads to high fees and sometimes delays. If you run the ERC-20 token and the network is overloaded, the transaction may hang.

However, it is not the problem of Ethereum only, it is one of the key disadvantages of the blockchain at all, but in the nearest future Ethereum 2.0 blockchain will be able to overcome it.

Anyway, for OTT tokens such disadvantages as lack of uniqueness and transaction speed are not critical at all.

First of all, the idea of a token to be a currency inside of the tip service with fixed exchange rate, so the complicated structure and features of the token will be a disadvantage according to the purposes.

Secondly, the transaction speed is not vital here, it is not a problem for the receiver to wait several hours until he gets his tips. For instance, the waiter in a restaurant will check his balance only at the end of the day, so, the transactions which were sent in the morning or afternoon have enough time to complete.

5. Reasons for ERC20 token

There are a lot of other standards for Ethereum contracts: ERC-721, ERC-223, ERC-1155, ERC-621, etc. Let's look at each of them separately and define why ERC-20 is the best choice for tips service purposes.

All of them have particular features that make them useful for specific tasks. In such a way ERC-721 gives the users ability to create their own unique tokens, ERC-1155 can operate with unique and non-unique (non-fungible) tokens at the same time.

Some of the standards focus on security, like ERC-223, to avoid accidental transactions. Also, there are contracts which allow you to burn or add some tokens, in other words to change the total supply number.

As it was mentioned above, the token in this project has the role of "dummy" unit, which has a fixed exchange rate, providing the possibility to track the transaction, sender and receiver addresses and the amount. The token is fungible and does not assume users to create their own tokens, transaction security is also under the service control, hence the ERC-20 base standards are enough.

6. Alternatives

For such service as tipping provider it is really sufficient to fight with commissions, since it is directly affects employees bonus salary. As OTT token implemented via ERC-20 standards it is working in Ethereum blockchain, which has a disadvantage, when the Mainnet is overloaded the commission for all transactions is rather high, which affects the cost of tip.

To solve this problem the token can be implemented via Moonriver parachain [10], which is the platform for smart-contracts, united with Ethereum, providing the "test" for the code in the real economics conditions, before it is going to the Moonbeam and Relay Chain.

The key advantage of this solution is that commissions in Polkadot Relay Chain, which unties parachains is rather small in compare to the Ethereum Mainnet. Polkadot has significatn improve of the scalability of the system, since each parachain or its own blockchain (connected via a bridge) acts in the Polkadot ecosystem as a single segment, which significantly increases the overall network bandwidth and, as a result, reduces commissions.

2.4 Tokenomics

2.4.1 Tips schema

For now it is sufficient to understand the whole process of the service and money transaction inside the business.

Let's start from the logic of the particular user: someone, would like to tip the waiter, courier or taxi driver for the service, then the person would go to our website and choose the cafe, taxi company or delivery app, which provides the service itself. Then, the user has to pick the employee and open the page with the employee's crypto address and the form where he can paste the amount of tokens, which he is going to send and leave a comment with the transaction.

Also, it is important to note that the tip is impossible if there are no tokens on the user's balance, so he has to buy it, with a fixed exchange rate which equals 100 rubles.

From the employee point of view the logic is simpler, after registration with crypto address confirmation, he is going to wait for tips from customers. Once, he got some amount of tokens on the balance so he can exchange it according to the same exchange rate which is 1 OTT = 100 rubles.

2.4.2 Business profit

The reasonable question arises, how does the business gain profit? According to the tests, the most convenient and comfortable for users and business commission is 3% per transaction. Let's get for instance, the average number of transactions per day equal 3 000m which is a real value for the startup service, and the expected value of tip equal 1 OTT, the service will gain 270 000 rubles per month.

Obviously, with growth of businesses which will connect OnlyTips service to their system the profit will also grow, which means that the key factor of success is expansion policy in business point of view and the marketing, which provide more users.

2.4.3 Token distribution

The total supply is 1 000 000 OTT tokens, which is not the final, as our token is both possible for mint and burnable, the token owner has the ability to burn some tokens or create more supply if the existing number is not enough.

Here is the distribution plan for all tokens:

- 55% of all token supply will go to the community and for tips services. This means that users will be able to buy at all 55% of all OTT tokens. As we give users tokens for real money and then provide real money for the tokens for employees, the 550 000 OTT is enough to end up the “transfer circle”. Since, we pay up only money that we earn from OTT sold, there is no need to attract investments to provide liquidity and the transactions will always be paid. So, the “transfer circle” is a process when users buy some amount of OTT, then tip all this amount to employees and employees exchange these tokens for money, the token owner burns them and mint new tokens to provide the same amount of token (550 000).
- 20% of all tokens will go to the investors, as at the start of the project, money for marketing campaign, airdrops and team salary. In addition to the part of the company's profit they will get some amount of OTT in proportion to the investments. In such a way it is the “insurance” of the investors that a particular amount of tokens which they can exchange will be on their wallets, until the investments are not paid.
- 15% will go for the marketing campaign, it includes free token allocation for the test groups, user's who will register in the app via the link, first users, bonuses, reward for the first tip, etc.
- 5% of the total token supply distributed and held on the team member's wallet's.
- 5% will be used as airdrops, which is a part of a marketing campaign, but it is provided in a specific way, so the pool for airdrop rewards should be restricted.

2.4.4 Metamask insertion

In the world of crypto currencies there are a lot of ways users can be deceived, so customers choose carefully and with a huge caution, which services to use. That is why asking users to give service both address and private key is a bad idea, which will scare away a lot of potential users. However, it is a well known fact that each transaction in blockchain needs a private key to confirm it, and the process of tipping also faces such a problem.

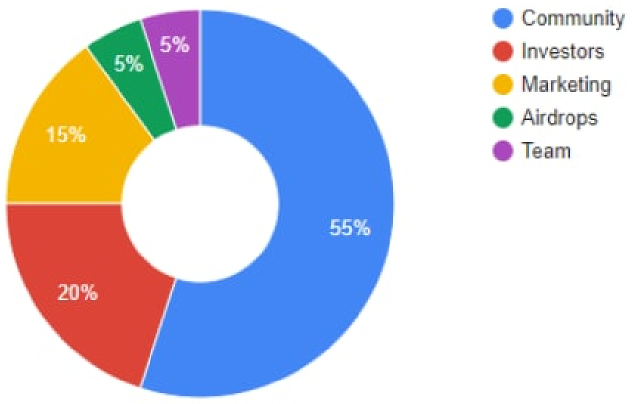


Figure 1: Token distribution diagram

Nevertheless, the solution for this problem is an online crypto-wallets [5], they protect users from theft of their assets, and provide their services for a serious period of time, so the client is not afraid to use them to sign transactions with his private key. Examples of such wallets are: TrustWallet, MetaMask, MyEtherWallet, Mist, etc.

The idea is that the user has to connect his MetaMask wallet (it is the most popular, but might be some other wallets will be able too), and once he bought some tokens they will be sent to his MetaMask address. If the user is going to tip someone, the pop-up window of MetaMask will arise and ask him to sign the transaction.

This works vice versa for the employee, if he would like to exchange tokens, then he sends tokens via signing transaction in the MetaMask and get the payment in fiat on his credit card.

2.5 Investment

As it was mentioned in the Tokenomics part, the project assumed to work without fiat liquidity pool as all tokens have to be bought before they are going to be exchanged. However, marketing campaigns, team salaries and fiat pool for exchange tokens that were gained through airdrops and bonus rewards should be taken into account.

For these purposes the investments should be gained. The plan for investments assumes that the project needs N rubles, which can be attracted from several investors. Each investor will get:

- OTT tokens in proportion $\frac{investments}{N}$ form the pool tokens for the investors (200 000 tokens). For instance, if the proportion of the investments is 10%, then the investor will get 10% of 200 000 = 20 000 tokens.

- The 50% of monthly profit will also be splitted in the same proportion between investors.

In such a way, part of the investments saved since they are locked in the token and other parts will return with profit each month.

2.6 Marketing

This part splits into two fields, the first is B2B search and the second is attracting customers.

The B2B part assumes search for restaurants, cafes, taxi companies, delivery services , etc. to connect with the OnlyTips app, adding their business and employees in the database and website.

Profit for businesses is the innovation in tips service, which is a good for business reputation and can be a good marketing campaign. Also, the employees, who gained extra money from not only fiat tips, but the crypto too will be glad and grateful, which gives them extra motivation to work hard. What is more, it is free for businesses and all costs are on the side of OnlyTips.

Once a huge number of businesses connect their employees to the service, the second stage of the marketing campaign begins. It also has two parts, the first one is advertisements in social media, referral programs and bonus programs. Parallel to this story, the airdrop is going to be conducted, which also assumes social media and word of mouth processes.

Applying all these parts correctly, the project will engage both businesses, employees and customers, who think in a crypto-enthusiastic way and want to touch the future in several clicks.

2.7 Main result

The main result of the project is to study principles of web application setting, dig into concepts and working processes of the blockhain technology, learn what may smart-contracts contain, how they are written and how they can be connected to a web app. Another aim is to analyse the crypto segment in general, define the level of perspective to create a such a startup, discuss marketing and investment nuances and create a business strategy for the the tips service.

2.8 The role of members of the project

Our team is consisted of three members that have approximately equal contribution to the project: Marianna Rybnikova who was responsible for the website, Gleb Zotov, who is the token creator, and Dmitry Kurbatov, who analysed the DeFi sphere and also connected smart-contracts with the website so that the service may functionalise properly.

To be more precisely, the roles of the members of the project are defined in the following way:

Marianna Rybnikova

- Website implementation

Frontend:

- User interface creation including web pages styling, animations, managing content on each page

Backend:

- Database and model development
- Implementation of user authentication systems (registration, login and edit profile forms) and main functions of a website

Gleb Zotov

- Tokens analysis
- Tokenomics. Calculation and estimation of token distribution
- Token development
- Backend implementation
- Drawing up a business plan for investors
- Development of marketing campaign strategy

Dmitry Kurbatov

- External market analysis
- Competitor analysis
- B2B and B2C analysis and identifying target audience for startup
- Backend web-site connection
- Business analysis implementation

3 Comparative analysis

3.1 Segment analysis

In order to analyse the crypto segment in Russia we decided to perform a PEST analysis that is concluded in detecting potential problems and risks in political, economic, social and technological spheres.

