SAPIENZA
UNIVERSITÀ DI ROMA

BulkyRewards

A decentralized application on the Algorand blockchain

Department of Computer Science
Blockchain and Distributed Ledger Technologies

Marco Raffaele
ID number 1799912

Advisor
Prof. Claudio Di Ciccio

Academic Year 2022/2023

Contents

Chapter 1

Introduction

1.1 Presentation of the DApp

BulkyRewards is a system based on the circular economy, aimed at solving the problem of bulky waste on city streets by granting virtual coins as a reward for the proper disposal of this type of waste. Users will be able to spend these tokens on the purchase of various public services.

1.2 Main responsabilities

As an individual project, I took care of everything related to the application development, including:

- Smart contract programming using TEAL/pyTeal, and its deployment using javascript.
- Implementation of server-side, using NodeJs
- Front-end realisation using React, HTML and CSS.

1.3 Outline of the report

- **Background**: Brief introduction, history and concepts regarding blockchain technology, and application domain.
- **Presentation of the context** DApp purpose, project description and motivation for using blockchain.
- **Software architecture**: Description of the software architecture and its general overview, and illustration via UML diagrams, of the system components and their interaction, structure and content of smart contracts, use cases, token life cycle.
- **Implementation**: Illustration of the system GUI, and description of the implementation of smart contracts.
- **Conclusion**: Description of conclusions concerning the development from the application and implementation point of view, known issues and limitations and some possible future remarks.

Chapter 2

Background

2.1 Introduction

Distributed ledger technologies (DLT), such as the blockchain, are databases that allow information to be shared and duplicated on a network of computers called nodes. All nodes have access to the ledger and the information it contains is always verified. The blockchain is a specific type of DLT that uses cryptography and mathematical algorithms to create and manage a data structure in the form of a chain of blocks. In this data structure, the entered information is immovable, distributed, and public.

2.2 History

The concept of digital payment protocols has been around since 1980, with David Chaum proposing the first model. However, the first practical implementation of an electronic cash system, known as e-cash, occurred in 2009 with the creation of Bitcoin. This event also marked the emergence of cryptocurrency, a digital currency that exists only online and is recorded on a blockchain. Bitcoin's unique feature is the use of a proof-of-work mechanism to secure a chain of blocks, or transactions, through cryptography. Since its inception, Bitcoin has garnered significant attention and adoption, with it being used as a store of value, a means of exchange, and a means of investment. It has also inspired the creation of numerous other cryptocurrencies and blockchain-based applications.

2.3 Types of Blockchain

Blockchain is a technology that can take several forms, such as:

- **Public blockchains**: A public blockchain is a decentralized, distributed ledger technology that allows anyone to participate in the consensus process, validate transactions, and contribute to the network. Examples include the Bitcoin and Ethereum networks.
- **Private blockchains**: A private blockchain is a decentralized, distributed ledger technology that is restricted to a specific group of individuals or organizations. Unlike a public blockchain, a private blockchain requires permission to join and participate in the consensus process. This means that access to

the network is controlled by a central authority or group of authorities, who have the ability to grant or revoke access to the network.

- **Consortium blockchains**: These are networks that are partially decentralized, with only certain pre-selected nodes being able to validate transactions. Consortium blockchains are often used in industries where multiple parties need to come to consensus on transactions, such as in the financial sector.
- **Hybrid blockchains**: These are networks that combine features of both public and private blockchains, and can be either permissioned or permissionless.

2.4 Blockchain Concepts

Here are some common concepts in the world of blockchain:

- **Distributed ledger**: A distributed ledger is a database that is shared and synchronized across a network of computers. In a blockchain, the distributed ledger is used to record transactions in a secure and transparent way.
- **Cryptography**: Cryptography is the practice of secure communication, in which messages are encoded in such a way that they can only be read by the intended recipients. In blockchain, cryptography is used to secure the transactions and ensure that they cannot be altered or tampered with.
- **Transaction**: A transaction in a blockchain is a transfer of value or data from one party to another. In the context of cryptocurrency, a transaction refers to the transfer of cryptocurrency from one address to another. Transactions are recorded on the distributed ledger and are validated and added to the blockchain through a process called “mining“.
- **Block**: In a blockchain, a block is a record of one or more transactions that have been verified and added to the distributed ledger. Each block contains a unique code, called a “hash“, that links it to the previous block in the chain.
- **Consensus mechanism**: A consensus mechanism is a process by which the participants in a blockchain network reach agreement on the state of the distributed ledger. Different blockchain platforms use different consensus mechanisms, such as proof-of-work (used by Bitcoin), proof-of-stake (used by Ethereum) or pure-proof-of-stake (used by Algorand).
- **Smart contract**: A smart contract is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein are stored on the blockchain.
- **Blockchain Trilemma** The creation of blockchain platforms is based on three main aspects, decentralisation, scalability, and security. It is on these concepts that the so-called "blockchain trilemma" was born. The blockchain trilemma states that if you want to create a DLT, you cannot support the presence of all three of these concepts at the same time, but only two of them. From this statement three different cases emerge:
  - Without **decentralisation**: A system without decentralisation is exclusive and secretive, very similar to the systems prevalent to date based on central authorities central authorities.

- Without **security**: A system without security does not guarantee transparency on the transactions that are generated, which implies that we could be deceived by receiving counterfeit transactions.
- Without **scalability**: A non-scalable system cannot cope with the increase in users and transactions, which inevitably leads to a slowdown in the system and possible congestion. of the system and possible network congestion.

2.5 Application domain

The applications domain of blockchain are various and include areas such as finance, supply chain management, voting systems, real estate, and more. Blockchain technology has the potential to revolutionize many different industries by providing a secure and transparent platform for recording and verifying transactions. In the financial industry, for example, blockchain could be used to facilitate secure and efficient cross-border payments, while in the supply chain industry, it could be used to track and verify the provenance of goods. The potential applications of blockchain are vast and are still being explored and developed.
Bulky Rewards is a system based on the concept of circular economy. This economic model sets as its main objective the reduction of the use of new resources, favouring a policy of recycling and reuse. The proposed application aims to create a system based on rewards for the proper recycling of bulky materials. The rewards are expressed as digital tokens and are reusable for the purchase of public services, which promotes a lower environmental impact within cities and an increase in the use of urban services.
Before concluding this chapter, it is important to say that Bulky Rewards was implemented on the Algorand blockchain. Algorand is a decentralized, open-source blockchain platform designed to be fast, secure, and scalable. It uses a unique consensus algorithm called Pure Proof of Stake (PPoS) that allows for high transaction throughput and low transaction fees. Algorand is designed to support a wide range of use cases, including asset transfer, decentralized finance (DeFi), and decentralized applications (DApps).

Chapter 3

Presentation of the context

3.1 Aim of the DApp

The chosen scenario falls within the circular economy, a sustainable economic model that aims to minimize waste and maximize the use of resources. It is based on the principles of reducing, reusing, and recycling, with the goal of minimizing the waste and the extraction of new raw materials.
In urban areas, there is often a problem with bulky waste on the streets, such as furniture, appliances, and electronics. These items cannot be collected by traditional waste management systems and can create logistical challenges for pedestrians and vehicles. Bulky Rewards aims to educate users about the proper disposal of bulky waste at collection points in urban areas, in order to keep cities clean and efficient and promote the principles of the circular economy.
To encourage people to dispose of bulky waste correctly, I have implemented a rewards system using virtual coins called Algorand Standard Assets. These assets can be redeemed for public services, such as subway and bus tickets, or to request home pick-up of bulky items that require specialized handling and disposal methods. To participate in this program, users need to have an Algorand wallet, which is a digital wallet that allows them to store, manage, and transact with Algorand Standard Assets and other digital assets. It is typically accessed through a smartphone app or online platform and requires a unique login and password for security.

