Bilkent University

Department of Computer Engineering

Senior Design Project

Project Short-name: So FarM So Good

Final Report

Giray Baha Kezer, Fazilet Simge Er, Melih Ünsal, Kaan Atakan Öztürk

Supervisor: Prof. Dr. Halil Altay Güvenir

Jury Members: Prof. Dr. Özcan Öztürk, Prof. Dr. Uğur Güdükbay

Innovation Expert: Kerem Erikçi

May 27, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS492.

Table of Contents

1 Introduction

1.1 Summary of Project and Problem Introduction

SoFarmSoGood is farm friendly software that intends to encourage farmers to produce and sell their harvest. In Turkey producers face to quota problem as a main problem. We focus sugar beet as our project base. Encouraging local farmers of a harvest into whole purchasing system is an undeniably important issue for agriculture supported economy. We offer a blockchain system that encourage local farmers who could not exceed purchasing quota limit of huge companies, named SoFarmSoGood. Into our solution we have three actors, farmers as producers, cooperatives as virtual farmer communities to be part of purchasing, and companies. These three main actors are the core of blockchain integrated purchasing and selling system. Our farmers with differentiated with IDs, are identified with unique or current harvest by its ID, then be part of transaction related with its concept. If he is lower the initial quota, he could be part of a virtual cooperatives in terms of his harvest value then cooperative sell their product to the company, thus balance of farmer will be changed into system. Cooperative can be thought as the new cover of transaction to be able to pass the quota. As an example, Ahmet is one of the sugar beet farmers; however, he has not enough soil for plant culture, either he has not enough cultivation money for its expenses. In that cases, what would Ahmet do? Not cultivate his soil, appropriate for sugar beet, just related with not having the enough amount of harvest. In that cases, we offer him an option; find the Mehmet be together as virtual cooperatives with unique ID exclusive for that transaction or process. Thanks to that cooperation, sell their products to the company, take the money and meet the need and continue to cultivate. Otherwise, farmers are discouraged from planting their products, so it decreases the total production in the national scope.

1.2 Project Scope

We focus sugar beet as the main topic for process. Nevertheless, we have that occasion into several products, such as hazelnut, tea plant, etc. All of them are undeniably and critically important for national agriculture. Blockchain is a specific concept that is totally appropriate and effective for agriculture itself, thus project scope is only related with demand. It can be integrated with new products, new projects and become larger into scoping.

2 Requirements Details

2.1 Functional Requirements

A functional requirement is exactly what the user needs the product to do. Functional requirements are separated into two parts below in terms of their specification.

2.1.1 User functionality requirements

- Farmers could have the ability to create virtual cooperation according to the demand.
- The platform encourages farmers to produce and sell their harvest as a total participated manner.
- The platform provides users to form their cooperatives to sell the same type of product.
- The platform provides companies with cooperatives related to their needs in production.
- The platform recommends farmers to property-based sell options to directly companies if they have adequate harvest in terms of different quotas.
- The cooperatives take material of the farmers, then the transaction is stalled until the company purchase materials from the cooperative, after that step money will be transferred to the farmer total balance.

2.1.2 System Functionality Requirements

- The platform needs to be a web-based served blockchain platform.
- There needs to be two different user type, cooperative, farmer and companies as actors.
- Each user could participate into platform with related harvest.
- Cooperative names must be unique for that transaction into transaction pool.

- Farmers’ farmerID, cooperatives’ coopID, companies’ companyID, harvest’s harvestID must be unique.
- A database is set up for the retrieved data from actors.

2.2 Non-functional Requirements

2.2.1 Usability

The platform has ease of use for our main customers, farmers, which are the most important part of agriculture. Farmers demand more ease of use for this technological move. Thus, they are able to use the platform easily and willingly.

- Features of the platform and user interface of composer’s related components are easily understandable and functional.

2.2.2 Security

- The platform ensures the security of data retrieved from all actors and private information about companies by blockchain system.

Blockchain transactions are also secured by cryptography. Each transaction is protected with a private key and then can be further verified with a public key. If transaction data changes, the signature becomes invalid.

2.2.3 Scalability

- The platform is scalable enough to handle the huge data volume of production actors and data processing work.

2.2.4 Robustness

- Blockchain is indeed very promising for its high-quality encryption and monitoring. It also holds some other features, like its robustness. In fact, it has a built-in robustness and durability functionality. It does not have a single point of failure because of its distributed nature. Hash keys encrypts input of letters and numbers into a fixed length output.

