isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 3 ,Apr,2025 ,ISSN_NO: 2321-3337

ECOMMERCE BASED BUSINESS OF FARMER TO CLIENT AND PASS FOR BUSINESS HEADWAY ON THROUGH MOBILE APPLICATION

Prem kumar H[1] , Mahesh K[2] , Mahammad Ali S K[3] , Vinayagam T A[4*]

[123]UG Scholar-Dept CSE, GRT Institute of Engineering and Technology, Tiruttani , India.

[4*]Assistant Professor-Dept CSE, GRT Institute of Engineering and Technology, Tiruttani , India.

itspremkumar18@gmail.com , naidumahesh004@gmail.com, crazymahammad@gmail.com

[4*] Corresponding Author : vinayagamta@grt.edu.in.

Abstract- In these portable application, the modules of rancher and client in these application, the rancher register in these application the check otp shipped off email and verify with required arrangement of versatile the client login in to application, the rancher add the result of harvests or vegetables, the additional items view in the client side the client of client purchase the item in these application they ready to talk with rancher in these application.

The business improvement development of assess of business the board in these use of advancement , they talk with others to examine with the rancher module , the arranged subtleties in the application view in client and rancher both of these arrangement of utilization.

1. INTRODUCTION

In these convenient application, the modules of farmer and client in these application, the farmer register in these application the browse otp delivered off email and check with required course of action of adaptable the client login in to application.

The farmer add the after effect of harvests or vegetables, the extra things view in the client side the client of client buy the thing in these application they prepared to converse with farmer in these application.The business improvement of survey of business the board in these utilization of headway , they talk with others to analyze with the farmer module , the organized nuances in the application view in client and farmer both of these game plan of usage.

The application aims to bridge the gap between ranchers and clients by facilitating a seamless interaction platform where ranchers can list and manage their produce while clients can browse, purchase, and communicate with ranchers directly.

It seeks to enhance business management and development through streamlined communication and transaction processes. The application includes modules for both ranchers and clients. Ranchers can register and verify their accounts using OTP sent to their email. They can then list their produce, which becomes available for viewing by clients. Clients can log in, browse the listed items, make purchases, and initiate communication with ranchers.

2. MODULE DESCRIPTION

2.1. Customer Register and Login

This module is designed to manage the registration and login processes for customers. It begins with the collection of customer details, including personal information and contact details. After submission, the system sends a verification email to the customer. This step is crucial for confirming the authenticity of the user's email address and preventing fraudulent registrations. Once the customer verifies their email, they can access the application by logging in with their credentials.

The module ensures that the data is securely handled and stored, maintaining the integrity and confidentiality of user information. This setup creates a secure entry point into the application and forms the foundation for all subsequent user interactions.

2.2. Farmer Register and Login

This module operates similarly to the customer registration system but is tailored for farmers. It involves the collection and verification of farmer details, including their business information and contact details. After farmers submit their registration information, they receive an email with a verification link or OTP. This process confirms their email address and activates their account. Once verified, farmers can log in to the application to manage their profile and access other functionalities. The module ensures that only verified and legitimate farmers can use the platform, enhancing security and reliability.

2.3. Farmer Add Products

In this module, farmers have the capability to add new products and manage their existing listings. Farmers can input detailed information about their produce, such as product name, description, price, and images. This information is then saved in the database and displayed on the platform for customers to view. The module also allows farmers to update or remove products as needed, ensuring that the information remains current and accurate. This feature is essential for enabling farmers to effectively market their products and attract potential buyers.

Volume: 6, Issue: 3,Special Issue: 3 ,Apr,2025 ,ISSN_NO: 2321-3337

2.4. Customer View

The Customer View module is designed to provide an organized and user- friendly interface for browsing products. Customers can explore a list of products added by farmers, with options to filter and sort based on various criteria such as product type, price, and availability. Each product listing includes detailed information, including descriptions, prices, and images, allowing customers to make informed decisions. By presenting product information in a clear and structured manner, this module helps customers easily find and evaluate items of interest.

2.5. Customer Chat

This module facilitates direct communication between customers and farmers through a chat interface. Before making a purchase, customers can use this feature to ask questions, seek additional information, or clarify any doubts they may have about a product. The chat functionality promotes transparency and fosters a better understanding between buyers and sellers. It allows for real-time interaction, which can help resolve issues quickly and enhance the overall customer experience. This direct line of communication is crucial for building trust and ensuring customer satisfaction.