3.2 Description of the DApp

Bulky Rewards is a system that uses digital tokens, specifically Algorand Standard Assets (ASA), as rewards or incentives. These ASAs are based on the Algorand blockchain and there are three ASAs being used in the Bulky Rewards system. In addition, the Algo token, which is the native cryptocurrency of the Algorand blockchain, is also being used. The use of digital tokens can provide a flexible and efficient way to incentivize participation and encourage desired behaviors within a system or community.
The asa included in the system are listed and briefly described below:

- **EcoAsset**, or ECO, is the asset that is rewarded to users following the proper disposal of bulky waste.
- **TicketAsset**, or TCKT, is the the asset that can be used to purchase tickets for public transport.

- **TransportAsset**, or TRAN, is the asset that can be obtained through the exchange of ECOs and allows users to request the home collection of bulky waste.

Bulky Rewards involves the use of 3 different smart contracts each relating to the different operations that the system enables:

- **Recycle contract**: Smart contract that manages the distribution of rewards following disposal by users. The contract takes care of calculating the compensation based on the weight and recyclability values provided to it.
- **Exchange contract**: Smart contract dedicated to managing the exchange of assets, this contract aims to verify if the exchange conditions of a specific asset are respected.
- **Escrow contract**: Smart contract for requesting home collection of bulky waste.The purpose of this contract is to hold the payment in TRAN made by the user until one of two possible conditions occurs: a claim request from the operator containing the correct password, or, the expiry of the relevant timeout for collection, at which point the user may request his payment back.

The proposed system is divide into two sections, Admin and User. The Admin is responsible for various tasks including

- **Bulky waste disposal**: The Operator is responsible for collecting bulky waste from users, measuring its weight and level of recyclability, and entering these values into the system. The system will then calculate the value of the reward in a coin called EcoAsset. After this, a scanner will read the QR code associated with the user's address, allowing the smart contract to send the reward to the user. This operation helps to facilitate the disposal and recycling of bulky waste in an efficient and transparent manner, while also rewarding users for their efforts to reduce waste and promote sustainability.
- **Acquiring funds from the escrow contract**: The system allows users to request home collection of bulky waste. This service is managed through an escrow contract, in which the user deposits the payment for the transporter. The Operator is able to access the funds deposited by the user in the contract by entering the contract ID and password provided by the user at the time of collection. This ensures that the transporter is paid for their services, while also providing a secure and transparent process for managing the payment.
- **Managing the system's smart contract**: The Operator is able to perform several actions related to managing the system's smart contracts:
  - Send EcoAssets to the bulky materials recycling contract.
  - Send TransportAssets or TicketAsset to the asset exchange smart contract.
  - Recover funds from the asset exchange application.

The User is able to perform the following operations within the system:

- **Opt-in to receive EcoAsset, TicketAsset, or TransportAsset**: The user will be able to send opt-in transactions for assets in the system directly from their personal section, in order to add the assets to their wallet. Only by doing so will they be able to send and receive Eco, Tckt, and Tran units.

- **Exchange TransportAsset, and TicketAsset**: The user will be able to exchange EcoAssets for TransportAssets, or TicketAssets, with different exchange rates, using a smart contract that will ensure that the process is conducted correctly. This smart contract will verify that all necessary conditions are met before allowing the exchange to take place. This helps to ensure the integrity and security of the exchange process
- **Request collection of bulky waste**: To request the home collection of bulky waste, the user can pay with TransportAssets. Payment is made in advance through an escrow contract, in which the user specifies a password to release the funds and the city zone where the user resides so that the transporter operating in that area can take care of it. The contract will expire after one week.
- **Sender Claim**: If the collection has not occurred by the expiration time, the user can recover its payment through the appropriate section.

3.3 Why using a blockchain

Bulky Rewards uses blockchain technology for several reasons. The system needs to store data on recycling activity, Algo, EcoAsset, and TransportAsset balances of users, and smart contracts. It is also important to record the ownership of each contract in the contract's state to ensure the security and integrity of system operations. The escrow contract is a special case in which the address of the receiver (the transporter) must also be recorded in the contract's state, along with the creator. In addition to these functions, the system is used by various actors including the general user who receives rewards, the collection center operator who handles the recycling processes, and the transporter. These actors all play a role as writers in the system, as they send transactions and create smart contracts, contributing to the overall operation of the system.

Having a reliable third-party (TTP) that is always online can be important for a system that handles asset exchange, as it allows users to access and use the system at any time. This can make the process of exchanging assets and spending them more convenient and efficient.

It is not necessary for a system like Bulky Rewards to be known to all writers in order for anyone to be able to participate and use it.

Permissionless blockchains, such as those used for cryptocurrencies like Bitcoin, can be a suitable platform for implementing a system like Bulky Rewards. These types of blockchains allow anyone to participate and make transactions without the need for permission, which can make them more open and accessible. It is important to consider the trade-offs of using a permissionless blockchain, such as the potential for reduced privacy and increased risk of fraud, as well as the benefits.

Chapter 4

Software architecture

4.1 Overview

In this chapter, the software architecture of Bulky Rewards is described, starting with a general description of the system, followed by a series of UML diagrams describing the content and structure of the smart contracts, the components involved in the system, the functionalities offered, and the lifecycle of the various proposed use cases. The figure 4.1 shows a general view of the software architecture through which the system was implemented, starting from the realisation of the smart contracts, up to the generated webApp.

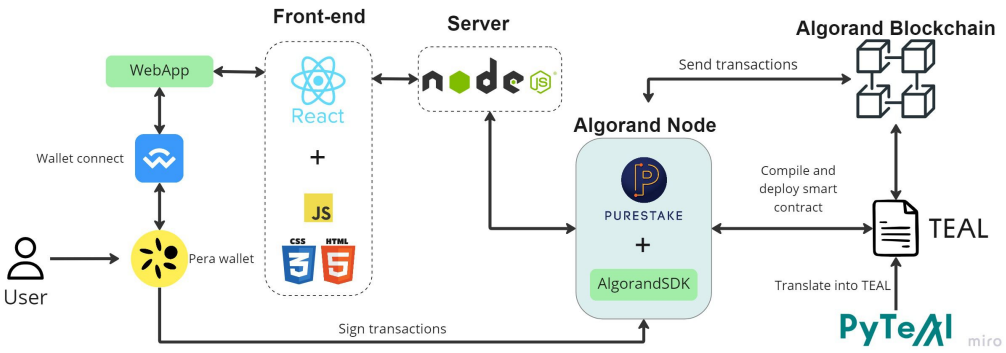


Figure 4.1. Software architecture

- **Front-end**: The front-end was implemented using React, HTML, and CSS. As a JavaScript library for building user interfaces, React is designed to make web development easier and more efficient by allowing developers to create reusable components that can be easily rendered in the browser. In this case, it was used in conjunction with HTML and CSS to create a dynamic and interactive web application. React is known for its efficient updates and rendering of changes to the user interface, making it a great choice for building applications like BulkyRewards.