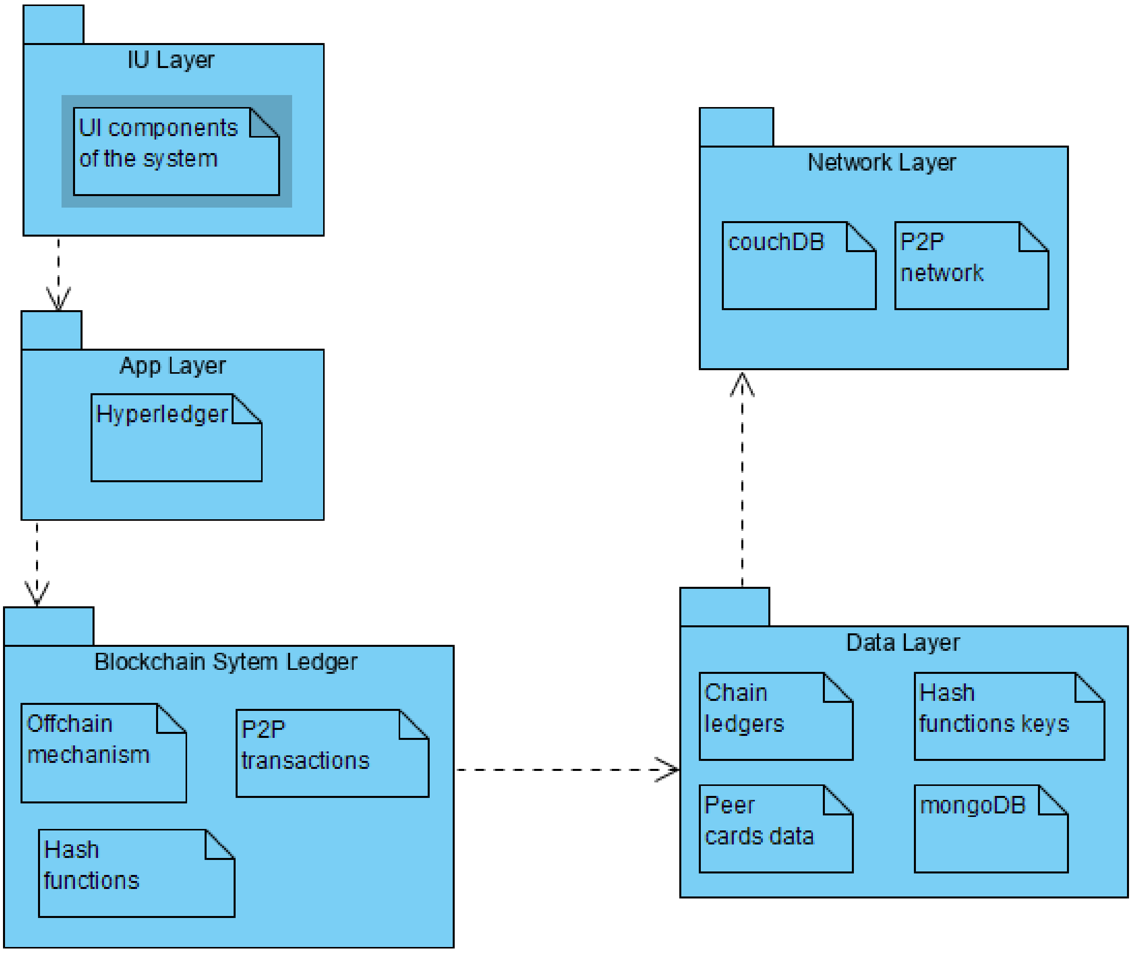
2.2.5 Extensibility

- The platform is extensible in the UI helps to show more relevant and organized data for production actors.
- Extensibility in the backend helps you perform custom business logic and processing. Custom logic could be driven by integration with other applications in further versions, with changed use cases, and other special requirements of an application.
- Backend extensibility allows you to change the server-side Restful API to perform the customized business logic into harvest purchasing system.

3 Final Architecture and Design Details

3.1 Overview of Architecture

SoFarmSoGood application provide users decentralized data system. System has MongoDB since the Blockchain technology has naturally it. CouchDB is also database that uses JSON and API actions [1]. The data in the system will not disappear when a member of the system is crushed.



3.1.1 Data Persistence Subsystem

The subsystems are MongoDB, Hyperledger Composer, Hyperledger Fabric and CouchDB by docker.

3.1.2 Recommendation Subsystem

This subsystem occurs where the calculations of recommendation procedure is completed. Recommendation subsystem uses our Consensus Algorithm with decentralized system.

4 Development/Implementation Details

4.1 Developed Solution in SoFarmSoGood

Selling harvest is an undeniably important branch for national production and economy. Farmers are the dynamic actors of this branch. For the process of harvest production, farmers need to maintain initial transactions with different actors. We integrated that specific transaction steps with unique actors into our blockchain system. However, recent harvest production and selling system is only appropriate for bigger local farmers or larger institutional actors. We focus smaller local farmers, which can only produce their harvest into related purchase quota. In our country we have some purchase quotas of huge companies that shape harvest production, such as Konya Şeker for sugar beet, Oltan Gıda for hazelnut, Çaykur Çay İşletmeleri tea plant, etc. In our main focus group smaller local farmers and all producers can only produced below purchase quota, we developed virtual cooperative solutions. In our virtual cooperatives we build initial required cooperatives for selling harvest with several farmers. To illustrate, Ahmet is a passionate sugar beet farmer, but he has lower potential of required quota of purchasing limited by huge companies. What would Ahmet do except for selling his harvest to the larger producer by its lower value? We offer Ahmet and other farmers in similar situation. We let Ahmet and Mehmet be together and provide them an opportunity with selling their harvest with real value. Additionally, SoFarmSoGood’s offered solution could provide huge companies with bigger market volume with participation of local farmers. Thanks to that bigger market volume could trigger bigger production chain and directly higher exportation values. To summarize through total participation of local farmers into active purchasing harvest, larger production chain with win-win situation between farmers and companies, could be built.

In our service, a harvest is identified into system and some farmers can be gathered and sell their product to meet the demand of company purchasing quota. To meet that function, we need trigger as a selling ratio to the farmer from cooperative so as to take farmers’ harvest. Nevertheless, in blockchain system we do not have triggers into communication between data and related database. Thereby, we created a system with data holder for harvest transaction between triangle of farmer, cooperative and company. We took last owner of initial harvest between them. When the last owner of harvest return to the company from farmer to cooperative and lastly to company, the determined harvest price is transferred to the farmer balance.

4.2 Used Tools of Developed Solution and Related Technology

4.2.1 mongoDB

Hyperledger Blockchain technology uses MongoDB as its base to store data in system. In the hyperledger composer data retrieved and stored into its base database named MongoDB [2].

4.2.2 Hyperledger

Hyperledger is a useful Blockchain technology that eases coding of blockchain system into project requirements [3]. Hyperledger contributes developers related RESTFUL API and its integrated Angular UI component-based frontend for Blockchain logic.

4.2.3 Docker

Docker container is used for distribution the data of actors among the system peers [4]. Peers are the actors of harvest production and purchasing system. Through the Dockers’ Couch Database system, we can retrieve related data via GET, insert them into related JSON format via POST, update it using PUT also DELETE them. Through ‘crud’ structure.

4.2.4 Angular

Hyperledger composer is integrated with related Angular frontend support with the version of Angular 2.0 into initial REST Server. It can be tested into related local server for deploying. It is appropriate for making connection between RESTFUL API and actors via peer card [5]. HTML protocols of JS scripting file are also used for system by related angular app with npm package manager via Yeoman of Hyperledger Fabric.

The UI of the website which is still in progress is being written in Angular as well.

4.2.5 Typescript

Angular functions are implemented in TypeScript and it is appropriate for Angular 2.0 integrated frontend components of blockchain backend logic [6].