2.6. Buy Product

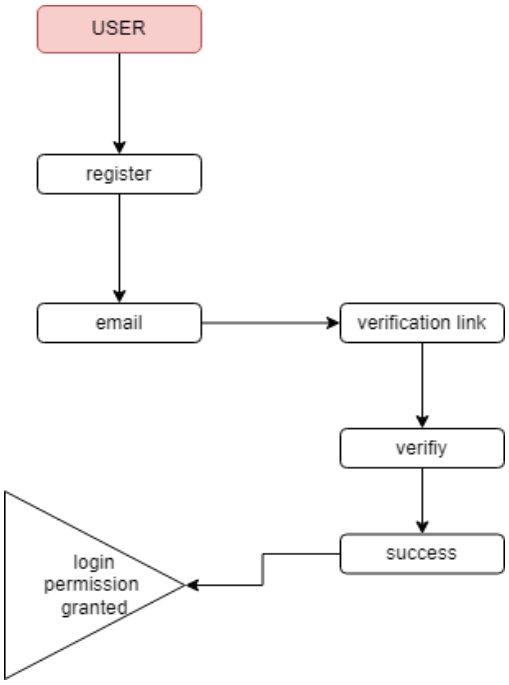
The Buy Product module manages the purchasing process for customers. It provides a streamlined checkout experience, often featuring a banner or popup on the purchase page to highlight important information or promotions.

Customers can select their desired products, review their choices, and proceed to checkout. The module guides users through payment options and order confirmation.

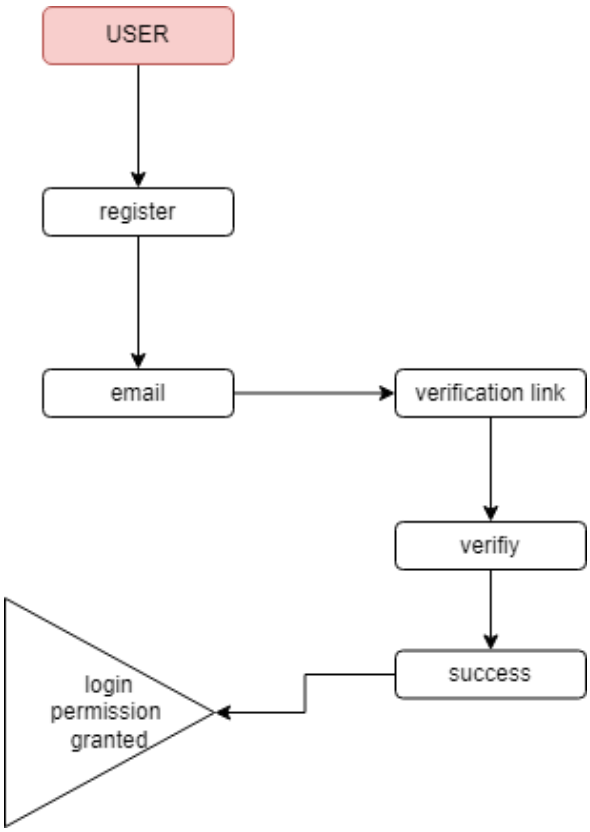
By facilitating an intuitive and efficient buying process, this feature ensures that customers can complete their transactions smoothly and securely, contributing to a positive overall shopping experience..

2.7. Module Diagram

2.7.1 Farmer



2.7.2 Client



3. DESIGN ENGINEERING

3.1. General:

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

3.2 Use Case :

The use case diagram for the described agricultural platform illustrates the interactions between its primary user types: farmers and clients, through a series of well-defined functionalities and processes. Farmers initiate their engagement with the platform by undergoing a secure registration process.

This is followed by identity verification via an OTP (One-Time Password) sent to their email, ensuring that only verified and authorized users can access the system. Once authenticated, farmers gain access to the Rancher module, where they can manage and update details about their crops or vegetables.

This includes inputting new information and making real-time updates, which are then visible to clients through the Client module. This real-time data synchronization ensures that clients always have access to the most current crop listings. Farmers and clients can interact directly through the platform, leveraging features designed for direct communication.

