Benefits Blockchain Formal Report

Brett Egan
Dennis Reed
Paul Barstad
Chris Haynes

Abstract

USDA's Supplemental Nutrition Assistance Program (SNAP) is the program that manages welfare benefits in the United States. Unfortunately this program faces many issues, the most important of which is tracking money assigned to participants and where they can use that money. In 2018 the Economic Research Service analyzed design issues within this program and found six main flaws with it. SNAP impacts roughly 14% of the population and costs nearly $70 billion annually. SNAP is a program that is closely examined by the government and its officials because of its significant stake in the United State budget. There are six main areas in which this study found flaws within the current SNAP system. The first one being that the government gives each state a "lump sum" of money and lets each state decide how the money should be allocated. The problem with this is that two people in different states that have identical circumstances will likely receive differing amounts of money. Second issue is that SNAP doesn't limit the foods that participants can buy so anything with a nutrition label is free game. Reducing the items that participants choose from would increase the amount of nutritional benefit they get from the allotted amount of money they have. Also SNAP members and TAFDC (cash assistance) programs from the government both give participants an EBT card which people can purchase both food and non food items with this card. Third, stores have to become SNAP authorized which means they have to meet certain nutritional requirements. This list is constantly modified and needs to be easily standardized. Fourth, concern over how adequately people get compensated for factors such as: variation in food prices by location, variation in age and nutritional requirements for members of a household, and the cost associated with preparation of

a given food. Fifth, states can choose to use BBCE SNAP which helps state save money on administrative costs but allows higher income people to enroll in the program and raise costs of SNAP. Sixth, Individuals participating in SNAP must comply with the requirements of work. This includes applying for job, not quitting jobs, and/or show progress toward getting a job. This can be difficult to enforce and is hard for the program to track. (Oliveira, 2018)

Introduction

We created a token, called *SnapToken*, based on the current government program called SNAP. It automatically distributes government benefits monthly, for example - food stamps. Users are required to be already approved by the government before registration on our DApp. Once users have registered, they are allowed to spend their *SnapTokens* at authenticated vendors. Vendors are then be able to exchange the tokens they receive over time, using our DApp, for Ethereum. Prices are based on the current exchange rate of Ethereum, which are fetched using an oracle. We recognize the centralized behavior of the government (which is the authorized owner of the smart contract) by approving vendors and users to register in the contract, as well as the distribution of the tokens. However, this system is more transparent than the current system as record of use is permanently saved on the blockchain.

Motivation

Our motivation for this project are the issues in the article stated above. Although there are six problems with SNAP currently, it is unrealistic for us to solve all of them in the limited amount of time that we have. As a team we chose to focus on one current issue with the SNAP program. The problem is that the SNAP program and the cash assistance program use the same debit card. While SNAP benefits are only supposed to be spent on food, if someone is utilizing both programs they will be able to spend money on non food items using money that should be only for food as well. Our contract only focuses on SNAP members and therefore our SnapTokens will only be allowed to be spent at food-only vendors. This eliminates the use of debit cards and rather participants have to redeem their *SnapTokens* at verified vendors. Then in-turn vendors can redeem *SnapTokens* for an Ethereum amount equivalent to the price of the food. If given more time to work on this project we would address another issue with SNAP which is when some participants buy items that are too expensive or unnecessary. By tracking their purchase history of food to verify their food changes are approved SNAP will be able to prevent more fraud and abuse in the program.

Goals

- Create a mint-able token SnapToken
    - We focused on non elderly or disabled beneficiaries to avoid unnecessary challenges.
- Create smart contracts for registered users and authenticated vendors.
- Connect contracts to a DApp.

High Reach Goals

- Fetch current exchange rates via an oracle
- Integration with KeepKey

Issues Encountered

The first large issue we encountered was being unable to import files from OpenZeppelin's GitHub. We were unsure if we needed to download the repository, and import the files locally or if we could use the website link. By using a more recent compiler version for

Solidity (5.7) we finally got the imports functioning. There was a learning curve for us to use the files we imported. We we went back and forth on whether to use burn() or _burn(), have the contract be an ERC20 or an IERC20... After some trial and error we got it all sorted out. Developing in Remix was a struggle throughout the past few weeks. Remix was laggy, you could type faster than the site would display it. There were two instances when the IDE would not compile, forcing us to leave earlier than the time we set aside to work as a group. Dziugas was very helpful when we wanted to connect Oraclize to our contract. We were unaware we needed a _callback() function, and did not know the return structure of an oraclize_query. Once Dziugas assisted us, we were able to successfully grab the current dollar value of one ether for our conversion. The last hurdle we came across was getting ether into our contract. We were under the impression that any function that interacts with ether (depositing or removing) needed to be a payable function. Once finding out only the functions adding ether to the contract need to be payable, we were able to successfully transfer ether to a vendor in exchange for their SnapTokens. An incorrectly placed _burn() was setting the amount to zero and causing no ether to be transfered. By moving the _burn() to after the transfer we were able to successfully transfer the correct amount and then burn the correct amount of SNAP tokens. On the front end, the hardest part was interacting with the contract and correctly storing the state returned from call(). The asynchronous nature of the calls was difficult to handle since we have not encountered async calls yet.

Conclusion

Overall we reached all our baseline goals. For our high reach goals we were able to achieve fetching current exchange rates of Ethereum so we can give vendors an accurate amount of cash when they exchange their SnapTokens to Ethereum. We did not reach our goal of integrating KeepKey into our authentication. This was due to the fact that KeepKey's presentation was delayed and we decided it was a far fetched goal for us to put effort into. We were able to successfully implement all of the features we set out to achieve. This includes working through the many issues we had in Solidity and deploying our website tying our smart contract to Metamask.

References

1. Oliveira, Victor. "Design Issues in USDA's Supplemental Nutrition Assistance Program: Looking Ahead by Looking Back." United States Department of Agriculture. Jan. 2018. 01 Apr. 2019 <https://www.ers.usda.gov/webdocs/publications/86924/err-243_summary.pdf?v=0>.