5 Testing Details

Unit testing is used for whether the data retrieving or not in the objects. Retrieving and validating process can be implemented in this way. The platform can be tested through admin endpoint for actors such as Farmers, Cooperatives, Companies in terms of the reliability, usability, and security. Unit testing also provide eliminating of typos or logic errors in the business logic. Automated system tests are also be implemented within HyperLedger Composer. Automated system test may be implemented after deploying business network, creating assets, and submitting transactions. Testing may be implemented also from command line. Developers and testers may use **composer network list** command to see the state of asset and participant registries. **Composer transaction submit** command can be used for submitting transactions. The last testing method for HyperLedger composer is interactive testing. Testers can use playground to test creating participants, assets and submitting transactions.

The user interface of the website, which is still in progress, is being written by testing it in Google Chrome inspection. Further work will be done on the testing of UI of the website in the future, and these will be further explained in Part 9 of the report.

6 Maintenance Plan and Details

If the project's future is determined and desired, the related unique maintenance plan will be drawn. SoFarmSoGood agriculture project is on the local server for that point. It will be maintained with worldwide web services, clouds if required. However, current system does not serve everything we offer before the implementation, it could be sustained with advance versions into professional web services, cloud supports such as Amazon Web Service. Data consistency and project availability can be progressed with required amount of effort in the future of project. Firstly, we could divide our project into parts then schedule them into smaller pieces. Secondly, allocate related human resource into effort-based scheduling after that process if it happens add actions for spare parts or unexpected cases then related with permissions of workers allocate and work them with each other. Thus, we could maximize our efficiency into work allocation and future plan of SoFarmSoGood project.

7 Other Project Elements

7.1 Consideration of Various Factors

7.1.1 Security

We consider safety issue by using Blockchain technology. This technology is innately secure because it utilizes powerful cryptography. Each transaction is signed with the private key and then can be further verified with a public key. If transaction data changes, the signature becomes invalid. As a result, the block is ignored and will not make it to the chain.

7.1.2 Welfare

Our main aim is to support native farmers by establishing cooperatives. Since farming is one of the locomotives of the Turkey’s economy, welfare is causally related to the farming and farmers. In our project, sugar beet farmers come together and resolving quota problem. After solving quota problem, farmers are not going to sell their products to crop chandlers. These crop chandlers purchase the crops much cheaper than the government or other sugar beet cooperatives.

7.1.3 Social Factors

SoFarmSoGood helps to the farmers to control and manage crop distribution. Our software also helps the farmers to sell sugar beets even if the amount of the crop is less. Moreover, farmers hold the data by the Blockchain and make contributions to the economy of the country.

7.2 Ethics and Professional Responsibilities

Data privacy and security will be provided with blockchain system into access permissions among production actors. Blockchain logic are supported with related access and permissions between actors. It means different actors will not reach the other process. Transactions between virtual cooperatives and companies are unique for every transaction with unique IDs and then new required cooperatives will be built, dependent present cases. Ledgers, blocks are also encrypted with 64 key randomized hash chain-keys to sustain data privacy.

7.3 Judgements and Impacts to Various Contexts

Judgement Description	The platform will be used by many farmers, companies, and cooperatives. Moreover, farm industry will grow directly by producing.

	Impact Level	Impact description
Impact in global context	0	The project is designed for the Turkey in first step, so it

		has no global impact for now.
Impact in economic context	8	Platform will give rise growth as farm economy to the farmers, cooperatives, and companies.
Impact in environmental context.	2	There will be no direct correlation between environment and platform. Fields of the sugar beets will be controlled and planted regularly by producing sugar beets.
Impact in societal context	8	The platform will have significant social effect on the society. If the local farmers develop, economy of the villages, towns, cities and also country will grow.

SoFarmSoGood is blockchain based software which has decentralized infrastructure. This platform is open to every farmer, company and cooperative. It is totally free. We do not price a money for its usage. Since our aim is to reach local farmers, we do not have economic concerns. Farm is the one of the main locomotives of the Turkey economy. If economy of the farmers develops, Turkey's economy will grow directly. Namely, SoFarmSoGood has a significant impact on the social context of the farmers.Moreover, transactions of the farmers will be encrypted by using hash keys. As this technology is extremely secure because of its nature, farmers, companies, and cooperatives will not suspect on the security concerns.

7.4 Teamwork and Peer Contribution

With the remote education taken into place, it became harder for us to work together since face-to-face meetings play a big role in projects. Although it was hard, we tried our best to maintain the teamwork.

Below are the contributions of each member in the group throughout the project:

Melih Ünsal: Blockchain system research, RESTFUL API and databases of Hyperledger and Docker.

Giray Baha Kezer: All related research about blockchain systems, composer's convenient versions-using cli with package manager, building RESTFUL API and related connections with internal mongoDB and external couchDB, databases of Hyperledger and Docker. Built RESTFUL API of hyperledger composer. Built Hyperledger composer blockchain logic with related ledger connections and decentralized system peer actors blocks. Built related integrated Angular apache v2.0 into RESTFUL API via yeoman generator through localhost for development, deployment and testing. Built core Hyperledger Composer, RESTFUL API and frontend Angular communication.

Kaan Atakan Öztürk: All related back end part of the project was implemented. Research about HyperLedger Composer and Docker. Installation of the Hyperledger and Docker was completed. After building RESTFUL API, database connections were established. Actor blocks connections also were established with ledgers and network. In the script file of the projects, various changes were made. 3 main transaction types were added to the system as logic of the project. These transactions become adapted with actor blocks after implementation. Version incoordination problems were also solved within the system.

Fazilet Simge Er: Basic blockchain research, just enough to understand what the project will be about, to be able to come up with the diagrams and the mockups of the system. Angular research for the UI of the project and the frontend implementation of the website, which is still in progress.