- **Algorand Node**: The Algorand node is represented by the Purestake service in conjunction with the Algorand Software Development Kit (AlgorandSDK). Purestake is a provider of infrastructure for the Algorand blockchain, and it offers a service that allows users to easily spin up and maintain their own

Algorand nodes(Algod API). In addition Purestake offers a fast and efficient way to index the blockchain data and make it easier to query and retrieve information from the blockchain. This tool is the Indexer and can be used with the Indexer API. The Algorand SDK is a set of tools and libraries that developers can use to build applications on Algorand blockchain. By using Purestake APIs in conjunction with the Algorand SDK, it is possible to interact with the blockchain and build Algorand-based applications.

- **Algorand Blockchain**: Bulky rewards uses the Algorand blockchain, on which all transactions generated by the system and deployed smart contracts are recorded.
  The smart contracts are written in TEAL (Transaction Execution Approval Language) which is the native programming language of the Algorand blockchain. It is a simple, stack-based language that is used to write smart contracts and other types of decentralized applications on the Algorand platform.
  Furthermore, smart contracts are written using a python library called PyTeal which offers the possibility to write smart contracts on the Algorand platform with a more popular, simple and intuitive language such as python.

- **WalletConnect and Pera wallet**: The connection between the user and the webApp was achieved through the use of WalletConnect, an open protocol for connecting decentralized applications to mobile wallets using secure, encrypted QR code scans. The mobile wallet used for this system is PeraWallet, a cryptocurrency wallet that is specifically designed to hold and manage Algorand (ALGO) tokens.

4.2 UML Diagrams

4.2.1 Concept diagram

Fig. 4.2 illustrates the conceptual diagram, which is a visual representation of an idea or concept. This model describes the smart contracts involved in the system and the assets they exploit to perform the operations the system offers.

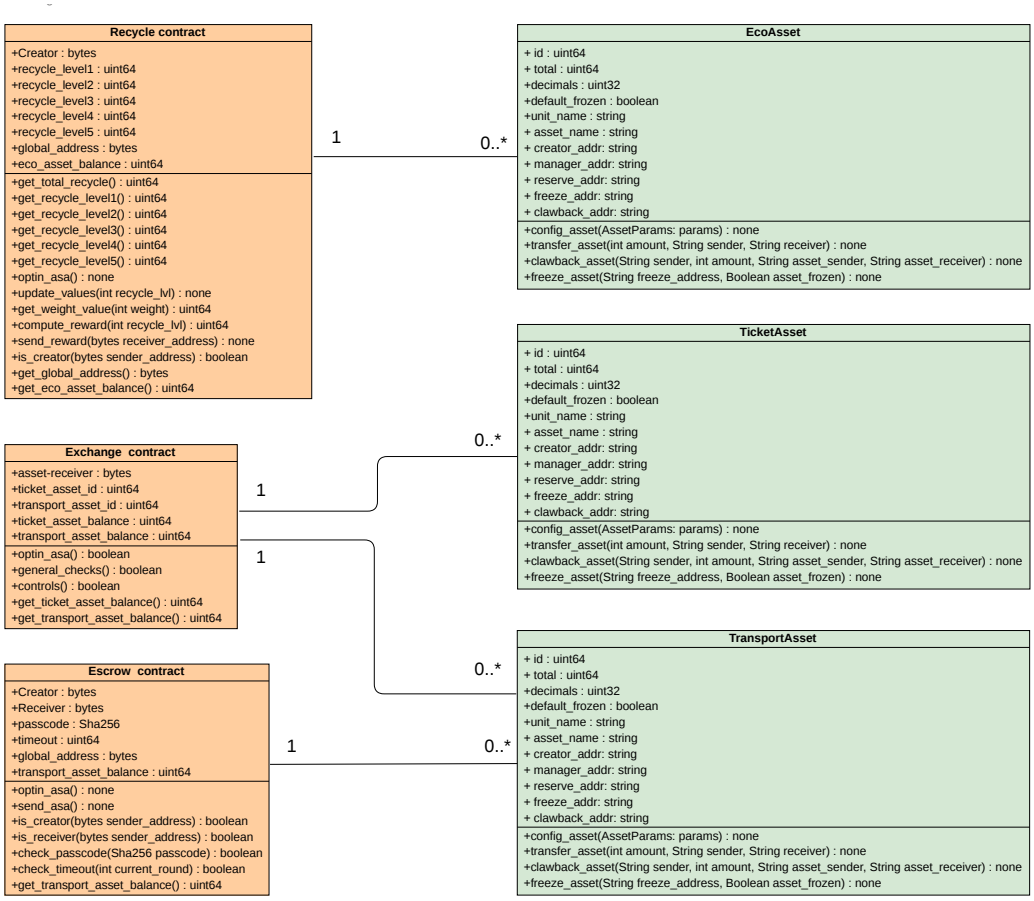


Figure 4.2. Concept diagram

4.2.2 Component diagram

Figure 4.3 shows the component diagram, which is used to display the relationships between the various components of a system. The component diagram illustrates the structure of the Bulky Rewards system, the dependencies between its components, and the interfaces between them.

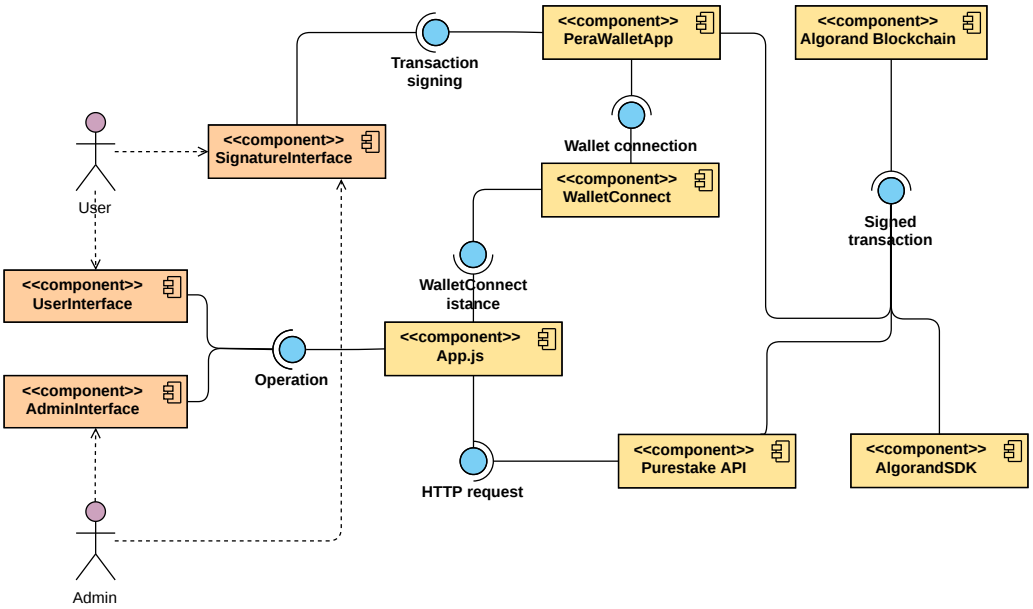


Figure 4.3. Component diagram

4.2.3 Collaboration diagram

The collaboration diagrams are used to show the relationship between the components in a system. The Figures 4.4, 4.5, 4.6, 4.7, and 4.8, illustrate the interactions between Bulky Rewards components and their data stream.

