

# Smart Agricultural System-an application to maximize farmer's profits and increase productivity by using UI/UX methodologies and Agile Development

Dr.C.G.Balaji, Vankayala Jahnvi, Nagasruthi Uppalapat,  
Kambamhimabindu , Ushasree Polamarasetty

Date of Submission: 01-05-2023

Date of Acceptance: 10-05-2023

## ABSTRACT

This paper is going to introduce a proposed system to help farmers increase their productivity and effectiveness by helping them procure good quality seeds, and agricultural equipment, get interest-free loans from banks and NGOs, encourage peer-to-peer communication among farmers, provide them with viable and sustainable methods of farming such as organic farming and provide a platform like Big Basket to farmers where they can directly sell their products without the help of middlemen. We have built a smart agricultural application using UI/UX methodologies and have used Agile Software Development Process. The technologies we have used to build the smart agricultural application are HTML, CSS, React JS, Express JS, Python, Anaconda, Tensor Flow and MongoDB(No SQL), and NodeJS.

**KEYWORDS:** Smart Agricultural Application, UI/UX, Agile, HTML, CSS, React JS, Express JS, Python, Anaconda, TensorFlow, MongoDB, NodeJS.

## I. INTRODUCTION

Agriculture is the backbone of India. But there are a lot of problems in the agricultural sector in India. These problems include lack of proper agricultural equipment like machinery, lack of proper seeds, lack of money, lack of proper water harvesting systems, high debts of farmers, lack of education, increased dependence on middlemen to sell produce, etc. One way to solve these problems is to use technology. Technology has been rapidly developing and can produce wonders. The technology that is going to be used is based on web development using UI /UX methodologies and languages such as HTML, CSS, React JS, MongoDB, and Python. A smart agricultural application is built to help farmers

increase their productivity through viable and sustainable methods of farming such as organic farming, enhance peer-to-peer communication among farmers, help farmers get interest-free loans from certain banks willing to provide such loans or from NGOs, help farmers procure food quality seeds and agricultural equipment at minimal costs, provide a platform like Big Basket to farmers to help them sell their products without the need for middlemen, and last but not least, to provide farmers with unconditional support and counseling so that if they ever feel down, they know that they are not alone.

## II. RELATED WORK

1. Agro App: An application for healthy living published by Mayank Agarwal, Apoorv Kaushik, and Arlington Sengar talks about a mobile application that keeps farmers updated with all the information related to crops, insecticides, pesticides, financial sector, etc. It provides detailed information about which crops to grow in which season and which crops are suitable for the particular area in which the farmer is living.
2. Krishi Ville: Android based solution for Indian agriculture published by Manav Singhal, Kshitij Verma, and Anupam Shukla talks about an Android based mobile application using information and communication technology (ICT) that would take care of the updates of the different agricultural commodities, weather forecast updates and agricultural news updates. The paper proposes that the advancement of ICT could be utilized to provide accurate and timely relevant information and services to the farmers, thereby facilitating an environment for remunerative agriculture.
3. Android App to Connect Farmers to Retailers and Food Processing Industry published by Pranav Shiram and Sunil Mhamane talks about an Android

app that provides market information to a farmer using its easy interface on mobile application. The mobile application is intended to be used for fast and updated information delivering system for farmers. Also, it has native language support to make the transaction easy for farmers. The mobile application treats farmers as a seller and a buyer. The intention behind this paper is to help farmers as they buy or sell their agriculture goods and products. Market prices provided by data.gov.in lets the system to keep the selling and buying prices in control. As the products are to be browsed and there may be plenty of products for the user. To make browsing easy many filters are provided. Farmers face many problems while selling their goods and products, this system promises to provide an easy and recreational way to sell the products. The system lets the farmers to sell goods at a reasonable price and makes business even fair and transparent. Consumers are the opposite side of the same coin. This system lets consumers choose from a wide variety of products, select the product as per their requirement and also to apply price filters. Location is one of the parameters for consumers and producers- while selling or buying their product it will help the user to get the product nearby their location. The basic objective of the system is to consider everyone's needs and fulfill their requirements with fair and transparent agriculture business.

### **III. PRELIMINARIES**

#### **A. Login and Authentication**

This is the first module of our Smart Agricultural Application. This is where the farmers sign up to the app and create an account. After an account is created it has to be authenticated to verify whether the user is genuine or not. That is whether it is a robot. Websites and apps need user authentication nowadays. Authenticating users is crucial, but writing all this code ourselves is difficult. Firebase simplifies this. Firebase Authentication is responsible for everything that needs to be done on the server in order to authenticate the user. Using the Firebase SDK makes the Authentication process simple. It simplifies the process of using the API. Firebase Authentication's user interface libraries offer screens for logging. Firebase supports passwords, phone numbers, Google, Facebook, Twitter, and other identity providers. Firebase UI is a plug-and-play authentication service that can be added to apps and websites with ease.