7.5 Project Plan Observed and Objectives Met

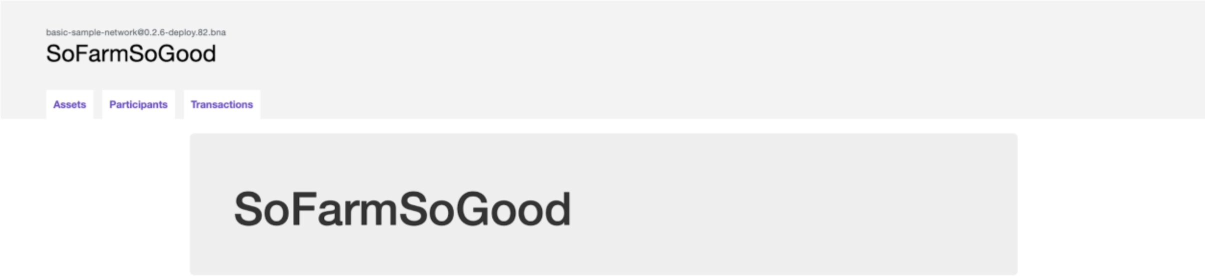
Farmer actors are identified with their harvests and balance. Company and virtual Cooperative actors are identified, their connections and transactions are satisfied into blockchain system. Sell-buy process are satisfied, transactions are recorded with blockchain diaries from past to current. Related basic Angular based frontend UI are satisfied but friendship relationships, social features of our application are not satisfied. Most of project plan are satisfied but not social features, GPRS-coordination finder system are not satisfied, one total functionality are not satisfied, named online server-based web application with farmer actor usage with related location-based algorithm suggestions.

7.6 New Knowledge Acquired and Learning Strategies Used

Hyperledger is an umbrella project, appropriate for blockchain technology. We developed our project through using its features of fabric. Hyperledger fabric provides us its composer named Hyperledger composer with its related playground, served in localhost:8080 address as its base. We developed our blockchain logic into that channel and built RESTFUL API to communicate with it. This communication raises up the connection between them and logic is converted into ‘crud’ structure-based functionality. It is served into localhost:3000 as its base, with this channel project is ready for serving UI integrated with convenient Angular. As the last layer we raised Angular up thanks to using fabric provided package managers, tool managers named yeoman. Through yeoman project is converted into frontend-based project of blockchain logic and so, package manager of npm can be appropriate for start the stream. Docker is a network container for Hyperledger Fabric cores, we start our fabric.sh scripting files into that container. Their names are start, stop and teardown fabric scripting files are working in Docker image and thus, network can be raised up via container. Raising up network into Docker, with the help of creating related peer card, as an admin, transformed our network into convenient for blockchain logic. Then through learning related blockchain system, actors’ communications via peer cards and giving related permissions for their transactions is enough for creating our system. After learning new type of parameter based, two step-based transaction models we could create our blockchain logic.

8 Users Manual

In this section, activity flow of user action will be shown so as to illustrate how SoFarmSoGood, blockchain solution for harvest purchasing and gathering farmers into virtual cooperatives system, works and to contribute user manual for our actors for providing them ease of use.

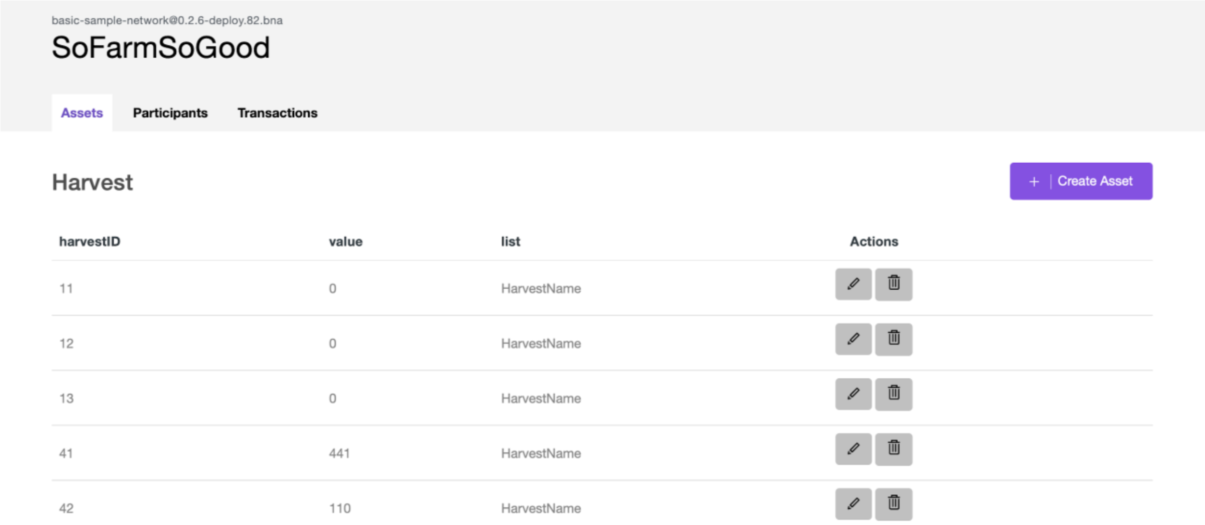


basic-sample-network@0.2.6-deploy.82.bna

SoFarmSoGood

SoFarmSoGood

In the first page actor experience with basic UI for our project. Assets, Participants can be entered or picked, or transactions can be investigated from saved into blockchain records/diaries.



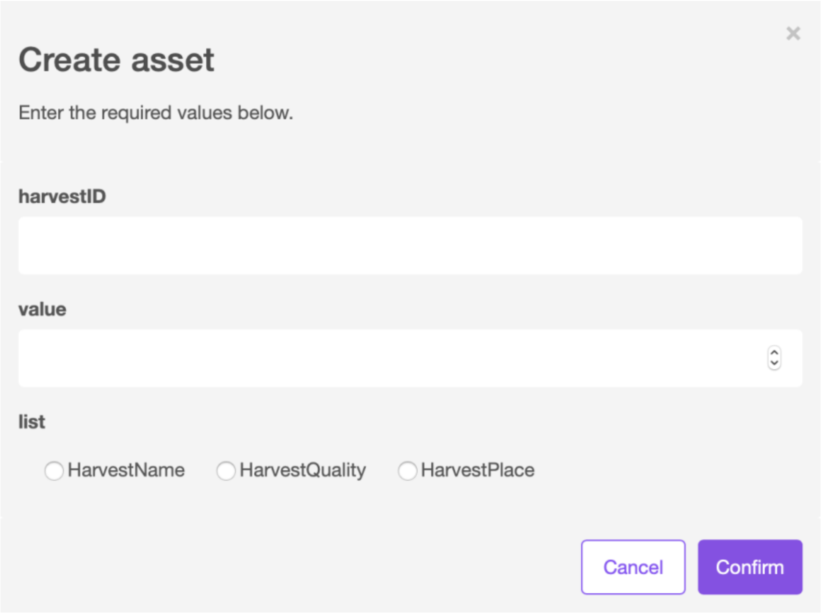
basic-sample-network@0.2.6-deploy.82.bna