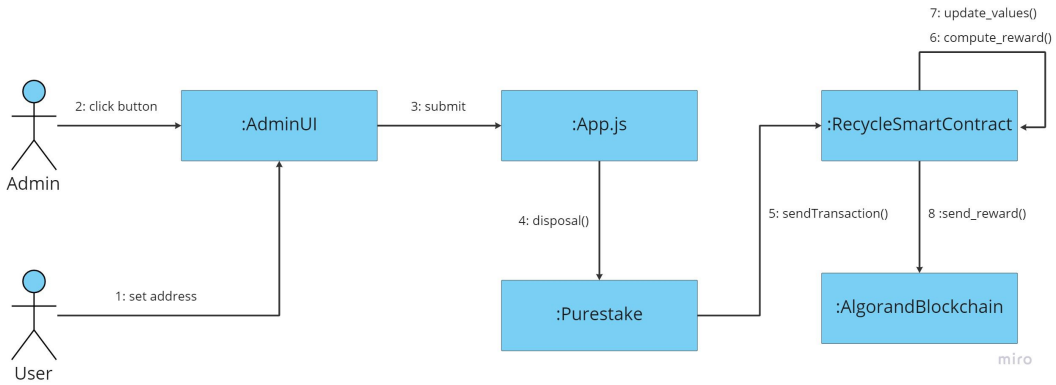


Figure 4.4. Disposal collaboration diagram

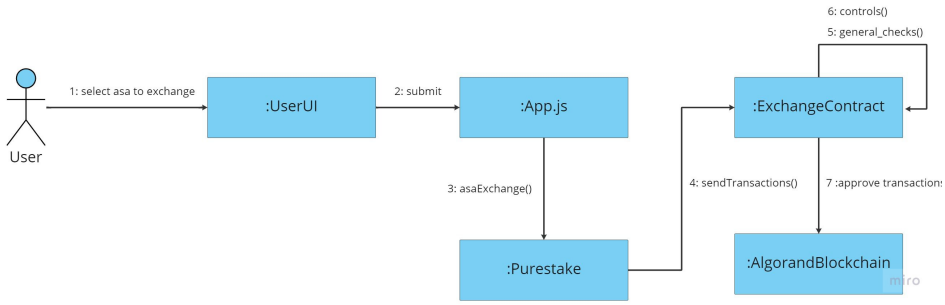


Figure 4.5. Asset exchange collaboration diagram

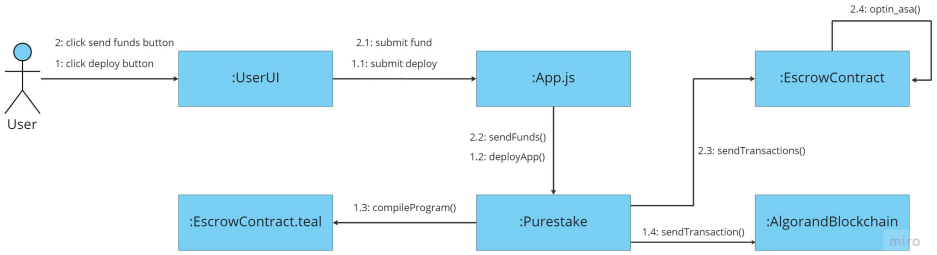


Figure 4.6. Escrow contract creation collaboration diagram

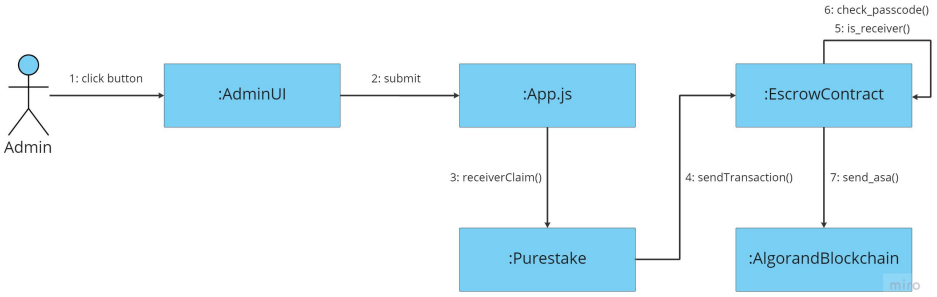


Figure 4.7. Receiver claiming collaboration diagram

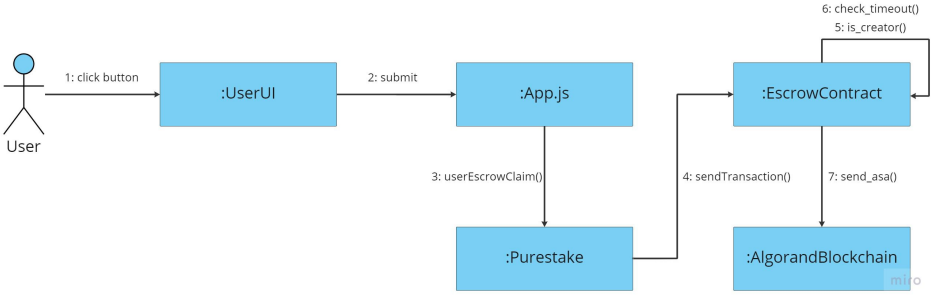


Figure 4.8. User claiming collaboration digram

4.2.4 Use case diagram

The use case diagram, illustrated in fig. 4.9, shows the interactions between actors and the system. In this use case diagram, the possible actions that each actor can take within the appropriate sections are shown, providing an overview of the Bulky rewards system.