#### **B. Creating Backend Routes and Initializing MongoDB**

There is a number of React routing frameworks available, but one of the most well-

known is React Router. The library is built with user-friendly modules that facilitate the creation of a declarative routing framework. Routers in React facilitate the generation of and movement between the various URLs that constitute a web application. They make it easy for users to switch between different parts of our program while their progress is saved, and they can even give each part of our app its own URL so that it may be easily shared. By reducing the number of clicks required to complete a task, we may enhance our app's user experience with the help of routers. For the purposes of setting up MongoDB, we shall refer to the official documentation. The MongoDB system is a document-based database. The BSON data format is used for storage. Documents in MongoDB are the unit of data storage; they are structured like JSON objects and consist of key-value pairs.

#### **C. Integration of Front end**

It is necessary for us to use the Express.js framework in order to compile our code before we can run our JavaScript code on the backend. Express.js is a node.js HTTP framework that assists us in the development of RESTful APIs that are completely functional. Web app and API development is Express JS's intended use. The Node.js community generally considers it to be the preferred server framework.

### **IV. SMART AGRICULTURAL APPLICATION BY INTEGRATING FRONT END, BACK END AND DATABASE**

In this section, we integrate the front end of the application with the backend and database. The technologies used here are HTML, CSS, React JS, Express JS, Python, Anaconda, Tensor Flow, MongoDB and Node JS. In the front end, we also talk about UI UX techniques which simplify our application and provide the best user experience to the farmers.

#### **A. SYSTEM OVERVIEW**

The purpose of our system is to provide a smart agricultural system for farmers with a user friendly and simple user interface. UI Design Principles such as the structure principle, the simplicity principle, the visibility principle, the feedback principle, the tolerance principle and the reuse principle. System boundary is detected by both functional and technical perspectives

1. single application system will normally have the following characteristics:
2. Consistent user interface design with and application behaviour.

3. Common architecture(e.g. Program language, program design).
4. Shared application system components (e.g. visual objects, programs, database tables).
5. When undertaking an analysis of computer systems, it is important to verify the presumed scope of the application system and to identify all interface or integration points it has with other application systems.
6. The scope of the application system affects the systems analysis effort. Scope is also significant when formulating system maintenance, enhancement or replacement options.

**Inputs:** User selecting a button, feature or tapping the screen.

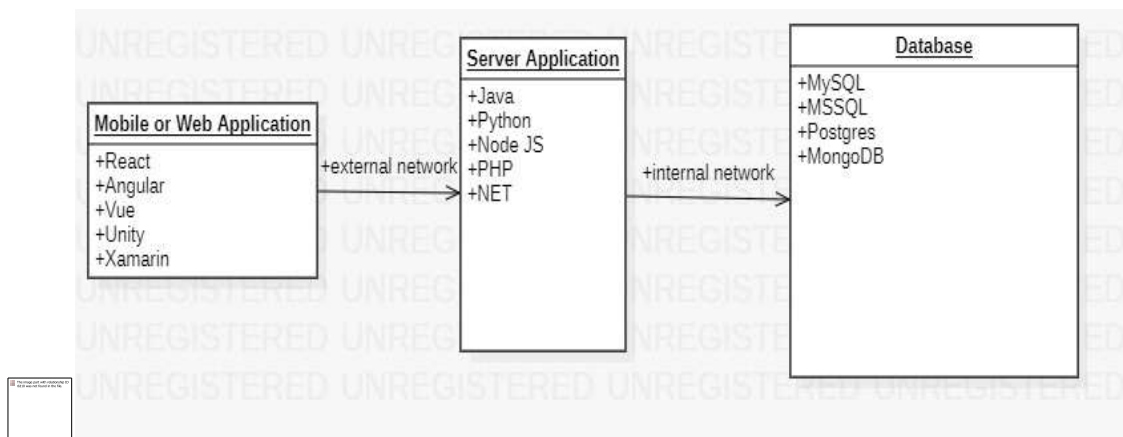
**Output:**An image, message or sound being displayed.

**Subsystem-**The purpose of the application subsystem is to facilitate easy software delivery and management across all ONOS instances in a cluster. The subsystem uses ONOS eventually consistent map and the inter-node communication mechanism to fully replicate the inventory of applications across the entire ONOS cluster.

**Feedback loops-** The performance of the app and user satisfaction is validated through feedback. Any negative feedback is looked at and changes are made accordingly.