- **Politics**

  In distinction from many European countries, cryptocurrencies in Russia are not as popular as they might be. It is still impossible to pay in supermarkets, restaurants, shops by bitcoin for example, whereas in Germany, Italy or Spain anybody can make transaction in cryptocurrencies in order to pay for something. Moreover, Italy is considered to be one of the most attractive and popular regions for crypto payments, more than 15% of the entire crypto operations are made in this country. Countries neighbouring to Russia (Belarus, Ukraine) also have crypto assets officially legalised and even more, in Belarus such transactions are not obliged with any taxes. In Ukraine a federal act permitting crypto transactions was published in 2021 and for now no tax obligation exists, however, in the near future it is going to be introduced. Talking about Russia, the legislation contains definitions of cryptocurrency, crypto assets, wallets and etc., mining is permitted and considered as entrepreneurial business as in other countries, however, no official law allowing payments in crypto exists.

  Another political problem that currently exists is the 5-th sanctions package from the European Union with respect to Russia regarding freeze of any crypto assets for individuals under sanctions and for those who proceeded operations with crypto wallets via foreign banks. Due to these facts there is a risk that government decides to forbid any operations with cryptocurrency within the territory of Russian Federation that is perhaps the greatest concern for our project right now.

- **Economics**

  Our service does not demand high number of employees in order to manage the performance and support the functionality. Unless the tip service is popular enough, no one except the group of creators is demanded for proper working platform. This implies we do not depend on unemployment rates and approximate salaries in the region. We are aiming at first steps at collaborating with cafes and restaurants which are highly developed in the central part of Russia.

  The main issue actually does not depend on the specific region or country as it is connected with the exchange rate of cryptocurrencies in general. As such currencies are not the primary ones, they are not even the reserve currencies that leads to high volatility and unstable exchange rates. As commonly used Bitcoin and Ethereum cost more than 10 000 $ in equivalent, their fluctuations affect relative prices in crypto much more than compared to the usual prices in dollars or euros.

- **Socio-Cultural**
  Crypto sphere is brand new in Russia as the main development of mining and thus, operations with cryptocurrencies, began in 2017. Nevertheless, the greatest share of Russian population is still not aware of the concept of cryptocurrencies and blockchain. According to the survey, conducted in 2021 by "tripleA" [11], only 12% of Russians own crypto assets and less than 5% understand its concept and working principles. These facts state that people do not trust cryptocurrencies sufficiently and therefore, tend to pay or send tips using credit cards. On the other hand, in 2021 Russian government claimed that in the near future there is a possibility of appearance of a legal DeFi system and therefore, in next 5-10 years there is a chance that many services will become crypto directed. In addition, in many schools and universities today teachers try to tell the students the basic concept of cryptocurrencies to improve their IT and blockchain knowledge.

- **Technological**
  From the technological point of view Russia is an attractive market because despite the fact that at the legislation level crypto transactions as a payment method are prohibited, mining is allowed with several restrictions and conditions. In 2017 the number of mining farms rocketed and now the approximate quantity of farms exceeds 300 000, in 2021 miners earned more than 4 billion US dollars that is the third place in the world after USA and Kazakhstan. In addition, Russia has good reputation among crypto experts around the world and highly unlikely any restrictions would be imposed on local mining farms.

To summarize, from PEST analysis it can be stated that Russia can be considered as an attractive market for tips service based on blockchain as there are numerous mining farms and it is not a problem to create a tipping platform. The main risk here is that at the moment customers will probably be afraid of such a service because of the global level of knowledge and awareness of cryptocurrencies among the Russian population. However, in case of lack of competition, in a 3-5 year perspective the sphere may be rather compelling.

3.2 Competitors analysis

Today, tipping platforms have become a great alternative to cash tipping which society has had for years now. There are several ways how tipping can be done cashlessly: by using cryptocurrency or via a customer's credit card. We will start with the tipping platforms that have been used in media and social networks and work with cryptocurrency only and then proceed to services that operate with the usage of mobile-payment apps and credit cards.

First of all, we would like to discuss tipping services on cryptocurrencies.

- **Tippr**

  That is a Bitcoin Cash tipping bot [12] which allows users to reward authors of good content on Twitter and Reddit [13]. The tipping transaction can be done by using the following command: "$0.50 @tipprbot" in response to the twitter / post of the content-maker. A few years ago, it became a well-known platform among users in social networks and forums (like Twitter and Reddit): they spread cryptocurrency micropayments to the authors they supported. The bitcoin cash transactions have really low network fees (around 0.003$) and thus were in use. Moreover, all the transactions were tracked: the founders of Reddit made it possible to follow the statistics which reflected the popularity of the bot. People were able to have a look at the top tippers and analyze the amount of payments done throughout a day. However, 2 years ago both Reddit and Twitter bots stopped operating. Luckily, Chaintip appeared and became a better alternative, since the money went back to the sender if unclaimed, and it forwarded funds directly to the receiver's wallet.

  Advantages:

  1. Relatively low fees.
  2. Simple in usage.

  Disadvantages:

  1. Narrow range of usage: tips only for Twitter and Reddit users.
  2. The service appeared several years ago when crypto tipping platforms were not popular.

- **Chaintip**

  Chaintip [14] is a tip service working on Bitcoin Cash and that guarantees the gratuity will reach the destination. The process is followed by a special bot that sends private messages to both tippers and tippees in order to obtain understanding between parties. The service allows to make operations both privately and publicly, thus, if a tipper desires not to uncover

his or her name, a tippee will get the gratuity with the notification excluding the sender information. Another advantage of such a service lies in the opportunity to get money back if there is an error in tip delivery to the receiver. As well as the other tip services, Chaintip requires the receiver to have a BCH wallet and a special link to address the tip. Unfortunately, the tippee needs to create the wallet and the link at least one week in advance as otherwise the operation will not be executed.
Advantages:

1. Possibility for the sender to tip anonymously.
2. In case of troubles during the transaction money will be returned.

Disadvantages:

1. Necessary to have a Bitcoin Cash Wallet linked in order to transfer tips.
2. No public information on commission available.

- **Gitcash**
  Github is probably one of the most famous and popular websites for programmers and other people connected with IT. As GitHub is absolutely free service people who share the code with others are not paid off, however, their knowledge may be extremely useful for some other programmers and the last one may desire to tip the code creator for assistance and help and here the Gitcash [15] service arrives. Using the Gitcash.io [16] platform developers may tip someone by connecting to the Bitcoin Cash wallet and transferring via the platform gratuity. Interesting thing is that Gitcash was at first sponsored by Bitcoin and the tip platform was designed by the Chaintip development team.
  Advantages:

  1. Open source code for the whole project.
  2. First tipping service for programmers.

  Disadvantages:

  1. Narrow range of usage: tips only for GitHubs users.
  2. Necessary to have a Bitcoin Cash Wallet linked in order to transfer tips.

- **CoinText**
  The main idea of this tipping platform is that there is no need to install any apps in order

to use it. Via a CoinText service, a Bitcoin Cash [18] tip can be sent directly to someone's mobile phone via SMS and the only thing that is needed for both sender and recipient is the access to the Internet. This platform is available for citizens of more than 100 countries now. In 2019 founders of CoinText added the ability to pay Bitpay Invoices using SMS [19] via utilizing the BIP70 protocol. This basically means that now users are able to copy a URL of the protocol payment and pay the resulting invoice through the mobile text messaging. Moreover, the same BIP70 protocol is supported by another tipping platform which is called Anypay Global. It creates a Bitcoin Cash invoice in the specific app, after that the service provides a CoinText code for a payment. The main advantage of the CoinText so far is that it can be used even by people who do not have a smartphone and who use basic feature phones. The fee taken from each transaction cannot be considered as too high, however, compared to Tippr it is 10 times greater: 0.05$ per each tip.
Advantages:

1. Tip can be sent using the SMS code without any Internet connection.
2. Reasonable fee.

Disadvantages:

1. SMS messages fees depend on the operator and in case of some each message may be quite expensive.
2. Necessary to type the receiver‘s address manually that is totally inconvenient.

These are the most popular tips services in the world right now. Some of them have already disappeared, however, most of them are still operating successfully. There can be global advantages and disadvantages highlighted. Let us start from the **advantages** first:

- The blockchain technology makes cryptocurrency tipping platforms much more secure. Every transaction is tracked, allowing users to check whether a particular payment has reached the definite recipient.
- All operations using cryptocurrencies are traceable and thus, they are more legal and more convenient for the government to monitor monetary movements.

Disadvantages:

- Cryptocurrency makes fees extremely high (especially true for Bitcoin and Ethereum).

- All the platforms that are now available on the market specify only on a particular cryptocurrency and cannot be applied to any other.

However, all above example regard foreign services only as in Russia we did not succeed to find any similar suggestions. But electronic tips in Russia are popular and therefore, there are numerous different tips services. Here are few of them:

- The idea of the most popular service - **Чаевые просто**, which is now **Yandex.Tips** is rather simple, the waiter registers on the website and receives a virtual Visa card for free. If a visitor wishes to tip, the employee shows him his personal QR code, which can be displayed on the bill, table tent or mobile phone screen. The guest scans the QR code, enters the tip amount on the payment site and confirms it with a card. The money is immediately credited to the waiter's account. The commission for money withdrawal is 1%.
  For restaurants this service has a more convenient function - free setup of QR-code integration into the bill and the development of layouts for business cards and table-tents. The main advantage of this particular service is low commission for the transactions and a plethora of payment methods.

- The next service to study is **SberTips** [20], which is the strongest Yandex Tips competitor, here the personal employee code for entering into the application is specified in the bill. After displaying the order on the screen of his smartphone, the guest pays for it with a card, adding a tip to the waiter. The service differs from many similar applications in that, in addition to paying tips, you can book a table, pay the bill, order the preparation of a dish by a certain time. Commission in the service is higher compared to Yandex.Tips - here it reaches 5% per each transaction.

- **Tips+** [21]. One of the few services that does not involve installing an application and even scanning a QR code. The working principle is as following: tips can be transferred by card or using a smartphone using the NFC function via a contactless terminal located in the restaurant hall. It is enough to choose the tip amount and transfer it directly to the waiter. A significant disadvantage: the restaurant will have to rent a terminal and pay a commission of 10-20% for it. And it is also impossible to bring terminals to the guest's table, since they do not have batteries.