SoFarmSoGood

Harvest

harvestID	value	list	Actions
11	0	HarvestName	
12	0	HarvestName	
13	0	HarvestName	
41	441	HarvestName	
42	110	HarvestName	

After the first step, we could experience our activity flow in farmer scope at that step. Farmer needs to enter initial harvest as the new one, either previous recorded ones. After that step of harvest actions can be done by transactions between different actors.



Create asset

Enter the required values below.

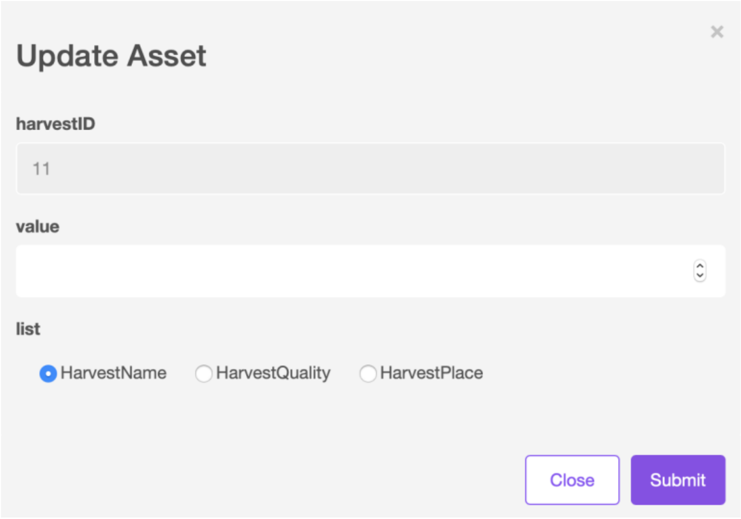
harvestID

value

list

○ HarvestName ○ HarvestQuality ○ HarvestPlace

Enum based list can be entered into name, quality, and place for harvest. After that harvest step transaction can be done.



Update Asset

harvestID

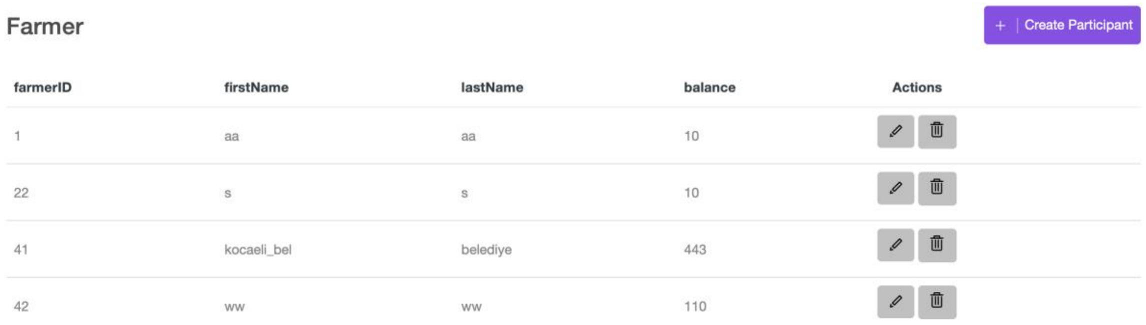
11

value

list

● HarvestName ○ HarvestQuality ○ HarvestPlace

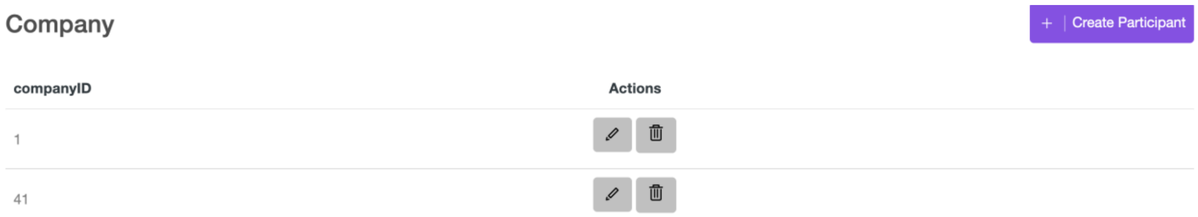
Assets can be updated with new values in terms of name, quality, and place differentiations. Through picking related harvest, taking it with unique IDs, give actor changing feature.



Farmer

farmerID	firstName	lastName	balance	Actions
1	aa	aa	10	✎ 🗑
22	s	s	10	✎ 🗑
41	kocaeli_bel	belediye	443	✎ 🗑
42	ww	ww	110	✎ 🗑

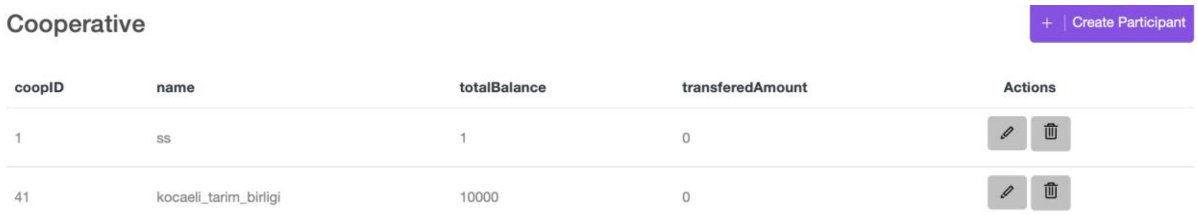
Farmers can be created as new participants or delete old ones. Farmers can be edited into related harvests via differentiating with IDs.