This functionality allows users to discuss product details, negotiate terms, and resolve queries, facilitating a more personalized and effective transaction process.

3.3 State :

The state diagram for the agricultural platform represents the various states and transitions that occur within the system for both farmers and clients, detailing how the platform responds to user interactions and system processes.Initially, farmers enter the Registration State, where they provide necessary details to create an account. Upon completing this step, they transition to the Verification State. Here, the system sends an OTP to their email for identity verification.

The farmer remains in the Verification State until they input the correct OTP, at which point they transition to the Authenticated State.

In the Authenticated State, farmers can access the Rancher Module, where they have the ability to Input/Update Crop Information.

This state allows farmers to manage their crop data, which is then reflected in real-time on the client side. The farmer remains in this state, making updates as needed, until they choose to log out, which returns them to the Logged-Out State.

3.4 Activity :

The activity diagram for the agricultural platform outlines the sequence of activities and decision points involved in the interactions between farmers and clients, illustrating how the platform facilitates and manages these interactions through a series of coordinated processes.

The process begins with the Farmer Registration activity, where farmers input their details to create an account. Following registration, the system initiates the Identity Verification activity by sending an OTP (One-Time Password) to the farmer's email. The farmer must then complete the OTP Verification step by entering the received code. If the OTP is correct, the system transitions the farmer to the Authenticated State, allowing them to access the Rancher Module.

3.5 Class :

At the core of the diagram are two primary classes: Farmer and Client. The Farmer class includes attributes such as `farmer ID`, `name`, `email`, and `crop Details`. Methods for this class encompass `register()`, `verifyIdentity()`, and `updateCropInfo()`. The `register()` method initiates the account creation process, while `verifyIdentity()` handles OTP verification, and `updateCropInfo()` allows farmers to input or modify their crop data. This class is connected to the Crop class, which holds attributes like `cropID`, `type`, `quantity`, and `price`. Methods in the Crop class include `addCrop()` and `editCrop()`, used for managing crop details.

The Client class features attributes such as `clientID`, `name`, `email`, and `purchasedCrops`. Methods for this class include `register()`, `login()`, and `viewCrops()`. The `register()` and `login()` methods handle client account creation and authentication, while `viewCrops()` allows clients to access and browse available crop listings.

The Client class is associated with the Transaction class, which includes attributes like `transactionID`, `clientID`, `cropID`, and `quantity`. Methods for the Transaction class include `makePurchase()` and `viewTransactionHistory()`, enabling clients to perform and review their transactions.Additionally, there is a Communication class that represents interactions between farmers and clients. This class contains attributes such as `messageID`, `senderID`, `receiverID`, and `messageContent`.

Methods include `sendMessage()` and `receiveMessage()`, facilitating direct communication between users..

4. PROPOSED SYSTEM

The proposed system aims to develop The application. the business improvement advancement of survey of business the board in these utilization of headway , they talk with others to look at with the farmer module , the organized nuances in the application view in client and farmer both of these course of action of usage.

4.1 Business Development through technology :

The application acts as a digital platform for agricultural business. Farmers can showcase their products (crops/vegetables), increasing visibility and reach beyond local markets. Clients can discover, compare, and purchase directly from farmers, promoting direct trade and reducing middlemen.

4.2 Communication and Collaboration :

A built-in chat feature allows clients and farmers to communicate directly. They can discuss product details, prices, availability, delivery, and more before making a purchase decision. This two-way communication enhances transparency and trust, which is essential for business growth.

4.3 Organised Information Sharing :

Product information, pricing, availability, and farmer contact are well-structured and visible to both clients and farmers. Both parties can manage their own dashboards to track: Listings (for farmers) **,** Orders (for both)**,** Messages Purchase history (for clients).

5. DEVELOPMENT TOOLS :

5.1Flutter:

Flutter is an open-source UI software development toolkit created by Google. It is used to build natively compiled applications for mobile, web, and desktop from a single codebase. Flutter was first introduced in 2015 and has gained popularity for its ability to enable developers to create high-performance, visually appealing applications with a flexible and expressive user interface.

Here are key aspects of Flutter:

5.1.1 Single Codebase:

Flutter allows developers to write code once and deploy it on multiple platforms, including iOS, Android, web, and desktop. This helps in reducing development time and effort.