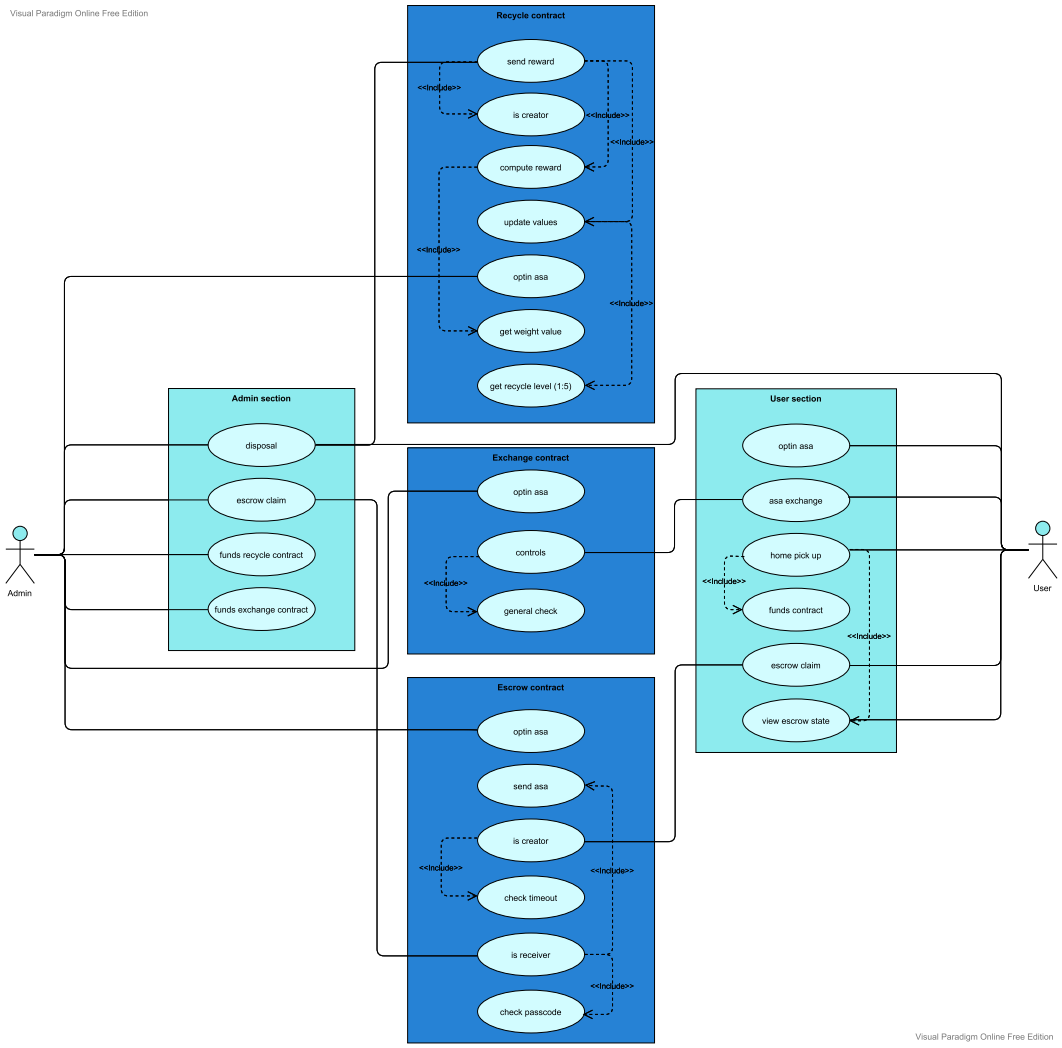


Figure 4.9. Use case diagram

4.2.5 Activity diagrams

activity diagrams are part of the behavioural diagrams, and are intended to show the flow of an activity from a starting point to an end point. Figures 4.10, 4.11, and 4.12, respectively, show the life cycle of the assets involved in the system.

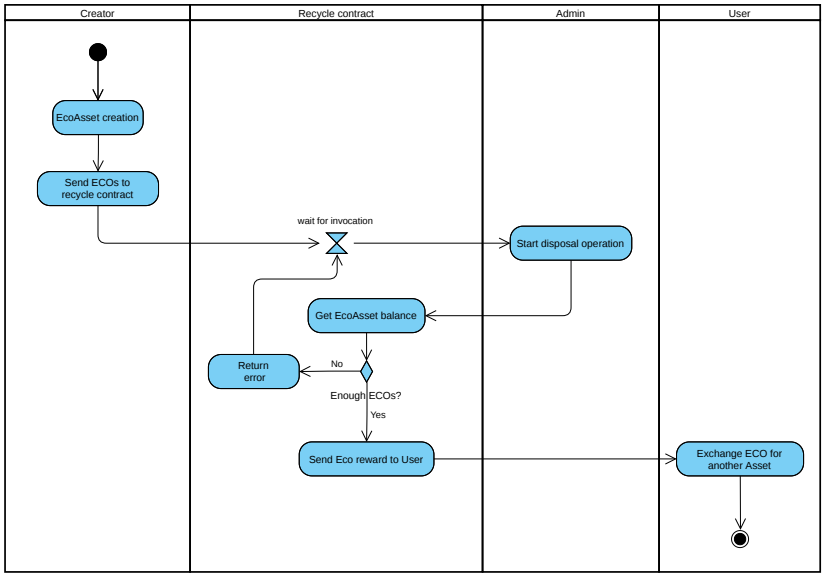


Figure 4.10. EcoAsset activity diagram

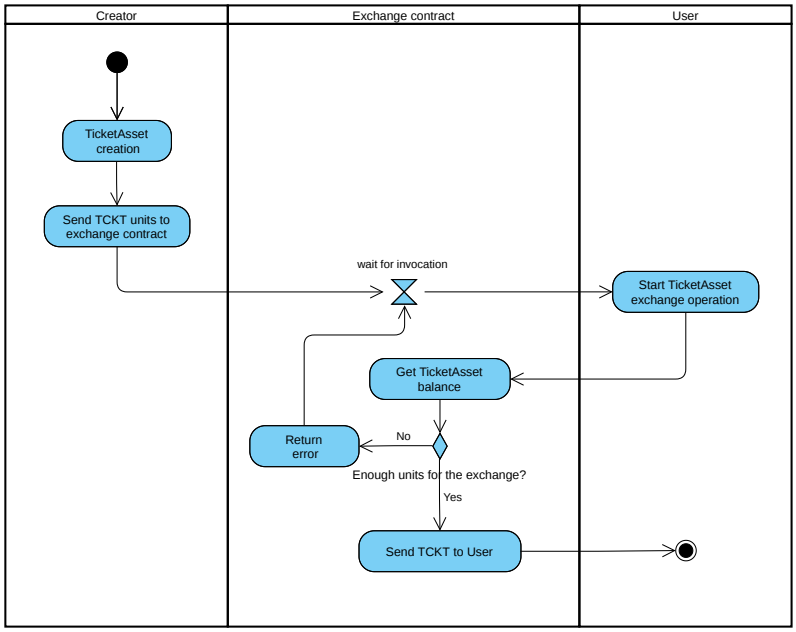


Figure 4.11. TicketAsset activity diagram

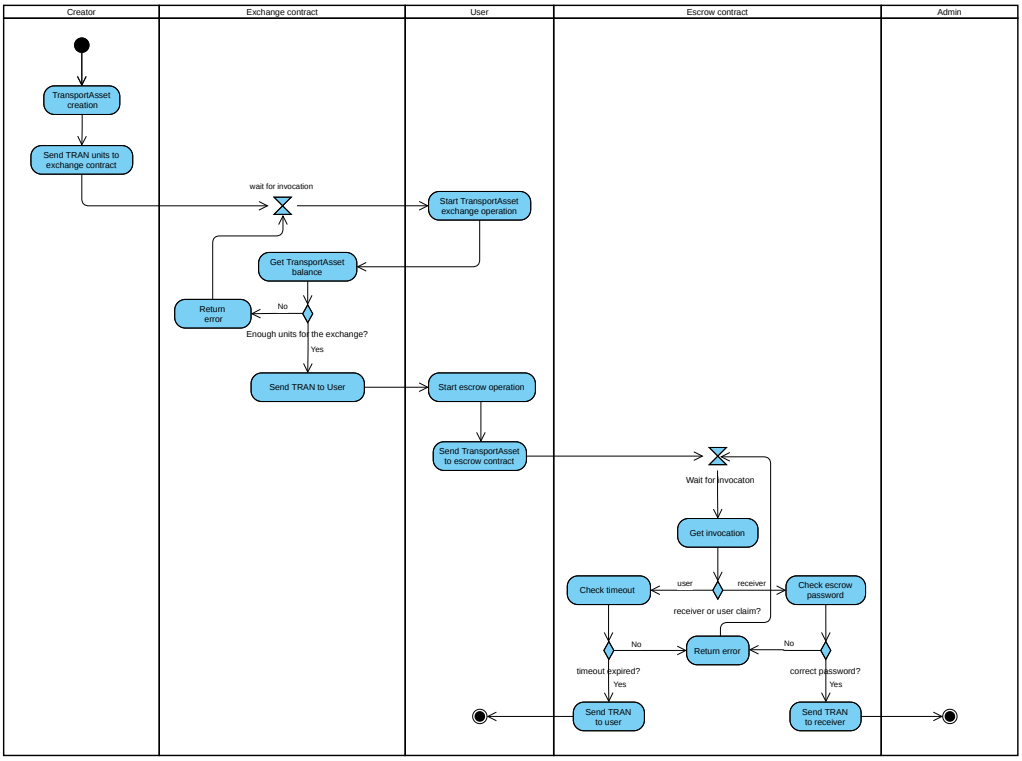


Figure 4.12. TransportAsset activity diagram

Chapter 5

Implementation

5.1 Frontend

This chapter describes the implementation of the DApp. In particular, the user interface realised through the use of React js, Html, and Css is illustrated. The structure is very simple and is based on three different, very intuitive sections, Home, User, and Admin.
It is important to say that only the crucial functionalities within the system are described, avoiding those that do not affect the use of smart contracts.