Company

companyID	Actions
1	✎ 🗑
41	✎ 🗑

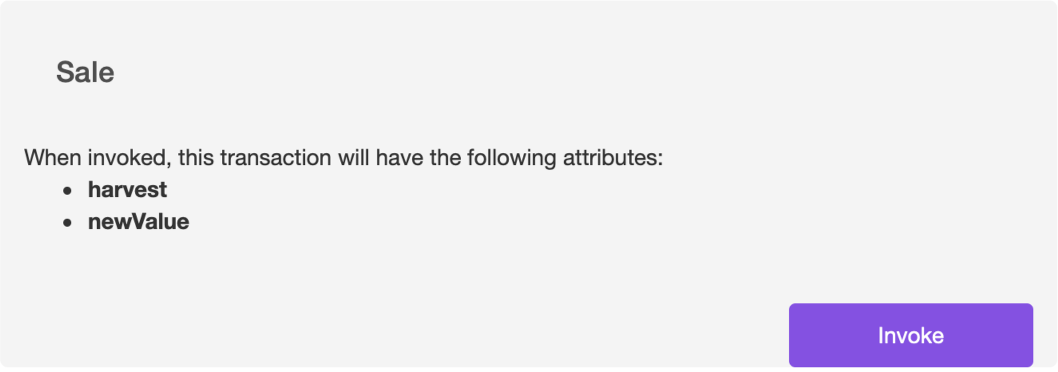
Companies can be created as new participants or delete old ones. Companies can be edited into balance. Companies can be interacted with related harvests of cooperatives or directly into the farmers via differentiating with IDs.



Cooperative

coopID	name	totalBalance	transferedAmount	Actions
1	ss	1	0	✎ 🗑
41	kocaeli_tarim_birligi	10000	0	✎ 🗑

Cooperatives can be created as new participants or delete old ones. Companies can be edited into new harvests of new farmers. Cooperatives can be interacted with related harvests' transactions with initial farmers and companies. Thereby, related balance updating can be done via retrieve cooperatives as parameter of process between indirectly in companies and bounded farmers.

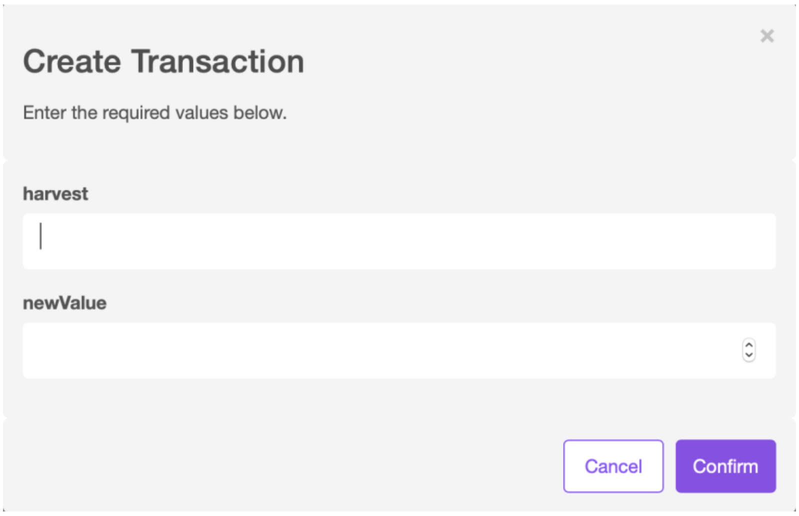


Sale

When invoked, this transaction will have the following attributes:

- **harvest**
- **newValue**

Transaction can be invoked by UI component-based interactions. Thanks to entering related harvest IDs and new values transactions between different actors, can be progressed into system.



Create Transaction

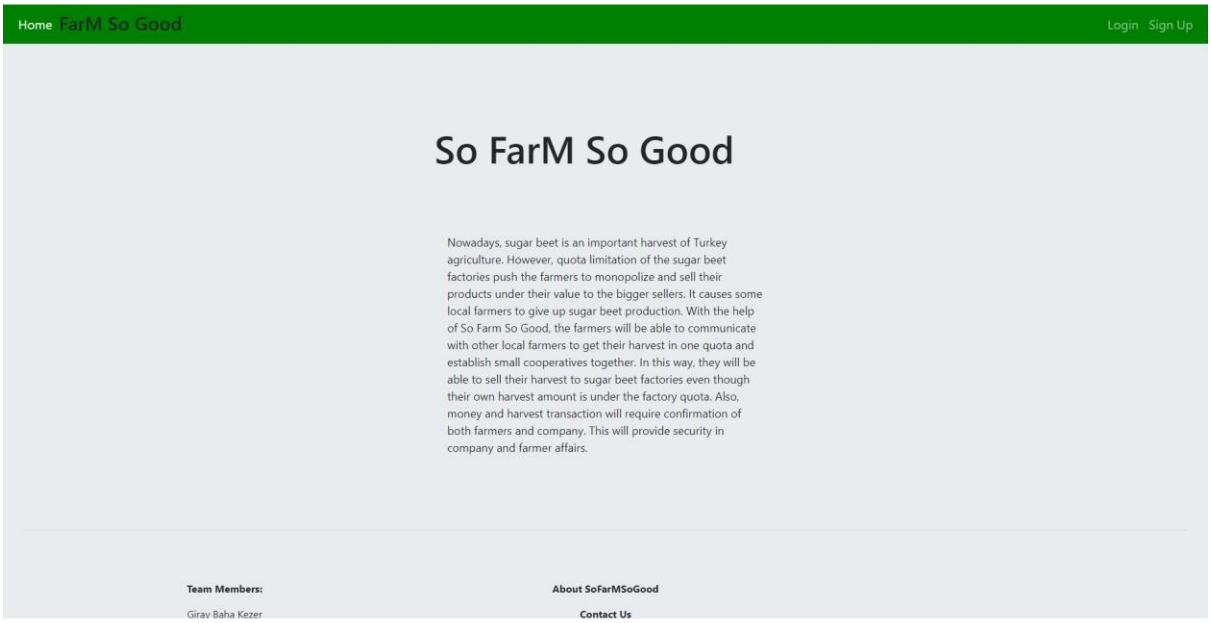
Enter the required values below.

harvest

newValue

SoFarMSoGood will also have a user-side frontend, and this part is still in progress. At the moment, the pages have navigation between them, and a fake backend written only to test them. These pages are not connected to the pages shown above, and they are not connected to the database, which the data are being used. This will be the next thing to be done.