5.1.2 Dart Programming Language:

Flutter apps are primarily written in the Dart programming language. Dart is a modern, object-oriented language developed by Google.

5.1.3 Widgets:

Flutter uses a widget-based architecture for building user interfaces. Everything in Flutter is a widget, including structural elements like buttons and layout elements like rows and columns.

Widgets are reusable and can be combined to create complex UIs.

5.1.4 Hot Reload:

One of Flutter's most praised features is hot reload, which allows developers to instantly see the effect of code changes during development without restarting the entire application.

This accelerates the development process and makes it easier to experiment with UI changes.

5.1.5 Rich Set of Pre-designed Widgets:

Flutter comes with a rich set of pre-designed widgets for common UI elements, making it easier for developers to create consistent and attractive interfaces. Custom widgets can also be created to suit specific design requirements.

5.1.6 Material Design and Cupertino Styles:

Flutter provides widgets that implement the Material Design guidelines for Android apps and the Cupertino style for iOS apps. This allows developers to create platform-specific designs while sharing most of the code.

5.1.7 Performance:

Flutter compiles to native ARM code, providing high performance on both iOS and Android.

The framework also includes a GPU-accelerated rendering engine, making it suitable for graphics-intensive applications.

5.1.8 Integration with Native Features:

Flutter allows developers to integrate with native features and APIs of the underlying operating systems, providing access to device-specific capabilities.

5.1.9 Community and Ecosystem:

Flutter has a growing and active community of developers who contribute to its ecosystem.

There is a wide range of third-party packages and plugins available through Dart's package manager (pub.dev) that can be used to extend Flutter's functionality.

5.1.10 Web and Desktop Support:

Flutter extends its platform support beyond mobile devices. It includes experimental support for building web applications and desktop applications for Windows, macOS, and Linux.

6. FEASIBILITY STUDY

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success.

In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

6.1 Economical feasibility

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not.

6.2 Technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.

6.3 Operational feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

7. VARIOUS LEVELS OF TESTING

7.1 White Box Testing

White-box testing, sometimes called glass-box, is a test case design method that uses the control structure of the procedural design to derive test cases. Using White Box testing methods, we can derive test cases that Guarantee that all independent paths within a module have been exercised at least once Exercise all logical decisions on their true and false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structures to assure their validity.

7.2 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested.

Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it.

The test provides inputs and responds to outputs without considering how the software works.

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

7.3 Unit Testing

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application.

In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method.

Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process.

Unit testing is software verification and validation method in which the individual units of source code are tested fit for use. A unit is the smallest testable part of an application. In this testing, each class is tested to be working satisfactorily.

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated.
It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

7.4 Functional Testing

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does.

Functional testing differs from system testing in that functional testing "verifies a program by checking it against ... design document(s) or specification(s)", while system testing "validate a program by checking it against the published user or system requirements" (Kane, Falk, Nguyen 1999, p. 52).

Functional testing typically involves five steps. The identification of functions that the software is expected to performThe creation of input data based on the function's specifications , The determination of output based on the function's specifications , The execution of the test case , The comparison of actual and expected outputs.

isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 3 ,Apr,2025 ,ISSN_NO: 2321-3337

8. Layout:

Earlier in client- server computing, each application had its own client program and it worked as a user interface and need to be installed on each user's personal computer. Most web applications use HTML/XHTML that are mostly supported by all the browsers and web pages are displayed to the client as static documents.

A web page can merely displays static content and it also lets the user navigate through the content, but a web application provides a more interactive experience.

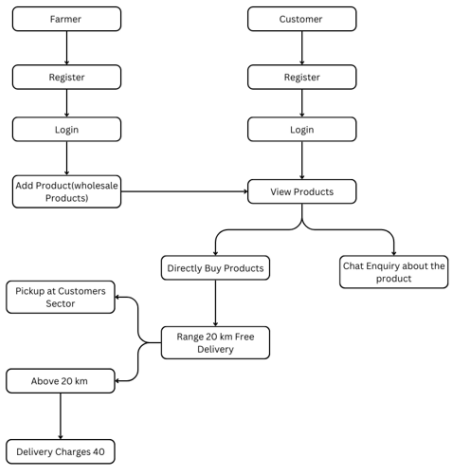
Any computer running Servlets or JSP needs to have a container. A container is nothing but a piece of software responsible for loading, executing and unloading the Servlets and JSP.