5.1.1 Home

The home section is the simplest in the application and has no particular functions. Its only purpose is to welcome the user and guide them to the most relevant section of the app for their needs and it is illustraded in Fig. 5.1.

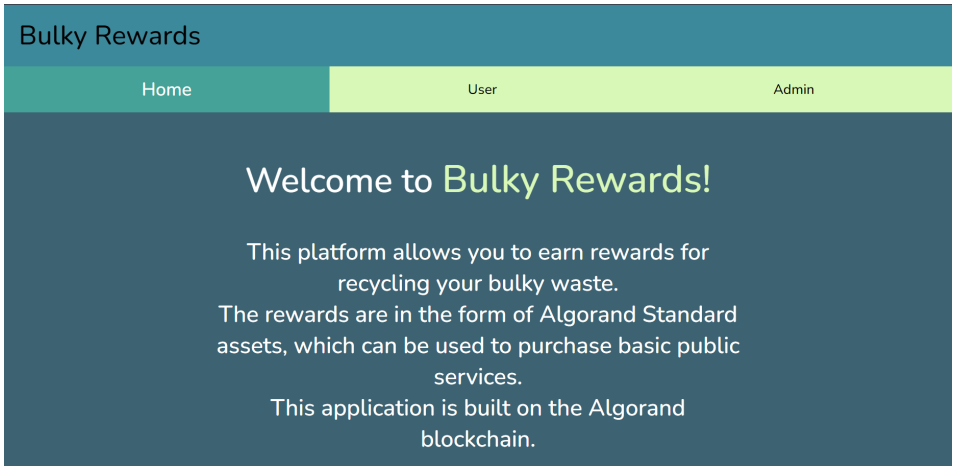


Figure 5.1. Bulky Rewards home page

5.1.2 Admin

Following the welcome shown on the home page, it is possible to reach the admin section via the navigation bar. The first thing the app proposes is what is shown in the Figures 5.2 and 5.3.

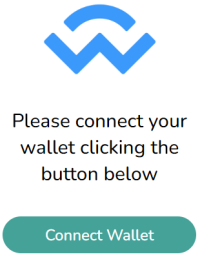


Figure 5.2. Wallet connection box

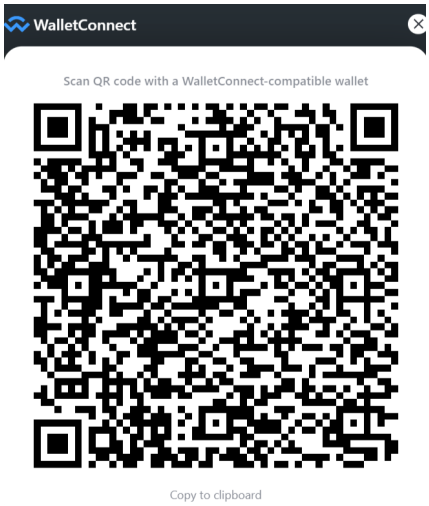


Figure 5.3. WalletConnect QRcode

After connecting their wallet through the wallet interface, the app will display the screen shown in Fig. 5.4. On this screen, the admin can perform all operations that the page provides.
On the right side of the page, there is also an information panel that shows the real-time status of contracts in the system and their balance in terms of assets.

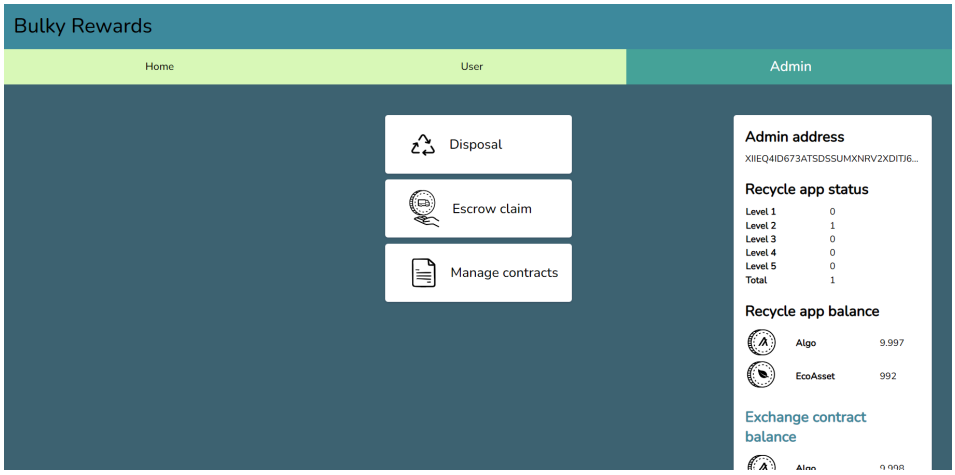


Figure 5.4. Admin section

Bulky waste disposal

Clicking on the "Disposal" button results in what is shown in the fig 5.5.

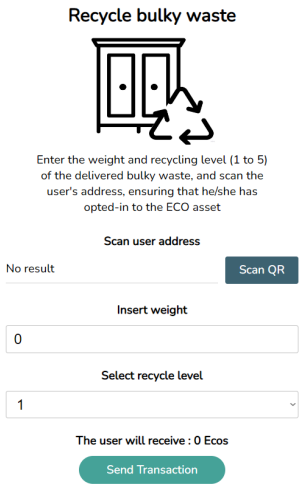


Figure 5.5. Bulky waste disposal

In this section, the admin can:

- Scan the QR code for the user's address.
- Enter the weight of the bulky materials delivered by the user and select the level of recyclability of the materials.
- Send a transaction with the entered data to the smart contract.

In addition, the user's reward, displayed in the bottom part of the panel, changes in real-time.

Escrow claim

Clicking on the "Escrow claim" button results in what is shown in the fig 5.6.

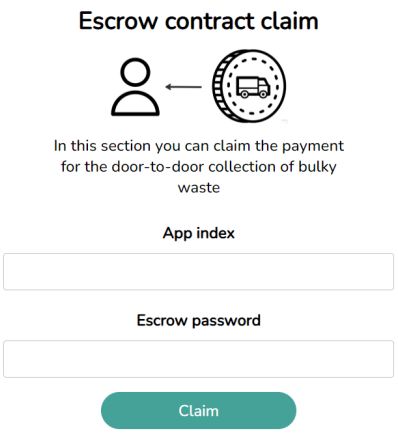


Figure 5.6. Admin escrow claim

In this section, the admin can:

- Enter the escrow contract application id and the escrow contract password provided by the user when the materials were collected.
- Send transaction with the entered data to the smart contract.

5.1.3 User

The first thing shown once the user get to this section is the wallet connection interface, as already shown in Fig. 5.2 and Fig. 5.3. After connecting the wallet through the wallet interface, the app will display the screen shown in Fig. 5.7. On this screen, the user can perform various actions that are available to them. On the right side of the page, there is also an information panel that shows the balance of the user wallet and the indexes of created smart contract.