### B. SYSTEM DESIGN

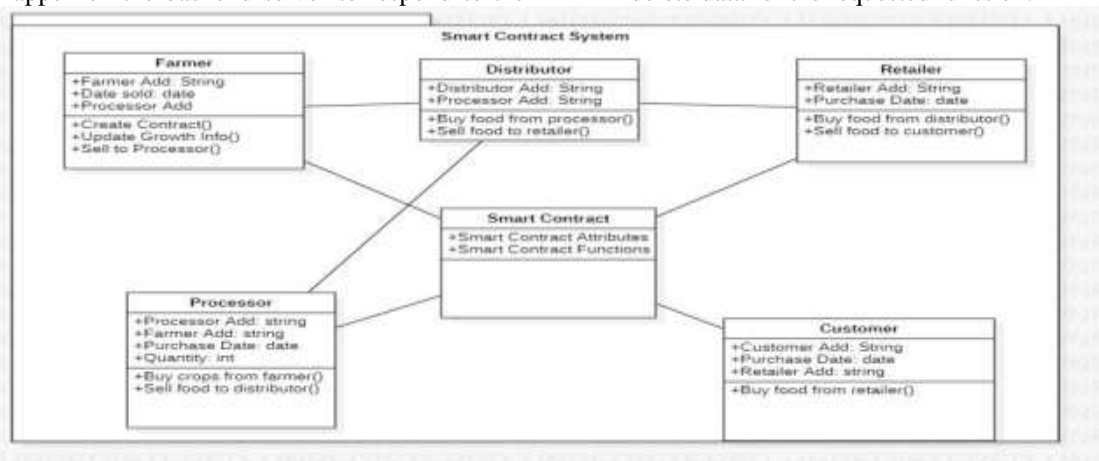
At the very basic level you have a front-end client application, a backend server application, and a database for storing your application data. The client makes requests to the server on behalf of the user and provides applications functionality.



1. Imagine a user clicking a link or tapping a button in the user interface of the front-end client application. This application runs on the user's computer or mobile device.
2. Based on the tap, some action may need to happen on the backend server to respond to the

user's request. In most cases this is done by sending an HTTP request over the internet to the backend server.

3. The backend server application may then need to connect to the database and fetch, create, update, delete data for the requested function.



### B. ENTITY SEQUENCE DIAGRAM

The relationships between entities, as shown in Fig. 2, show some of the key properties and capabilities of the smart contracts, as well as the relationships between entities and smart contracts. Each participating entity in the agricultural food supply chain participates by calling a function in the smart contracts. The smart contracts are created by the farmer, who grows the crop and uploads the growing environment, details and images to the IPFS by calling update-GrowthInfo (), which is stored in the IPFS hash, and updates the update-Growth Info () until the crop is ready for harvest. When the crops are harvested, the trade begins between the farmer and the processor. Once the farmer and the processor have negotiated the details of the agreement, the

farmer agrees and sells the crop to the processor. Fig. 3 shows the sequence diagram of the farmer and processor executing the sellToProcessor () and buyCropFromFarmer () functions, respectively. First, the processor executing the buyCropFromFarmer () Function, passing processor address, quantity and sales date Parameters to activate the smart contract trigger the CropRe-Quested () event to notify the participants, and passing and recording these parameters. Then, the farmer executing the sellToProcessor () function, passing the farmer address, processor address, quantity and sales date parameters, the smart contract trigger the CropSold () event to notify closing the transaction, and passing and recording these parameters.

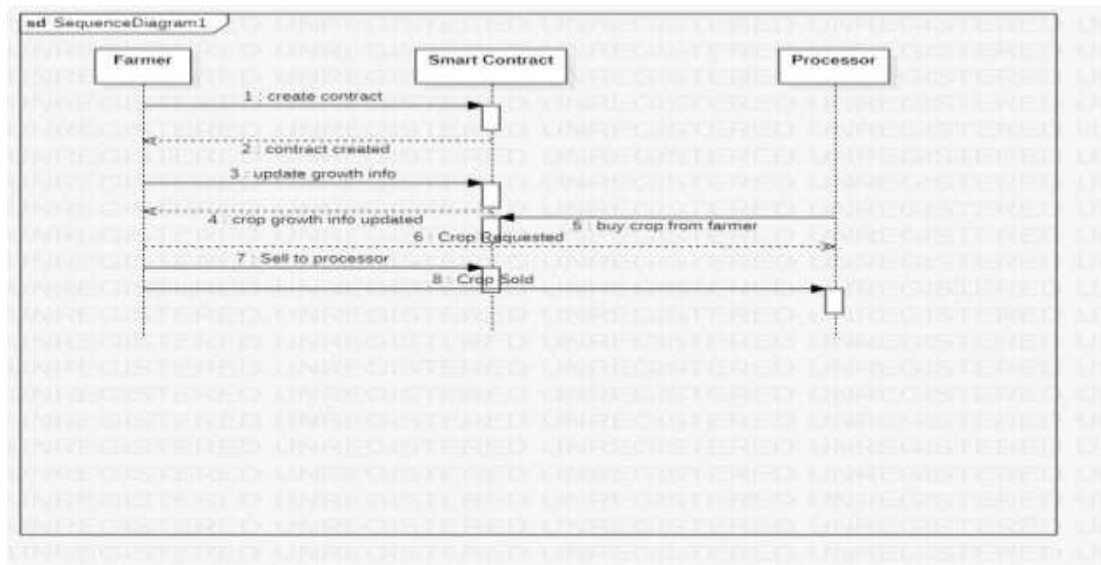
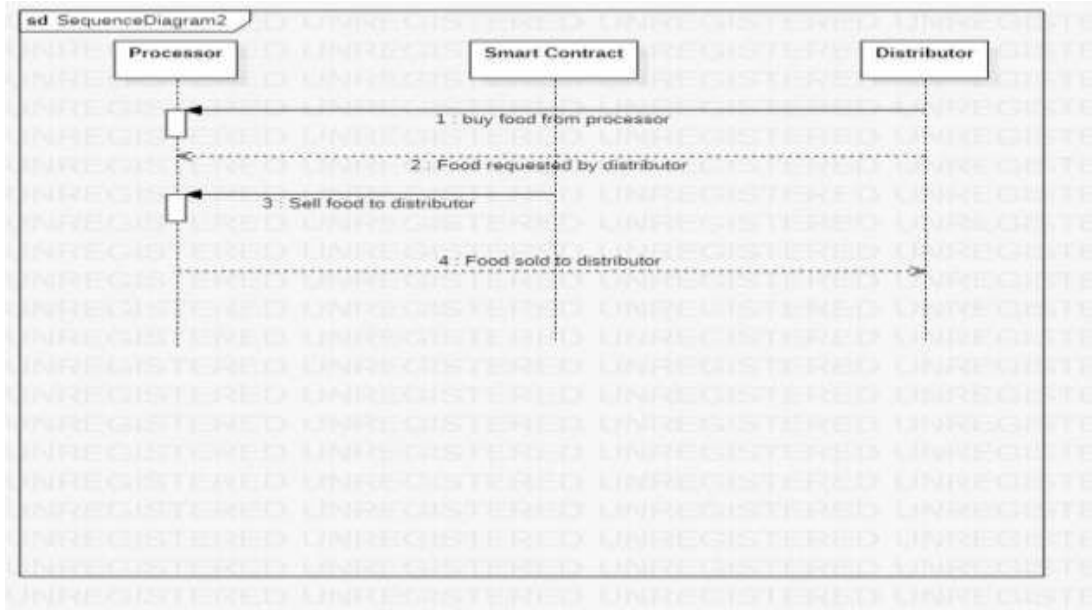