While servlets can be used to extend the functionality of any Java- enabled server. They are mostly used to extend web servers, and are efficient replacement for CGI scripts.

CGI was one of the earliest and most prominent server side dynamic content solutions, so before going forward it is very important to know the difference between CGI and the Servlets.

8.1ARCHITECTURAL DIAGRAM

The system architecture diagram for the agricultural platform outlines the overall structure and organization of the system, depicting how different components interact to deliver the platform's functionalities.



8.2 DATA FLOW DIAGRAM:

LEVEL **0:**

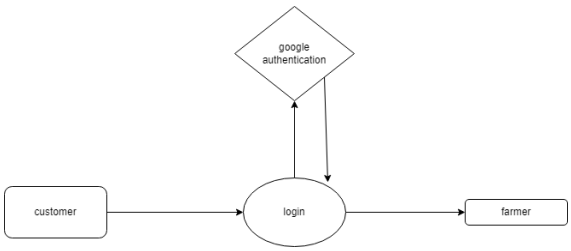


Fig. 6. 4.1. Diagram of Level 0

LEVEL 1:

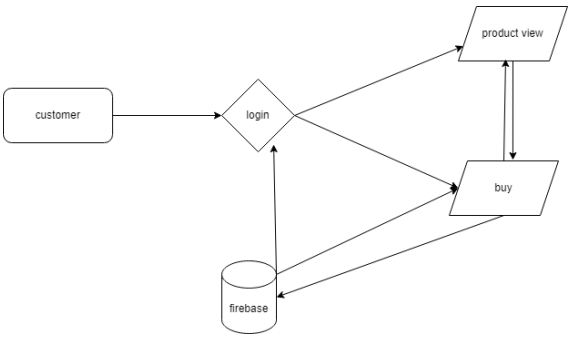


Fig. 6.4.2. Diagram of Level 1

LEVEL 2:

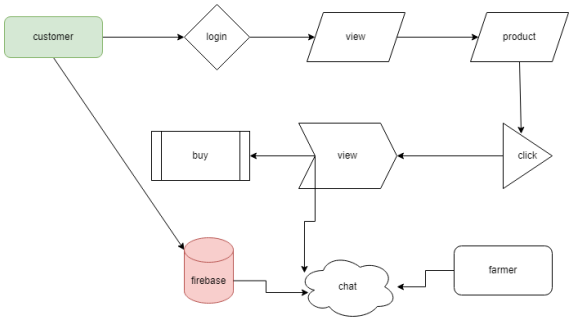


Fig. 6.4.3. Diagram of Level 2

7.EXPERIMENTAL ANALYSIS

This result discusses about the implementation of the e-commerce mobile application connecting farmers directly with clients can boost business headway by streamlining transactions, reducing costs, and expanding market reach, potentially leading to increased profits for farmers and better access to fresh produce for consumers. below as Fig. 7.1., Fig.7.2. and Fig. 7.3.