Below, you can see a few implemented pages with basic visuals:



So FarM So Good

Nowadays, sugar beet is an important harvest of Turkey agriculture. However, quota limitation of the sugar beet factories push the farmers to monopolize and sell their products under their value to the bigger sellers. It causes some local farmers to give up sugar beet production. With the help of So Farm So Good, the farmers will be able to communicate with other local farmers to get their harvest in one quota and establish small cooperatives together. In this way, they will be able to sell their harvest to sugar beet factories even though their own harvest amount is under the factory quota. Also, money and harvest transaction will require confirmation of both farmers and company. This will provide security in company and farmer affairs.

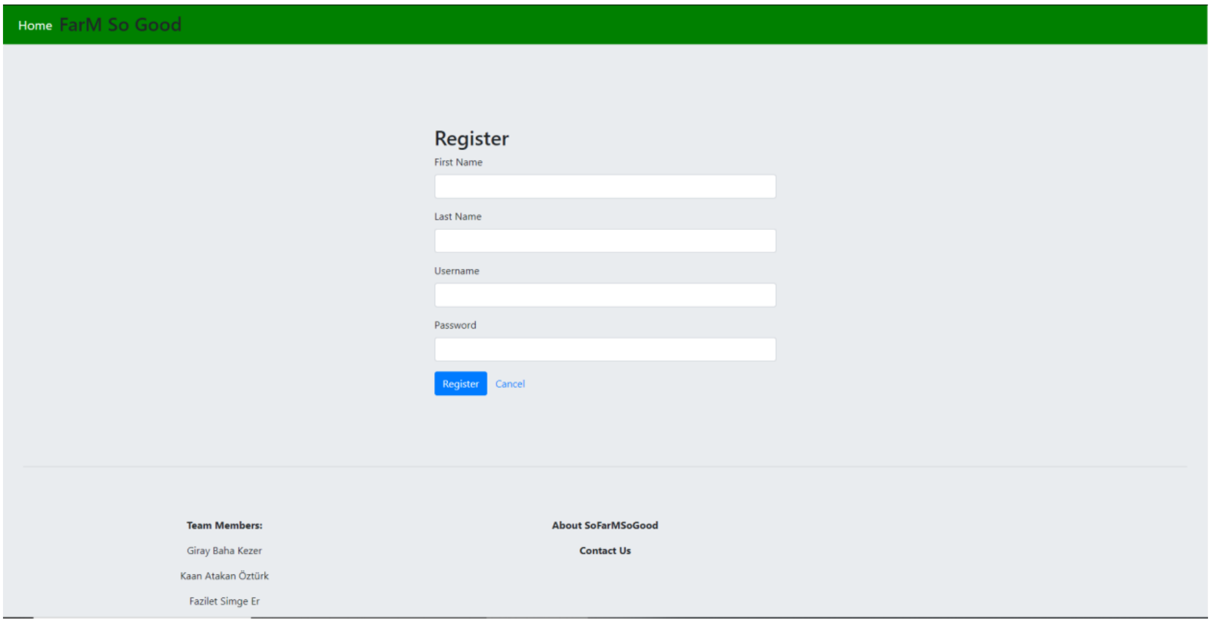
Team Members:

Giray Baha Kezer

About SoFarMSoGood

Contact Us

This will be the home page of the website. Information on the system, a logo and visually appealing pictures and components will be added later.



Register

First Name

Last Name

Username

Password

Team Members:

Giray Baha Kezer

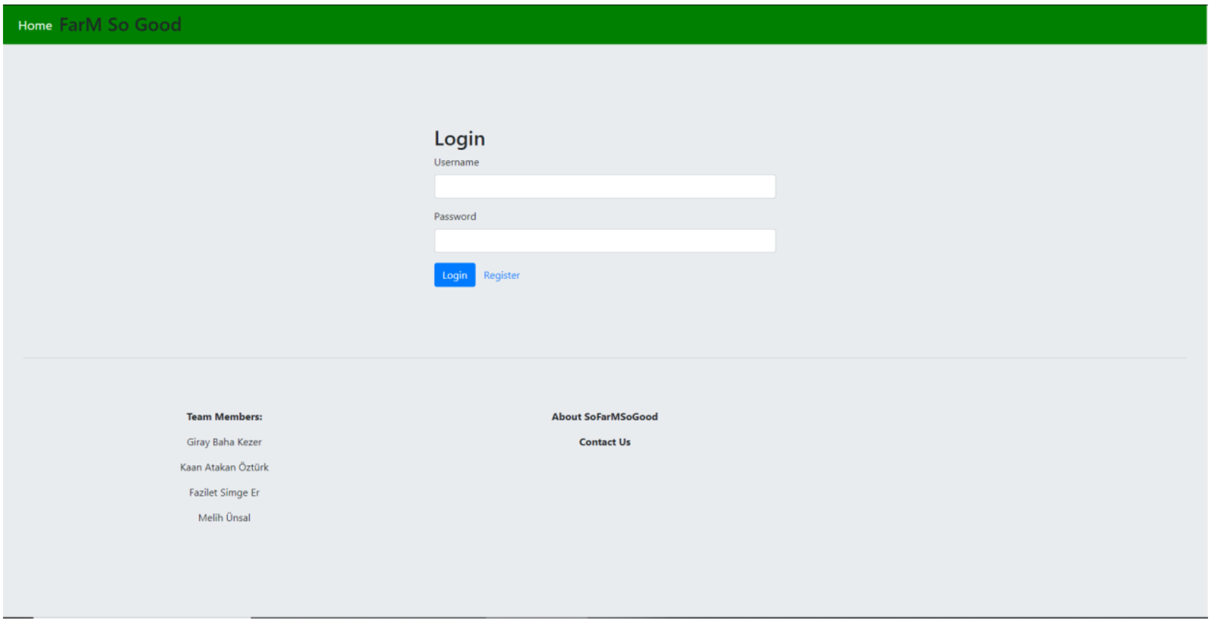
Kaan Atakan Öztürk

Fazilet Simge Er

About SoFarMSoGood

Contact Us

This will be the sign-up page, which the user can reach from the home page.