Actually, there are other services similar to those described above, for instance, **TipJar, EasyTip, NetMonet, Tips Delivery** and etc., but we decided to take a closer look at the most popular ones.

Advantages:

- The main advantage of the cashless tips is the availability. Nowadays almost everyone has either a card for payments, or Apple/Google Pay services, in both cases it is much more convenient than tipping with cash.
- Most cryptocurrencies are volatile , so it can be the case when 100$ of tips during the week will drop to 80$ for example. Tipping via simple currencies is much more safe.
- To cash out crypto currencies people have to use currency exchange services, which also take the commission for the transactions ($\sim$ 5%). Cashless tipping services only take their built-in commission.
- There is a huge advantage for restaurants and other businesses which are strongly linked with tip services, as cryptocurrencies are still illegal in most countries, hence it is impossible to use tips via crypto officially. While the cashless service is legal and accessible from the law point of view, there are a plethora of examples of restaurants, barbershops, pubs, etc. which use Yandex Tips for instance.

Disadvantages:

- Some employees do not have access to banking tools and thus cannot use digital payments.
- Impossible to track whether the money goes directly to the employee – for this reason many customers do not trust cashless tips.
- Platforms charge some amount for their services, so the employee may receive less tips at the end.
- No platforms that are easy and convenient for both employer and customer.

To sum up, today there are no platforms and services for tipping cashlessly that would benefit both the customer, the worker and the founders of the tipping system. As we can see, some services take huge commissions, some employees may not receive the whole amount of tips and some customers do not trust the existing platforms. Cashless tipping services, both those that work as separate apps, and those that are integrated into the ecosystem (for example, Delivery Tips) have enough disadvantages that are worth working on, and it is tipping on a cryptocurrency basis that can solve these problems. Talking about cashless tipping platforms that are based on the mobile-payment apps, they do have a less complicated structure without a need to trace all the monetary operations. Today, tipping via crypto still has not gained enough popularity and still does not have a convenient platform, thus, we arrive at the lack of demand. The share of restaurant visitors, as an example, the number of tips with a Bitcoin Cash wallet is extremely small. Cryptocurrency tip services also have a huge room for improvement, as it was mentioned above, there were a lot of attempts to create a stable, balanced, comfortable and legal tips service, however today it is still a point to achieve. So, in our coursework we are aiming at creating a system that will take into account all the listed disadvantages and will overcome some of them in order to make tipping by cryptocurrency available and convenient for all users.

4 Project implementation

4.1 Frontend functionality for web application

4.1.1 Design Specification

Main color

The world of cryptocurrency is usually associated with dark color palettes. After looking through the suggested palettes of the services from financial and trading & investment sphere, it was suggested to try a combination of three main colors that were chosen according to analogous color schemes (colors that are located next to each other on the color wheel) and complementary color schemes (colors that are on the opposite side from each other on the color wheel).

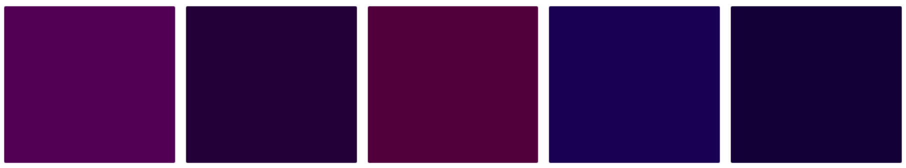


Figure 2: Main colors scheme

Accent color

Plus, a neon accent color was suggested in proportion of 90 - 10: 90% of the palette consisted of the dark background colors and only 10% of the elements were filled with the contrast ones. Neon palette was applied to button animation and to some headers only. As a result, customers managed to navigate easily in the web app and were staying focused on the main functional features of the crypto service at the same time.

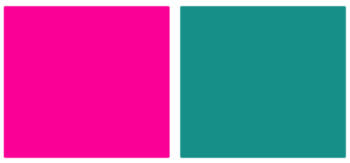


Figure 3: Accent colors scheme

Main fonts

A combination of two various fonts was applied to all the content of the website. The choice was based on several facts:

- Readable
- Legible design

- One of the fonts is used for both long and short sections of the text
- One of the fonts is for headers only and thus can be less readable

Another point to mention: the fonts were taken from Google Fonts. The choice was made on this service, since it is free and includes dozens of fonts to choose from. As a result, we selected **Comforta** for some small headers, buttons and text sections, while **Great Vibes** was used for a main header only.

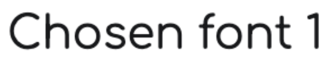


Figure 4: Comforta font sample

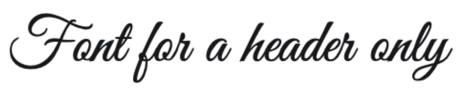


Figure 5: Great Vibes font sample

Main instruments for styling

The implementation of the website design was done by using HTML and CSS. First of all, a few words about HTML. The Hypertext Markup Language is a well-known tool among web designers and is used to create most of the web pages and applications online. HTML describes the markup of the created page. Different types of elements basically show the browser how the page should be shown to the user. CSS in its turn, stands for Cascading Style Sheets and is used to describe the way how HTML elements are displayed on a screen. It can control the layouts and styles for multiple pages that should be of the similar appearance. Moreover, using CSS made all the HTML files much more readable and helped to add some small fancy details to the interface at the same time. For instance, animated buttons were implemented to make the website more up-to-date and more reactable to users' actions on a page as well as the design for the whole page. Finally, CSS was used to create a common template for several web pages that share some content with each other and thus are quite similar from the design point of view.

4.1.2 Developing and maintaining the user interface

There are several types of the web pages in the implemented service.Some of them are needed for reading purposes only, while others require filling in some forms from the user. This section is dedicated to different types of web pages used: we will have a closer look at their structure and design.

Here it is crucial to understand the logic of the client that uses OnlyTips. To make the project a little bit easier, the interaction of the user with this service was assumed to be via the computer only: in the future versions of OnlyTips a mobile website will be added as well as the access from a variety of gadgets (for now it works locally to be less focused on data security). The main focus was made on a so-called multimedia user interface. For instance, a customer that wants to leave a tip uses a mouse to click on the buttons and choose some particular elements on the pages and a keyboard to leave some personal data for future operations, to pick the amount of tips they want to leave and to write some comments to the waiter. Finally, a cursor is used to navigate on each page. The first two are combined into a tactile input, while the last one is followed by visual effects (e.g. animation on a screen). Together these three instruments help the user to interact with the service correctly and support an effective functionality of the website.

Form-Based user interface plays an important role in this case. Before a waiter can receive any tokens and a customer can leave a tip, both of these user classes have to fill in a form with their details: first name, last name, email, password, photo, crypto wallet, additional information and a cafe (the last two are for waiters only). The design is quite minimalistic and simple for understanding. Now, let us look at each of the implemented form:

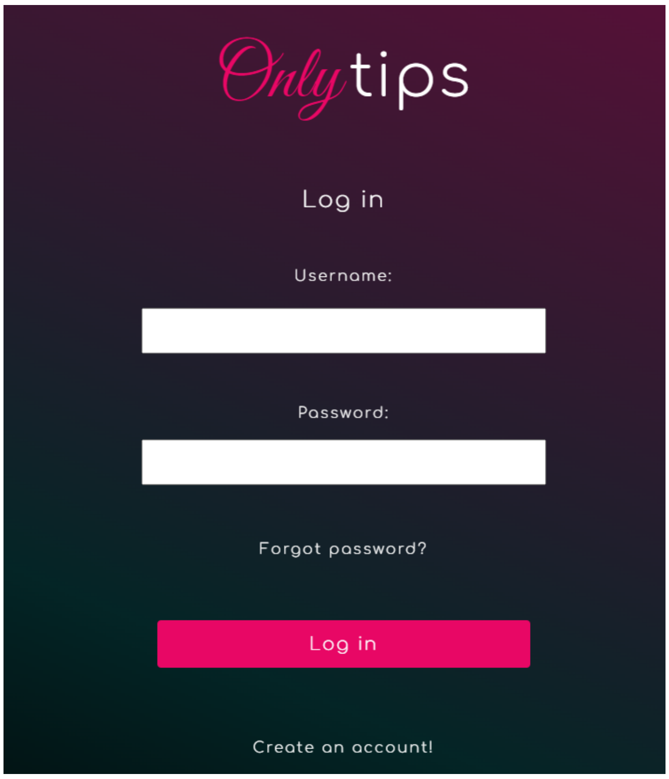


Figure 6: Login form

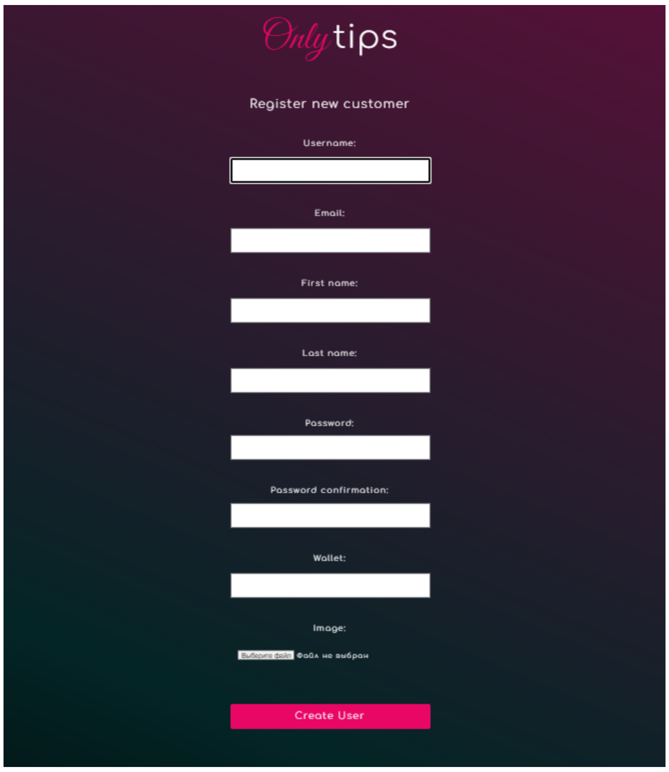
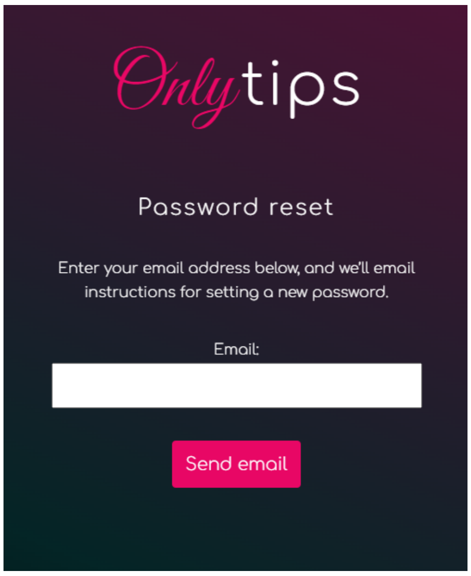