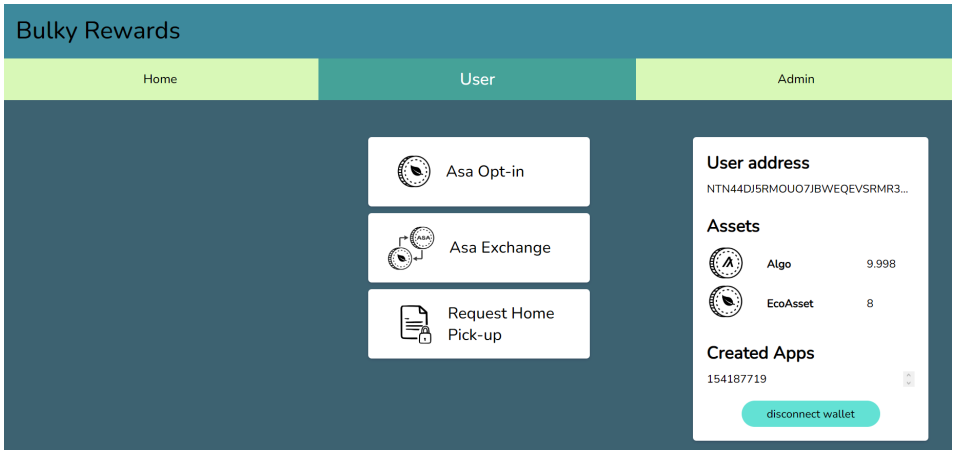


Figure 5.7. User section

Asa exchange

Clicking on the "Asa exchange" button the system display what is shown in the fig 5.8.

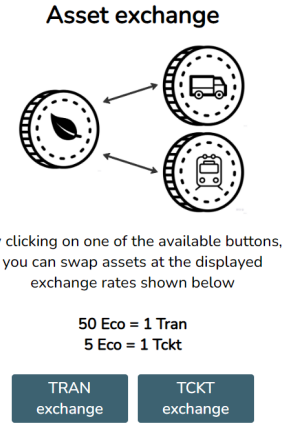


Figure 5.8. Asa exchange

In this section the user can, click one of the two buttons displayed and send the exchange transactions for the assets at the corresponding exchange rate.

Request home pick-up

Clicking on "Request Home Pick-Up" will cause the system to display the panel shown in Fig 5.9 related to creating an escrow contract for home pick-up. The system will then also display the panel shown in Fig 5.10.

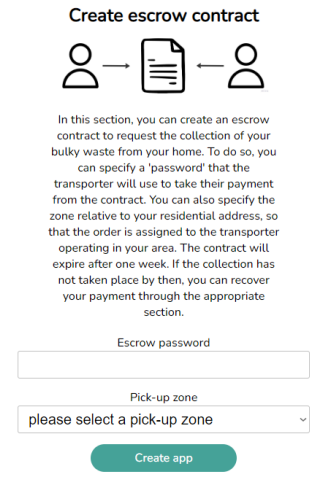


Figure 5.9. Escrow contract creation

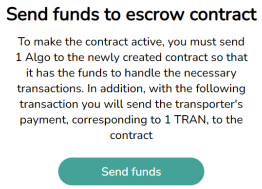


Figure 5.10. Escrow contract funding

In this section the user can:

- Enter a password to lock funds on the escrow contract.

- Select their residential area in order to contact the competent operator.
- Click the "Create app" to send the app creation transaction.
- Click the "Send Funds" button to provide the contract with the asset to be locked and the funds needed to manage the transactions.

User escrow claim

Clicking on the id of a contract shown in the panel illustrated in Fig 5.11, the system will display to the user a summary of the escrow contract and its activity status as shown in Fig 5.12.

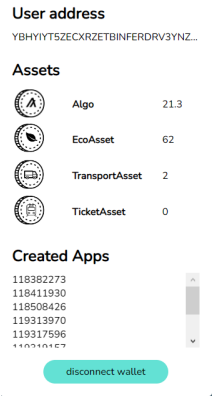


Figure 5.11. User wallet panel

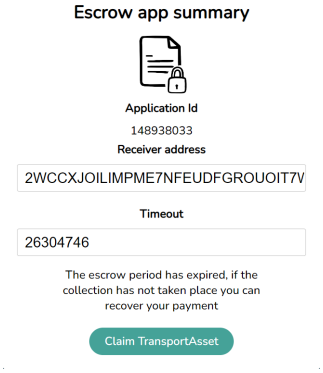


Figure 5.12. Escrow contract summary

Regarding Fig 5.11, the user can view the balance of assets in their wallet, while in Fig 5.12 they can check if the contract has expired and then proceed to request the previously made payment back if the home pick-up has not taken place.