Fig. 3 shows a sequence diagram of the processor and distributor collaboration using smart contracts. The distributor is a warehouse that buys processed produce in bulk from various processors and sells it to retailers. Firstly, the distributor trigger AgriFoodRequestedByDistributor () event, passing

distributor address, processor address, quantity and sales date parameters to notifies the processor selling agricultural food to it, then the farmer performs thesellAgriFoodToDistributor () function, passing processor address, distributor address

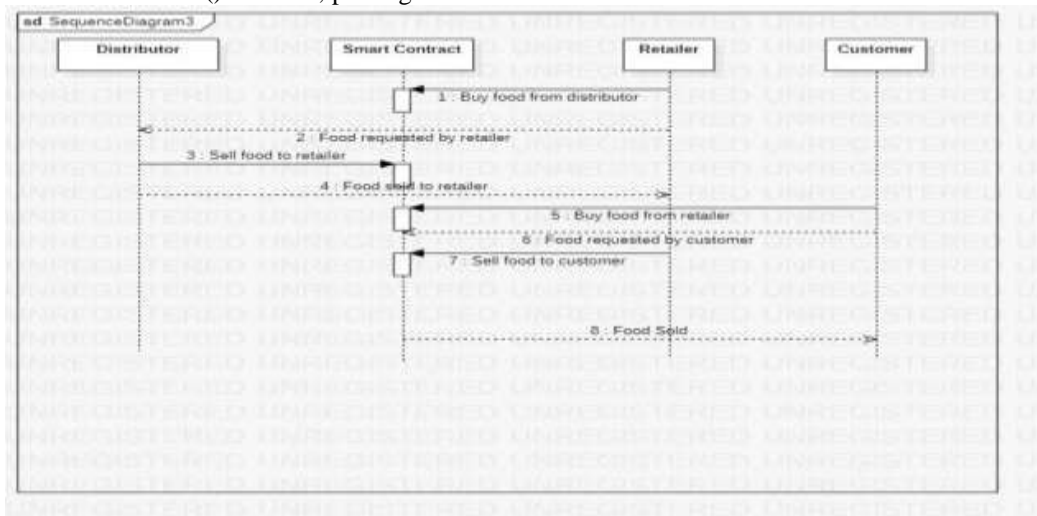


Sales quantity and sales date parameters to activate the AgriFoodSoldToDistributor() event to notify interaction entities. Retailers buy agricultural food from distributors, executing the buyAgriFoodFromDistributor() function, and passing retailer address, distributor address, quantity parameters. The activation AgriFoodRequestedByRetailer() event notifies the distributor, the distributor then performs the sellAgriFood To Retailer() function to sell agricultural food to the retailer, and activated event AgriFoodSoldToRetailer() notifies the relevant participant of this process. At the same time, passing the addresses of both parties, quantity, batch number and sales date parameters. Finally, the retailer sells the agricultural food to the customer by executing the sellAgriFoodToCusTomer() function, passing retailer

address, customer address, Agri-food name and sales date parameters, and broadcasts the process for the agricultural food via the AgriFoodSold() event. Fig. 5 shows a sequence diagram of distributor, retailer and customer.