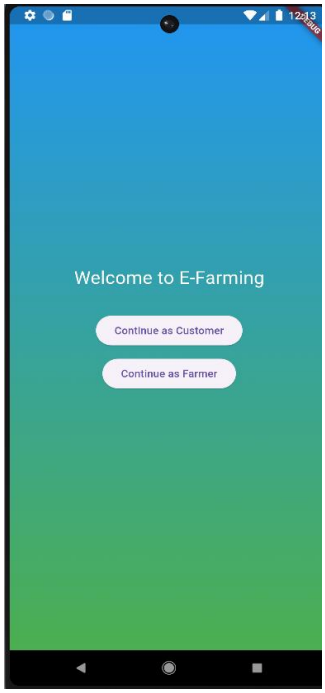


Fig. 7.1. Shows the Page of client and farmer

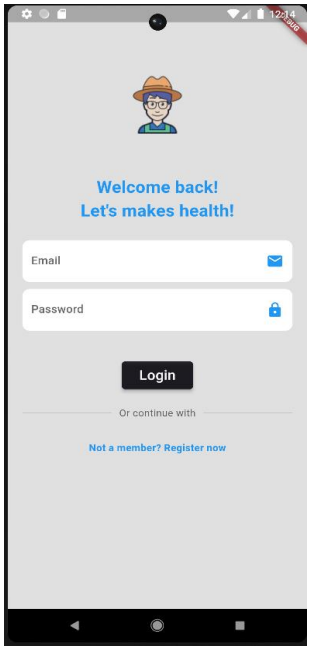


Fig. 7.2. Shows the Login Page of Farmer

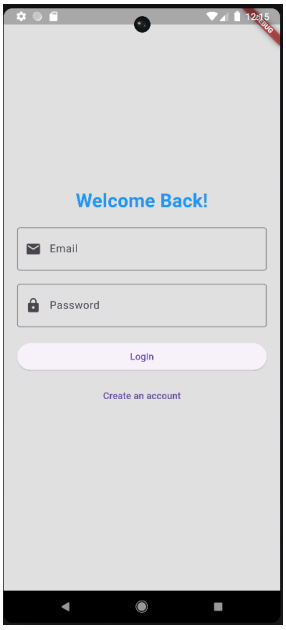


Fig. 7.3.Shows the Login Page of Client

9. CONCLUSION:

In conclusion, the integrated platform described represents a advancement in the agricultural sector by bridging the gap between farmers and clients through its tailored modules and real-time capabilities. By providing a secure and user-friendly interface for both farmers and clients, the platform ensures that all interactions are efficient and effective. The farmer module facilitates seamless registration, crop management, and direct communication, while the client module offers up-to-date information and options, enhancing decision-making and marketplace responsiveness.

Furthermore, the platform's analytical tools support business development by offering valuable insights into transactions and operational data, thereby fostering better strategic decisions and improving overall operational efficiency. This comprehensive approach not only optimizes the transactional process but also contributes to a more dynamic, efficient, and responsive agricultural marketplace, ultimately benefiting both producers and consumers.

10. REFERENCE

[1] O. S. Manfreda et al., "On the use of unmanned aerial systems for environ-mental monitoring," Remote Sens., vol. 10, no. 4, 2018, Art. no. 641.

[2] M. F. McCabe, R. Houborg, and A. Lucieer, "High-resolution sensing for precision agriculture: From earth-observing satellites to unmanned aerial vehicles," Proc. SPIE, vol. 9998, pp. 346–355,2016.

[3] K. Wang, S. E. Franklin, X. Guo, and M. Cattet, "Remote sensing of ecology, biodiversity and conservation: A review from the perspective of remote sensing specialists," Sensors, vol. 10, no. 11, pp. 9647–9667,2010.

[4] F.-H. Tseng, H.-H. Cho, and H.-T. Wu, "Applying Big Data for intelligent agriculture-based crop selection analysis," IEEE Access, vol. 7,pp. 116965–116974, 2019.

[5] M. Maniadaki, A. Papathanasopoulos, L. Mitrou, and E.-A. Maria, "Reconciling remote sensing technologies with personal data and privacy pro-tection in the European Union: Recent developments in Greek legislation and application perspectives in environmental law," Laws, vol. 10, no. 2,p. 33, 2021.

[6] A. R. White, "Human expertise in the interpretation of remote sensing data: A cognitive task analysis of forest disturbance attribution," Int. J.Appl. Earth Observ. Geoinf., vol. 74, pp. 37–44, 2019.

[7] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linearmixing model to address spectral variability for hyperspectral unmixing,"IEEE Trans. Image Process., vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[8] P. Mairota et al., "Challenges and opportunities in harnessing satellite remote-sensing for biodiversity monitoring," Ecological Informat.,vol. 30, pp. 207–214, 2015.

[9] R. Sharma and R. Arya, "UAV based long range environment monitoring system with Industry 5.0 perspectives for smart city infrastructure,"Comput. Ind. Eng., vol. 168, 2022, Art. no. 108066.

[10] P. Fraga-Lamas, S. I. Lopes, and T. M. Fernández-Caramés, "GreenIoT and edge AI as key technological enablers for a sustainable digital transition towards a smart circular economy: An Industry 5.0 use case,"Sensors, vol. 21, no. 17, 2021, Art. no. 5745.