Figure 7: Registration form

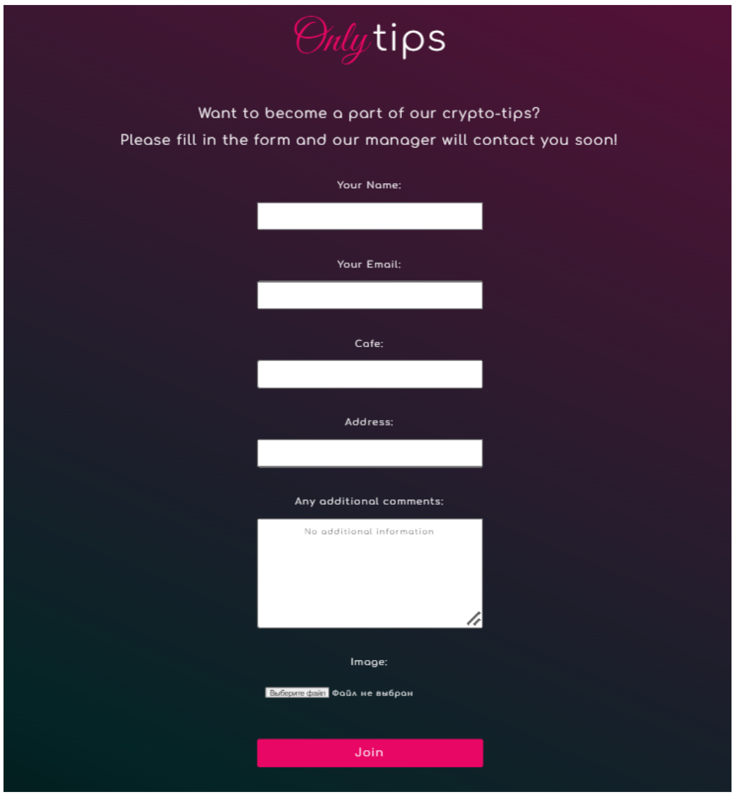
One of these forms should be completed before accessing personal information like first name, last name and a photo of the waiters that are already connected to our service. After that, all the input data is saved in the database according to a type of the user (waiter or customer type). The structures of data will be discussed in great detail in the section *Backend functionality*.

If for some reason the password was forgotten, the user has the opportunity to reset it via a reset-password form:



Enter your email address below, and we'll email instructions for setting a new password.

Figure 8: Login form



Want to become a part of our crypto-tips?
Please fill in the form and our manager will contact you soon!

Figure 9: Registration form

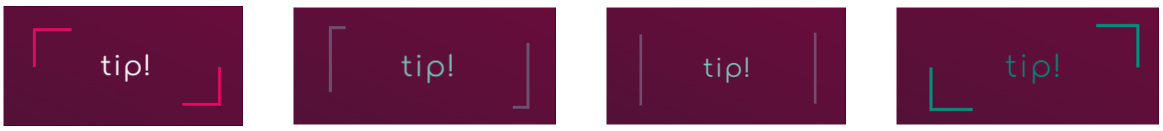
That is a key functionality which exists in every user-friendly service. It consists of an email field only. After the form is submitted, a user receives an email with the following instructions of a password resetting procedure. After all the stages are completed, a user can authenticate with a recovered password and their username.

"Join us" form

In order to use the suggested server, a user needs to register and to choose a cafe they are working at. The OnlyTips database has a separate table that includes the list of all the restaurants that have joined us before. New location can be added only by the administrator, that is why the following form is introduced. If a waiter is from a cafe that is not included in the database, then they or their employer can fulfill the form, and their restaurant will be added to the service very soon.

After the discussion of the form-based user interfaces, proceed to the graphical user interface, that helps to stay focused on the content and actions the user is doing and makes the navigation easier.

Button "tip"



(a) "tip!" button (1) **(b)** "tip!" button (2) **(c)** "tip!" button (3) **(d)** "tip!" button (4)

Figure 10: "tip!" button animation

Once a user hovers the cursor over the button, the animation occurs in response to a user's actions. Since the main function of the service is to leave / receive a tip, a button that leads a user to this action must be noticeable: it is not only animated, but also surrounded by the neon color that was chosen as an accent color. When users open a main page for the first time, they understand what they should do next immediately.

Small navigation buttons

OnlyTips website includes several web pages that can be discovered. For that reason, all the links are provided with a minor animation which responds to the cursor movement and makes the user focused.

(a) "join us" button (1)

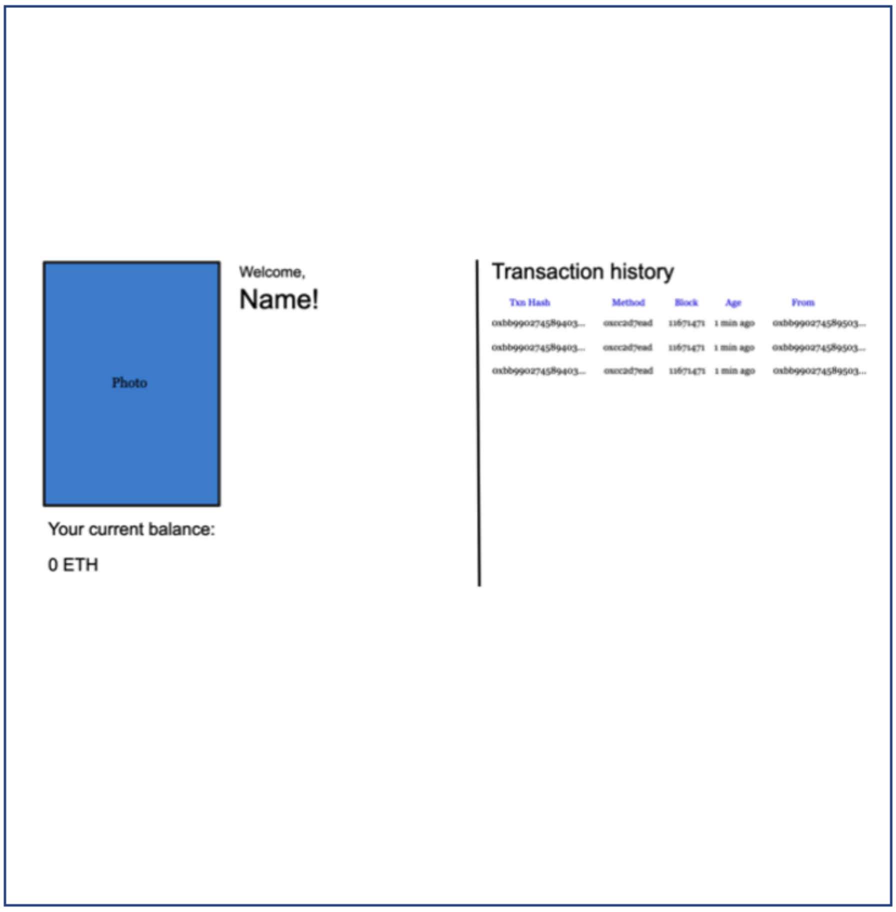
(b) "join us" button (1)

Figure 11: "join us" button animation

4.1.3 Page Templates

After the project implementation began, a few analogues have been discussed. Several of them are overly complicated in some sense, from our point of view. A customer that wants to tip is not aiming at surfing a website for a long time: everything must be quick, logical and straightforward. The idea behind our service is that only essential functions must be included. A customer is here to thank a waiter, therefore this action must be done as soon as possible.

From the beginning we suggested this idea, and it can be actually seen from the comparison of the draft and a final waiter's page:



Your current balance:

0 ETH

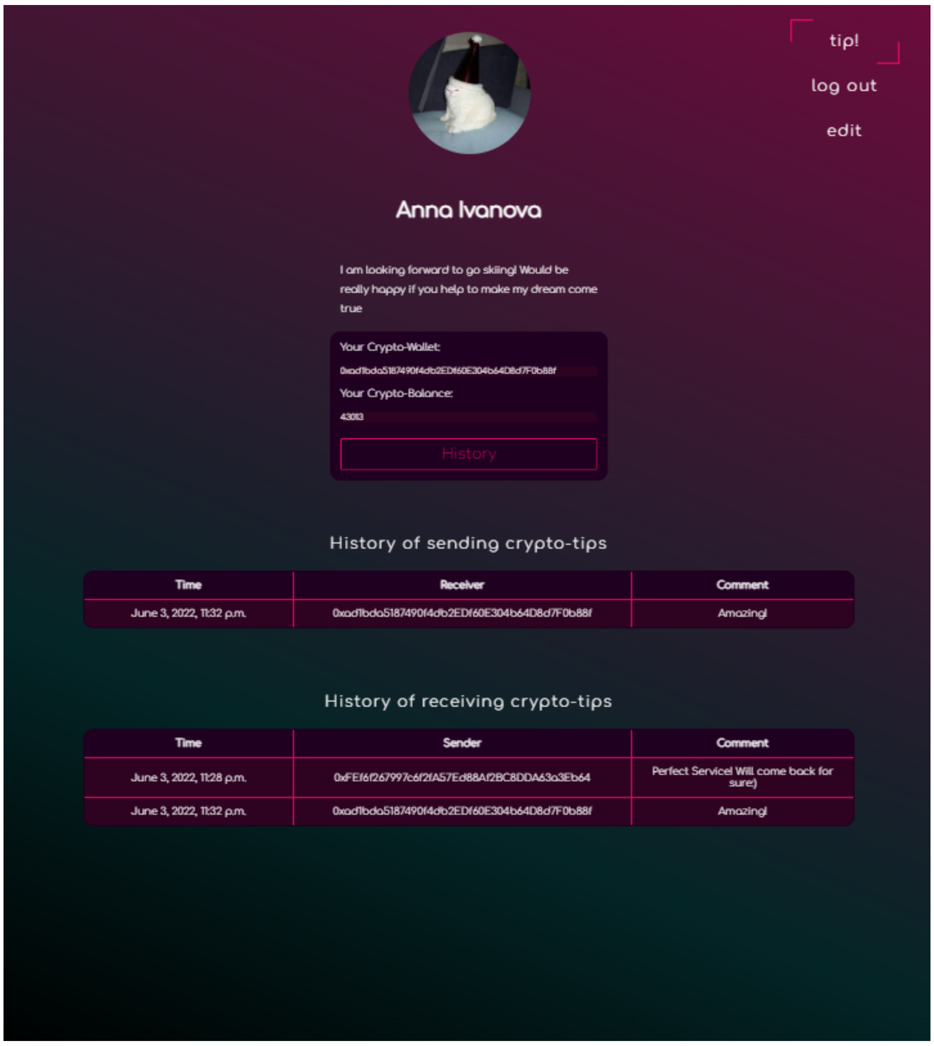
Transaction history

Txn Hash	Method	Block	Age	From
0xbb990274589403...	0xcc2d7ead	11671471	1 min ago	0xbb9902745895o3...
0xbb990274589403...	0xcc2d7ead	11671471	1 min ago	0xbb9902745895o3...
0xbb990274589403...	0xcc2d7ead	11671471	1 min ago	0xbb9902745895o3...

Figure 12: Web application draft

Note that even though the design and the position of the elements changed a lot, the functionality remained the same. All the needed requirements are included, however, no irrelevant information is present. This will help users to navigate much faster and, as a result, they will likely not have any negative experience over the session on the website.

Now we are ready to proceed to a backend part, where all the relations and functions will be described.



Anna Ivanova

I am looking forward to go skiing! Would be really happy if you help to make my dream come true

Your Crypto-Wallet:
0xad1bda5187490f4db2EDf60E304b64D8d7F0b88f

Your Crypto-Balance:
43013

History of sending crypto-tips

Time	Receiver	Comment
June 3, 2022, 11:32 p.m.	0xad1bda5187490f4db2EDf60E304b64D8d7F0b88f	Amazing!

History of receiving crypto-tips

Time	Sender	Comment
June 3, 2022, 11:28 p.m.	0xFEf6f267997c6f2fA57Ed88Af2BC8DDA63a3Eb64	Perfect Service! Will come back for sure!)
June 3, 2022, 11:32 p.m.	0xad1bda5187490f4db2EDf60E304b64D8d7F0b88f	Amazing!

Figure 13: Profile page

4.2 Backend functionality

4.2.1 Database management

The following models were created over this project:

1. **Cafe**

   Each Cafe object has a unique key – cafe slug – and is described by the following features: its location (that is, a city and the exact address of the cafe) and its title. Moreover, each cafe is provided with its photo or a logo to help users navigate faster.

2. **User**

   Each User object has a unique key – username. Moreover, this model consists of a first name, last name and the email as well as the password to the tipping account.

3. **Waiter**

   Waiter model can be considered as a one-to-one relation with the User model. This means that each registered person can be either a waiter or a customer, plus, he / she is a user. It is more convenient to store this data in two distinguished tables that are connected by a username, since Django Framework suggests quite secure and rather simple solutions for storing passwords of a User class (default one). Moreover, each waiter has a picture.

4. **Customer**

   Similarly to a Waiter model, this one can be considered as a one-to-one relation with the User model. This means that each registered person can be either a waiter or a customer, plus, he / she is a user. Additionally, a customer can provide a photo, but it is not obligatory.

5. **Transaction**

   The final model was created to store all the transactions effectively. Each of them has a unique id, which helps to identify a particular transaction. Then, each object of this class is described by a sender crypto wallet number, a receiver crypto wallet number,a date of the transaction, its amount and any additional comments a receiver decided to leave (there are two possible cases: waiter send tips to a waiter or customer sends tips to a waiter).

6. **Join**

   That is a table that is filled through sending a form "join us". It is needed for the administrator only. It includes such fields as the name of the form sender, a cafe name, its photo and address as well as any additional comments and the email address. As soon as a new observation is added to the table, the administrator adds a new instance into the database Cafe with all the needed details. After that, waiters from a newly added cafe can create their profiles in our system.

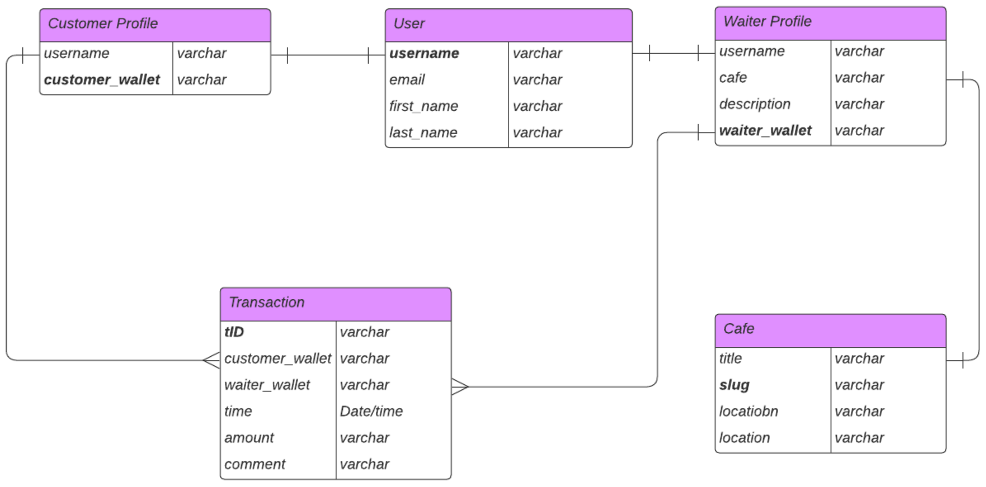


Figure 14: Database structure

4.2.2 Project structure

The implemented project is huge enough and includes a variety of implemented forms, classes, models and functions. In the next subsections the main ones will be covered to provide the reader with a detailed overview of the key points.

Begin with a short description of a project structure and the included files that were constantly changing throughout a project:

1. **blockchain.py**

   The file includes all the functions related to the transaction of the token, balance checking, creation of a testnet, etc. This is described in great detail in the Smart Contract Section by my colleague.

2. **models.py**

   This file includes all the models that were implemented in the project: `Cafe, WaiterProfile, CustomerProfile, Join` (model that is needed to store all the inquiries about the inclusion of a new cafe into the system) and `Transaction`. Each class is defined more or less in a similar way. First of all, each field is assigned to a variable type (for example, `CharField, DateTimeField, FloatField, TextField`, etc). Then, a function `__str__` should be defined. This will allow to change a string representation of the project.

3. **forms.py**

   That is the file where all the project forms are stored. As it can be seen from the previously discussed sections, quite a huge number of forms were implemented: registration form for waiters, registration form for customers, update form for both types of users, join form and a token transaction form. The key idea behind all the classes in this file is that we use a Meta subclass to describe a behavior that is expected from a class which is described.

4. **urls.py**

   In order to make a web page visible in the browser, a URL path should be created. We used a dynamic way of assigning it – each URL path has its own unique name. As a result, changes of the path appearance itself does not break the whole code. In addition, each path instance includes a particular function from views.py file that will be run when the user enters the web page.

5. **views.py**

   This file describes functions that get a web request from a browser, implement some tasks and return a web returns (redirection to some other page, HTML template, etc)

6. **admin.py**

   Is used for the admin interface creation. Every implemented data class is registered in this file in order to be visible to the admin on the administrator side of the website. It manages the content and helps to structure the frontend content around.

7. **setting.py**

   That is a core file where a particular app is registered in (in our case, the app is called Tips). Moreover, it stores all the database settings, logging and registering configurations as well as the static files, if present.

8. **styles (include CSS files)**

   As mentioned before, CSS [8] files are needed to make HTML files more readable and to style an HTML template. The appearance of each page is controlled by CSS files from this folder. Some pages share the same CSS, since the arrangement of the elements and the content are quite similar.

9. **templates (include HTML files)**

   This folder consists of all the web page templates. Since the main tool for a markup is

HTML, each web page has its own file, where all the blocks are assigned to some element type.

10. **uploads (include images)**
A folder that stores all the uploaded images. If a waiter or a customer adds a profile picture, it will appear here as soon as the form is saved in the database.

4.2.3 User Authentication

1. Registrations

As discussed earlier, a User model was taken as a base for the authentication procedures. This model provides very flexible and secure settings for password storage: raw password is not saved in the Django database [22] and only its hashed version is kept. By default, Django framework uses the PBKDF2 algorithm with a SHA256 hash, which is recommended by NIST. From this moment a problem was faced: both a Waiter and a Customer models require some fields that were not included in the default User model (for instance, a Cafe field or the description of the waiter). The following solution was found: our custom models can be in a one-to-one relation with the User model. For that reason, when some authentication form is completed by the user, Django manages two forms at once: a username, last name, first name, email address and a password are stored for the User model, while all other fields including a Cafe title, a description and a wallet number refer to a Waiter or a Customer model that are connected to the default User model with a unique username field. This rather simple solution has provided a good level of security that we can deal with and has made a Database more readable. The most complicated structure belongs to the User Registration Form (class `ExtendedRegistrationForm`).It consists of several parts. Firstly, all the required fields are described. All the required data structures and the sizes of each field are assigned. Then, a Meta class with the User model is defined. This shows to Django Framework the exact storage, where the data from the form should be saved. After that, all the default help text from the fields "username" and "password" is removed , as the form is big enough and such blocks of text take too much space. Finally, save() function is defined. Django receives the instructions on the data that have to be stored when the registration form will be called on a website. The other registration forms (either `WaiterProfileForm` for Waiter registration or the `CustomerProfileForm` for the Customer registration) have a very simple structure and straightforward approach. In both cases all the completed fields are just saved into the required database. If the form was successfully sent and it included all the information (required fields that cannot be empty and a picture), the user is logged in and redirected to the page with the cafes.