### V. IMPLEMENTATION

As mentioned above, the smart contracts are created by the farmer. In the initial state of establishing the smart contracts, the smart contracts will check whether the farmer is registered. The processor then issues a purchase request, at which time the contract status is buyCropFromFarmer, and two conditions need to be checked: (1) Whether the requested





Processor is a registered entity; (2) Whether the processor has paid the fee. If these two conditions are satisfied, the contract status changes to CropRequestAgreed, the processor status is now WaitForCropFromFarmer, the farmer status changes to SellCropToProcessor and all active entities receive information from the farmer about selling crops to the processor.

If the above two conditions are not met, the contract state becomes CropRequestFailed, the processor state is Request-Failed, and the farmer state is CancelRequestOfProcessor.

#### Algorithm 1

Describes the process by which farmers sell their crops to processors.

**Input:** 'rp' is the list of registered Processors  
Address of Processor,  
Address of Farmer,  
Quantity, Date Purchased, Crop Price  
1 Contract state is buyCropFromFarmer  
2 State of the processor is Crop Requested  
4 Restrict access to only  $rp \in$  Processor  
5 if Crop Sale is agreed and Crop Price = paid then  
6 Contract state changes to CropRequestAgreed  
7 Change State of the processor to WaitForCropFromFarmer  
8 Farmer state is SellCropToProcessor  
9 Send a notification of crop sale to processor  
10 end  
11 else  
12 Contract state changes to CropRequestFailed  
13 State of processor is Request Failed  
14 Farmer state is CancelRequestOfProcessor  
15 Send a notification stating request failure  
16 end  
17 else  
18 Reset contract and displays an error message  
19 end

The processor then sells the processed crop to a distributor, who in turn sells it to retailer, as shown in algorithm 2. At this point, the production date, sales quantity and purchase date of the agricultural food are important parameters of the current stage. First, with respect to recognition address and the states of the distributor and retailer, due to the distributor having just finished the trade with the processor, the smart contract status is AgriFoodSoldToDistributor, and the state of the distributor is AgriFoodReceivedFromProcessor. The status of the retailer is ReadyToPurchase, which must satisfy two conditions: (1) Whether the requested retailer is a registered entity; (2) Whether to agree to the sales agreement and whether the agricultural food

payment has been completed. If these two conditions are satisfied, the contract will automatically execute the transaction with the contract status changed to SaleRequestedSuccess, distributor status changed to AgriFoodSoldToRetailer, and retailer status changed to AgriFoodDeliveredSuccess. Upon completion of the transaction, the deed will send a notification of successful deliver to the retailer. If the above two conditions are not satisfied, the contract status is changed to SaleRequestDenied, the distributor status is changed to RequestFailed, the retailer status is changed to AgriFoodDeliveryFailure, and the contract sends a notification of failure to all participants.

#### Algorithm 2

Distributors Sell Agri-Food to Retailer

**Input:** 'rr' is the list of registered Retailer  
Address of Distributor,  
Address of Retailer,  
Date Manufactured, Quantity,  
Date Purchase  
1 Contract state is AgriFoodSoldToDistributor  
2 State of the distributor is AgriFoodReceivedFromProcessor  
3 Retailer state is ReadyToPurchase  
4 Restrict access to only  $rr \in$  Retailer  
5 if Sale is agreed and Price = paid then  
6 Contract state changes to SaleRequestedSuccess  
7 Change State of the distributor to AgriFoodSoldToRetailer  
8 Retailer state is AgriFoodDeliveredSuccess  
9 Send a 'success' notification to retailer.  
10 end  
11 else  
12 Contract state changes to SaleRequestDenied  
13 State of distributor is Request Failed  
14 Retailer state is AgriFoodDeliveryFailure  
15 Send a 'failure' notification to all participants.  
16 end  
17 else  
18 Reset contract and displays an error message.  
19 end

Algorithm 3 describes the algorithm for consumers to purchase agricultural food from retailers. First, the consumer's initial state is ReadyToBuy. Thanks to the successful dealings between retailers and distributors, the smart contract state is SaleRequestAgreedSuccess, while retailer status is AgriFoodDeliveredSuccess. Similarly, smart contracts restrict customers who register with retailer to make purchase requests. The important parameters at this stage are customer address, retailer address, purchase date, sales ID, and AgriFood ID. When consumers successfully pay agricultural food prices, contract status changes to AgriFoodSoldToCustomer,

retailer statueto  
 SuccessfulPurchaseAgriFoodSaleSuccess, and  
 customer status to SuccessfulPurchase. If the  
 payment is not successful or the paid price is  
 incorrect, the contract status will be changed to  
 SaleOfAgriFoodDenied, the retailer status will be  
 AgriFoodSaleFailure, and then the customer status  
 will be changed to Failed Purchase.