5.2 Smart contract implementation

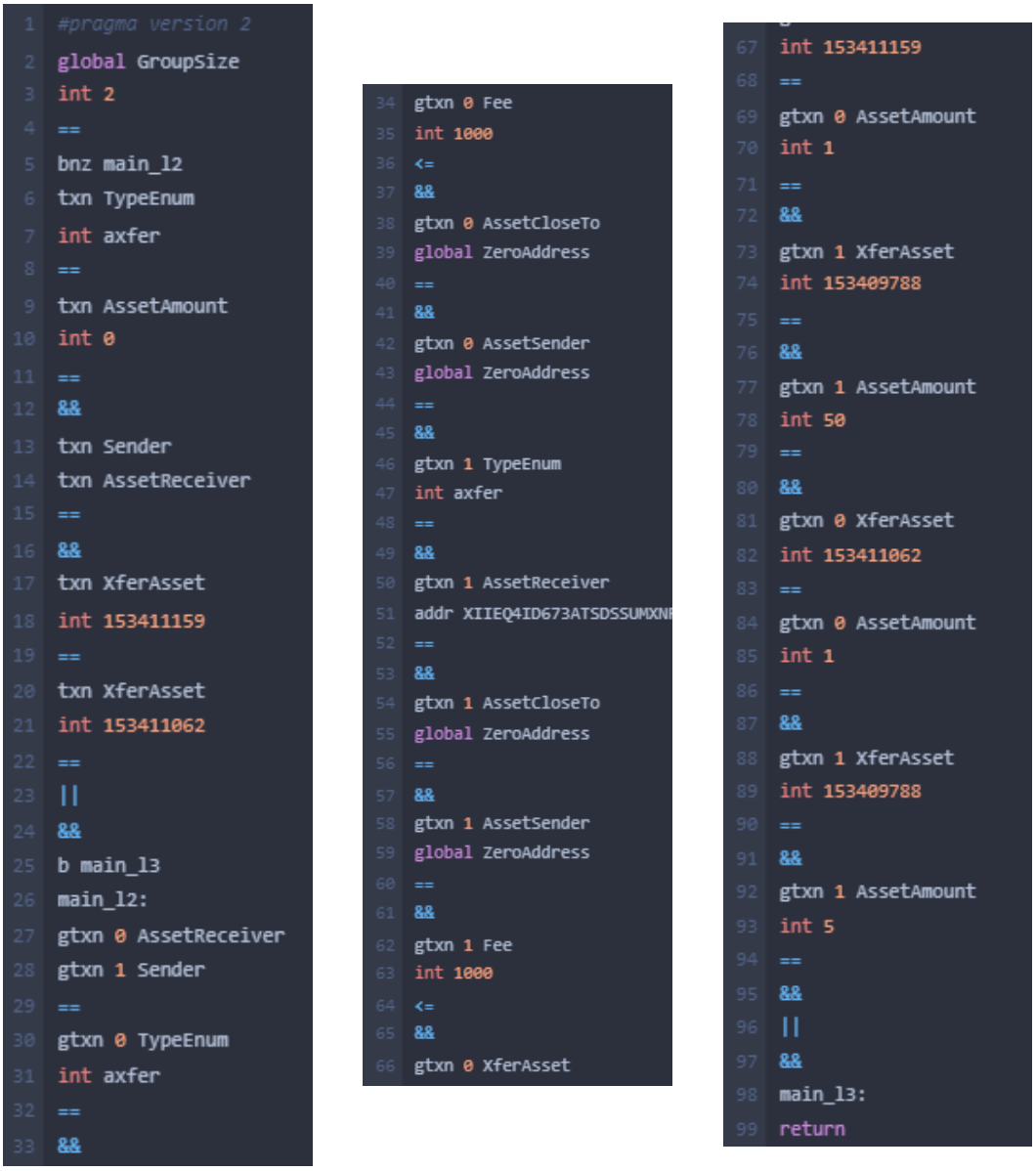
Algorand introduced the Algorand Virtual Machine (AVM) in 2020, with the release of the Algorand 2.0 network. The AVM is a smart contract platform that allows developers to build and deploy smart contracts on the Algorand blockchain. The language used by the AVM is called TEAL (Transaction Execution Approval Language), and the Bulky Rewards smart contracts have been implemented using this language through the AVM pragma version 6.
In order to facilitate the coding of smart contracts, a library called PyTeal has been chosen. PyTeal allows you to write stateless and stateful smart contracts in Python and then compile them into TEAL code. It's important to note that the compilation performed by PyTeal does not return byte code, but rather a simple translation of the contract into TEAL. Figure 5.13 shows the implementation through the PyTeal library of the asset exchange contract, while Figure 5.14 shows the corresponding compilation in TEAL language. The exchange contract was chosen for illustrative purposes only, as it has more compact functionalities.

```python
import os
from pyteal import *

def transportExchange():

    optin = And(
        Txn.type_enum() == TxnType.AssetTransfer,
        Txn.asset_amount() == Int(0),
        Txn.sender()  == Txn.asset_receiver(),
        # <ID TransportAsset> or <ID TicketAsset>
        Or(Txn.xfer_asset() == Int(153411159), Txn.xfer_asset() == Int(153411062))
    )
    general_checks = And(
        # User is the same that receive the asa and send the eco to stateful
        Gtxn[0].asset_receiver() == Gtxn[1].sender(),
        Gtxn[0].type_enum() == TxnType.AssetTransfer,
        Gtxn[0].fee() <= Int(1000),
        Gtxn[0].asset_close_to() == Global.zero_address(),
        Gtxn[0].asset_sender() == Global.zero_address(),
        Gtxn[1].type_enum() == TxnType.AssetTransfer,
        # <Addr secondo stateful>
        Gtxn[1].asset_receiver() == Addr("XIIEQ4ID673ATSDSSUMXNRV2XDITJ6HUXRZSX4J5SKKBQIZSW5GZBHGPX4"),
        Gtxn[1].asset_close_to() == Global.zero_address(),
        Gtxn[1].asset_sender() == Global.zero_address(),
        Gtxn[1].fee() <= Int(1000)
    )
    controls = And(
        general_checks,
        Or(
            And(
                # <ID TransportAsset>
                Gtxn[0].xfer_asset() == Int(153411159),
                Gtxn[0].asset_amount() == Int(1),
                # <ID EcoAsset>
                Gtxn[1].xfer_asset() == Int(153409788),
                Gtxn[1].asset_amount() == Int(50),
            ),
            And(
                # <ID TicketAsset>
                Gtxn[0].xfer_asset() == Int(153411062),
                Gtxn[0].asset_amount() == Int(1),
                # <ID EcoAsset>
                Gtxn[1].xfer_asset() == Int(153409788),
                Gtxn[1].asset_amount() == Int(5),
            ),
        )
    )
    return If(Global.group_size() == Int(2), controls, optin)
```

Figure 5.13. PyTeal Exchange contract



```
#pragma version 2
global GroupSize
int 2
==
bnz main_l2
txn TypeEnum
int axfer
==
txn AssetAmount
int 0
==
&&
txn Sender
txn AssetReceiver
==
&&
txn XferAsset
int 153411159
==
txn XferAsset
int 153411062
==
||
&&
b main_l3
main_l2:
gtxn 0 AssetReceiver
gtxn 1 Sender
==
gtxn 0 TypeEnum
int axfer
==
&&
gtxn 0 Fee
int 1000
<=
&&
gtxn 0 AssetCloseTo
global ZeroAddress
==
&&
gtxn 0 AssetSender
global ZeroAddress
==
&&
gtxn 1 TypeEnum
int axfer
==
&&
gtxn 1 AssetReceiver
addr XIIEQ4ID673ATSDSSUMXN
==
&&
gtxn 1 AssetCloseTo
global ZeroAddress
==
&&
gtxn 1 AssetSender
global ZeroAddress
==
&&
gtxn 1 Fee
int 1000
<=
&&
gtxn 0 XferAsset
int 153411159
==
gtxn 0 AssetAmount
int 1
==
&&
gtxn 1 XferAsset
int 153409788
==
&&
gtxn 1 AssetAmount
int 50
==
&&
gtxn 0 XferAsset
int 153411062
==
gtxn 0 AssetAmount
int 1
==
&&
gtxn 1 XferAsset
int 153409788
==
&&
gtxn 1 AssetAmount
int 5
==
&&
||
&&
main_l3:
return
```

Figure 5.14. TEAL Exchange contract

Chapter 6

Conclusion

The goal of the completed project was to increase awareness among citizens about the proper way to dispose of waste in cities, with the aim of improving the livability of urban areas and the health of the environment. Additionally, implementing projects like this one helps to spread blockchain technology, with the goal of creating more secure and reliable systems that can improve our daily lives.

6.1 Known issues and limitations

The main issues and limitations are the following:

- Improper exchange of assets: There is the possibility that users may exchange assets with each other, which does not increase the number of assets in circulation. This can be somewhat limited by using revocation operations implemented by the asset creator if a bad actor is detected.
- Compromise of reward calculation data: If the data used to calculate rewards is compromised by a malicious actor, it may result in incorrect rewards being distributed.

6.2 Future developments

Bulky Rewards is designed to be used in large metropolises, small towns or even just in neighbourhoods. However, since there could be untrustworthy authors in such a system, one possibility for future development could be to use a private permissionless blockchain to create a system that can only be used by a limited group of people, for example, exclusive for people living in a specific city. Below are also listed some possible future developments regarding the functionality of the system, all of which include an asset reward:

- The possibility of reporting the presence of litter on the streets.
- Recycling of non-bulky waste, such as paper, plastic, or metal.

6.3 References

1. Blockchain history: Mastering Blockchain, Imran Bashir
2. Algorand blockchain: https://www.algorand.com

3. Algorand development: https://developer.algorand.org
4. Algorand foundation: https://www.algorand.foundation.com
5. PyTeal: https://pyteal.readthedocs.io
6. Circular economy: https://en.wikipedia.org/wiki/Circular_economy