2. Login

For a registered user, a login form is provided. For this authentication procedure a default `login()` function was applied. Since the login is completed by using the username and a password only (both fields are from a User Class), Django manages everything itself. An important remark: users cannot search for a particular waiter or a cafe until they are not authenticated. This secures personal information of the people that made posts on the website and decreases the number of users that can look at it.

3. Edit profile

The approach here is somehow similar to the registration form. The main difference is that the username cannot be changed. A user fills in only the field that they would like to change. The function `edit_profile()` checks each field for the presence of some data. If no data was written, then the field value remains the same as it is now.

4. Log out

The log out procedure is completed in the same way as the log in. The default Django function `logout()` is used. Since the default User model was applied, it supports this functionality.

4.2.4 Navigation

Now, the main functionality of the website will be discussed.

Choosing a cafe

As soon as the User logs in, they are able to choose a cafe they are sitting at. A list of available locations is shown to them. Each Cafe instance consists of its title and a photo. The cafes are ordered by their title in the alphabetic order, so the search will not take too much time. To show a list of all the cafes that are connected to our service, a function `Cafe.objects.all().order_by('title')` is called.

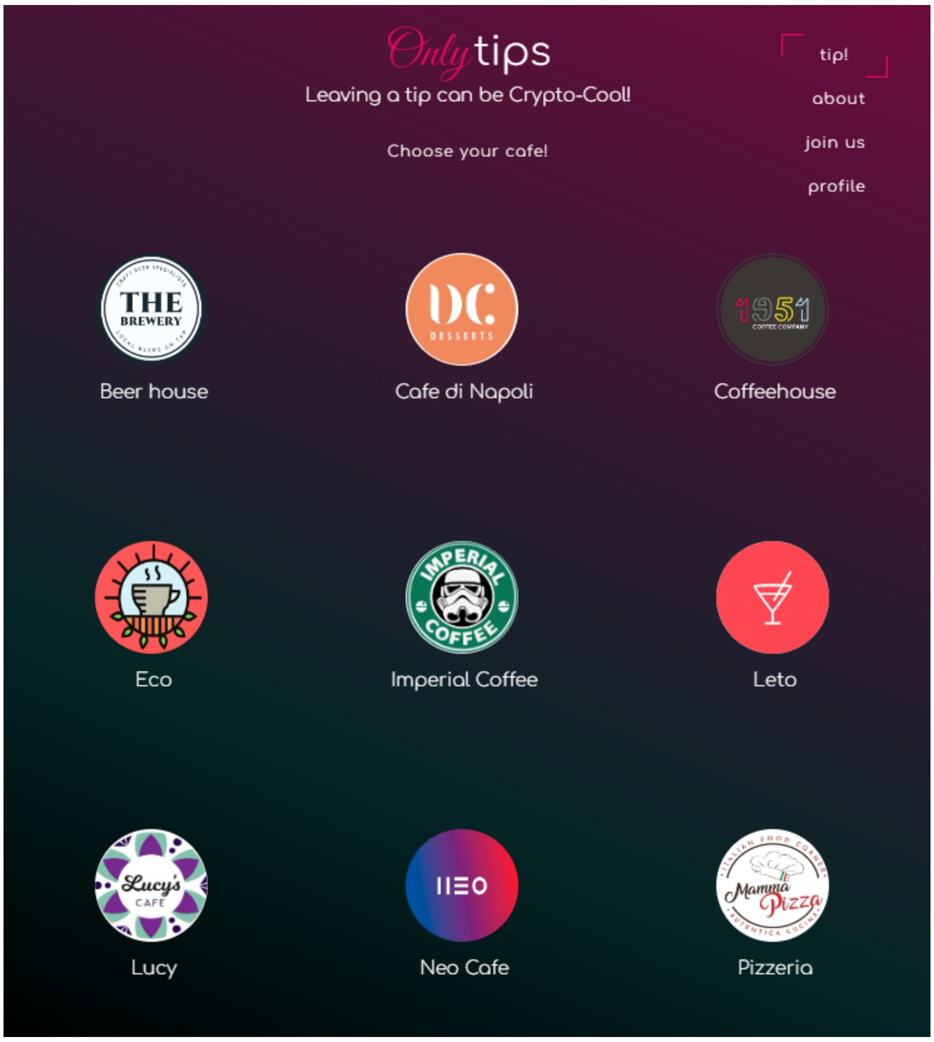


Figure 15: Cafes page

Choosing a waiter

Similar interface is seen when a user clicks on a particular cafe title. After that the list of waiters who are working at this location is shown in alphabetic order to a customer. Each waiter instance consists of a photo and a name – this will help the user in the navigation process, as a customer remembers the face of the waiter and can find this person in the list. An array of waiters who work in a particular cafe received by the the following function:

```
list(WaiterProfile.objects.filter(
cafe__title=selected_cafe.title
).order_by("user__last_name"))
```

All the waiters' photos that are seen on a website were generated by the AI. We care about personal data and images of other internet users and do not upload them without permission.

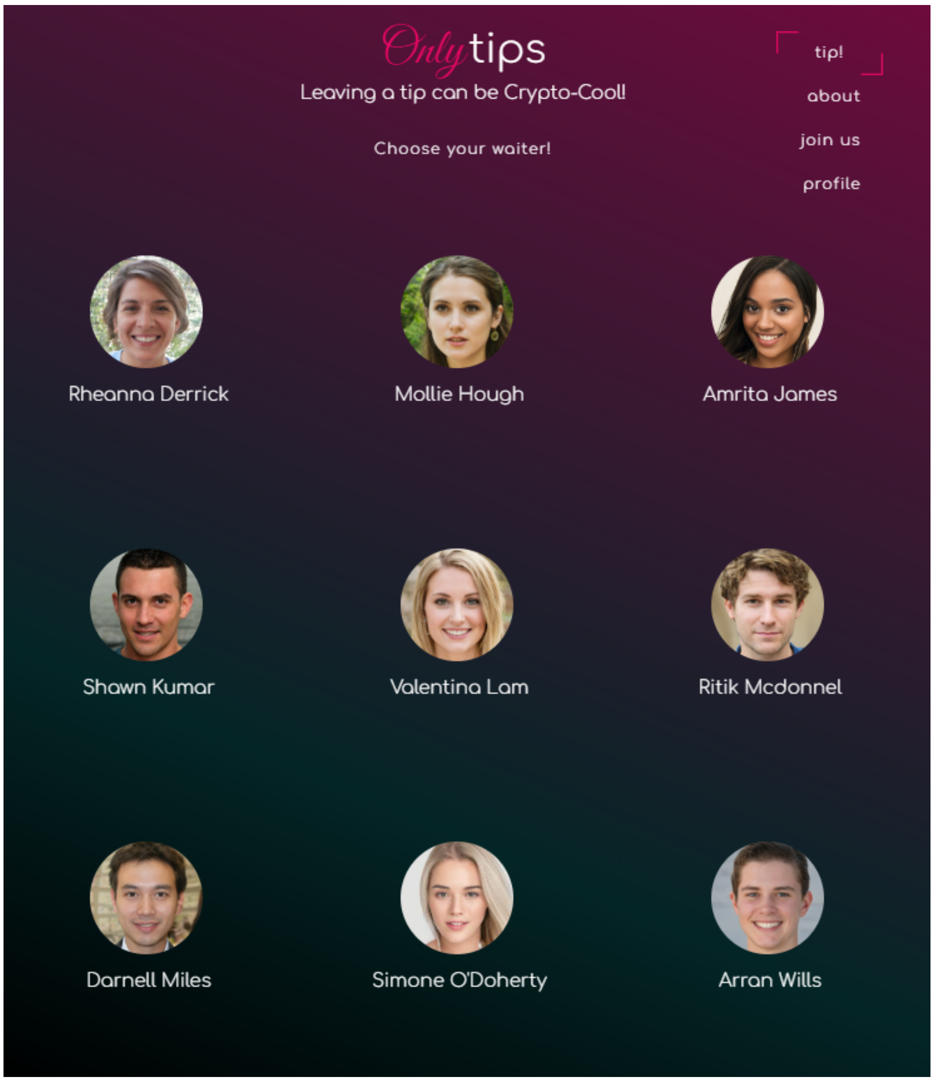
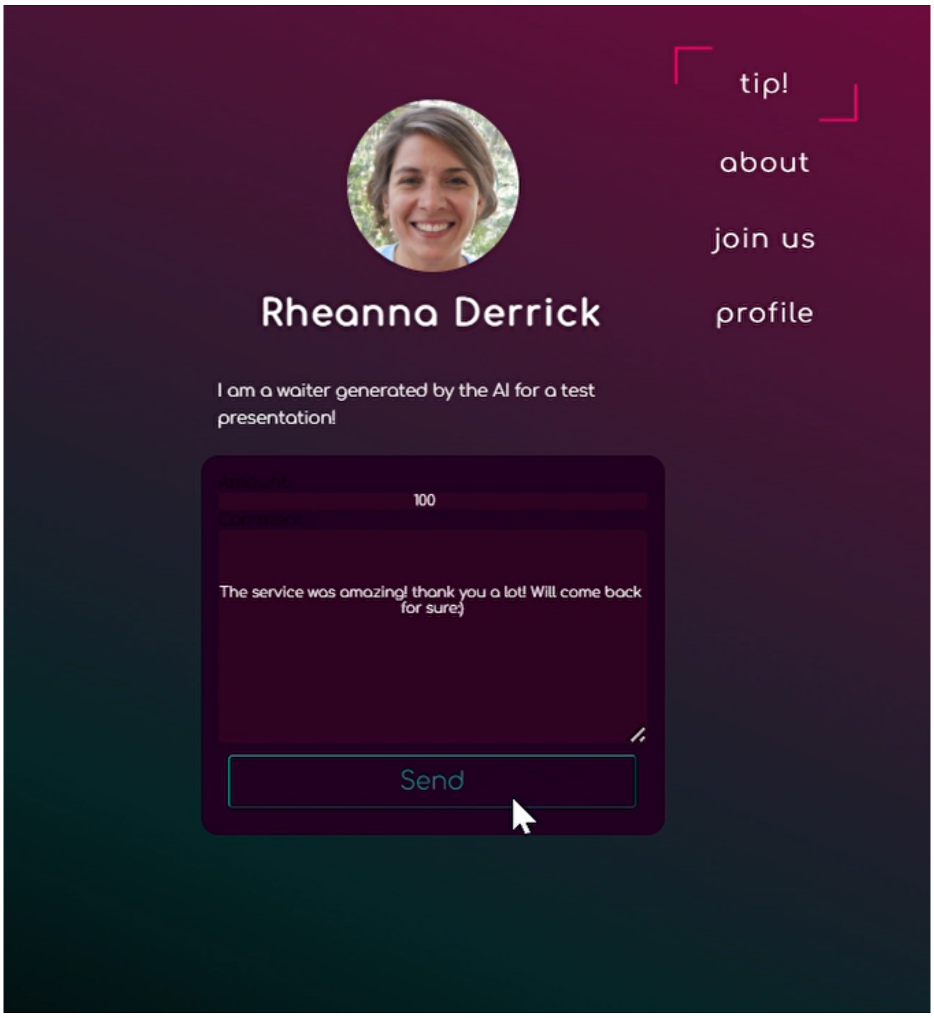


Figure 16: List of waiters page

Waiter page structure

Finally, a needed waiter was found and a customer clicked on their name. After that, a waiter's personal page is visible. The structure is extremely simple and includes only the essential buttons and data. A customer can look through the waiter's personal information that was preferred to be posted: their description (for instance, a waiter can save up for something, and so this can be mentioned here), their picture and a name. Below that, a form for tip sending is placed. It consists of the two graphs: the amount of tip that a customer would like to send and a field for some additional comments (if any). The principle of operation of this form will be discussed in great detail in the next section of the report.