**Algorithm 3**

Customer Buys From Retailer

**Input:** Address of Retailer,  
 Address of Customer,  
 SalesID,  
 AgriFoodID,  
 Date Purchased

- 2 State of the retailer is AgriFoodDeliveredSuccess
- 3 Customer state is ReadyToBuy
- 4 Restrict access to only Customers
- 5 if Price = paid then
- 6 Contract state changes to  
 AgriFoodSoldToCustomer
- 7 Change State of the retailer to  
 SuccessfulPurchaseAgriFoodSaleSuccess
- 8 Customer state is SuccessfulPruchase
- 9 Send a 'purchase success' notification.

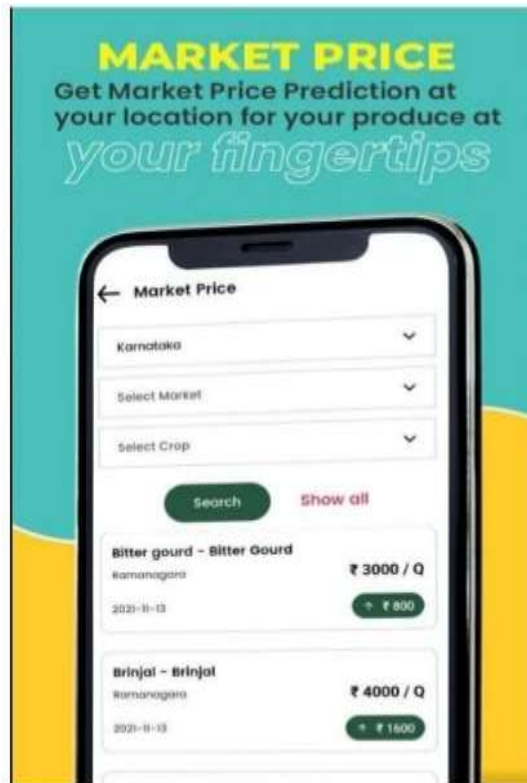
- 10 end
- 11 else
- 12 Contract state changes to SaleOfAgriFoodDenied
- 13 State of retailer is AgriFoodSaleFailure
- 14 Customer state is Failed Purchase
- 15 Send a 'purchase failure' notification.
- 16 end
- 17 else
- 18 Reset contract and displays an error message.
- 19 end

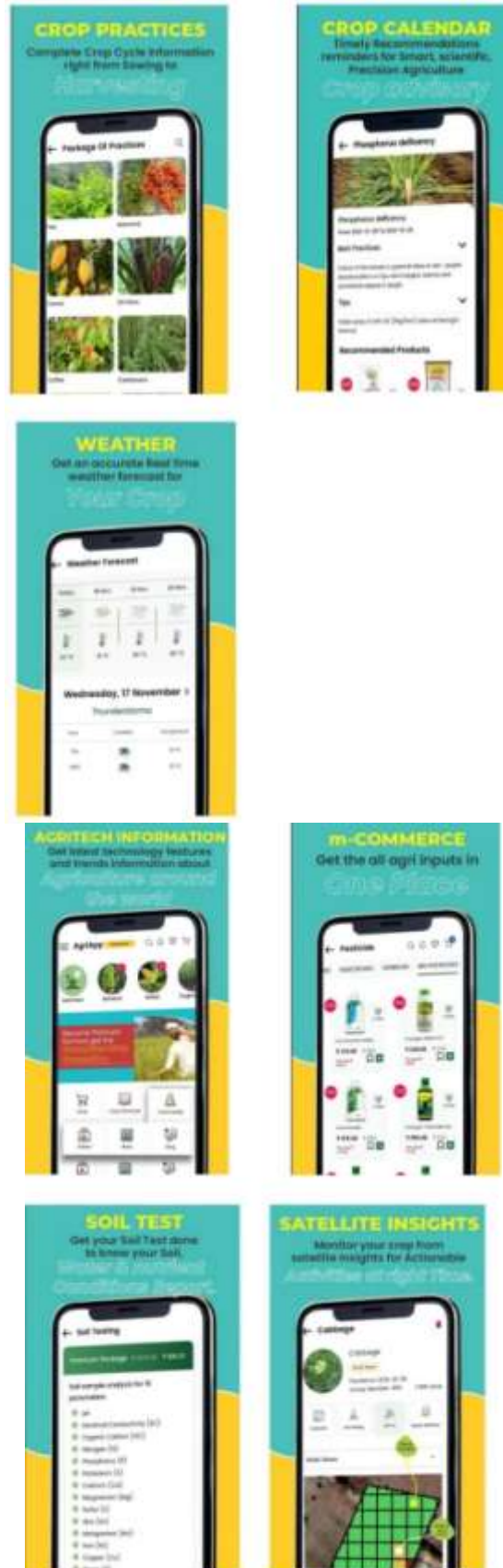
**VI.RESULTS AND ANALYSIS**

This project is not only limited to current technology but can be used by updating technologies for better farming ways and help the consumers to save their health.