After registering, user will be able to login from the login page above.

New pages such as user home page will be written and added to the system and they will be connected to the rest of the project in the future.

9 Conclusion and Future Work

Our project purpose is offering a helper system for agriculture in terms of locality. Local smaller farmers cannot exceed the quota limitation of purchasing put by huge buyer companies. We focused sugar beet as the core harvest for project, we investigate huge buyer company; Konya Şeker. [7] For that harvest and company, we take local smaller farmer into consideration to make them be part of the whole production chain. We serve them making virtual cooperatives into exceeding quota limitations occasions, then we bind Ahmet with Mehmet Uncle into each other and count them as one seller actor. That seller actor can be communicated with company to sell their harvest easily and directly to the company. It provides real valued sales guarantee to the local farmers and bigger supply chain for the company itself. That idea would give a win-win situation to production with secure and sustained blockchain integration.

9.1 Future Work on User Interface of the Website

As it is stated in various parts above, the UI of the project website is still unfinished. Hence, in the future, the first thing to be done on the UI part is to complete the pages the website will have and connect the frontend to the backend, which is the whole Blockchain system.

In the progress of the implementation of the frontend, the written frontend will be tested in three different web browsers, which are Google Chrome, Internet Explorer and Microsoft Edge. After making sure that the website is functional and appealing to the eye, further work will be done on integration with cell phone browsers.

9.2 Future Work on AI Part of the Project

At the beginning of the project, we planned to append AI modules to the project so that it is going to be more functional, more interactive without any human effort. However, after it is decided to use Blockchain technology in our project, AI related parts of the project has been delayed so as to focus on completing the core part of the project which is our Blockchain system.

In the future, as related to AI, we plan to detect the sugar beet lands from satellite images by using one of the most popular deep learning framework which is Pytorch and the state of the art double stage object detection algorithm which is Cascade R-CNN right now.[8] By using this algorithm, the registered farmers will be able to assign their lands to the appropriate area in the satellite map of the website by simply adding a point inside of his/her landing. After the landing is assigned, all the companies in the website will be able to see the area and boundary of the appended land calculated by our model. To train the model, we plan to collect open source datasets consisting of bounding boxes of sugar beet lands and satellite images from different areas.

The second AI module that we are eager to add to the project in the future is the harvest predictor where a state of the art recurrent neural network architecture will be used for the network and also the subnetwork used in the module. This module will predict the amount and the quality of the harvest of selected farmer’s sugar beet landing. This module will inform the big companies wanting to make a trade with a cooperation about the landings of this cooperation. This information consists of the predicted quality and amount of sugar beet to be collected. The module will use the previous year’s data and another subnetwork responsible for predicting the weather forecast of the weeks from an open source dataset. The main network will be trained by an open source dataset then the predictions will be done by the entered information by the farmers in the website. This module is also refresh itself by the upcoming data in the database so that it predicts better as time goes on.

The third AI module that we plan to append to the project in the future is the recommendation system module. This module will recommend both the best corporations for a particular farmer in terms of the average distance of the farming areas in the cooperation landings and also recommend the best cooperation for a particular company according to some criteria determined by the company itself. This module will be consisting of an integer programming algorithm to calculate the shortest possible cycle within the landing areas because of the importance of fuel cost. The module will also upgrade the scores of corporations and also particular farmers based on their past votes by companies and other farmers and also the quality of past harvest which had been gotten in their landings.

9.3 Future Work on Formalization of the Project

In the future, we plan to formalize the website by cooperating with government to collect the sugar beet landing and amount of sugar beet data taken in those areas. By collecting the data, the object detection model planned to develop above may be excluded from the project because of existence of the ground truth data instead of predicted farming areas. If possible, we also plan to register all the farmers having a farming account in government automatically so that the website will be more formal and reliable.

9.4 Future Work on the Scope of the Project

Right now, the scope of our project consists of only sugar beet but in the future, we plan to add several products, such as hazelnut, tea plant, etc. The more product we add, the more data to train our networks will be needed in the planned deep learning modules above. We plan to start with collecting open source dataset to train our models then, continue with the data we collected in our website.

10 Glossary

MongoDB	Blockchain uses MongoDB to store values in the system.	7,8,10,14
Docker	Docker container is used for distribution the data of actors among the system peers.	8,11,13,14
Angular	Angular is an application design framework and development platform for creating efficient and sophisticated single page apps.	11,13,14
Typescript	Typescript is used for utilization of the Javascript effectively in the big projects.	11
HyperLedger Fabric	Hyperledger Fabric is intended as a foundation for developing applications or solutions with a modular architecture.	8,11,14
Hyperledger Composer	Hyperledger Composer is a set of collaboration tools for building blockchain business networks.	8,10,11,12,13,14

11 References

[1] “Data Where You Need It”, *Apache CouchDB,* [Online]. Available: https://couchdb.apache.org/. [Accessed: 26/5/2020]

[2] “Building Enterprise-Grade Blockchain Databases with MongoDB”, [Online]. Available: https://www.mongodb.com/collateral/building-enterprise-grade-blockchain-databases-with-mongodb. [Accessed: 26/5/2020]

[3] “About Hyperledger”, [Online]. Available: https://www.hyperledger.org/about. [Accessed: 26/5/2020]

[4] “Why Docker?”, [Online]. Available: https://www.docker.com/why-docker. [Accessed: 26/5/2020]

[5] “Writing Web Applications”, [Online]. Available: https://hyperledger.github.io/composer/v0.19/applications/web. [Accessed: 26/5/2020]

[6] “Angular”, [Online]. Available: https://www.typescriptlang.org/docs/handbook/angular.html. [Accessed: 26/5/2020]

[7] "TOPLUMSAL SORUMLULUK." *Konya şeker*. N.p., n.d. Web. 27 May 2020.

[8] Cai, Zhaowei, and Nuno Vasconcelos. “Cascade R-CNN: Delving Into High Quality Object Detection.” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, doi:10.1109/cvpr.2018.00644.