Rheanna Derrick

I am a waiter generated by the AI for a test presentation!

Amount:
100

Comment:
The service was amazing! thank you a lot! Will come back for sure:)

Figure 17: Waiter profile page

As soon as the form is submitted, a new transaction instance is added to a Transaction Database. A function that does all the work is called `waiter_details()` and is defined in **views.py** file. It consists of several parts:

1. **Form filling**

   Let us quickly discuss what information is needed for a transaction to be completed. A sender and a receiver wallet number, a unique id of a transaction, the amount a sender would like to share and any additional comments. We assume that some of the listed fields should be filled in not by the customer, but by the machine itself. When a user enters a waiter's page, a function can already receive all the needed information including the two crypto wallet numbers (since they are unique and can be considered as the id number for each customer and waiter). The only fields that should be indicated is the amount of tokens and the commentary, so, this is left for the user. The id of the transaction is done by setting a default value of `uuid.uuid4` for a variable `tID` (transaction ID). This generates a random sequence of the length 36. They can be considered as unique identifiers, as the probability of receiving the same sequence twice is extremely small.

2. **Blockchain transaction**
   Please see the section **4.4 "Token development"** for detailed information.

3. **Form saving**
   If the transaction was successful, its instance should be saved in the database to be visible in the users' history of transactions. More detailed information can be found in section **4.5 "Web app and blockchain connection"**.

4.3 Token development

As it was mentioned above, the OTT token is implemented on ERC-20 [6] standards. For the implementation processes Remix IDE was used, which provides a comfortable software to write, compile and deploy smart-contracts for tokens. Also, the OpenZeppelin Wizard, which provides templates and patterns for ERC-20 tokens, simplifies the process of implementation.

Token deploy starts from `constructor()` and `mint()` functions:

```
contract OnlyTipsToken is ERC20, ERC20Burnable, ERC20Snapshot, Ownable, Pausable,
    ERC20Permit{
    constructor() ERC20("OnlyTipsToken", "OTT") ERC20Permit("OnlyTipsToken") {
        _mint(msg.sender, 10000000 * 10 ** decimals());
    }
```

In this part the name - OnlyTipsToken, symbol - OTT and the total supply, which is 1.000.000 with 10 decimals values, are set.

Next, there are 3 more sufficient functions, which should be implemented inside of smart-contract, which makes a token that will meet the requirements necessary for the purposes of the project.

First function is `mint()`, which allows owner to create additional supply of tokens:

```
function mint(address to, uint256 amount) public onlyOwner {
    _mint(to, amount);
}
```

It is necessary for the project, since when employees exchange their tokens to fiat, then the tokens should be burned and at the same time the amount, which was burned should be minted, to refill the pool of tokens used for tips purposes.

Secondly, is a ERCBurnable parameter, which is already implemented during the `construct()` implementation:

```
contract OnlyTipsToken is ERC20, ERC20Burnable, ERC20Snapshot, Ownable, Pausable,
    ERC20Permit{
    constructor() ERC20("OnlyTipsToken", "OTT") ERC20Permit("OnlyTipsToken") {
        _mint(msg.sender, 10000000 * 10 ** decimals());
    }
```

As it was mentioned, it is a vital feature for a token, because in other case the pool of tokens which was selected for community purposes can be exhausted.

Finally, the `pause()` and `unpause()` functions:

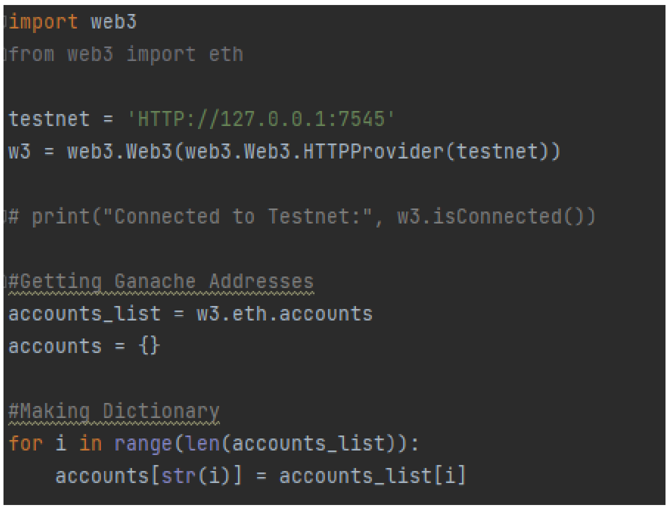
```
function pause() public onlyOwner {
    _pause();
}

function unpause() public onlyOwner {
    _unpause();
}
```

It is an important function, since it allows the token owner to pause all operations with a token, in case of some unstable situations. This tool helps to overcome some problems with hacker attacks, avoid problems with transactions when the technical problems in the mainnet appears and create a strong security level.

Now, let us talk about implementation of the key functions, which allow users to provide transactions and check their balance. The Ganache app was used for test purposes of the transaction system and the whole process with a token. It allows to create 100 addresses on the local host and provide 100 test ETH on each address to operate with. It is rather simpler and convenient than using testnets, since before the token will go to the mainnet, it is sufficient to be sure that everything works well and all the functions work.

The first and the most important function is obviously transaction. It is realized via python and web3 library, which actively used to operate with Ethereum smart-contracts.

1. Import web3 library and get Ganache addresses



```
import web3
from web3 import eth

testnet = 'HTTP://127.0.0.1:7545'
w3 = web3.Web3(web3.Web3.HTTPProvider(testnet))

# print("Connected to Testnet:", w3.isConnected())

#Getting Ganache Addresses
accounts_list = w3.eth.accounts
accounts = {}

#Making Dictionary
for i in range(len(accounts_list)):
    accounts[str(i)] = accounts_list[i]
```

Figure 18: Web3 import

   At this step, all addresses and private keys from localhost Ganache are obtained, to operate with them in the next steps.

2. ABI

   ABI is an Application Binary Interface, or in other words it represents how the functions are called and the data received.

   Since, smart-contract is represented in blockchain as a bytecode, with several functions inside. The role of ABI is providing the encode for the solidity smart-contract function which should be invoked and vice versa, how to read the data which is returned form the function.

   The ABI code was taken from the Remix IDE after deploying the contract, which means that this ABI represents all functions that was mentioned in the token smart-contract.

3. Transaction function

   Here is the code of the function, which transfers some amount of tokens from one address to another. This function takes three input parameters: `sender_address` - which is the address from which tokens are going to be transferred, `reciever_address` - address which will receive tokens, and the `transferAmount` - the number of tokens which is going to be transferred.

```
def transac(sender_address, receiver_address, transferAmount):
    return token.function.transfer(receiver\_address,
    int(transferAmount)).transact({'from': sender\_address})
```

Calling the function it returns the web3 function, which call for transaction to the blockchain with input data.
Second function is balance check, which allows the user or employee to check how much tokens are stored on their addresses.
The way how it is implemented has the same steps 1) and 2) as the transaction function, however the 3) step differs:

```
def balance(address):
    return print(token.functions.balanceOf(address).call()/10**10, 'Tokens')
```

Here the only one input feature is obligatory - the address itself. Then, the function returns the value of tokens via the web3 library function which sends the request.
All this function works well with the connecting MetaMask and going to the mainnet, so at this step it can be said that the tests are done, and the token satisfies necessary conditions for the business.

4.4 Web app and blockchain connection

As soon as we have discussed the functionality of our web application and smart-contracts, we can move to the last iteration of the technological implementation of the project. Obviously, there has to be some backend functions that execute the blockchain transaction by clicking a button on a website.

4.4.1 Preliminary step

We created a function `waiter_details()` that calls other functions responsible for two actions: token transfer to and from addresses specified in the web app and adding information on the transaction to the database. It is important to mention that it was decided by the team that operations with database are executed primarily.

As it was already mentioned, firstly the information about the upcoming transaction is generated. As the website allows to send tips only being logged in, the application obtains sender address in advance as soon as one authorises. We have also considered the case when a waiter visits a restaurant and desires to tip somebody else. Therefore, an exception for range of receivers addresses is made and it is impossible to send tokens to the same address as the sender has. When a sender chooses the desired waiter, his public key or wallet address is automatically being found in the users database and saved in the transactions database. Additionally, date and amount of tips that was set in the tipping form are added in the database. Another extremely important check is concluded in ensuring that the tipping amount is entered correctly, thus, the tip does not exceed the sender‘s balance. The form supposes the amount is real number and does not support letters or special signs. If all conditions for transaction are satisfied, mentioned before information is added to the database, and in profiles of the sender and the receiver the transaction history table is being replenished with a new transaction. As soon as these operations complete, the blockchain technology starts working.