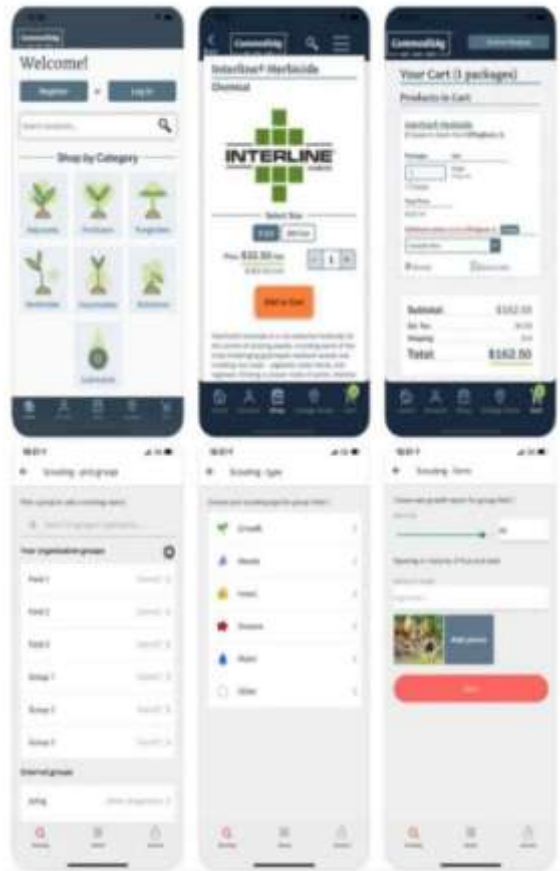
With the data gathered through this work, we can use this for creating a survey by analyzing how much percentage of farmers are able to use this technology and how many can understand by providing navigation

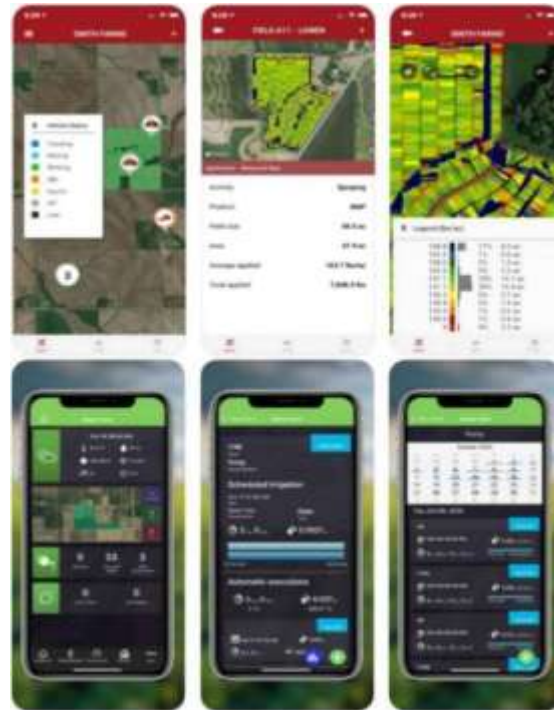
We created this app to make farmers utilize the technology for farming and avoid loss, making farmers owners of their farming and making them independently make them into profit ways.











## VII.CONCLUSION

We conclude that by using machine learning techniques to validate the providing information to the farmers based on the collected information and rules of the sectors and we are arranging a platform between farmers and consumers. By conforming with banks we will provide a platform for farmers to get loans online. We are also providing a facility where they can order the pesticides and seeds according to their requirements. We will guide farms to grow organic farming and make profits and make sure the health of the public increase.

They can call our advisers directly and ask their doubts. Overall our app is providing information on Land test and geographic data, Details about soil and soil fertility, Seeds, Pesticides, Organic farming guidelines, Crop rotation, Loans from private bank, Loan offer I government banks, Vehicles for rent, Workers nearby, Irrigation method suitable for a particular crop, selling sector, weather updates and news regarding farmers, Call to adviser and finally problems and suggestion from agriculture experts. By creating an account in Agriculture in one tap they can access our services

## REFERENCES

- [1]. [https://www.researchgate.net/publication/348178814\\_Smart\\_AgricultureSystem/link/5ff2b3a7a6fdcccb82a851e/download](https://www.researchgate.net/publication/348178814_Smart_AgricultureSystem/link/5ff2b3a7a6fdcccb82a851e/download)
- [2]. [https://www.researchgate.net/publication/315770657\\_Smart\\_Agricultural\\_System\\_Better\\_Accuracy\\_and\\_Productivity](https://www.researchgate.net/publication/315770657_Smart_Agricultural_System_Better_Accuracy_and_Productivity)
- [3]. <https://irejournals.com/formatedpaper/1700219.pdf>
- [4]. <https://www.fao.org/climatechange/297900178d452d0ca9af024aad1092d4b78b1d.pdf>
- [5]. [https://play.google.com/store/apps/details?id=com.criyagen&hl=en\\_IN&gl=US](https://play.google.com/store/apps/details?id=com.criyagen&hl=en_IN&gl=US)
- [6]. <https://www.croplife.com/editorial/matt-hopkins/best-agriculture-apps/>
- [7]. <https://core.ac.uk/download/pdf/295538594.pdf>
- [8]. <https://www.google.com/url?sa=t&source=web&rct=j&url=https://ieeexploreieee.org/abstract/document/7786002&ved=2ahUKEwiIyJLK7Yz7AhV7SWwGHVpUDVcQFnoECIQAAQ&usg=A0vVaw0h2Looyz9I9tLPB3HrYkz>
- [9]. <https://www.google.com/url?sa=t&source=web&rct=j&url=https://ieeexploreieee.org/document/6965213&ved=2ahUKEwiZrYz17Yz7AhXoSWwGHbUIDEoQFnoECA8QAAQ&usg=AOvVaw3ISP1VOsMvNYttabpNWL2a>
- [10]. <https://www.google.com/url?sa=t&source=web&rct=j&url=https://ieeexploreieee.org/document/6757170&ved=2ahUKEwiZrYz17Yz7AhXoSWwGHbUIDEoQFnoECAoQAQ&>

- usg=AOvVaw0\_  
QhHWCy5uoTj7N1MvFlkB
- [11]. <https://www.google.com/url?sa=t&source=web&rct=j&url=https://ieeexplore.ieee.org/document/6163685&ved=2ahUKEwiZrYzl7Yz7AhXoSwwGHbUIDEoQFnoECB4QAQ&usg=AOvVaw324gYKerABuvBj8skscbY>
- [12]. <http://www.aui.ma/sscapstonerepository/pdf/spring2020/SMART%20FARMING%20SYSTEM.pdf>
- [13]. [https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.irt.org/wizardforfarmersusingmobileandwebapplicationuzhavu&ved=2ahUKEwiQx4f6-Jb7AhW5RWwGHViBAHMQFnoECCsQAQ&usg=AOvVaw2r\\_i\\_c-J8PIA3iveEg6b66v](https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.irt.org/wizardforfarmersusingmobileandwebapplicationuzhavu&ved=2ahUKEwiQx4f6-Jb7AhW5RWwGHViBAHMQFnoECCsQAQ&usg=AOvVaw2r_i_c-J8PIA3iveEg6b66v)
- [14]. [https://www.google.com/url?sa=t&source=web&rct=j&url=https://researchjournal.co.in/upload/assignments/14\\_334342.pdf&ved=2ahUKEwiQx4f6Jb7AhW5RWwGHViBAHMQFnoECCwQAQ&usg=AOvVaw3Fh33UGk9esaQCgAye0dU7](https://www.google.com/url?sa=t&source=web&rct=j&url=https://researchjournal.co.in/upload/assignments/14_334342.pdf&ved=2ahUKEwiQx4f6Jb7AhW5RWwGHViBAHMQFnoECCwQAQ&usg=AOvVaw3Fh33UGk9esaQCgAye0dU7)
- [15]. [https://www.google.com/url?sa=t&source=web&rct=j&url=https://codeshy.com/shop/product/farmer-information-app-for-agriculture/&ved=2ahUKEwjz-ZuE-Zb7AhWYTWwGHebJBe44ChAWegQIDhAB&usg=AOvVaw2jt pDZ5pBAv64MM\\_mQr3Xf](https://www.google.com/url?sa=t&source=web&rct=j&url=https://codeshy.com/shop/product/farmer-information-app-for-agriculture/&ved=2ahUKEwjz-ZuE-Zb7AhWYTWwGHebJBe44ChAWegQIDhAB&usg=AOvVaw2jt pDZ5pBAv64MM_mQr3Xf)
- [16]. <https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.vteam.space/blog/how-to-build-an-agriculture-app/&ved=2ahUKEwiD8NrR5b7AhVuS2wGHedRBCoQFnoECEgQAQ&usg=AOvVaw0IUacqrUjcaDmdt2dMTGr>
- [17]. [https://www.google.com/url?sa=t&source=web&rct=j&url=https://kodytechnolab.com/agriculture-farmer-app-development&ved=2ahUKEwiD8NrR5b7AhVuS2wGHedRBCoQFnoECEIQAQ&usg=AOvVaw2Qxn\\_qKRQNc217x7zK6R-N](https://www.google.com/url?sa=t&source=web&rct=j&url=https://kodytechnolab.com/agriculture-farmer-app-development&ved=2ahUKEwiD8NrR5b7AhVuS2wGHedRBCoQFnoECEIQAQ&usg=AOvVaw2Qxn_qKRQNc217x7zK6R-N)
- [18]. [https://www.google.com/url?sa=t&source=web&rct=j&url=https://rspsciencehub.com/article\\_12023.html&ved=2ahUKEwiVo\\_rk5b7AhX2TGwGHYHHCBI4ChAWegQIFhAB&usg=AOvVaw0d w4bJsnlGs2-G64nW91OM](https://www.google.com/url?sa=t&source=web&rct=j&url=https://rspsciencehub.com/article_12023.html&ved=2ahUKEwiVo_rk5b7AhX2TGwGHYHHCBI4ChAWegQIFhAB&usg=AOvVaw0d w4bJsnlGs2-G64nW91OM)