4.4.2 Transaction step

We have already talked about a function for transferring tokens in **section 2.3.1** and mentioned that it takes receiver‘s address and amount as input values. However, this function considers that sender‘s address is known, and in our situation we have to pass the sender‘s address to the function. That is done with the function:

```
transac(instance.sender, instance.receiver, instance.amount)
```

where `instance.sender` is the tipper‘s address, `instance.receiver` is the tippee‘s address and `instance.amount` is the tip size itself. Sender‘s address is taken from the logged in account data, the tippee‘s address is obtained basing on the waiter the tipper chose, and the amount is filled in the form. When necessary information for function variables is available, `transac()` begins operating and transferring tokens from one account to another. If everything is successfully executed, the customer will be automatically redirected to the cafes page.

But there might be several errors while tipping. Let us discuss them more precisely:

1. Sender‘s balance is less than the tip amount
   In this case the page will be reloaded and therefore, the tip will not be sent. Obviously, transaction database will not consider this action as completed.

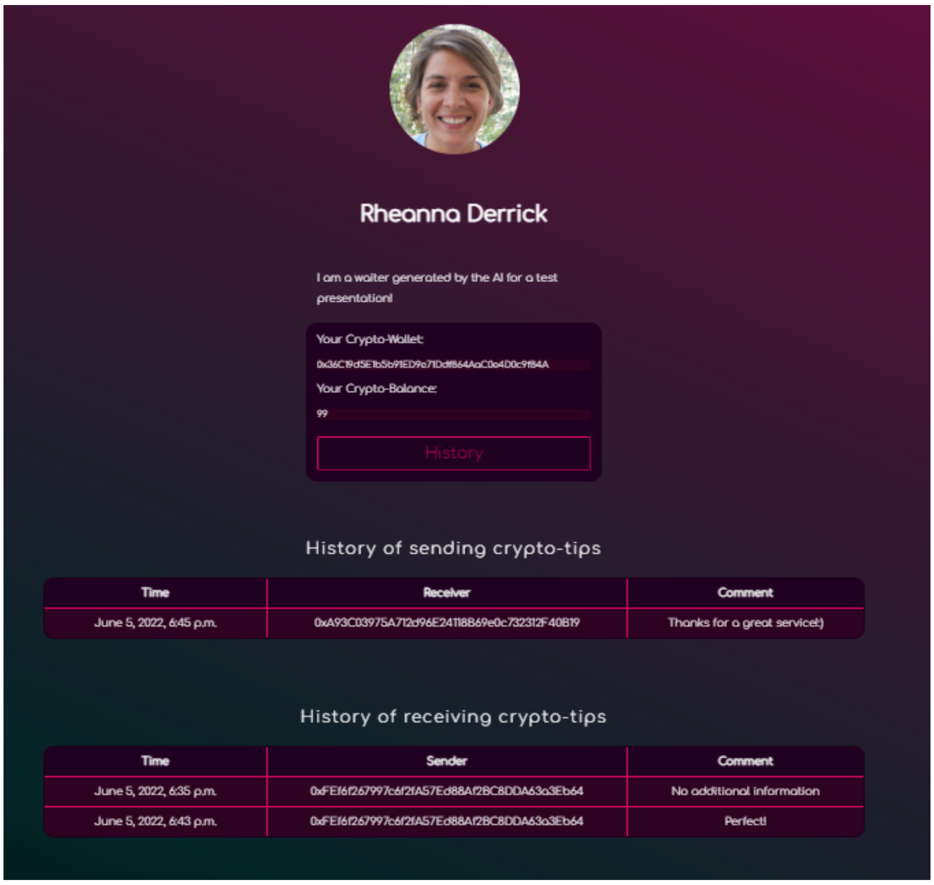
2. Chosen tip amount is incorrect

   Excepting the small balance, this situation may happen if a tipper enters amount he/she desires to send in incorrect format, for example, by typing special signs or letters. The page will be reloaded and transaction will be denied. Transaction database will also remain unchanged.

3. Incorrect receiver‘s address

   This case is unlikely to happen because it is impossible to change manually the tippee‘s address, and customer shall not choose the unavailable waiter. However, if something indeed goes wrong, the user will be redirected to the page with "Waiter does not exist!" text. Transaction, apparently, will not be completed and the transaction database will remain unchanged.

4.4.3 Profile data



Rheanna Derrick

I am a waiter generated by the AI for a test presentation!

Your Crypto-Wallet:
0x36C19d5E1b5b91ED9e71Ddf864AaC0a4D0c9f84A
Your Crypto-Balance:
99

History of sending crypto-tips

Time	Receiver	Comment
June 5, 2022, 6:45 p.m.	0xA93C03975A712d96E24118B69e0c732312F40B19	Thanks for a great service!:)

History of receiving crypto-tips

Time	Sender	Comment
June 5, 2022, 6:35 p.m.	0xFE16f267997c6f21A57Ed88Af2BC8DDA63a3Eb64	No additional information
June 5, 2022, 6:43 p.m.	0xFE16f267997c6f21A57Ed88Af2BC8DDA63a3Eb64	Perfect!

Figure 19: Profile page with both sided transactions history

Apart from simple transaction there are also several functions that can be called by the website. First of all, in the user profile one may have a look at his/her balance, the field for which is located right under the personal crypto wallet address. This is done by making a variable `balance_wallet` in the `profile()` function, that was also described before. The wallet balance is obtained by executing the code:

```
    balance_wallet = str(balance(request.user.customerprofile.wallet))
```

`balance` function is similar to `balanceOf` function. `transaction` and `transactions_received` are responsible for the list of executed transactions for senders and receivers respectively. Without doubt, a case when a waiter tips somebody else is also considered and the transactions history represents outgoing operations first and consuming ones secondly as it can be seen in the figure below.

Presence of both types of transactions is defined by comparing the number of operations for each kind with zero. If a user did both send and receive tips then the parameter `len(transactions)` and `len(transactions_received)` will be nonzero and thus, both tables containing transactions history are going to be shown.

4.5 Conclusion

All in all, sending crypto can be considered as one of the most transparent ways to tip the employee. Every user can make sure that if the transaction has been completed, the worker will receive the full amount (excluding commission). The idea of traceable money is ideally applicable to a services sector: each transaction stays forever in a blockchain and in theory can be viewed by any other user. At any time the client can check particular details of a committed transaction and see, whether the recipient got all the tokens sent.

Our team have provided a web-based tipping service, which allows clients to appreciate waiters, couriers, drivers and all people which work in a service industry. For B2B part the service provides an innovative way to gain tips, motivate employees who are working hard and attract new customers that are enthusiastic about crypto and innovations.

The instruction for users is simple:

1. The registration should be completed, linking the MetaMask wallet to provide transactions.
2. OTT tokens are then bought in web-app via MetaMask.
3. A particular business, e.g. cafe, restaurant, delivery service, etc. should be chosen.
4. The tippee is selected.
5. The amount of OTT tokens that should be sent to a tippee is typed and a comment is left.

6. As soon as the transaction is committed, the user can check it out in the history of transactions.

A huge work has been done in a business and competitors analysis part, as well as DeFi analysis, integration in tokens mechanics, study of tokenomics and finance part of the business. We have created the marketing and investment campaign, prepared style and designing. Then, a web-app was implemented together with both front-end and back-end development. Finally, all separate parts were connected and integrated into each other.

5 References

[1] Чудеса хэширования [Electronic Source] / Kaspersky daily.
Available at: www.kaspersky.ru

[2] Блокчейн Blockchain [Electronic Source] / TAdviser.
Available at: www.tadviser.ru

[3] Что такое Web3? Децентрализованный интернет будущего [Electronic Source] / Habr.
Available at: habr.com

[4] Что такое Utility tokens? Как они работают? [Electronic Source] / Bytwork.com
Available at: https://bytwork.com

[5] Криптокошелек: как выбрать и какие бывают [Electronic Source] / VC.RU.
Available at: vc.ru

[6] Токены стандарта ERC-20 на смарт-контрактах Ethereum [Electronic Source] / Майнинг криптовалюты.
Available at: https://mining-cryptocurrency.ru

[7] Meet Django [Electronic Source] / Django.
Available at: https://www.djangoproject.com

[8] Advantages and disadvantages of CSS [Electronic Source] / Tech Quintal.
Available at: https://www.techquintal.com

[9] В чем разница между Utility tokens и Security tokens? [Electronic Source] / International Wealth.
Available at: https://internationalwealth.info

[10] Moonriver (MOVR): криптовалюта — обзор, отзывы / INP.ONE.
Available at: https://inp.one

[11] Accept Crypto Payments, Grow Your Business / tripleA.
Available at: https://triple-a.io

[12] Tippr [Electronic Source] / GitHub.
Available at: https://github.com

[13] Tippr Bot Distributes Over $100K in Bitcoin Cash Across Reddit Forums [Electronic Source] / Bitcoin.com.
Available at: https://news.bitcoin.com

[14] On-chain tipping [Electronic Source] / ChainTip.
Available at: https://www.chaintip.org

[15] GitCash [Electronic Source] / GitHub.
Available at: https://github.com

[16] Websites for you and your projects / GitHub Pages.
Available at: https://pages.github.com

[17] Introducing GitCash.io the Bitcoin Cash powered Github tipping app [Electronic Source] / Bitcoin.com.
Available at: https://news.bitcoin.com

[18] Tipping platforms [Electronic Source] / Bitcoin.com.
Available at: https://www.bitcoin.com

[19] How to send Bitcoin Cash via text messages to anyone with a mobile phone [Electronic Source] / Bitcoin.com.
Available at: https://news.bitcoin.com

[20] СберЧаевые – безналичные чаевые вашим сотрудникам [Electronic Source] / СБЕР бизнес.
Available at: https://www.sberbank.ru

[21] Самый удобный способ оставить чаевые [Electronic Source] / Tips+.
Available at: https://tipsplus.ru

[22] Password management in Django [Electronic Source] / Django.
Available at: https://docs.djangoproject.com

6 Appendix

List of Figures

1 Token distribution diagram 21

2 Main colors scheme 32

3 Accent colors scheme 32

4 Comforta font sample 33

5 Great Vibes font sample 33

6 Login form 35

7 Registration form 35

8 Login form 36

9 Registration form 36

10 "tip!" button animation 37

11 "join us" button animation 38

12 Web application draft 38

13 Profile page 39

14 Database structure 41

15 Cafes page 45

16 List of waiters page 46

17 Waiter profile page 47

18 Web3 import 50

19 Profile page with both sided transactions history 53